universität wien

MASTERARBEIT | MASTER'S THESIS

Titel | Title

Beats, Bars, and Bad Words: A Comparative Analysis of Profanity Detection in Code-Switched German Rap Lyrics

verfasst von | submitted by

Karin Niederreiter BA

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien | Vienna, 2024

Studienkennzahl lt. Studienblatt | Degree programme code as it appears on the student record sheet:	UA 066 587
Studienrichtung lt. Studienblatt | Degree programme as it appears on the student record sheet:	Joint-Masterstudium Multilingual Technologies
Betreut von | Supervisor:	Assoz. Prof. Mag. Dr. Dagmar Gromann BSc

Acknowledgements

I would like to express a big thank you to my supervisor, Assoz. Prof. Mag. Dr. Dagmar Gromann, BSc, for her patience and support throughout the entire writing process. Her guidance, encouragement, and expertise have been truly invaluable to me.

I also want to express my special appreciation to my friends and family, whose belief in me has been a constant source of motivation. Lastly, I want to thank everyone in the MLT24 cohort for creating such a motivating and supportive environment. I am very grateful to have shared this journey with them.

Abstract

The exponential growth of the social media user base has led to an alarming rise in the use of offensive language and profanity, necessitating effective detection mechanisms. This thesis explores the challenges and advancements in automatic profanity detection, focusing on the limitations of existing methodologies, particularly in addressing the low-resource code-switched domain and the evolving language dynamics. Notably, it investigates the effectiveness of integrating word-level annotated colloquial English data alongside German in fine-tuning XLM-R with token classification to enhance profanity detection performance. Additionally, zero-shot domain transfer is employed to comparatively evaluate the performance of the monolingual and the bilingual fine-tuned models on previously unseen data, specifically low-resource code-switched German rap lyrics. Through experimentation and comparative analysis, the thesis showcases significant performance improvements in profanity detection with the bilingual fine-tuned model outperforming the monolingual one across various metrics. In relative comparison to the monolingual fine-tuned model, the bilingual one exhibited approximately 14.82% higher recall, 17.55% higher precision, and 16.35% higher F1 score in the zero-shot domain transfer setting. Additionally, the thesis highlights the bilingual fine-tuned model's superior ability to recognize profanities in both German and English, as well as its effectiveness in detecting neologisms, a crucial capability given the constantly evolving nature of natural languages. This research contributes to advancing profanity detection methodologies while addressing critical challenges in cross-lingual and code-switched contexts. Additionally, it presents a unique word-level annotated dataset, providing a valuable resource for further research in this domain. **WARNING: This thesis contains offensive and profane language.**

Zusammenfassung

Die wachsende Anzahl von Social-Media-Nutzer*innen und die zunehmende Verwendung von beleidigender Sprache und Schimpfwörtern machen es immer wichtiger, wirksame automatische Erkennungssysteme zu entwickeln. Diese Masterarbeit untersucht die Herausforderungen und Fortschritte in der automatischen Erkennung von Schimpfwörtern, insbesondere im ressourcenarmen, codegemischten Bereich. Hierfür wird die Effektivität der Integration von englischen umgangssprachlichen und auf Wortebene annotierten Trainingsdaten zusätzlich zu deutschsprachigen Daten für das Fine-Tuning von XLM-R zur Verbesserung der Schimpfworterkennung ermittelt. Darüber hinaus werden die Ergebnisse beider Modelle mittels Zero-Shot-Domain-Transfer in einer neuen Domäne, nämlich codegeswitchten deutschen Raptexten, verglichen. Im Vergleich zum monolingual finegetunten Modell erzielt das bilingual finegetunte Modell eine um 14,82% höhere Trefferquote, eine um 17,55% höhere Genauigkeit und einen um 16,35% höheren F1-Score. Das bilingual finegetunte Modell erkennt darüber hinaus sowohl mehr englischsprachige als auch mehr deutschsprachige Schimpfwörter. Diese Masterarbeit trägt somit zur Weiterentwicklung der Methoden in der multilingualen Schimpfworterkennung bei. Darüber hinaus wurde im Zuge dieser Masterarbeit ein einzigartiger, auf Wortebene annotierter Datensatz erstellt, der eine wertvolle Ressource für weitere Forschungen in diesem Bereich darstellt. **WARNUNG: Diese Masterarbeit enthält anzügliche und beleidigende Sprache.**

Contents

List of Tables

List of Figures

1 Introduction

With more than 3.6 billion social media users worldwide (Kanchan and Gaidhane, 2023, p. 1) and about 360k user-generated postings per minute on X (Domo, Inc., 2023), formerly Twitter, social media plays an important role in our daily lives. As the social media user base expands, cyberviolence has become increasingly prevalent, characterized by the extensive use of offensive language and vulgarity on social media platforms, often facilitated by the anonymity provided by the internet (Wang et al., 2020, p. 1448). Considering the impact of offensive language and online hate speech on our society, there has been a noteworthy surge of interest within the Natural Language Processing (NLP) community regarding the detection of hate speech (Poletto et al., 2021, p. 478) and a particular emphasis has been placed on investigating and developing robust and efficient automatic hate speech detection models using Deep Learning (DL) methodologies, such as Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) (Bachfischer et al., 2018; Jonathan and Setiawan, 2023), and state-of-the-art (SOTA) Pre-Trained Language Models (PLMs) (Mukherjee and Das, 2023; Wang et al., 2020).

Despite recent promising advancements in automatic hate speech detection, the task remains largely unsolved primarily because of the numerous challenges associated with this task (Hahn et al., 2021, p. 6). Typically treated as a supervised text classification task, the detection of hate speech entails training Machine Learning (ML) systems on manually labeled datasets (Ataei et al., 2023, p. 2787). Consequently, advancement in hate speech detection heavily relies on the availability of annotated data. However, the dynamic evolution of new hate speech domains and their associated expressions, influenced by real-world occurrences, such as worldwide crises, makes creating exhaustive datasets for every language and domain unfeasible (Montariol et al., 2022, p. 347).

In addition, current approaches in hate speech detection mainly concentrate on document level analysis, such as employing either a binary coarse-grained text classification method, where models are trained on annotated postings and label the entire post as either offensive or non-offensive (Zampieri et al., 2019b, p. 76), or a more fine-grained multi-class text classification approach, distinguishing postings between different types of hate speech categories, such as homophobia, misogyny, or xenophobia (Chakravarthi et al., 2023). This focus on evaluating entire posts results in the creation of datasets commonly annotated either at the sentence or document level, with limited availability of datasets that specifically focus on

offensive vocabulary and include annotations at the word level.

Furthermore, languages are inherently dynamic; they evolve over time and are subject to constant change (Aitchison, 2013, p. 3), as underscored by the increasing use of code-switching, i.e., "the use of two or more linguistic varieties in the same conversation or interaction" (Scotton and Ury, 1977, p. 5), on social media platforms (Barman et al., 2014; Mewengkang and Fansury, 2021). Notwithstanding the heightened interest in research on multilingual hate speech detection and the emergence of hate speech datasets encompassing languages beyond English in recent years (Poletto et al., 2021, p. 480), datasets featuring code-mixed and code-switched content remain notably limited (Jose et al., 2020, p. 138). This scarcity may be attributed to the significant variations in perceptions of hate speech across diverse languages and cultures. What might be perceived as informal yet non-offensive discourse in one language could be considered offensive or even hateful in another (Montariol et al., 2022, p. 347). These linguistic and cultural differences in perception make the task of detecting hate speech within code-mixed content particularly challenging.

Another significant challenge arises from the fundamental complexity of defining the concept of hate speech, worsened by the widespread ambiguity surrounding the use of related expressions, such as offensive, abusive, and vulgar language, which frequently intersect and are subject to highly subjective interpretations (Poletto et al., 2021, p. 479). Despite this definitional vagueness, the importance of profanity and vulgar expressions for hate speech detection is undeniable, as they are commonly utilized as a specialized lexicon for classifying hate speech. Such lexicons typically incorporate words indicative of targeted classes. In the context of identifying abusive language, these lexicons feature explicitly offensive expressions, such as *cunt*, *idiot*, and *nigger*, which are used during the classification process to assess the presence of any of them within a comment, determining whether it constitutes hate speech or not (Wiegand et al., 2018a, p. 21).

Examples of hate speech detection tools using predetermined lists of offensive words encompass Python libraries, such as profanity[1] and better-profanity[2]. These tools solely identify posts containing words that exactly match those in the respective lists as instances of hate speech. However, this approach fails to accommodate the evolving nature of language since it struggles to accurately identify posts containing derogatory language not included in these hard-coded lists, resulting in limited accuracy. PLMs, that are based on the Transformer architecture (Vaswani et al., 2017), such as Cross-Lingual Language Model-RoBERTa (XLM-R) (Conneau et al., 2020) and Bidirectional Encoder Representations from Transformers (BERT)

[1] `https://github.com/ben174/profanity/blob/master/profanity/data/wordlist.txt` [Accessed: 09.05.2024]

[2] `https://github.com/snguyenthanh/better_profanity/blob/master/better_profanity/profanity_wordlist.txt` [Accessed: 09.05.2024]

(Devlin et al., 2019), offer a significant improvement over these hard-coded lists due to their semantic representations of text sequences leading to SOTA performance across a range of sequence classification tasks (Mukherjee and Das, 2023, p. 278).

1.1 Research Questions and Assumptions

The influence of the English language on German has been steadily growing since the end of World War II, which is particularly evident in colloquial language, such as in popular music and youth culture (Mair, 2018, p. 57). Given this linguistic landscape, it is reasonable to anticipate the presence of English vulgarities within colloquial German language data. Consequently, augmenting German word-level annotated training data, covering single-word as well as multi-word profanities, with English colloquial data and conducting bilingual fine-tuning is anticipated to yield superior offensive word detection performance.

Furthermore, in low-resource settings, where code-mixed and code-switched training data is scarce (Jose et al., 2020, p. 138), employing zero-shot transfer techniques shows potential for enhancing the detection of hate speech (Castillo-López et al., 2023; Plaza-del-arco et al., 2023). Hip hop music is characterized by extensive utilization of code-switching, exemplified by renowned Austrian and German hip hop artists in their stylistic expression and lyrical creations (Mair, 2018, p. 61f.). Moreover, rap music, characterized by its foundational elements of beats (rhythmic pulses) and bars (time segments defined by beats) (Berry, 2018, p. 1f.), has a history of pushing cultural and linguistic boundaries through spoken rhymes, including numerous bad words. This exploration is particularly evident in gangsta rap since the late 1980s and early 1990s, which confronts controversial topics, such as misogyny, violence, and homophobia (Littlejohn and Putnam, 2010, p. 120). As a result, German rap offers an ideal source for evaluating the efficacy of zero-shot domain transfer in detecting profanities in a previously unseen code-switched domain, by assessing the abilities of a model fine-tuned exclusively on German colloquial training data against one fine-tuned on both German and English data.

These considerations lead to two research questions. Firstly, to what extent does incorporating colloquial English training data, such as social media posts, alongside colloquial German training data during fine-tuning enhance the efficiency of a multilingual PLM in a token classification task for detecting profane expressions in either language? Secondly, to what extent do the colloquial style, cultural and linguistic diversity, as well as the heavy use of English vulgar vocabulary affect the zero-shot performance when contrasting the monolingual and bilingual Fine-Tuned Models (FTMs) in a previously unseen domain, specifically code-switched German rap lyrics? The research questions are based on the assumption that fine-tuning at the word level and in two languages will improve the performance of a multilingual

PLM in detecting profane expressions in both languages in the domain used for fine-tuning as well as on a code-switched dataset from a different domain.

1.2 Motivation and Objectives

Identifying explicit profanity frequently hinges on the presence of specific keywords (Waseem et al., 2017, p. 81). Thus, collecting and manually annotating offensive content along with these keywords is pivotal for detecting hate speech (Nozza and Hovy, 2023, p. 3897). However, the process of manual annotation can be time-consuming and may possibly induce symptoms similar to those of post-traumatic stress disorder in human annotators (Wang et al., 2020, p. 1448). Furthermore, the differences in perceptions of hate speech across diverse languages (Montariol et al., 2022, p. 347) along with the lack of code-switched and code-mixed datasets (Jose et al., 2020, p. 138) exacerbates the challenge of progressing in the field of cross-lingual hate speech detection.

To address these issues, this thesis seeks to explore how integrating colloquial English training data, along with German data, can improve the effectiveness of a PLM in identifying vulgar expressions, while consciously avoiding the use of code-switched data during fine-tuning. Subsequently, the monolingual German-only and bilingual German-English FTMs will undergo zero-shot domain transfer to compare their performances in a novel domain characterized by heavy use of profanities and English words. The objective is to determine whether enhancing German data with readily available English data leads to improved profanity detection results when confronted with heavily code-switched cross-lingual content. This experiment addresses not only the challenge of limited code-mixed data, but also assesses the models' capability to recognize unfamiliar offensive expressions by using the contextual patterns they acquired during fine-tuning, drawing from the manually labeled vulgarities in the training datasets.

To achieve this, both models have undergone extensive fine-tuning using manually word-level annotated datasets in colloquial German and English language, consisting of sentences taken from various social media sources. To the best of the author's knowledge, such datasets do not yet exist and can thus be considered unique. Therefore, in addition to contributing new insights in profanity detection in the low-resource code-switched domain, this thesis also presents another significant contribution in the form of the developed datasets. With German being among the ten most used languages on X (Hong et al., 2011, p. 519) and in light of the growing necessity to moderate user-generated toxic content, which poses a widespread and costly challenge for companies facilitating user interaction (Risch et al., 2018, p. 39), the contributions of this thesis are significant for the NLP research community as well as for the industry at large.

2 Theoretical Background

This master's thesis explores several core concepts of NLP, specifically focusing on hate speech and profanity detection, language contact phenomena, such as code-switching, and the cultural and linguistic aspects of German hip hop. Additionally, it examines fundamental ML learning paradigms, alongside popular DL technologies, such as Artificial Neural Networks (ANNs), SOTA PLMs and the Transformer architecture. Furthermore, it discusses techniques, such as pre-training and various forms of transfer learning, with a specific focus on fine-tuning and zero-shot domain transfer. The chapter concludes by examining classification in NLP, particularly discussing token classification and classification evaluation metrics.

Section 2.1 provides an overview of swearing, hate speech and profanity, focusing on the growing prevalence of profanities across social media, followed by an attempt to define hate speech, differentiating it from related concepts, and highlighting the importance of profanities in hate speech detection. Section 2.2 explores language contact phenomena and language change with an emphasis on code-switching and highlights the various different functions and types of code-switching. Section 2.2.4 then differentiates code-switching from its related concepts, namely borrowing, code-mixing, and neologisms, and examines the relevance of code-switching and neologisms for NLP in general and profanity detection in particular.

Section 2.3 further investigates profanity and code-switching within the context of German hip hop. It offers a brief linguistic and cultural analysis of German rap lyrics, introduces the concept of English as the Hip Hop Nation Language (HHNL) and emphasizes the linguistic diversity, cultural nuances, and prevalent use of code-switching and swear words within this domain. Section 2.4 and Section 2.5 provide the ML fundamental basis required for the comprehensive understanding of PLMs, starting by presenting four main ML learning paradigms followed by a brief overview of ANNs and the Transformer architecture, the underlying framework of most recent PLMs.

Section 2.6 provides an overview of various different types of PLMs, distinguishing them based on their pre-training objectives and whether they operate monolingually or multilingually. Section 2.7 investigates transfer learning, including an overview of different types of transfer learning, namely fine-tuning and zero-shot domain transfer. Section 2.8 concludes with an introduction to classification in NLP, focusing on token classification, word-level dataset labeling, and evaluation methods for classification.

2.1 Swearing, Offensive Language and Hate Speech

Swearing refers to the use of taboo or offensive language to express one's emotional state (Pamungkas et al., 2020, p. 6237). Despite acknowledging the presence of profanity, especially in informal conversations, scholars in the fields of psychology and linguistics had neglected to incorporate vulgarity and swearing into their language theories for more than a century. This stems from swearing being considered too taboo for scholarly exploration. The resulting dearth of research perpetuates the taboo and results in language theories that disregard such sensitive topics (Jay, 2000, p. 10). However, their inclusion in a theory of language is necessary as the use of offensive language serves as means through which feelings and emotions can be expressed effectively, surpassing the limitations of conventional language (2000, p. 243). Additionally, swear words are used for emotional or connotative purposes and provide essential linguistic information about emotions, thus enhancing comprehension (2000, p. 11). Despite its potential to offend, swearing also serves various other purposes, such as bantering, joking, or engaging in sexual talk. Socioculturally, it is influenced by cultural norms and can thus be a means for bonding with others (2000, p. 244).

The emotional turn in large parts of the human and social sciences, including linguistics, has resulted in a growing interest in the emotional aspects of language (Schwarz-Friesel, 2013, p. 15), thus paving the way for fresh perspectives on offensive language, emphasizing swearing as a multifaceted social phenomenon with intricate pragmatic functions instead of deeming it inappropriate, taboo, or illicit (Beers Fägersten, 2012, p. 20). With the increasing scholarly focus on the pragmatics of offensive language, it is noteworthy to mention that swear words significantly contribute to our everyday language usage, representing approximately 0.5% to 0.7% of the words in our daily conversations (Mehl and Pennebaker, 2003; Wang et al., 2014).

With social media platforms, microblogging applications and chat forums providing people with the opportunity to instantly and extensively express and share their opinions and thoughts online (Jahan and Oussalah, 2023), the use of offensive language extends beyond traditional offline face-to-face conversations nowadays. Swearing is frequently encountered in online discussions spanning various languages and platforms, with social media platforms, such as X standing out for their casual atmosphere and spontaneous content (Pamungkas et al., 2020, p. 6237). Additionally, Wang et al. (2014, p. 418ff.) observed that on English X swear words are used at a rate of approximately 1.15%, which is about double the rate observed in daily offline conversations. Moreover, 7.73% of all the postings in their dataset contained at least one swear word and a surprisingly low number of only seven swear words, i.e., *shit*, *fuck*, *ass*, *nigga*, *bitch*, *whore*, and *hell* accounted for more than 90% of all swear word occurrences in their dataset.

This heavy use of swear words on social media, coupled with the desire to understand the pragmatics of swearing in naturally occurring conversations, and the easy access to large amounts of colloquial user-generated content, makes social media an interesting source for investigations into swearing and offensive language from an NLP perspective and has piqued the interest of computer scientists, who are increasingly exploring ways to explicitly model offensive language in downstream NLP tasks (Cachola et al., 2018, p. 2927). While a consensus on the precise definitions of hate speech and the related concepts remains elusive within the NLP community (Wiedemann et al., 2018, p. 86), the term hate speech serves as the primary and most commonly employed expression for this phenomenon (Schmidt and Wiegand, 2017, p. 1). Consequently, any studies related to the automatic identification of abusive or hurtful content are mostly referred to under the umbrella term of hate speech detection (Jahan and Oussalah, 2023; Schmidt and Wiegand, 2017).

2.1.1 Hate Speech Definition

Hate speech is is a complicated phenomenon and as such it presents challenges in its recognition for humans and machines. One significant challenge stems from the difficulty in precisely defining hate speech, compounded by the ambiguity surrounding related concepts, such as abusive, aggressive, offensive, or toxic language. These related fields frequently intersect and are susceptible to subjective interpretations (Poletto et al., 2021, p. 478f.). Consequently, even though the concept of hate speech is frequently mentioned in legal and policy contexts, a uniform definition for hate speech and its related concepts remains elusive (Assimakopoulos et al., 2017, p. 3), which often results in a varying nomenclature for identical linguistic phenomena, or alternatively, the application of identical labels to distinct phenomena (Poletto et al., 2021, p. 481).

Schmidt and Wiegand (2017, p. 1) define hate speech as "a broad umbrella term for numerous kinds of insulting user-created content". Additionally, the term also serves as the primary and most commonly employed expression for this phenomenon. Teh and Cheng (2020, p. 228) expand this broad definition of hate speech and include the aspect of hostility towards individuals or groups, usually based on certain characteristics, such as gender identity, race, or sexual orientation. Silva et al. (2016, p. 689) identified nine main characteristics and categorized them into distinct hate categories, with the addition of one extra category, namely Other, to encompass anything not captured by the main categories. Table 1 illustrates the hate categories as well as some corresponding examples of hate targets.

Table 1: The distinct hate categories based on specific characteristics (Silva et al., 2016, p. 689)

Categories	**Example of hate targets**
Race	nigga, black people, white people
Behavior	insecure people, sensitive people
Physical	obese people, beautiful people
Sexual orientation	gay people, straight people
Class	ghetto people, rich people
Gender	pregnant people, cunt, sexist people
Ethnicity	chinese people, indian people, paki
Disability	retard, bipolar people
Religion	religious people, jewish people
Other	drunk people, shallow people

Fortuna and Nunes (2018) provide an even more extensive definition, based on four dimensions, i.e., specific targets, inciting violence or hate, attacking or diminishing language, and the status of humor, and conclude that, in addition to targeting specific individuals or groups based on certain characteristics, hate speech also aims to incite violence or hate towards these minorities, involves attacking or diminishing language and can even occur in a humorous linguistic environment.

Waseem et al. (2017, p. 79) present a more fine-grained dual-layer typology that, on the one hand, distinguishes between whether the abuse is generalized or is targeted towards a specific individual or group and, on the other hand, whether the abuse is explicit or implicit. Explicit hate speech uses clear and direct derogatory or harmful language, such as homophobic or sexist remarks, or overtly offensive language. In contrast, implicit hate speech does not use derogatory expressions but still conveys discriminatory or prejudiced attitudes often through subtle forms of discrimination, insinuations, or stereotypes that imply negative sentiments.

Directed hate speech targets a specific individual, group, or entity and involves language aimed at that particular person, organization, or identifiable entity. In contrast, generalized hate speech does not target a specific individual, but makes derogatory or discriminatory remarks about entire groups. Actions, such as cyberbullying or trolling are considered directed hate speech, whereas generalized examples include hate speech targeting racial categories or sexual orientations without mentioning specific individuals. Table 2 provides examples, taken from the Social and Language Technologies (SALT) Implicit Hate Speech Dataset (ElSherief et al., 2021), that illustrate the differences between explicit and implicit, as well as the differentiation between directed and generalized hate speech, according to the dual-layer typology (Waseem et al., 2017).

Table 2: Examples of explicit, implicit, directed, and generalized hate speech

	Directed	**Generalized**
Explicit	lol you aren't being wiped off the planet! you're a paranoid racist white supremacist!	#jews & #niggers destroy & pervert everything they touch #jewfail #niggerfail
Explicit	you should be put behind bars you stupid devil - you influenced breivik to kill hundreds of innocents - you terrorist devil	so it targets faggots and blacks looks like god is trying to help out the white race.
Implicit	anti-white jew david sirota blocked me when i asked him why israel wasn't taking in any refugees lol	a certain 1400 years old religious ideology has proven to be the biggest killing machine of innocent people in human history
Implicit	zuckerberg is an open borders zealot and only cares about cheap labor to enrich himself.	both north america and europe belong to us. white men conquered and colonized both

The use of derogatory expressions, such as *racist*, *white supremacist*, *stupid devil*, *niggers* and *faggots* as well as the accusation of influencing Breivik to kill are considered explicit expressions of hate speech. In contrast, the implicit hate speech does not use explicitly offensive words and is based on subtle implications instead. It manifests through the insinuation of negative stereotypes or biases against individuals based on their identities, such as the implication that David Sirota's Jewish identity is relevant to the discussion, coupled with the suggestion that he is anti-white, which carries implicit anti-Semitic sentiments. Additionally, the use of *lol* suggests a dismissive attitude toward the topic of refugees, which could imply a lack of empathy for those affected by the refugee crisis.

The expression *open borders zealot* suggests a negative portrayal of Mark Zuckerberg, implying that his support for open borders is extreme or unreasonable and the assertion that he only cares about cheap labor to enrich himself could be interpreted as an attack on his character and motivations, which may contain implicit biases or prejudices. Lastly, the instances of directed hate speech focus on specific individuals, notably those directly addressed using the word *you*, as well as David Sirota and Mark Zuckerberg.

The definitional consistency within the NLP community diminishes even further when attempting to contextualize hate speech alongside its related concepts, and delineating the disparities between them (Srba et al., 2021, p. 320). Due to the ambiguous distinctions between hate speech and its associated ideas, hate speech is often mislabeled or used inter-changeably with abusive language (Nobata et al.,

2016, p. 147) or harmful speech (Faris et al., 2016, p. 5). Zampieri et al. (2019a, p. 1415), on the other hand, consider hate speech a component of offensive language, alongside other concepts, such as abusive language or (cyber)-bullying.

2.1.2 Profanity

Despite the "Definitional Quagmire" (Faris et al., 2016, p. 5) of hate speech and its related fields, a noteworthy concept to mention in this context is profanity (Jahan and Oussalah, 2023). Profanity is acknowledged as closely linked, yet not synonymous with hate speech, as hate speech often involves profane expressions (Teh and Cheng, 2020, p. 228). It is defined as socially offensive language, commonly known as bad, vulgar, or inappropriate language (Teh et al., 2018, p. 66), which includes swearing and curse words, and is frequently observed among younger people (Wiegand et al., 2018, p. 2). Moreover, profanity can also constitute abusive language (Founta et al., 2018, p. 495; Jahan and Oussalah, 2023; Nobata et al., 2016, p. 147) and serves as an expression of abusive behavior (Kaur et al., 2021, p. 247). Wiegand et al. (2018b, p. 1046) present three instances of offensive sentences with the related profane words, i.e., *dumbass*, *stupid*, *bimbos*, and *scum*, highlighted in bold.

1. stop editing this, you **dumbass**.
2. Just want to slap the **stupid** out of these **bimbos**!!!
3. Go lick a pig you arab muslim piece of **scum**.

The dynamic nature of language (Aitchison, 2013, p. 3) leads to new abusive words continuously finding their way into everyday language. One example is the expression *twunt*, which results from combining the two profanities *twat* and *cunt*. Another instance is the newly created word *gimboid*, which originates from the British TV series Red Dwarf and characterizes an incompetent individual. The word likely stems from the word *gimp*, with the addition of the suffix *-oid*. A more recent creation is *remoaner*, referring to someone who is dissatisfied with the outcome of the 2016 EU referendum in the United Kingdom (UK) that led to Brexit. The expression is considered pejorative and consists of the words *moan* and *remainer*. These examples highlight that creating profanity lexicons is a dynamic process that requires continuous effort (Wiegand et al., 2018b, p. 1046f.).

As interest in abusive language detection continues to rise within the NLP community, the significance of profanities and swear words in tasks associated with hate speech detection has become increasingly evident (Pamungkas et al.,

2023, p. 160). Given that hate speech often manifests through the use of profane vocabulary (Teh and Cheng, 2020, p. 228), profanity has been extensively employed in tasks related to detecting hate speech (2020, p. 230), with the use of profanities often serving three main purposes: create lexicons, categorize hate speech, and identify the hate targets.

Firstly, profane words are used as a resource for hate speech dictionaries and lexicons (Wiegand et al., 2018, p. 5), which can be used to classify offensive data (Siegel and Meyer, 2018, p. 17). Although the utility of lexicons is limited to explicit hate speech detection, lexicon-based classification methods have demonstrated high reliability in cross-domain hate speech detection, attributed to their diminished susceptibility to overfitting (Wiegand et al., 2018b, p. 1054).

Secondly, using such abusive lexicons as features enables the classification of profane words into one or more distinct hate categories. For instance, the expression *fuck* may be associated with the category of sexual orientation, *idiot* with disability, and *racist* with behavior. This classification facilitates a deeper comprehension of profanity and its usage within specific hate contexts, particularly across various demographic groups, such as men and women (Wong et al., 2020, p. 4ff.).

Lastly, profanities can be used for deducing hate targets. Understanding the hate categories and the contexts in which profane language occurs allows for the identification of the corresponding hate targets. Expressions, such as *I hate*, or *I am so sick of*, when combined with the template *<one word> people*, aid in the detection of potential hate targets, which could be *Mexican people*, *black people*, or any other combination of *<term> people* (Silva et al., 2016, p. 688f.). Moreover, profane keywords can help differentiate whether the hate target is a group of people or an individual (Zampieri et al., 2019a, p. 1416).

However, the mere utilization of profane words does not automatically result in hate speech (Madukwe and Gao, 2019, p. 345). In fact, the use of profanity can serve different communicative functions. These functions vary depending on their contexts, spanning from causing offense, to intensifying an emotion, or merely serving as an expression of a certain degree of informality in a conversation (Cachola et al., 2018, p. 2927).

The fact that they can be used for positive sentiments to convey emphasis (Wiegand et al., 2018, p. 2), as well as for negative sentiments to offend, places them on opposing sides of the emotional spectrum. Furthermore, their usage is influenced by respective cultural backgrounds and demographics, making them a particularly challenging field of research within the NLP domain (Holgate et al., 2018, p. 4405). Table 3 shows example postings that illustrate six different communicative functions of the word *ass*.

Table 3: Overview of six different communicative function of the swear word *ass* (Holgate et al., 2018, p. 4405)

Function	**Tweet**
Express aggression	<USER> You are an **ass** Your industry is full of assholes and you do nothing to improve (...)
Express emotion	There are so many things I want to do, But investing in equipment is a pain in the **ass**
Emphasise	today is a good **ass** day <URL>
Auxiliary	Wish <USER> could save my **ass** on these exams like he used to
Signal Group Identity	Now this is a group of **ass** kickers!
Non-vulgar	Kick **Ass** 2 - Red Band Trailer <URL>

While in the first example the word *ass* expresses aggression towards another individual through verbal abuse, the same word can also be used in non-abusive environments, such as to express emotions, as illustrated in the second example, to emphasize emotions as shown in the third example, and to strengthen group affiliation, as exemplified in the second to last posting (Holgate et al., 2018, p. 4405).

Given the importance of profane words for hate speech detection (Siegel and Meyer, 2018), improving the understanding of vulgar language and its contextual nuances can aid hate speech detection models to identify and correctly categorize hate speech, especially when profanity is employed, making the task even more complex (Cachola et al., 2018, p. 2928). Consequently, specifically addressing the usage of profanity is anticipated to enhance the effectiveness of practical endeavors, such as the implementation of profanity filters (Holgate et al., 2018, p. 4405).

2.2 Code-Switching

Code-switching is the act of shifting between two or more languages or different language varieties, referred to as codes, within the same conversation or even within the same sentence (Auer, 1998, p. 1). Linguists adopted the term code from the communication technology domain, in which code-switching refers to "a mechanism for the unambiguous transduction of signals between systems" (Gardner-Chloros, 2009, p. 11), and use the term analogous to spoken and written signal transfer between two different linguistic systems, i.e., languages or languages varieties. In linguistics, code nowadays serves as umbrella term that covers languages, styles, dialects and registers and replaced the traditional term variety (2009, p. 11). The switched elements span from single words to multi-word expressions and can even include entire sentences (Riehl, 2014, p. 24). Hofweber et al. (2020, p. 910) present

two examples of German-English code-switching with the switched elements *because I'm ill* and *Schuhwerk* highlighted in bold.

(1) Ich kann heute nicht kommen **BECAUSE I'M ILL**.
(I cannot come today because I'm ill.)

(2) We didn't bring **SCHUHWERK** for hiking.
(We did not bring shoes for hiking.)

Code-switching occurs in nearly all contact situations and is common in diasporas, among minority communities, as well as in indigenous multilingual groups (Gardner-Chloros, 2020, p. 188). The phenomenon has gained significant scientific interest over the past few decades, evolving from a niche topic to a subject of importance across various linguistic disciplines (Auer, 1998, p. 1).

Even though code-switching is sometimes mistakenly viewed as a compensation strategy for language gaps, as suggested by the use of negatively connoted expressions, such as Spanglish or ingleñol for this phenomenon, linguists believe that code-switching actually requires a high level of multilingual proficiency (Müller et al., 2015, p. 11). It demonstrates two main characteristics supporting this assumption. Firstly, it features a seamless transition between languages, devoid of interruptions, such as hesitations or long pauses, evident in the smooth flow between preceding and following elements surrounding the switched item. Secondly, it is characterized by a clear lack of awareness regarding the change, as evidenced by the absence of any repetition of the preceding part, nor is it echoed in the subsequent part. Consequently, code-switching demands a strong bi- or multilingual skill set, which involves understanding and correctly using the shared grammar repertoire of both languages (Poplack, 1980, p. 601). Moreover, code-switching is considered an excellent example of a language contact phenomenon, which helps to better understand the unique structural characteristics that arise from language contact situations (Müller et al., 2015, p. 11).

2.2.1 Language Contact and Language Change

The prerequisite for language contact is the existence of a particular contact situation, which initially often occurred due to cultural, demographic, economic, and political reasons (Oksaar, 1988, p. 204; Sankoff, 2002, p. 640). Thus, language contact is not restricted to linguistic aspects, but rather constitutes a combination of linguistic and extra-linguistic factors (Haugen, 1956, p. 39). Moreover, language contact is omnipresent in all areas of human interaction and there are hardly any languages whose speakers did not have contact situations with other languages. This makes the phenomenon of language contact typical rather than rare (Thomason, 2001, p. 10).

According to Thomason (2001, p. 1) language contact refers so "the use of more than one language in the same place at the same time". Trudgill (2003, p. 74) solidifies this superficial and trivial definition by extending it to social contact situations between groups or individuals, who lack a shared native language. However, language contact is not restricted to a bilingual or multilingual environment, it can also occur intralingual, i.e., between different varieties of a single language. The phenomenon of language contact can thus be defined as a contact situation in which "two or more different languages, varieties, or even just people from different linguistic backgrounds interact with each other across time and space" (Coronel-Molina and Samuelson, 2017, p. 379).

Language contact is not a short-term process and even though communication challenges between these individuals or groups may arise initially, they can potentially lead to long-term language influence and change as a consequence of bilingualism among certain speakers (Trudgill, 2003, p. 74). These linguistic transformations and the fusion of languages, which often occurs between geographically proximate ad-stratum languages, reflect the prolonged interactions spanning numerous years and stem from diasporic movements and historical migrations over centuries (Földes, 2010, p. 136f.). For instance, English is renowned for assimilating a substantial number of loanwords, with certain estimates indicating that as much as 75% of its vocabulary originates from French and Latin (Thomason, 2001, p. 10). In summary, using more than one language, whether by an individual or a group of people, within the same conversation, leads to a dynamic interaction between the involved languages, resulting in different language contact phenomena, such as code-mixing, code-switching, or borrowing (Coronel-Molina and Samuelson, 2017, p. 379).

2.2.2 Types of Code-Switching

Code-switching can occur at any linguistic level (Poplack, 2015, p. 918). From a grammatical perspective, Poplack (1980, p. 605) identifies three categories of code-switching, based on the location of the switching in the sentence: (i) inter-sentential switching, (ii) intra-sentential switching, and (iii) extra-sentential, also referred to as emblematic or tag-switching. Figure 1 graphically illustrates these three different categories and degrees of code-switching.

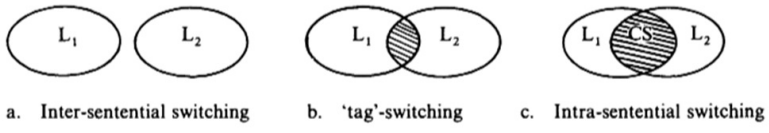

Figure 1: Overview of the different degrees of code-switching (Poplack, 1980, p. 615)

L1 and L2 refer to sentences in the respective languages and the gray area marks the degree of contact. Inter-sentential switching describes the act of switching between several utterances within a conversation, i.e., a change in transition from one sentence to the next without language mixing on sentence level and without direct integration of linguistic elements of L2 in a sentence of L1 (Riehl, 2014, p. 33). Conversely, tag-switching involves inserting only a tag or an interjection from L2 into a speech predominantly in L1 (Koban, 2013, p. 1175). Intra-sentential switching pertains to changes occurring within the boundaries of a single sentence, including a higher degree of language combination than tag-switching (Riehl, 2014, p. 33).

In the case of inter-sentential switching, a sentence in one language or language variety is syntactically complete without being bound to the previous sentence in a different language. Consequently, its prosodic features, including rhythm, pitch, and intonation, remain unique and uninfluenced by the meaning or structure of the preceding sentence, thus ensuring clear linguistic separation between the two sentences (Manfredi et al., 2015, p. 294). Muysken (2000, p. 1ff.) defines intra-sentential switching as code-mixing and further divides it into three distinct patterns, namely alternation, insertion and congruent lexicalization, which is also referred to as dense code-switching. Table 4 presents multiple examples of the three distinct categories of intra-sentential code-switching and the corresponding English-only expressions.

Table 4: German-English intra-sentential code-switching examples (Hofweber et al., 2020, p. 910)

Code-switching type	**Example**
(1) Alternation	Ich kann heute nicht kommen BECAUSE I'M ILL. I can today not come BECAUSE I'M ILL. I cannot come today BECAUSE I'M ILL.
(2) Insertion E > G	Wir suchen noch VOLUNTEERS fuer das Projekt. We search still VOLUNTEERS for the project. We are still looking for VOLUNTEERS for the project.
(3) Insertion G > E	We didn't bring SCHUHWERK for hiking. We didn't bring SHOES for hiking. We didn't bring SHOES for hiking.
(4) Dense	Wir haben FRIENDS gemacht mit'm SHOP OWNER. We have FRIENDS made with th' SHOP OWNER. We have made FRIENDS with th' SHOP OWNER.

Alternation (1) involves toggling between two languages with minimal connection, such as a shift from the primary statement to a supporting clause that is not directly connected to another segment of the sentence. Insertion (2, 3), on the other hand, is asymmetric, indicating that elements from one language, i.e., the embedded language (L2), get incorporated into the structure of another, the dominant language, also referred to as matrix language (L1). While this interplay can happen in both directions, the dominant language typically aligns with the bilingual person's primary language. Congruent lexicalization (4) predominantly occurs between closely related languages. The involved languages exhibit profound grammatical and lexical intertwining and speakers select words and structures based on contextual appropriateness rather than adhering strictly to the rules of a single language. Switching is more frequent than in alternation but lacks clearly traceable transition points. Unlike insertion, identifying a dominant language is challenging, as speakers integrate rules from both languages (Hofweber et al., 2020, p. 910).

Tag-switching is frequently observed among individuals who are not fully fluent in both languages. These individuals adeptly transition between languages, maintaining grammatical accuracy in each by incorporating particular words or expressions, known as tags, from one language (L2) into their speech in the other language (L1) (Poplack, 1980, p. 613). Tags are flexible elements in a sentence that can be placed almost anywhere without strict syntactic constraints or breaking any grammar rules (1980, p. 589). This distinguishes them from intra-sentential switches, which must conform to the syntactic rules of the internal structure of the sentence. Tags can be single words or multi-word expressions and include colloquialisms, such as *no way!*, fillers such as *I mean*, and interjections such as *shit!* (1980, p. 596). Poplack (1980, p. 605) places these three types of code-switching along a continuum based on the level of language proficiency needed for each, underscoring the varying degrees of linguistic proficiency essential for the diverse types of code-switching. Tag-switching, which necessitates only a minimal grasp of grammatical understanding in L2, is at the foundational level. Progressing along the continuum, the intermediate level, inter-sentential switching requires a deeper comprehension of the linguistic structures of the second language. The pinnacle of this continuum is marked by intra-sentential switches, where bilingual speakers adeptly integrate multiple languages within a single sentence. Such advanced switching requires an intricate understanding of the grammatical structures of each language and their interplay to maintain syntactic integrity.

2.2.3 Functions of Code-Switching

Code-switching can serve various purposes. Aligned with Jakobson's (1960) delineation of the six functions of communication, Appel and Muysken (2006) introduce

a functional model for code-switching that mirrors Jakobson's (1960) communication model. This alignment underscores the premise that bilingual individuals may assign distinct functions to different languages within their bilingual environment and conversations. Thus, the choice of language in bilingual discourse may signify the prioritization of specific communicative functions at any given moment. The functional code-switching model is divided into six functions, namely the expressive, referential, phatic, directive, poetic, and metalinguistic function (Appel and Muysken, 2006, p. 118ff.).

The referential function involves the use of switching between languages to express concepts more effectively, especially when discussing certain topics or when specific words from one language are more semantically appropriate. Bilingual speakers often switch languages consciously to better convey ideas or because they lack proficiency in one language on a particular subject. This particular kind of switching is common in contexts such as immigrant radio or television broadcasts, where words from the majority language are adopted to describe concepts unique to the society of the host country. It is also observed in discussions on technical subjects in many languages of the Third World (Appel and Muysken, 2006, p. 118f.). Additionally, instances of referential language switching tend to increase when speakers experience distractions, fatigue, or emotional distress (Crystal, 2010, p. 365).

The directive function involves directing communication towards specific individuals by either excluding certain people or including others through language choice. This form of code-switching can be seen in examples, such as parents speaking a foreign language to prevent their children from understanding. However, overusing this strategy may lead to unintended consequences, such as children learning the second language or creating a new language to exclude their parents (Appel and Muysken, 2006, p. 119). Among speakers of minority languages, it often serves as a means to highlight their shared cultural background and to express solidarity within the community, which ultimately leads to more acceptance (Crystal, 2010, p. 365).

The expressive function involves expressing a blended identity by utilizing two languages within a single conversation (Appel and Muysken, 2006, p. 119). Poplack (1980) illustrated this function by examining the interchange between Spanish and English within the Puerto Rican community and found out that proficient bilingual Puerto Ricans in New York develop a unique speech pattern when they engage in conversations rich with code-switching, wherein individual switches lose their specific intended purposes within discourse. Moreover, code-switching can serve as a method to demonstrate solidarity with a minority group, thereby fostering greater acceptance within that community, or to convey a specific attitude, such as irritation, friendliness or distance, towards a speaker (Crystal, 2010, p. 365).

The phatic function includes using code-switching within a conversation to signal a shift in its tone, thus serving a primarily social function. This phenomenon can be illustrated by the example of a stand-up comedian delivering their routine predominantly in a standard dialect but switching to a vernacular, such as an urban dialect, for the punchline (Appel and Muysken, 2006, p. 119).

The metalinguistic function involves using various languages or linguistic codes to directly or indirectly comment on language. Essentially, it involves switching between languages to highlight linguistic abilities or to convey a message about language. This phenomenon is commonly observed in public settings, such as performances, circus acts, or sales pitches (2006, p. 120).

The poetic function goes beyond mere information transmission, highlighting creativity and expression. It involves skillfully employing bilingualism, weaving puns and jokes to stir emotions and craft aesthetic experiences. However, despite these general approaches to define the functions of code-switching, it is important to note that the functions of code-switching may vary significantly across different communities. For instance, Puerto Ricans in New York may engage in code-switching for reasons distinct from those of Flemish speakers in Brussels (2006, p. 120).

2.2.4 Delimiting Code-Switching

In the context of definitional considerations, one recurring topic in linguistics is "that efforts to distinguish codeswitching, codemixing and borrowing are doomed" (Eastman, 1992, p. 1). Consequently, distinguishing code-switching from its related language mixture phenomena proves challenging, on the one hand, due to a lack of consensus within the discipline, and, on the other hand, due to methodological inconsistencies among researchers in this field (Poplack, 2015, p. 923).

Code-mixing Similar to code-switching, code-mixing, in its most trivial definition, can be described as "the use of two or more languages that are mixed in one utterance" (Perlina and Agustinah, 2022, p. 1). Consequently both code-switching and code-mixing involve integrating at least two languages or dialects within a single conversation. Pfaff (1979, p. 295) introduces the word mixing as a neutral umbrella term that encompasses both borrowing and code-switching.

Meisel (1994, p. 414f.) argues that code-mixing does not strictly adhere to the grammatical and discourse rules, but instead, often occurs when an expression is momentarily unavailable in one language but easily accessible in another. In contrast, code-switching is considered a skill used by bilingual speakers to choose the appropriate language based on the content of the conversation, the person they are speaking to, or the situation. This involves shifting between languages within a

conversation while adhering to sociolinguistic rules and maintaining grammatical accuracy, making it a more systematic and rule-governed process compared to code-mixing.

Similarly, Myers-Scotton (2002, p. 3) claims that code-mixing "suggests a jumbling of a more or less haphazard nature". Due to the negative perception of code-mixing as an unstructured form of language mixing, Saville-Troike (2003, p. 50) strongly opposes using the expression code-mixing synonymous with intra-sentential code-switching. In contrast, as discussed in Subsection 2.2.2, Muysken (2000, p. 1) equates code-mixing with intra-sentential code-switching and advocates to use code-mixing rather than code-switching to refer to instances where words and grammatical rules from two different languages are combined within one sentence. Consequently, akin to insertion in intra-sentential code-switching, insertional code-mixing involves the integration of elements from another language or language variety into a base language. Hamers and Blanc (2000, p. 260) support the idea of using code-mixing interchangeably with intra-sentential code-switching. To sum up, there is disagreement within the academic community regarding whether code-mixing and code-switching refer to the same phenomenon or represent distinct concepts.

The rap data examined in this thesis predominantly uses German as base language, incorporating English vocabulary to different extents within the sentences. Moreover, the analysis focuses on a sentence-by-sentence approach, specifically addressing intra-sentential code-switching. Therefore, the definition of code-mixing provided by Muysken (2000) and Hamers and Blanc (2000) is highly pertinent to this thesis, as they equate code-mixing with intra-sentential code-switching. As a result, the term code-switching is used in this thesis to signify both code-switching and code-mixing.

Borrowing The differentiation between code-switching and borrowing in bilingual conversations remains a contentious topic in academic discussions. Consequently, what Poplack and Meechan (1998, p. 127) described as "the heart of a fundamental disagreement among researchers", continues to hold true in current efforts to scientifically distinguish between code-switching and borrowing (Deuchar, 2020) and numerous criteria have been suggested to differentiate between these two concepts (Boztepe, 2003, p. 4).

Unlike code-switching, which can include single words, multi-word expressions as well as whole sentences (Riehl, 2014, p. 24), borrowing pertains solely to the adoption of individual words or brief, fixed idiomatic phrases from one language or language variation into another (Gumperz, 1982, p. 66). However, while longer foreign language segments are more easily identified as examples of code-switching, categorizing elements as borrowing merely because they consist of single words

is insufficient (Deuchar, 2020). As a result, distinguishing between borrowing and code-switching becomes more challenging the shorter the foreign language expressions are (Poplack, 1988, p. 220).

To facilitate a more precise distinction between code-switching and borrowing, an additional criterion, namely the frequency of the foreign language elements, has been investigated. Salmons (1990, p. 465) concludes, that items that appear frequently are rather categorized as borrowings, whereas less frequently occurring items are more likely to be considered examples of code-switching. Myers-Scotton (1997, p. 207), on the other hand, argues, that this criterion has its limitations, as determining the frequency of a word can be challenging and often leads to arbitrary conclusions. To address the limitations of the two previously discussed differentiation criteria, more robust criteria have been examined.

In contrast to code-switching, borrowing involves integrating linguistic elements from one language or system into the linguistic framework of another (Gumperz, 1982, p. 66; Matras and Adamou, 2020, p. 237). More specifically, it involves phonological and morphological adaptations to the sound system and grammar of another language. For instance, in the sentence *I am going to Los Angeles*, Saville-Troike (2003, p. 53) differentiates between borrowing and code-switching based on the pronunciation of *Los Angeles*. When pronounced in an anglicized manner, indicating phonological integration, the expression is considered borrowed from Spanish. In contrast, maintaining the original Spanish pronunciation indicates code-switching.

It follows then, that incorporating elements from a foreign language into the linguistic, phonological and morphological structure and lexicon of the borrowing language often masks their foreign origin. Consequently, while code-switching requires some level of familiarity with both languages (Coronel-Molina and Samuelson, 2017, p. 2), monolingual speakers unaware of the source language can use borrowed elements seamlessly (Lipski, 2005, p. 13). Additionally, due to their integration into the lexicon, borrowed words are commonly utilized not only by individuals but also the broader community.

Lastly, Poplack (1980, p. 585f.) argues that, in contrast to borrowing, code-switching is not possible within a word. The principle of the free morpheme constraint suggests that it is acceptable to switch languages during a conversation as long as non-independent elements of words, such as prefixes or suffixes, are not mixed. In summary, linguists have examined a range of criteria, from weak to more robust, to distinguish between borrowing and code-switching. Despite this exploration, reaching a consensus on a definitive distinction between these two concepts has proven elusive, with the boundary often appearing ambiguous.

Neologisms Natural languages are dynamic and undergo constant evolution and variation. The German vocabulary reflects this dynamic nature, with new words emerging and gaining popularity while others fade into obsolescence. For instance, words such as *Oheim* [uncle] and *Gemach* [chamber] are now considered archaic and are primarily found in literary texts (Putterer, 2019, p. 192). The term neologism traces its roots back to Ancient Greek. It is a fusion of néo- meaning new and lógos which translates to word (Khan, 2013, p. 819). Consequently, it essentially pertains to the introduction or adoption of new lexical items within a language framework.

During the emergence and acceptance of a neologism, the majority of language users views it as something new and distinctive. As it gains traction, this novel linguistic expression becomes widely adopted and recognized as a standard part of the language. This process of adoption and acceptance marks the lifecycle of a neologism within the evolving dynamics of language. In the fields of lexicology and lexicography, the term neologism gained wide-spread recognition only around the mid-20th century, which is relatively late compared to other linguistic terms (Herberg et al., 2012, p. XIf.).

Neologisms can be categorized into two main types, namely new lexical units and new meanings. New lexical units encompass both single-word and multi-word expressions, that were not part of the lexicon until a certain point in time. When no new lexical units are formed, but instead established words take on added meanings, these new meanings expand upon or alter the existing semantic scope of the word within the language. There is no inherent distinction between newly formed lexical units within a language and those borrowed entirely from other languages (2012, p. XI).

A distinguishing feature of neologisms lies in their emergence as newly coined words, terms, or meanings that arise during a specific period of language evolution within a community of speakers (2012, p. XIf.). As such, they serve as a means to satisfy the growing demand for new vocabulary that usually accompanies such developments (Kananaj and Rushiti, 2024, p. 3). Among the various different word formation processes[3], compounding and blending stand out as notable techniques for creating new vocabulary at specific socially or politically important points in time, that shape people's experiences and perspectives, such as wars and pandemics (Al-Salman and Haider, 2021, p. 34f.).

Compounding and blending both involve the creation of a new word by combining two independent words. However, they differ in their approaches to word formation. Compounding, which frequently occurs in German an English, involves merging entire words to form a new, composite expression, while blending entails a truncation process, in which elements from the beginning and end of two words are combined

[3]This thesis only focuses on the word formation processes most relevant to the thesis. A comprehensive overview of all word formation processes can be found in (Yule, 2023)

to create a new word. In the case of fingerprint and good-looking, both words remain intact to form a new word. Conversely, smog and brunch are combined by taking the beginning of the first word and the ending of the second word (Yule, 2023, p. 64).

The Russia-Ukraine war (Aleksandruk et al., 2023; Kramar and Ilchenko, 2023), the COVID-19 pandemic (Kananaj and Rushiti, 2024; Klosa-Kückelhaus and Kernerman, 2022), the UK's exit from the EU (Lalic-Krstin and Silaski, 2018), and numerous refugee crises and migration movements (Heselhaus, 2022; Putterer, 2019; Šinjori, 2019) can be considered examples of such specific periods in language evolution that often lead to the emergence of neologisms (Kramar and Ilchenko, 2023, p. 15). With compounding and blending being recognized as the most productive and hence the most widely used word formation methods (Al-Salman and Haider, 2021, p. 34f.), they are frequently employed to formulate words aimed at describing emergent concepts or phenomena. Table 5 illustrates multilingual neologisms that evolved from global social and political events.

Table 5: Neologisms that evolved from social and political events

Event	**Neologisms**	
COVID-19	Kovidiot (Kananaj and Rushiti, 2024)	Impfdrängler (Klosa-Kückelhaus, 2022)
Russia-Ukraine war	Ліліпутін (Liliputin) (Aleksandruk et al., 2023)	Рашизм (Ruscism) (Kramar and Ilchenko, 2023)
UK's exit from EU	Brexit (Lalic-Krstin and Silaski, 2018)	bregret (Lalic-Krstin and Silaski, 2018)

Kovidiot (Covidiot), a blending of the words COVID and idiot, is a derogatory label aimed at individuals who exhibited inappropriate behavior during the COVID-19 pandemic, such as disregarding medical advice (Mihaljević et al., 2022, p. 167) or needlessly stockpiling essentials, thereby contributing to shortages (Neologismenwörterbuch, 2006ff.).

Impfdrängler (vaccination tailgater), a German neologism following the principles of German compound word formation, i.e., impf[en] (to vaccinate) and Drängler (tailgater), are individuals who sought to receive vaccinations ahead of their scheduled turn during the initial phase of vaccine distribution when doses were limited and specific groups were prioritized for medical reasons (Klosa-Kückelhaus, 2022, p. 34f.).

Ліліпутін (Liliputin), a blending of two individual words, is a derogatory reference specifically emphasizing Putin's height (Aleksandruk et al., 2023, p. 198). Lilliputian, commonly employed in a humorous context as a synonym for small,

derives its origin from the inhabitants of the fictional island of Lilliput (Merriam-Webster, 2024) in Jonathan Swift's novel Gulliver's Travels.

Рашизм (Ruscism), a blending of the expressions Russian and fascism, signifies "the expansionist, ultranationalist ideology of the Russian Federation that reached its apogee in the invasion of Ukraine" (Kramar and Ilchenko, 2023, p. 22). Although the expression surged in popularity following the Russian invasion of Ukraine on February 24, 2022, its roots extend back to the First Chechen War. First introduced in 1996 by Dzhokhar Dudayev, the first President of the Chechen Republic of Ichkeria, the word русизм (russism) was intended to represent "an anti-human ideology grounded on the Russian chauvinistic worldview and the world domination complex" (Kramar and Ilchenko, 2023, p. 22).

Brexit is frequently perceived as originating from a blending of either *Britain* and exit or British and exit, with the latter interpretation being more widely recognized. The term describes the UK's exit from the EU following a referendum conducted in 2016 (Lalic-Krstin and Silaski, 2018, p. 3ff.).

Bregret is a blending of the noun Brexit and the verb regret and, alongside other creations, such as bremain, has served as a foundational element for various suffix additions, resulting in the emergence of neologisms such as bregretter or bremainer (2018, p. 5f.).

These examples demonstrate that numerous expressions evolving from political and social events encompass derogatory or insulting connotations, e.g., *Covidiot*, *Impfdrängler*, *Liliputin*. Moreover, the rapid evolution of neologisms is evident as they increasingly saturate the internet and social media platforms (Corazza et al., 2018). When examining politically contentious subjects, such as refugee crises, it is not uncommon to encounter not just one newly coined term resulting from the fusion of two words, but rather an influx of comparable negative neologisms. For instance, expressions, such as *rapefugee*, *rapeugee*, and *rapugee* emerge, all stemming from the combination of rape and refugee. These neologisms are frequently utilized by critics of asylum-seeker-friendly policies as disparaging propaganda expressions (Würschinger et al., 2016, p. 35). Consequently, language alteration phenomena such as code-switching and (profane) neologisms must be recognized as crucial topics for hate speech detection research in the ML community. This recognition ideally results in the development of robust algorithms that push the boundaries of profanity detection, enabling precise identification of newly coined profane or insulting expressions without relying on manual inclusion in predefined lists of offensive language.

2.3 Hip Hop and German Rap

Rapper Kendrick Lamar's groundbreaking Pulitzer Prize for Music win in 2018 led to a notable shift in the public recognition of rap lyrics, resulting in increased attention to modern hip hop music songwriting. This trend was also evident in Germany in 2018 when Capital Bra, a rapper from the Russian-speaking minority, achieved unprecedented success with eight number-one hits in just one year, solidifying his status as the most popular musician of the 21st century in Germany (Tikhonov, 2020, p. 55). After having been marginalized in scholarly circles for a long time, hip hop has nowadays transitioned into an important subject of scholarly discourse and academic scrutiny. Its cultural impact and global reach provide rich material for sociologists, anthropologists, and linguists to explore and analyze (Terkourafi, 2010, p. 1f.).

Following the acknowledgment of hip hop as a subject of academic inquiry, it is necessary to delineate the nuanced relationship between rap and hip hop, as well as define two important aspects of rap, namely beats and bars. Rap, defined as "the rhythmic delivery of spoken rhymes" (Terkourafi, 2010, p. 14), constitutes one of the four central pillars of hip hop culture, alongside DJing, breakdancing, also referred to as street dance, and graffiti art (Dumitru and Tudor, 2022, p. 228; Yapondjian, 2005). Therefore, rap serves as the vocal component of hip hop music and is predominantly delivered by a Master of Ceremonies (MC), hence its alternative name, MCing. In contrast, hip hop extends beyond the mere vocal aspect, encompassing lifestyle elements, such as behavior, speech, appearance, and communication (Dumitru and Tudor, 2022, p. 228). Although theoretically distinct, in practice, these two concepts are often used interchangeably, primarily due to the central role of rap in hip hop culture (Terkourafi, 2010, p. 14).

Beats and bars are crucial aspects of rap, with the former often cited as the primary reason people enjoy it. The word beat carries two distinct meanings. Firstly, it refers to the musical track that provides the backdrop for a rapper's lyrics. Secondly, it denotes the regularly recurring pulse or rhythm in the music, often heard through lower-pitched instruments, such as the bass or kick drum. A bar is a time segment defined by beats, establishing it as a fundamental unit of musical time that aligns with a piece's structure. In this example, the hook, also referred to as chorus, consists of eight bars, which is four lines of text repeated once (Berry, 2018, p. 1ff.).

(3) 1 You better - lose yourself in the music, the moment
2 You own it, you better never let it go (go)
3 You only get one shot, do not miss your chance to blow
4 This opportunity comes once in a lifetime

5 You better - lose yourself in the music, the moment
6 You own it, you better never let it go (go)
7 You only get one shot, do not miss your chance to blow
8 This opportunity comes once in a lifetime... you better (Eminem, 2002)

2.3.1 Linguistic and Cultural Influences

Hip hop music originated in the early 1970s in the Bronx, New York City, as a unique cultural and musical expression. It emerged in multicultural neighborhoods that faced challenging economic conditions and were socially segregated from the predominantly white, middle-class mainstream America (Forman, 2002, p. 87). American soldiers stationed in Germany during the late 1970s and early 1980s introduced the hip hop culture to parts of Western Germany through dance clubs, and the American Forces Network (AFN) radio station, which was aimed at the United States (U.S.) military community. AFN broadcast hip hop music, making it accessible to a broader German audience (Munderloh, 2017, p. 192).

However, it was predominantly U.S. American hip hop films that accelerated the popularity of hip hop music among the German public. Influential movies such as Charlie Ahearn's Wild Style (1983) and Stan Lathan's Beat Street (1984) played a crucial role. These movies showcased the culture, music, and lifestyle associated with hip hop, sparking interest and enthusiasm among German viewers. Additionally, the presence of diverse ethnic backgrounds of the U.S. rappers resonated with the German multicultural audience and led to a shared feeling of recognition, promoting a sense of belonging and identification with the portrayed rappers (2017, p. 192f.).

Germany boasts established minority languages, such as Romani and Sorbian, along with a diverse array of migrant languages, such as Turkish, Russian, Greek, and Italian. With multilingualism playing a significant role in the sociolinguistic landscape of hip hop, young people of migrant heritage have significantly contributed to the evolution of German hip hop in the past two decades. Immigration and ethnicity are frequently explored topics in German rap lyrics, mirroring the diverse cultural and linguistic backgrounds within the German hip hop community (Androutsopoulos, 2010, p. 19).

This linguistic and cultural diversity is also reflected in the broad linguistic variety of German rappers, with Capital Bra using 14 different languages, and Olexesh integrating 11 languages into their lyrics (Tikhonov, 2020, p. 58). Table 6 provides an overview of the linguistic diversity of German-speaking rappers, exemplified by Capital Bra, who incorporates German, Russian, and Turkish within just four bars.

Table 6: Overview of the different languages used by Capital Bra (Tikhonov, 2020, p. 58)

Line	**Original**	**English translation & [original language]**
Capital Bra "Was 2, hol 10", album "Makarov Komplex" (2017)		
CBI	Bras, die für gut Geld gut auf dich aufpassen,	Bros [RUS], getting good money for looking good after you [GER].
CBII	Aber ich bin Ukrainer, я убью за брата,	But I am Ukrainian [GER], I will kill for my brother [RUS].
CBIII	Но тихо едешь, дальше будешь, говорит мне мама,	Slowly but surely, tells me my mother [RUS].
CBIV	Trotzdem brauch' ich Para, brechen aus, Räubeleiter	But I need [GER] money [TK], break out, crossed ladder [GER].

The languages are incorporated to varying degrees throughout the song ranging from entire bars spoken solely in one language, such as in line CBIII, to bars split between two languages, such as in line CBII, or even in form of intra-sentential code-switching where single words are integrated into sentences entirely in another language, such as in lines CBI and CBIV (2020, p. 58f.).

Consequently, despite the dominance of the German language, there is a notable incorporation of migrant languages into German rap lyrics. Moreover, the use of English alongside migrant languages and German is highly common, reflecting the multilingual trend, with frequent occurrences of German-English intra-sentential code-switching (Androutsopoulos, 2010, p. 37f.).

2.3.2 The Hip Hop Nation Language: English

The English language is an essential aspect of hip hop as it is considered "the original Hip Hop Nation Language (HHNL)" (Androutsopoulos, 2010, p. 19). Moreover, as a universal language used across the globe, it holds high sociolinguistic importance and serves as a crucial medium for international, cross-border hip hop communication both in the U.S. and worldwide (2010, p. 19).

The HHNL is a complex linguistic system that goes beyond syntax and includes a wide range of communicative practices, attitudes, and non-verbal elements, such as clothing, gestures, mimics, and graffiti. It has unique rules for grammar, pronunciation, vocabulary, and a distinctive way of communication. It blends elements of spoken language, literature, and musical expression into a cohesive style. Furthermore, the HHNL is widespread across the U.S., has been adapted by various ethnic groups both locally and internationally and is central to the identity of the Hip Hop Nation (HHN) as a whole, but also including regional and

individual variations influenced by personal experiences and sociopolitical contexts. It addresses the communicative needs of HHN members and serves as a medium for expressing resistance and subversion against dominant societal norms. Consequently, the HHNL is deeply intertwined with the socio-political challenges faced by the HHN, such as police brutality, incarceration, and gentrification, reflecting and responding to these circumstances through its evolving discourse (Alim, 2006, p. 71f.). Figure 2 presents example lyrics that convey the fundamental concepts of the HHNL and highlight central issues frequently encountered by members of the HHN, such as police violence and graffiti art.

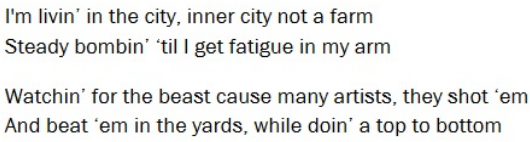

Figure 2: Example lyrics expressing the fundamental concepts of the HHNL (Cotgrove, 2018, p. 70)

In this example, familiar words are assigned new meanings that are specifically tailored to the hip hop community. The word Bombin', which translates to graffiti tagging, serves as an important means for the rapper to express their identity as an artist within the hip hop community. However, this artistic expression is constantly challenged by The Beast, i.e., law enforcement, who frequently target and harm African-American graffiti artists during their top to bottom graffiti spraying. The language used by the artist not only signals authenticity within the hip hop culture, but also acts as a form of communication that remains cryptic to people outside the hip hop community (Cotgrove, 2018, p. 70).

2.3.3 Code-Switching and Profanities in German Rap

While the HHNL may not align with the traditional academic criteria for languages (Bühler, 1934), the abstract concept of English as the HHNL presents intriguing opportunities to further understand German-English code-switching. The significant impact of U.S. American hip hop culture on the emerging German hip hop scene in the early 1980s is particularly evident in the prevalent use of English in the lyrics by the early German rappers, who used English to align with the authentic norms of the American hip hop community (Bower, 2011, p. 377).

Incorporating English words into otherwise German rap lyrics corresponds with the referential and the directive code-switching functions outlined in Subsection 2.2.3. In the context of German rap, the HHNL English serves as the predominant language, through which HHN-specific elements are introduced into the German

lyrics to reference distinct facets of the original U.S. hip hop culture, which reinforces the authenticity of German rappers within this culture. Furthermore, employing distinct HHN words in the lyrics directs the message towards fellow members of the hip hop community, rendering it incomprehensible to those outside the culture. Figure 3 provides an example of incorporating HHNL words into predominantly German rap lyrics to express identification with the original U.S. American hip hop culture.

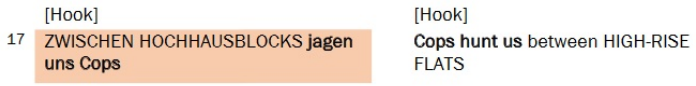

Figure 3: Example German rap lyrics with English elements (Cotgrove, 2018, p. 88)

By referencing the English slang expression *cops* the rapper establishes a connection with the transnational rap community. This connection suggests solidarity with others who have experienced similar struggles with law enforcement and societal oppression, thereby validating the rapper's authenticity within both the original American hip hop community and the hip hop community as a whole (Cotgrove, 2018, p. 89).

Among German-speaking rappers, Money Boy stands out for his distinct approach of seamlessly alternating between German and English. This distinctive blend of languages has gained attention not only from his core fan base but also from a broader audience of young listeners. Money Boy frequently integrates discourse markers such as *yeah*, interjections such as *Joke!* and hip hop slang terms such as *swag* and *fly* into his German lyrics, alongside employing English expressions such as *no way* for added emphasis (Mair, 2018, p. 61). Table 7 presents examples of English code-switched elements in the lyrics of Austrian rapper Money Boy, with the English words highlighted in bold. Additionally, it underscores the prevalence of vulgar and profane expressions in German rap, indicated by the underlines.

Table 7: English code-switched example lyrics of Austrian rapper Money Boy

Track	**Original line**	**Translation**
Deppen die rappen	Mit ein paar scheiß Studenten, wie soll dieses **Life** nur enden (Money Boy, 2023)	With a few shitty students, how is this life supposed to end
Deppen die rappen	Unter 'ner Brücke **maybe** (**yeah**) (Money Boy, 2023)	Under a bridge maybe (yeah)
Dreh den **Swag** auf	**Yeah**, ich bin der **Shit**, **Bitches** (Money Boy, 2010)	Yeah, I am the shit, bitches

The prevalence of profanities in German rap music is likely to be closely linked to two main characteristics of German rap. Firstly, the phenomenon of gangsta rap, emphasizing violence, aggression, and street confrontations, introduced the idea of the ghetto into the German setting, sparking significant discussions about the state of German urban areas and the decline of civic awareness. This cultural importation and thematic focus on brutal or aggressive topics, exemplified by prominent figures, such as Bushido, Fler, and Sido, has deeply influenced the thematic and linguistic landscape of German rap (Bower, 2011). Therefore, it can be reasonably inferred that German gangsta rap prominently features the recurrent use of vulgar and profane language within its lyrics.

Secondly, a fundamental aspect of European rap is the practice of dissing, which is a recurring and widely discussed element within rap and hip hop. It serves as a form of verbal attack intended to belittle or shame an opponent. Typically, it targets other artists, accusing them of lacking authenticity or integrity. However, it can also be directed at individuals outside the hip hop community, such as teachers or politicians (Androutsopoulos and Scholz, 2002, p. 15).

Table 8 presents three distinct diss tracks performed by German and Austrian rap artists, highlighting the frequent co-occurrence of English, emphasized in bold, and German, italicized, vulgarities within each track or even within the same bar.

Table 8: Sample lyrics from three distinct diss tracks by German rappers

Track	**Original line**	**Translation**
Samy De **Bitch**!! (7 Lektionen)	*Depp*, Deine *Scheisse* ist **pussy** und **wack** (Azad, 2001)	Dork, your shit is pussy and wack
Free Spirit	Ich bin in meiner Blüte, **Bitch**, es ist ein Frühlingsfest (Shindy, 2023)	I'm in my prime, bitch, it's a spring festival
Free Spirit	Du *Hurensohn*, schenk mir eine Orchidee (Shindy, 2023)	Son of a bitch, give me an orchid
MC Fetti du Biersäufer	Du bist eine Bitch mit einer Big *Fut* (**Pussy** (Money Boy, 2014)	You are a bitch with a big cunt (pussy)

2.4 Machine Learning Fundamentals

ML can be defined "as a set of methods that can automatically detect patterns in data, and then use the uncovered patterns to predict future data, or to perform

other kinds of decision making under uncertainty" (Murphy, 2012, p. 1). In essence, ML focuses on performance improvement of algorithms through experiential learning, emphasizing the creation of models with minimal human intervention, primarily relying on data for learning and improvement (Morales and Escalante, 2021, p. 112). When examining learning paradigms, three types stand out as particularly important: supervised, unsupervised, and reinforcement learning (Janiesch et al., 2021, p. 687). Additionally, self-supervised learning, another significant paradigm, is especially important for DL and PLMs (Kotei and Thirunavukarasu, 2023).

2.4.1 Supervised Learning

Supervised learning, also referred to as predictive learning, involves algorithms that learn from labeled data, receiving direct feedback during training. It is divided into two main categories: classification and regression (Chauhan et al., 2021, p. 582). Classification entails categorizing data into predefined classes or discrete categories, essentially recognizing patterns within the data. Regression, on the other hand, focuses on approximating a function that describes the relationship between input variables and continuous output values (Salman and Kecman, 2012).

In supervised learning the model is provided with a labeled dataset D which contains input-output pairs $D = \{(x_i, y_i)\}_{i=1}^{N}$, where x_i represents the input features, y_i represents the associated output labels, and N represents the number of training examples. The primary objective is to learn a mapping between the input features and the output labels, facilitating the model in accurately predicting the correct output for novel input instances. In the basic setup, each training input x_i is a D-dimensional vector of features, stored in an $N \times D$ design matrix X. These features can represent simple data, such as height and weight, or more complex data, such as images and sentences. The output variable y_i can be either categorical, such as male/female, for classification problems, real-valued, e.g., income level, for regression problems, or ordinal, e.g., grades, for ordinal regression problems, where the output has a natural order (Murphy, 2012, p. 2).

2.4.2 Unsupervised Learning

Unsupervised learning, also referred to as descriptive learning, involves learning patterns and structures within a dataset without explicit supervision in the form of labeled data. In this ML paradigm, the model is provided only with a dataset consisting of input features, denoted as $D = \{x_i\}_{i=1}^{N}$, where x_i represents the i-th input feature and N is the total number of input features. The goal of the model is to discover intrinsic relationships, meaningful patterns, or clusters within this data, without relying on any predefined labels or outputs (Murphy, 2012, p. 2).

2.4.3 Self-Supervised Learning

Self-supervised learning integrates aspects of both supervised and unsupervised learning paradigms (Rani et al., 2023, p. 2765) and has become increasingly popular due to the vast amount of available unlabeled data (Ohri and Kumar, 2021). In self-supervised learning the model is trained using segments of the input data to predict other segments, a process commonly referred to as predictive or pretext learning. This approach enables self-supervised learning algorithms to derive the supervision signal directly from the data itself, allowing labels to be generated from unlabeled data. As a result, this method supports supervised learning without the need for manually labeled datasets (Rani et al., 2023, p. 2765).

The main goals of self-supervised learning are to achieve the performance of supervised DL models without extensive labeled datasets, to learn meaningful and generalized representations that enhance downstream tasks, to utilize large, freely available datasets through self-supervised pre-training, and to adopt a practical, human-like learning approach. Self-supervised learning methods are effectively and successfully employed in widely used PLMs for NLP, such as BERT, Robustly optimized BERT approach (RoBERTa) (Liu et al., 2019), and XLM-R (Ohri and Kumar, 2021).

2.4.4 Reinforcement Learning

Reinforcement learning is an ML paradigm in which a system learns to accomplish a particular objective through interaction with its surroundings, rather than relying on predefined examples for learning. The core idea involves the system receiving information about its current state and taking actions based on a set of allowable choices. This approach is akin to a reward-punishment training mechanism, where the system learns through experimentation and adapts based on the received results, striving to maximize cumulative rewards over time. Consequently, this approach relies on the principle of reward maximization to guide the learning process (Janiesch et al., 2021, p. 687; Kaelbling et al., 1996).

2.5 Artificial Neural Networks

While there are differences in how biological and artificial neurons process information, the term neural originates from the neurons found in the human brain. Neurons gather input signals through their dendrites from neighboring neurons. When sufficiently stimulated, a neuron transmits a signal along its axon to multiple terminals, which then communicate with other neurons' dendrites. Artificial neurons mimic this impulse transmission, yet unlike natural ones, they require training rather than maturing autonomously (Koehn, 2020, p. 30f.). An ANN consists of

multiple interacting neurons organized in layers. These layers are connected by weights (Rückstieß, 2016, p. 18), and through the process of training, the network learns to adjust these weights to model complex patterns (Kelleher, 2019, p. 67). According to their structure, ANNs are categorized into single-layer models, also referred to as shallow networks, and multi-layer neural networks, commonly referred to as Deep Neural Networks (DNNs) (Montufar et al., 2014). Within single-layer networks, the input undergoes direct mapping to an output using a modified linear function. Conversely, in multi-layer ANNs, one or more hidden layers exist between the input and output layers (Aggarwal, 2018, p. 4).

2.5.1 Feed-Forward Neural Network (FFNN)

In a FFNN, data flows linearly in one direction from the input layer, traversing through the hidden layers before reaching the output layer. A neuron processes inputs in two stages. Initially, it computes a weighted sum of the inputs, which is subsequently transformed by a non-linear activation function to yield the neuron's final output. Various activation functions, ranging from simple to complex, can be employed. This is crucial for introducing non-linearity into the model, enabling it to learn and represent more complex patterns. This process repeats until the output layer produces a result, which is subsequently refined through backpropagation (Kelleher, 2019, p. 70ff.).

Figure 4 illustrates the structure of a multi-layered FFNN, consisting of an input layer, an output layer, and three hidden layers. The circles in the figure represent neurons that process information within a network. The arrows in the figure show the direction of information flow between neurons, with each connection only allowing one-way communication. These connections have associated weights, which influence how the neurons process incoming information (2019, p. 70).

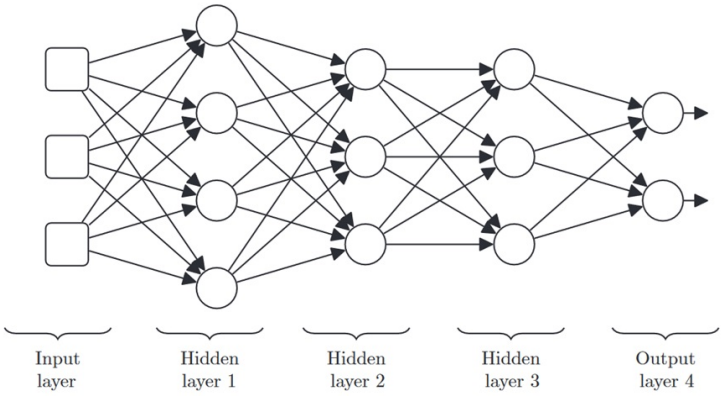

Figure 4: Illustration of an FFNN with three hidden layers (Kelleher, 2019, p. 68)

Backpropagation, a fundamental component of NNs, is a supervised learning algorithm crucial for iteratively adjusting weights to minimize errors, without which neural networks would not effectively learn. It utilizes the chain rule of calculus to efficiently compute error gradients. This approach comprises two main phases: a forward phase where model predictions are compared to gold standard outputs, and a backward phase, where errors propagate through the layers, facilitating weight adjustments. This iterative interplay of forward and backward propagation serves as the foundation of training neural networks, enabling them to learn and enhance performance progressively (Aggarwal, 2018, p. 21). FFNNs typically are not designed to directly manage sequential data due to their inability to consider the order of the sequence and interdependencies among its elements. Nevertheless, techniques exist to adapt them for sequential data tasks. One such approach involves employing a fixed-size sliding window, which segments the sequential data into smaller windows, treating each one as a distinct input. While this strategy can capture local patterns effectively, it may encounter difficulties with long-range dependencies unless the windows are sufficiently large, potentially leading to increased model complexity (Jurafsky and Martin, 2023, p. 157f.).

2.5.2 Recurrent Neural Network (RNN)

Unlike FFNNs, which propagate information strictly in a forward direction across layers and encounter difficulties with sequential data processing, RNNs introduce cycles, enabling a form of memory. This feedback mechanism creates an internal recursive structure, allowing information from previous time steps to be incorporated into current computations. In RNNs, each layer corresponds to a specific position in a sequence, allowing the network to dynamically adjust its layers based on the temporal context. By utilizing a shared set of parameters across these layers, RNNs ensure consistent modeling at each timestamp, maintaining a fixed parameter count while effectively processing sequences through their inherent memory capability (Aggarwal, 2018, p. 273).

However, during the training of RNNs using backpropagation, the vanishing gradient problem (Bengio et al., 1994) may occur, constraining the network's ability to effectively learn sequences of extended length (Rückstieß, 2016, p. 108). In DNNs, the vanishing gradient problem arises as gradients in earlier layers diminish significantly during training, hindering effective learning. This challenge originates from certain activation functions that inherently yield small derivatives, causing gradients to decay exponentially as backpropagation traverses through layers. Consequently, this imbalance results in minimal updates being applied to earlier layers compared to later layers. While solutions, such as using activation functions with larger gradients or adjusting weight initialization can alleviate this issue, they might results in gradient explosion and achieving the optimal balance remains a

difficult task (Aggarwal, 2018, p. 130f.). To address the vanishing gradient problem, RNNs were enhanced with a gating mechanism. The prevailing gating setups typically involve Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) and Gated Recurrent Unit (GRU) (Cho et al., 2014).

Long Short-Term Memory The gating mechanism embedded within LSTM networks offers a robust solution to the vanishing gradient problem pervasive in traditional RNNs. By incorporating specialized memory cells with input, output, and forget gates, they can selectively update, retain, and propagate information across sequential data. The input gate facilitates the incorporation of relevant information into the memory cell, while the forget gate selectively discards outdated or irrelevant data, enabling the retention of long-term dependencies crucial for sequential learning tasks. Additionally, the output gate regulates the flow of information, ensuring that only pertinent data is propagated forward. These gating mechanisms not only facilitate the learning of long-term dependencies but also regulate the gradient flow during backpropagation (Jurafsky and Martin, 2023, p. 201f.).

Gated Recurrent Unit GRU is another NN architecture designed to handle sequential data, providing a simplified alternative to the LSTM. GRU employs only two gates to manage information flow. The update gate combines input and forget gates, regulating the amount of the previous hidden state passed to the current hidden state. The reset gate determines the extent of past information to be forgotten. GRU networks update their hidden state by partially resetting and updating the previous time step's hidden states through these gates. This results in fewer parameters compared to LSTM, making them simpler and potentially more efficient for certain tasks. Their performances are similar, but GRU networks are easier to implement and can generalize better with less data. Nonetheless, LSTM are often preferred for handling longer sequences and larger datasets due to their extensive testing and established popularity (Aggarwal, 2018, p. 295ff.).

2.5.3 Encoder-Decoder Model

The encoder-decoder model, also referred to as Sequence-to-Sequence (Seq2Seq) model, is a framework employed for tasks that require processing an input sequence into an output sequence of different lengths and structures, such as in Machine Translation (MT), where the length and structure of sentences can vary significantly between languages (Jurafsky and Martin, 2023, p. 204). The encoder processes the input sequence and converts it into a fixed-sized contextual representation, encapsulating the essential information of the sequence. This context is a compact,

comprehensive representation of the input that captures its meaning and nuances. Following this, the decoder network takes this contextual representation and generates the output sequence. The task of the decoder is to produce a sequence that is not only coherent but also contextually appropriate, which often involves generating outputs of varying lengths and structures compared to the input (Forcada, 2017, p. 297f.).

2.5.4 Attention Mechanisms

It quickly became apparent that as the input length increased, performance decreased. Consequently, the encoder-decoder architecture underwent enhancement through the integration of an attention mechanism (Bahdanau et al., 2015) shortly after its introduction (Forcada, 2017, p. 299). Attention mechanisms are crucial components in improving the capabilities of NNs, particularly in tasks, such as MT and document classification (Kardakis et al., 2021). The attention mechanism allows the decoder to access information from all hidden states of the encoder, not just the last one. Instead of relying solely on the final hidden state of the encoder, attention dynamically computes a context vector for each decoding step. This is achieved by calculating a weighted sum of all encoder hidden states, with the weights emphasizing relevant parts of the source text for each token being decoded. This dynamic context vector, generated anew for each decoding step, provides richer information to the decoder, improving its ability to generate accurate outputs (Forcada, 2017, p. 299; Jurafsky and Martin, 2023, p. 208). Nevertheless, even though the attention mechanism pioneered by Bahdanau et al. (2015) stands as one of the earliest and most widely recognized, there exist several other types of attention mechanisms (Kardakis et al., 2021).

2.5.5 The Transformer Architecture

The Transformers architecture significantly influenced the field of Artificial Intelligence (AI), transforming how models are trained and marking a substantial advancement over earlier neural architectures (Guimarães et al., 2024), such as LSTM. Transformers are highly effective NNs, specifically well-suited for sequence-to-sequence tasks, such as MT. They rely exclusively on attention mechanisms without any recurrence or convolution, which eliminates the need for a fixed-length representation of the source sentence. Instead of using a single constant context vector to encode the entire source sentence, the model utilizes attention mechanisms to dynamically focus on specific parts of the source sentence that are relevant for generating the next token (Stahlberg, 2020, p. 349). Transformers are characterized by an encoder-decoder structure. Both the encoder and decoder utilize stacked

layers of self-attention and fully connected feedforward layers, enabling the model to capture dependencies within the sequence regardless of distance and build complex representations (Vaswani et al., 2017, p. 5999f.).

In the originally proposed version (Vaswani et al., 2017), the encoder is composed of six identical layers, each featuring a multi-head self-attention mechanism and a fully connected FFNN that processes individual positions. Additionally, residual connections and layer normalization are added to enhance the model's performance. Similarly, the decoder consists of six layers but includes an additional sub-layer for multi-head attention over the encoder's output and employs masking in the self-attention sub-layer to ensure predictions at each position depend only on preceding outputs (2017, p. 5999f.). Figure 5 provides a detailed visual representation of the Transformer architecture, showcasing the encoder positioned on the left-hand side and the decoder situated on the right.

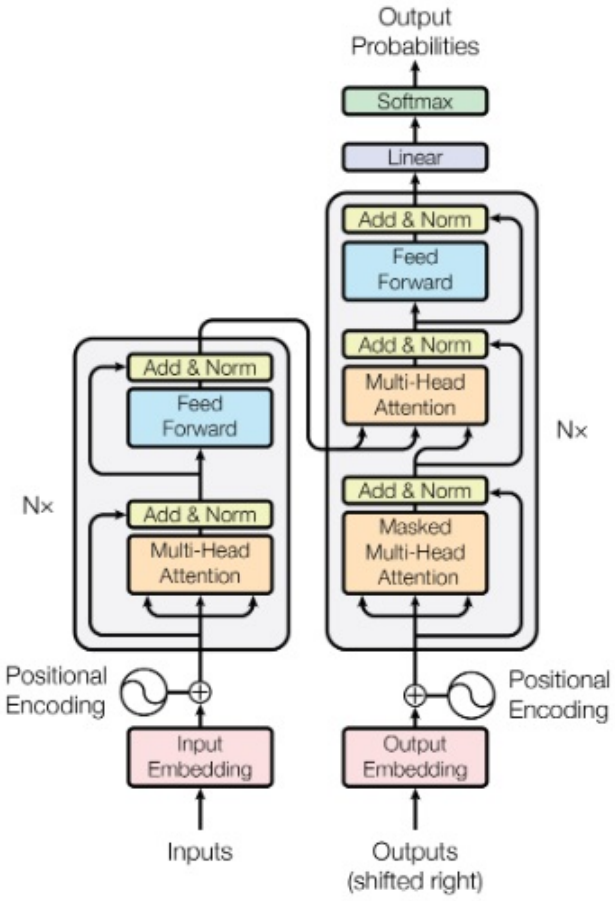

Figure 5: The Transformer architecture (Vaswani et al., 2017, p. 6000)

Attention mechanisms are methods that facilitate the mapping of queries, represented as vectors aimed at identifying pertinent information, to a set of key-value pairs. This process computes an output by calculating a weighted sum of the values, where a key signifies an item or piece of information used as a reference, and a value corresponds to the actual data or content linked with the key. The weights are determined by a compatibility function between the query and the corresponding key. Scaled Dot-Product Attention stands out as a specific form of attention mechanism widely adopted, mainly for its computational efficiency (Vaswani et al., 2017, p. 6000f.). It operates by computing dot products between queries Q and keys K, dividing them by the square root of the query's dimension d_k, and applying a softmax function to obtain weights on the values V as illustrated in Equation 1. These computations are commonly executed simultaneously on matrices comprising queries, keys, and values (Vaswani et al., 2017, p. 6000).

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

Multi-Head Attention improves the attention mechanisms. Instead of using a single set of queries, keys, and values, it employs multiple sets that are linearly projected and processed in parallel. This allows the model to attend to different aspects of the input simultaneously, improving its ability to capture complex patterns (Stahlberg, 2020, p. 351; Vaswani et al., 2017, p. 6000).

Positional encodings serve as a compensatory mechanism for the absence of recurrence and convolution, providing a method to recognize the sequence order. These encodings are incorporated into both the encoder and decoder stacks, maintaining consistent dimensionality with the embeddings and enabling them to be combined through summation (Vaswani et al., 2017, p. 6000).

2.6 Pre-Trained Language Models

PLMs have revolutionized NLP, shifting the focus from traditional supervised learning with labeled data to a two-step process involving self-supervised pre-training and subsequent fine-tuning (Min et al., 2024, p. 2; Paaß and Giesselbach, 2023, p. 384; Wang et al., 2024, p. 16). During the pre-training stage the model learns general language representations through self-supervised learning, for which large amounts of unlabeled training data are available. During the fine-tuning stage the pre-trained model can then be further adapted to various different downstream NLP tasks through fine-tuning or few-shot learning. This breakthrough has ignited a significant surge of research within the NLP community, with numerous researchers dedicating their efforts to further advance and improve these models (Wang et al., 2023, p. 51), consequently leading to the widespread application of PLMs across

diverse interdisciplinary NLP tasks, such as MT (Laskar et al., 2023; Weng et al., 2020), Named Entity Recognition (NER) (Chen et al., 2021; Naseem et al., 2021), question answering (Hu et al., 2023; Yoon et al., 2020), and text classification (Yadav and Kaushik, 2023; Zhao et al., 2021). The recent and ongoing success of PLMs can be attributed to innovations in architecture, refined fine-tuning techniques, as well as industry interest in providing computational resources for training particularly large PLMs.

Before the development of Transformers, RNNs were widely used for various NLP tasks. LSTM was the foundational architecture of language models such as Embeddings from Language Models (ELMo) (Peters et al., 2018), enabling the creation of contextualized word embeddings. However, with the introduction of Transformers, the attention mechanism replaced the resource-intensive recurrence mechanism of LSTM and convolution of CNNs, leading to significant improvements in efficiency and performance for various NLP tasks (Vaswani et al., 2017, p. 5999).

2.6.1 Pre-Training

Even though most SOTA PLMs, such as BERT and Generative Pre-Trained Transformer (GPT) are based on the Transformer architecture (Wang et al., 2024, p. 7), they differ in their pre-training language modeling objectives. Auto-regressive language modeling is unidirectional and focuses on predicting the next word given the preceding context, typically employed in decoder-only models, such as GPT. Masked Language Modeling (MLM) can either be unidirectional or, as seen in models, such as BERT, bidirectional. It includes masking a set of tokens and predicting the tokens adjacent to those that have been masked (Min et al., 2024, p. 4).

Auto-Regressive Language Modeling Auto-regressive language modeling, sometimes also referred to as Causal Language Modeling (CLM) (Conneau and Lample, 2019), is a unidirectional approach that involves the training of a Transformer model to predict the conditional probability of a word w_t given its context, denoted as $P(w_t \mid w_1, \ldots, w_{t-1}; \theta)$, where θ represents the model parameters (Min et al., 2024, p. 5). The primary goal in auto-regressive language modeling is to maximize the log-likelihood of the sequence of words, as mathematically represented in Equation 2, where θ_T denotes the parameters of the model, and the summation runs over all words in the sequence (2024, p. 5).

$$\sum_i \log(P(x_i|x_1, x_2, \ldots, x_{i-1}; \theta_T)) \tag{2}$$

In auto-regressive modeling, only the auto-regressive decoder component of the Transformer architecture is employed. Multiple multi-head self-attention layers are stacked, where each layer includes a mechanism to attend to preceding tokens only. This procedural restriction is crucial, as it ensures that the prediction for any word x_i only depends on the preceding words $x_1, x_2, \ldots, x_{i-1}$ (2024, p. 5). Auto-regressively pre-trained language models are primarily used for language generation tasks such as text summarization and translation (Wang et al., 2024, p. 10).

Masked Language Modeling MLM, sometimes also referred to as the Cloze task (Conneau and Lample, 2019; Wilson, 1953, distinguishes itself from auto-regressive language modeling in its approach to predicting words within a sequence. In the case of BERT, MLM employs a bidirectional approach, where masked tokens are predicted based on all other tokens in the sequence, i.e., those preceding and following the masked tokens (Min et al., 2024, p. 5).

The core idea is to randomly mask a portion of the input tokens and to subsequently train the model to predict these masked tokens. During training, the masked tokens are replaced in three ways: 80% of the time they are replaced by a [MASK] token, 10% of the time by a randomly chosen token, and 10% of the time they remain unchanged (Devlin et al., 2019, p. 4174). The primary goal is to maximize the likelihood of correctly predicting the masked tokens. The objective function used for this is typically the cross-entropy loss over the masked tokens (Min et al., 2024, p. 5). Formally, the loss can be represented as shown in Equation 3.

$$\sum_i m_i \log \left(P(x_i \mid x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n); \theta_T \right) \tag{3}$$

$m_i \in \{0, 1\}$ indicates whether x_i is masked, and θ_T represents the parameters of the Transformer encoder. This objective function is optimized during training to adjust the parameters, enhancing the model's ability to predict the masked tokens accurately (2024, p. 5). Models, such as BERT, RoBERTa, and XLM-R use the MLM pre-training approach and are primarily employed for tasks involving sequence labeling and text classification (Wang et al., 2024, p. 10).

Figure 6 provides an illustration of both pre-training objectives. It contrasts the prediction of the subsequent Token E, given the preceding sequence of Tokens A, B, C, D, in unidirectional auto-regressive language modeling with the prediction of the masked Tokens B and D, positioned between Tokens A, C, and E, in bidirectional MLM, as employed in BERT.

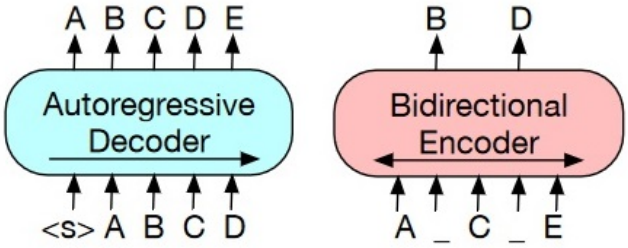

Figure 6: Contrasting visualization of auto-regressive and bidirectional MLM (Lewis et al., 2020, p. 7872)

2.6.2 Multilingual Pre-Trained Language Models

The prevailing focus on the English language during model pre-training becomes apparent, emphasized by the abundance of English benchmarks (Conneau and Lample, 2019). BERT is pre-trained predominantly using English language data sourced from two main corpora: the BooksCorpus (Zhu et al., 2015), consisting of 800 million words, and English Wikipedia texts, comprising 2,500 million words (Devlin et al., 2019, p. 4175). Multilingual BERT (mBERT) is a variant of the original BERT model designed to handle multiple languages. It is trained using the MLM objective on nonparallel Wikipedia articles in 104 different languages, with a shared vocabulary of 110k WordPiece tokens across all these languages. Each document used for training mBERT is in a single language, and the training does not rely on any cross-lingual dictionaries or specific methods for connecting the languages. The model learns the patterns of each language independently but within a shared framework, allowing it to perform well across a wide variety of languages (Paaß and Giesselbach, 2023, p. 108).

Similar to mBERT, XLM-R is a Transformer-based language model trained on the MLM objective. However, unlike mBERT, it uses a significantly larger CommonCrawl[4] corpus covering 100 languages. This extensive dataset allows XLM-R to learn from a more diverse and substantial text collection (Chen et al., 2021). Moreover, it does not use language-specific embeddings, enhancing the model's ability to handle linguistic phenomena, such as code-switching. The combination of high-resource and low-resource languages, such as Swahili and Urdu, makes XLM-R supposedly robust for low-resource languages (Conneau et al., 2020, p. 8441). Consequently, XLM-R shows notably better performance compared to mBERT across various cross-lingual evaluation tests (Chen et al., 2021), however, low-resource languages remain to be a challenge for PLMs.

[4] `https://commoncrawl.org/overview` [Accessed: 13.07.2024]

2.7 Transfer Learning

While PLMs learn generic language representations from large collections of general-purpose internet texts, such as Wikipedia articles during pre-training, they lack exposure to specialized corpora needed for specific fields (Qiu et al., 2020, p. 1882). Consequently, they may not perform as well when handling texts that require domain-specific knowledge (Wang et al., 2024, p. 12). A domain $D = \{X, P(X)\}$ is composed of two elements: a feature space X and its marginal probability distribution $P(X)$. The feature space X is defined as a set of instances, $X = \{x_1, x_2, \ldots, x_n\} \in X$ (Farahani et al., 2020, p. 345). Consequently, a domain can be a set of texts that share common characteristics, such as level of formality, style, or topic. More practically, it often refers to a body of texts that originate from a particular specific source (Koehn, 2020, p. 239). Texts from scientific journals, news articles, customer reviews, or social media posts each form their own domain due to their unique content and style (Luu et al., 2022, p. 5944).

The advantage of PLMs lies in their capability for transfer learning. This refers to their ability to transfer knowledge from a source domain and task to improve performance in a related target domain and task by utilizing the knowledge from the source to enhance the predictive function in the downstream task (Farahani et al., 2020, p. 345). This avoids the resource-intensive and time-consuming process of training new models from scratch (Qiu et al., 2020, p. 1891) Figure 7 graphically illustrates the process of transferring the pre-trained source model's knowledge to a new target domain or task.

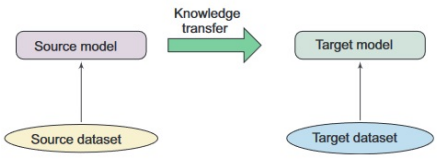

Figure 7: Transfer process (Qiu et al., 2020, p. 1886)

PLMs encounter two challenges when applied to downstream tasks: the task gap and the domain gap. The task gap emerges when the downstream task diverges from the pre-training language modeling objective. The domain gap, on the other hand, refers to the lack of domain-specific pre-training data crucial for a given task (Wang et al., 2024, p. 12). To effectively utilize PLMs for downstream tasks, both the domain and the task need to be adapted. In domain adaptation, fine-tuning occurs within the target domain while maintaining task consistency across different domains (Farahani et al., 2020, p. 344). Task adaptation, on the other hand, entails fine-tuning the model for specific tasks (Wang et al., 2024, p. 12).

2.7.1 Fine-Tuning

Model transfer is achieved through feature extraction or fine-tuning. Feature extraction involves extracting a PLM's the hidden representations of the last or last two layers of a PLM in order to obtain embeddings applied to another task. The pre-trained parameters in the model itself are not changed but applied to another task. In contrast, fine-tuning involves updating the pre-trained parameters (Qiu et al., 2020, p. 1886) by taking a model previously trained on a large dataset and refining it for a specific task or domain. This is achieved through additional training on a smaller, more targeted dataset (Min et al., 2024, p. 2). While both methods improve NLP tasks, feature extraction necessitates more intricate, task-tailored architectures, making fine-tuning generally more versatile and convenient (Qiu et al., 2020, p. 1886f.). To sum up, fine-tuning can be considered more effective than training from scratch (Paaß and Giesselbach, 2023, p. 136).

Hyperparameter Tuning Fine-tuning PLMs hinges significantly on the configuration of their hyperparameters (Liu and Wang, 2021, p. 2286). In ML, hyperparameters refer to the configurable settings or parameters of a model that are not acquired from data through learning but are pre-defined by the user or determined through experimentation (Kuhn and Johnson, 2013, p. 64f.). These parameters have a significant influence on the model's performance, affecting aspects, such as final accuracy and generalization capability. Minor deviation in these parameters may result in a considerable decline in performance (Liu and Wang, 2021, p. 2286). Typical hyperparameters in DNNs include *learning rate*, *number of layers*, *number of epochs*, *batch size*, and *weight decay* (Bartz et al., 2023, p. 59ff.; Nakamura and Hong, 2019). Table 9 summarizes these key hyperparameters, including a description of their roles in the training process.

Table 9: Key hyperparameters in DNNs and their roles in the training process

Hyperparameter	**Description**
Learning rate	Determines the size of the step at which the model's parameters are updated (Bartz et al., 2023, p. 63).
Number of layers	Refers to the depth of the neural network, specifying the number of hidden layers (Bartz et al., 2023, p. 59).
Number of epochs	Number of times the learning algorithm processes the entire training dataset (Yang and Shami, 2020, p. 301).
Batch size	The number of training examples used in one iteration before model paramater update (Bartz et al., 2023, p. 65f.).
Weight decay	Regularization technique to penalize large weights by adding a penalty to the loss function (Nakamura and Hong, 2019).

The optimal hyperparameter configuration varies depending on the specific data, task, and training objective. Various methodologies exist to determine the most suitable hyperparameter settings. Traditional methods involve manual adjustment, which includes referring to research findings for hyperparameter suggestions, drawing from personal experience, or conducting trial-and-error experiments (Probst et al., 2019). Alternatively, hyperparameter tuning can be automated through Hyperparameter Optimization (HPO) methods (Liu and Wang, 2021, p. 2286).

Hyperparameter Optimization HPO is the process of **automatically** tuning hyperparameters to improve model performance, minimizing human intervention (Feurer and Hutter, 2019, p. 3f.). Mathematically, it entails the optimization of an objective function that maps hyperparameters to an evaluation metric, usually seeking to maximize accuracy or minimize loss. The primary objective is to determine the optimal hyperparameter configuration, x^*, from a set of possible configurations, X, to minimize an objective function, $f(x)$ (Yang and Shami, 2020, p. 298) as illustrated in Equation 4.

$$x^* = \arg\min_{x \in X} f(x) \tag{4}$$

HPO algorithms, such as Random Search (RS) (Bergstra and Bengio, 2012), or more sophisticated methods, such as Bayesian Optimization (BO) (Snoek et al., 2012), or HPO frameworks, such as Optuna (Akiba et al., 2019) are frequently used to find the optimal set of hyperparameters. The performance of fine-tuning PLMs heavily relies on hyperparameter settings, with incorrect configurations potentially causing significant drops in performance (Liu and Wang, 2021, p. 2286). BO and RS remain standard HPO algorithms in ML and DL (Ilemobayo et al., 2024). For instance, they are used for optimizing hyperparameters in Neural Machine Translation (NMT) systems (Zhang and Duh, 2020) and Transformer-based PLMs (Liu and Wang, 2021). Additionally, RS is employed as a baseline in HPO to evaluate the efficiency of newly developed algorithms (Chen et al., 2022).

2.7.2 Few-Shot, One-Shot, and Zero-Shot Learning

While the inherent task-agnostic adaptability of PLMs has resulted in significant progress for tasks, such as reading comprehension, question answering, and textual entailment, attaining optimal performance in specific tasks frequently requires extensive fine-tuning on large, task-specific datasets (Brown et al., 2020, p. 1877). However, such datasets are not always readily available and acquiring them can be cost-intensive (Liu et al., 2023, p. 1097).

Few-Shot Learning (FSL), One-Shot Learning (OSL), and Zero-Shot Learning (ZSL) are innovative approaches in the field of DL, devised to tackle the challenge

of limited domain-specific data (Alzubaidi et al., 2023). FSL is inspired by human learning abilities, particularly the observation that humans can grasp new concepts with minimal examples (Yang et al., 2020, p. 177). This approach entails providing a model with a limited set of task instances during inference to influence its output (Brown et al., 2020, p. 1879).

Each example includes a context and a desired completion, such as an English sentence and its French translation. FSL significantly reduces the need for extensive task-specific data, though it still requires some data. OSL is akin to FSL, but includes only one training example. ZSL, on the other hand, is a technique where a model can perform tasks on new, unseen data or classes without specific training examples, by using auxiliary information or transferable knowledge (Sivarajkumar and Wang, 2022, p. 972).

2.8 Classification

Classification is a supervised learning task and entails predicting the category or class of a given text or pattern, where each text or pattern is presumed to belong to a specific class among several predefined classes. These patterns are characterized by features, which are attributes or properties that aid in differentiating between the different classes. In traditional ML, feature variables capture significant class-specific details, facilitating accurate prediction of a pattern's class based on their values (Theodoridis, 2015, p. 60).

In contrast to these traditional methods, DL models can automatically extract features from text as part of the fine-tuning process. This process typically involves supplying the model with labeled examples, enabling it to develop the ability to generalize and accurately categorize unfamiliar data. Consequently, classification in DL is about teaching a model to recognize and assign predefined categories or labels to input data based on the patterns it learns during the fine-tuning process (Zhou, 2020, p. 132f.).

2.8.1 Single-Label and Multi-Label Classification

Classification problems can be divided into two primary categories: single-label and multi-label classification, with single-label encompassing traditional binary and multi-class classification. The goal of binary classification is to categorize data into one of two possible classes, $y \in \{0, 1\}$, such as in spam detection (Meng et al., 2016). Multi-class classification assigns each instance to exactly one of k possible classes, ensuring that each instance belongs to only one class, $y \in \{1, 2, \ldots, k\}$.

Conversely, multi-label classification allows each instance to be associated with multiple labels. This enables it to belong to several classes simultaneously, denoted as $y \in 2^{\{1,2,\ldots,k\}}$, where $2^{\{1,2,\ldots,k\}}$ represents the power set of the set $\{1, 2, \ldots, k\}$.

This signifies the ability to form subsets with elements either included or excluded, reflecting the binary choices for each label. For instance, in document categorization, a document can be labeled with multiple topics, such as science and technology, indicating it belongs to both categories (Dekel and Shamir, 2010, p. 137).

2.8.2 Token Classification

Tokenization is often used to refer to the first step in preparing input data for NNs (Friedman, 2023, p. 380) and involves splitting text into smaller units known as tokens, typically representing words or sub-words. In contrast to text classification, which predicts a single label for an entire text, token classification involves assigning labels to each token within the text individually. This approach shifts the focus from macro-level classification of larger text segments to micro-level classification of specific words or sequences of words (Ettrich et al., 2024, p. 3).

In sequence classification, a type of text classification, the objective is to train a sequence classifier C that assigns a single class label $l \in L$ to a sequence s. This is expressed as $C : s \rightarrow l$, where l belongs to the set of class labels L. In this context, each sequence is paired with exactly one class label, and the classifier has access to the entire sequence prior to making its classification decision (Xing et al., 2010, p. 40). For instance, when using a PLM for a sequence classification term extraction task, the process involves providing the model with context/n-gram pairs during training. Positive examples consist of context/term pairs extracted from the corpus, while negative examples comprise randomly selected words or n-grams. These pairs are used to teach the model how to distinguish between sentences containing the target term (positive examples) and those containing other n-grams (negative examples) (Hazem et al., 2020, p. 97).

Figure 8 illustrates the difference between both concepts. Text and sequence classification assign a single label to the whole sequence. In contrast, in token classification, each token within the sequence is assigned an individual label.

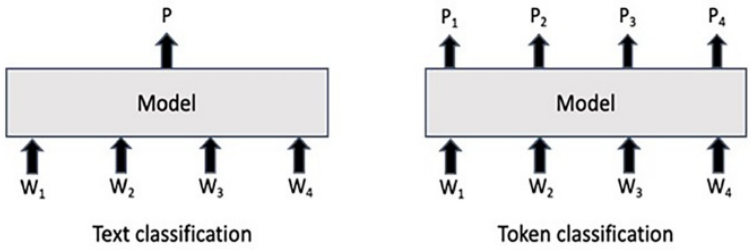

Figure 8: Contrasting text classification and token classification (Ettrich et al., 2024, p. 3)

Named Entity Recognition (NER) is a classification task that can be implemented with both sequence classification (Hu et al., 2022) and token classification (Mehta and Varma, 2023). The goal is to identify and categorizes specific entities, such as organizations, names of people, time and date, or locations, within a text (Mehta and Varma, 2023, p. 453). Currently, DL and neural networks provide the best performance for NER tasks and their numerous applications, including questions answering, information extraction, text summarization, and MT (Ettrich et al., 2024, p. 3).

There are various different annotation methods for NER. The simplest is IO, where I stands for words inside named entities and O for regular words that are not considered named entities. However, IO does not handle consecutive entities of the same type well. The BIO scheme, also referred to as IOB, improves on this by adding another label to differentiate if a word is the beginning B, inside I, or outside O of any recognized named entities. IOE is similar to the IOB scheme but includes an indication for the end E of the named entity. IOBES enhances entity boundary information by introducing tags for beginning B, inside I, end E, single-tokens S, and outside O. BI tags entities similarly to IOB but marks the beginning of non-entity words with B-O and the rest as I-O. IE works similar to IOE but marks the end of non-entity words with E-O. BIES combines entity encoding from IOBES with non-entity words, using the same method to denote the beginning B-O, inside I-O, and single non-entity tokens S-O between two entities (Alshammari and Alanazi, 2021, p. 295f.). Figure 9 illustrates the BIO NER token classification annotation scheme on an example sentence.

John	flew	to	New	York	to	watch	the	Super	Bowl	final	with	his	friends
B-PER	O	O	B-LOC	I-LOC	O	O	O	B-MISC	I-MISC	O	O	O	O

Figure 9: Example BIO annotation scheme for NER (Vacareanu et al., 2024, p. 323)

The Person tag is assigned to the name *John*, the beginning and inside Location tags are applied to *New* and *York*, and the beginning and inside Miscellaneous tags are attributed to *Super* and *Bowl*. Tokens, such as *flew*, *to*, *watch*, *with*, or *his* receive a O tag, indicating that they are not identified as named entities (Vacareanu et al., 2024, p. 323).

2.8.3 Classification Evaluation Metrics

Accuracy is a common metric for evaluating performance in classification tasks and is extensively used in popular benchmarks, such as General Language Understanding Evaluation (GLUE) (Wang et al., 2018). It measures the proportion of correctly

identified samples in a given dataset (Vickers et al., 2023, p. 498). However, using accuracy as a performance metric can be misleading because it does not truly reflect the classifier's effectiveness. An accuracy of 50% might initially seem acceptable, but it could be no better than random guessing. As a result, accuracy does not offer meaningful insights, particularly in scenarios with imbalanced class distributions or multi-class classification problems (Ben-David, 2007, p. 875).

In imbalanced datasets, the minority class is usually of greater interest, but its scarcity hinders the learning algorithm's ability to generalize effectively, leading to poor predictive performance. In such cases, conventional ML algorithms, which emphasize overall accuracy, tend to classify most observations as part of the majority class (Dal Pozzolo et al., 2015, p. 200f.). Consequently, it is important to consider additional metrics, such as precision and recall, which offer a more comprehensive evaluation of the classifier's performance and are frequently used to evaluate NLP tasks (Turian et al., 2003).

To compute precision and recall and to evaluate the performance, it is crucial to obtain certain indicators, namely True Positives (TP), True Negatives (TN), False Positives (FP), False Negatives (FN), and gold standard labels. TP are accurately predicted positive outcomes, TN signify accurately predicted negative outcomes, FP denote erroneously predicted positive outcomes, FN represent inaccurately predicted negative outcomes (Vujovic, 2021, p. 601). **Gold standard labels** refer to the human-defined annotations or classifications assigned to each document within a dataset, serving as the ground truth or reference standard against which automated systems' performance is evaluated.

Together with TP, FP, TN, FN, gold standard labels can be used to form a confusion matrix (Jurafsky and Martin, 2023, p. 70). A confusion matrix is a tabular visual representation that shows how well a classifier compares to human-defined gold standards. Its appearance changes based on the number of classes being predicted. As the number of classes increases, the confusion matrix grows in size and complexity, providing a detailed view of the model's performance across all classes (2023, p. 70f.). Figure 10 visualizes a simple binary classification example confusion matrix containing the system output and the gold standard.

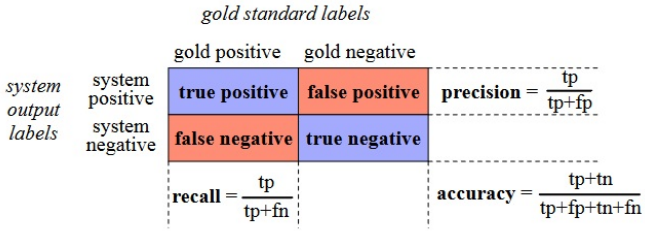

Figure 10: Binary classification confusion matrix (Jurafsky and Martin, 2023, p. 70)

Precision is the ratio of correct positive predictions to the overall number of positive predictions, measuring the accuracy of positive predictions made by the classifier (Vujovic, 2021, p. 602) and is mathematically represented as shown in Equation 5.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{5}$$

Recall, on the other hand, measures the capacity of a model to accurately detect all relevant instances within a dataset. Mathematically, recall is defined as the ratio of true positive instances to the sum of false negative and true positive instances (2021, p. 601) as shown in Equation 6.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{6}$$

In combination, precision and recall serve as the basis for calculating the F1 score, which represents their harmonic mean (2021, p. 602), as demonstrated mathematically in Equation 7.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{7}$$

The weighted F1 score is a specialized variant of the F1 score designed to handle class imbalance, as it involves the separate evaluation of the F1 score for each class label followed by the computation of their average. This average is weighted based on the number of true instances for each label, thereby mitigating class imbalance by assigning higher significance to classes with greater instance representation (Rusli et al., 2020, p. 30). The integration of accuracy, precision, recall, and F1 score serves as a robust foundation for assessing classifiers, with widespread application in the field of NLP, with accuracy and F1 score being used as evaluation metrics in seven out of nine GLUE tasks (Wang et al., 2018, p. 354).

However, there are many more classification evaluation metrics, such as Matthews Correlation Coefficient (MCC) and Area Under the Curve (AUC)-Receiver Operating Characteristic (ROC). AUC is a metric used to evaluate how well a classification model distinguishes between classes. It measures the area under the ROC curve, which plots the true positive rate against the false positive rate. A higher AUC indicates better overall ranking performance of the classifier, regardless of specific threshold choices or probability metrics (Vujovic, 2021, p. 602). To sum up, the choice of evaluation metrics depends on various factors such as the specific task, the goals of the analysis, and the characteristics of the data being evaluated.

3 Related Work

Detecting hate speech is a complex challenge that demands innovative approaches due to the intricacies and ever-evolving nature of language (Aitchison, 2013, p. 3). Thus far, most research in this field has centered on English (Fortuna and Nunes, 2018, p. 16), resulting in a dearth of datasets for low-resource and code-mixed contexts (Jose et al., 2020, p. 138). However there is a recent trend towards addressing hate speech in low-resource settings (Nkemelu et al., 2022; Roberts, 2024). This trend directly intersects with this thesis on automatic profanity detection in code-switched German rap lyrics, highlighting the crucial need to develop detection methods for varied linguistic environments where annotated data is scarce.

To bridge the low-resource data gap, Sharma et al. (2024) introduce a manually created and labeled dataset of YouTube comments in a Hindi-English code-mixed language, called Targeted Hate Speech Against Religion (THAR). The dataset includes 11,549 comments tagged for targeted hate speech against religions, with both binary (anti-religion or non-anti-religion) and multi-class classifications (targeting Islam, Hinduism, Christianity, or none). The quality of the annotations is upheld through a sufficiently high inter-annotator agreement, demonstrating substantial consensus among annotators and thereby underpinning the integrity of the data and annotations. The study implements and evaluates different DL models, such as CNNs, LSTM, and Transformer-based models, such as mBERT, and Multilingual Representations for Indian Languages (MuRIL) (Khanuja et al., 2021), a BERT-based language model pre-trained on a diverse set of 16 Indian languages, to determine their effectiveness in detecting religious hate speech.

For binary classification MuRIL outperforms all models with the highest weighted average precision, recall, and F1-score, 0.80, 0.78, 0.78. In multi-class classification, MuRIL again leads with weighted averages of 0.72, 0.71, and 0.72, respectively. mBERT also performs well, while CNNs show the poorest results. The study highlights class imbalance in the multi-class task, particularly with a high percentage of none class comments, affecting performance metrics. Overall, Transformer-based models demonstrate superior effectiveness over CNNs and LSTM in both tasks. The paper's contributions are significant as it provides one of the first large-scale annotated datasets in a Hindi-English code-mixed language and demonstrates its applicability for advancing hate speech detection research in low-resource languages.

Ghosh et al. (2023) address the problem of low-resource hate speech detection for Assamese and Bodo[5]. The lack of resources in these languages poses challenges for NLP tasks. The paper introduces the North-East Indian Hate Speech (NEIHS) dataset, which includes labeled hate and non-hate text for both languages. The dataset collection involved scraping comments from Facebook and YouTube, followed by manual annotation by first language speakers. The annotation process adhered to strict guidelines based on community standards, considering aspects, such as profanity, sexual orientation, personal attacks, gender chauvinism, religious criticism, political criticism, and violent intentions.

The results show that among the models evaluated on the NEIHS dataset for Bodo language, mBERT performed the best, achieving the highest weighted F1-score of 85%. On the NEIHS dataset for Assamese, mBERT and MuRIL emerged as the top-performing models, with the highest weighted F1-score of 69%. Transformer-based models outperformed traditional ML models, such as Naïve Bayes and SVMs in both low-resource languages. The authors emphasize the importance of weighted F1 score due to imbalanced class distribution in classification problems. Overall, both studies contribute to hate speech detection research in low-resource languages and provide valuable insights into model performance and dataset creation methodologies.

Castillo-López et al. (2023) compared monolingual and multilingual models emphasizing the challenges posed by linguistic variations and cultural differences within Spanish-speaking communities. Using mBERT and BETO (Cañete et al., 2020), a monolingual Spanish version of BERT, the study investigates hate speech detection across different variants of the same root language, namely Spanish, particularly targeting misogyny and xenophobia. The research questions address the effectiveness of language-specific models compared to multilingual models and the efficacy of zero-shot transfer learning across Spanish variants.

BETO consistently outperformed mBERT in both misogyny and xenophobia domains, with significant differences in macro-F1 scores. Compared to mBERT, BETO exhibited an 11-point higher score in misogyny detection and a 4-point higher score in xenophobia detection. Moreover, BETO demonstrated greater stability and consistency across different runs, as indicated by lower standard deviations. The study also highlights the impact of linguistic variations on model performance, with BETO exhibiting higher accuracy when trained and tested on the same Spanish variant.

Mishra et al. (2021) addressed the challenge of hate speech detection in multilingual settings and investigate different Transformer-based approaches using BERT

[5]Bodo, a Tibeto-Burman language, is spoken primarily in Western Assam's Bodoland region, with smaller communities in central and eastern Assam, in Arunachal Pradesh, Meghalaya, and Nepal. Assamese, an Indo-Aryan language, is prevalent in Assam's Brahmaputra Valley. Despite their linguistic differences, Bodo and Assamese share some historical and cultural connections due to mutual contact (Kaur, 2017, p. 1170)

models. Their discovery highlighted the effectiveness of multilingual training for detecting hate speech across diverse languages, revealing several advantages to effectively using multilingual models trained on multiple languages. They generalize well across various languages, making them useful for code-mixed content on social media. Additionally, they efficiently utilize a larger dataset without needing extra data points. Despite potential minor accuracy trade-offs compared to monolingual models, multilingual models still perform competitively, sometimes even better, especially with multi-task learning.

4 Methodology

This chapter outlines the methodology used to execute the experimental part of this thesis. It starts with the collection and annotation of the colloquial social media and rap data for the profanity detection token classification task, followed by the fine-tuning process of XLM-R for token classification using the word-level annotated colloquial datasets. After fine-tuning, the German monolingual FTM and the German-English bilingual FTM undergo zero-shot domain transfer with unseen cross-domain rap data to evaluate their generalization capability, adaptability, and performance in profanity detection across new domains. Figure 11 visualizes the experimental setup.

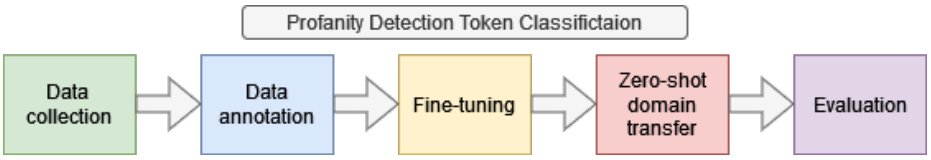

Figure 11: Overview of the experimental setup

4.1 Data Selection and Collection

To the best of the author's knowledge, there are no publicly available word-level annotated datasets in German and English that include both single and multi-word expressions for profanity detection. Consequently, appropriate datasets needed to be created for the experiment[6]. The colloquial language social media domain in German and English was chosen for fine-tuning and the code-switched German rap lyrics domain for zero-shot domain transfer. Both domains are rich in profanities and informal expressions, making them valuable for studying and analyzing the usage of such language.

4.1.1 Colloquial Language Data

The initial step of creating the German and English colloquial language datasets for fine-tuning involved compiling lists of profane words from multiple sources,

[6]The datasets are available on GitHub `https://github.com/Meraki89/Profanity-Datasets`

including the lexicon of abusive words (Wiegand et al., 2018b), the Carnegie Mellon University School of Computer Science bad word list[7], the List of Dirty Naughty Obscene and Otherwise Bad Words[8], and the List of Bad Words dataset[9] from Kaggle. The latter two sources additionally provided compilations of German profane vocabulary, which were utilized as sources for the German language profane word list. This diversity ensured a varied selection of vulgar single-word and multi-word expressions. Throughout the whole process, each language was treated and processed independently and the languages were not mixed in the datasets.

Following this, the word lists from the different sources were merged and duplicates were removed automatically. This resulted in one single comprehensive list of profanities for each language. To establish a well-organized and significant dataset, the profane expressions were selected based on three main criteria. The aim was to encompass (i) all expressions containing or stemming from the seven primary swear words, which collectively represent over 90% of online profanity instances (Wang et al., 2014) as outlined in Section 2.1, including variations, such as *apeshit*, *piece of shit* and their German equivalents, such as *Scheiße*, and *Scheißkerl*, (ii) all expressions with explicit sexual vulgar connotation, such as *cocksucker*, *jizz*, *Fotze*, and *Arschfick*, and (iii) all offensive insults categorized within the nine hate categories (Silva et al., 2016) presented in Subsection 2.1.1.

Table 10 presents a selection of insults in both English and German, sourced from the created profanity lists. Each insult is paired with its respective hate category.

Table 10: Sample insults in English and German from the profanity lists along with their corresponding hate categories

Category	**English examples**	**German examples**
`Race`	sandnigger, white trash, spic	Neger, Schlitzauge
`Behavior`	crybaby, pathetic	Weichei, Versager, Zicke
`Physical`	lardo, fatso, fugly	Hackfresse, fette Sau
`Sexual orientation`	lesbo, faggot, fairy	Homo, Schwuchtel, Transe
`Class`	hillbilly, redneck	Bonze
`Gender`	attention whore, cunt, chode	Schabracke, Miststück
`Ethnicity`	Afghani	Zigeuner, Kanake
`Disability`	retard, schizo, psycho	Krüppel, minderbemittelt
`Religion`	kike	Judensau

[7] `https://www.cs.cmu.edu/~biglou/resources/bad-words.txt` [Accessed: 31.05.2024]
[8] `https://github.com/LDNOOBW/List-of-Dirty-Naughty-Obscene-and-Otherwise-Bad-Words/blob/master/de` [Accessed: 31.05.2024]
[9] `https://www.kaggle.com/datasets/tushifire/ldnoobw` [Accessed: 31.05.2024]

This process yielded a total of 215 German and 373 English profanities. Using these lists, example sentences from the colloquial social media domain were extracted from the May 2015 Reddit Comments dataset[10] for English, and from the German sentiment dataset proposed by Sepideh Mollanorozy[11], and the German part of the Multi-lingual HateSpeech Dataset[12] for German.

A reasonable ratio was maintained between positive examples, i.e., sentences containing profane expressions, and negative examples, i.e., sentences without profane words. Although the sentences were extracted automatically, the final selection was performed manually to ensure high quality. The goal was to create comprehensive datasets that feature a wide range of profanities, including orthographic variations, such as *Scheiße* and *Scheisse*, while also ensuring an adequate number of negative examples. As presented in Table 11 the English data consists of 442 negative and 783 positive examples, showing a higher prevalence of positive instances. Likewise, the German data contains 192 negative and 393 positive examples, also demonstrating a predominance of positive instances.

Table 11: Number of Positive and Negative Examples by Language

Language	**Negative Examples**	**Positive Examples**
English	442	783
German	192	393

The larger number of positive instances ensures an ample supply of profane examples, while the inclusion of negative instances helps preventing the model from expecting profanities in every context. This distribution helps maintaining model performance by providing a comprehensive range of linguistic examples for training.

4.1.2 Rap Data

The rap dataset includes examples from a new and unseen domain: code-switched German rap lyrics. None of these examples are presented to the model during the fine-tuning phase. Exclusively reserved for evaluating the models' zero-shot generalization abilities, this dataset is solely utilized for assessment purposes. The example sentences for the rap dataset were manually chosen and extracted from

[10] `https://www.kaggle.com/datasets/kaggle/reddit-comments-may-2015` [Accessed: 31.05.2024]

[11] `https://huggingface.co/datasets/sepidmnorozy/German_sentiment` [Accessed: 31.05.2024]

[12] `https://www.kaggle.com/datasets/wajidhassanmoosa/multilingual-hatespeech-dataset` [Accessed: 31.05.2024]

Kaggle's German Rap Dataset[13], focusing on sentences that accurately represent the domain of German rap lyrics. The selection highlights the linguistic diversity and the frequent use of swear words typical of the genre.

The data features a significant number of both positive and negative code-switched examples, as well as purely German positive and negative examples. This mirrors the diverse landscape of German rap, where not all lines incorporate English elements, emphasizing German as the primary language. Among the total 1,105 words, excluding special characters, 20.09%, 222 instances, are in English, while 79.91%, 883 instances, are in German. Instances that are used in German and English, such as *rapper* or *ghetto*, were attributed to both categories.

The criteria for selecting vulgar expressions are consistent with those used for the colloquial language dataset. Table 12 illustrates examples for positive and negative code-switched and German-only examples from the rap dataset, with the vulgar expressions in the positive examples underlined and the code-switched English elements in bold.

Table 12: Code-switched and German-only examples from the rap dataset

	Code-Switched	**German-only**
`Positive`	Ich guck diese Nutte an, du bist **whack**, **Bitch**.	Ja, jetzt steigt deine Schlampe ein.
`Negative`	**All the way up** aus der Mittelschicht.	Falscher Ort, aber richtiger Zeitpunkt.

In alignment with the colloquial language dataset, the rap dataset illustrates a prevalence of positive instances, comprising 47 negative examples and 84 positive examples.

4.2 Data Annotation and Labeling Scheme

All datasets were consistently annotated using the same method, with sentences being labeled on word-level for a profanity detection token classification task using a modified BIO schema approach. The B-B label is used for single-word profanities and the first word in multi-word profanities, while the B label is used for subsequent words in multi-word profanities. All non-profane expressions are labeled n. This clear distinction in labeling helps in the effective identification and classification of single-word and multi-word profane expressions. Table 13 illustrates the labeling scheme using an example sentence from each dataset. *son of a whore* is labeled

[13] `https://www.kaggle.com/datasets/efrodl/german-rap-dataset` [Accessed: 31.05.2024]

as a multi-word profane expression, *scheiße* (shit) and *hurenschleimer* (whore sycophant) are both labeled as single-word profanities.

Table 13: Demonstration of the labeling scheme using sentences from the datasets

Dataset	**ID**	**Example Sentence and Labeling**
German	477	können (n) wir (n) heute (n) einfach (n) zu (n) hause (n) bleiben (n) und (n) nichts (n) tun (n) ? (n)
English	714	i (n) do (n) not (n) want (n) to (n) see (n) you (n) again (n) you (n) son (B-B) of (B) a (B) whore (B) . (n)
Rap	44	ey (n) , (n) du (n) bist (n) so (n) scheiße (B-B), du (n) bist (n) n (n) hurenschleimer (B-B) . (n)

4.3 Class Distribution

As outlined in Section 2.1, swear words make up only 0.5% to 0.7% of words in daily offline conversations and about 1.15% in online conversations (Wang et al., 2014). Due to this rarity of profane words compared to non-profane words in typical language use, token classification tasks for profanity detection face a significant inherent class imbalance.

The colloquial language datasets utilized in this experiment were sourced from the social media domain, thus reflecting typical online language use and inherently featuring the imbalance between profane and non-profane words. Table 14 illustrates the disproportion between profane words and non-profane words, with more than 90% of words in both languages being non-profane.

Table 14: Comparison of class distribution in the English and German colloquial datasets

Dataset	**B-B Tokens**	**B Tokens**	**n Tokens**	**n Tokens %**
English	1078	194	18800	93,66%
German	488	47	6668	92.57%

4.4 Model Selection

XLM-R was selected for the experimental part of this thesis because of the numerous advantages it provides for multilingual profanity detection token classification. It is proficient in encoding text representations across multiple languages due to its

extensive pretraining on a diverse corpus and its tokenizer efficiently handles various linguistic complexities, such as code-switching and morphologically rich languages, enabling strong generalization across languages and adaptable performance in downstream tasks with minimal data and computational resources. Consequently, XLM-R is well-suited for bilingual German and English fine-tuning and subsequent zero-shot domain transfer in the low-resource, code-switched German rap lyrics domain.

4.5 Data Preparation

After selecting two suitable domains, creating datasets for the profanity detection token classification, and selecting a suitable PLM, the next step was to properly prepare the data for fine-tuning and subsequent zero-shot domain transfer.

4.5.1 Data Loading and Preprocessing

The first step was to load the labeled datasets from .csv files, with each row containing a sentence ID, a token, and its corresponding label. After reading the files, tokens and labels were extracted and organized into a Python dictionary. Each key in this dictionary is a sentence ID, and the associated value is another dictionary containing lists of tokens and their labels. Finally, the data was structured into a list of tuples (words, labels), where words is a list of tokens and labels is a list of corresponding labels from the list of B-B, B and n for each token in a sentence.

4.5.2 Data Splitting

The colloquial language data was divided into training, validation, and test sets, with 80% allocated to training, 10% to validation, and 10% to test, resulting in 468 train, 58 validation and 59 test examples for German and 980 train, 122 validation and 123 test examples for English. Table 15 presents a tabulated summary of the train, validation, and test split in the colloquial German and English datasets, alongside the total counts used for fine-tuning.

Table 15: Distribution of examples in the colloquial train, validation, and test sets

Data	**German**	**English**	**Total**
Train	468	980	1448
Validation	58	122	180
Test	59	123	182

4.5.3 Tokenization and Label Alignment

The XLM-R tokenizer was initialized from the pretrained XLM-R base model[14]. The input data was tokenized, the labels were aligned with the corresponding tokens and appropriate label IDs were assigned to each token. Special tokens, e.g., padding, were ignored in loss calculations by assigning them a label of -100. For each word, the correct label was assigned to the first token and -100 to subsequent tokens, facilitating accurate fine-tuning of the token classification model.

4.6 Fine-Tuning

To investigate how the integration of English colloquial data alongside German colloquial data affects the efficiency of detecting profane single-word and multi-word expression, two pretrained XLM-R models were fine-tuned. The first model was fine-tuned only on German data, creating a monolingual FTM. Next, the German dataset was supplemented with English colloquial data, and the second model was fine-tuned on both datasets, producing a bilingual FTM. This facilitated a comparative assessment of their profanity detection performances.

4.7 Hyperparameter Tuning and Optimization

During the fine-tuning of the XLM-R models, extensive hyperparameter tuning and optimization was conducted to enhance the performance of both models. This process centered around four crucial hyperparameters that govern the fine-tuning process and greatly influence the model's performance: learning rate, weight decay, batch size, and number of epochs. To refine these hyperparameters, a combination of manual trial-and-error and automatic optimization using Optuna was employed. In the initial stage, manual adjustments based on empirical observations were made to evaluate the varying performances of the models. This process helped establish suitable search spaces for both models which were then used for subsequent hyperparameter optimization with Optuna. Table 16 provides an overview of the manually determined search spaces for both FTMs.

Table 16: Optuna trial search spaces

Hyperparameters	**Monolingual FTM**	**Bilingual FTM**
Number of Training Epochs	3 to 6	3 to 8
Number of Batch Size	4 or 8	8 or 16
Learning Rate	1×10^{-6} to 1×10^{-4}	1×10^{-6} to 1×10^{-4}
Weight Decay	1×10^{-6} to 1×10^{-4}	2×10^{-3} to 7×10^{-3}

[14] `https://huggingface.co/FacebookAI/xlm-roberta-base`

Optuna facilitated a systematic exploration of the hyperparameter search spaces, seeking combinations that minimized the loss function and maximized the validation F1 scores. Multiple Optuna studies, each comprising 5 trials, were conducted within the manually determined search spaces for both models, involving fine-tuning the models on the colloquial language training datasets and subsequently assessing their performance on the validation datasets. This iterative approach led to the identification of an optimal set of hyperparameters, thereby maximizing the performance of both XLM-R models for the specific data at hand. Table 17 and Table 18 present the hyperparameter combinations from the five trials of the studies that achieved the highest validation F1 scores for each FTM, with the best results highlighted. These highlighted hyperparameters were then selected for fine-tuning the models.

Table 17: Monolingual FTM's hyperparameter optimization results

Trial	Epochs	Batch Size	Learning Rate	Weight Decay	Val F1
1	5	4	1.747150×10^{-5}	2.21840×10^{-5}	0.8687
2	**4**	**4**	$\mathbf{1.293912 \times 10^{-5}}$	$\mathbf{7.29137 \times 10^{-5}}$	**0.9293**
3	4	4	1.477943×10^{-5}	2.73571×10^{-5}	0.8723
4	5	4	2.926722×10^{-5}	2.52559×10^{-6}	0.8958
5	5	4	2.926722×10^{-5}	2.52559×10^{-6}	0.8958

Table 18: Bilingual FTM's hyperparameter optimization results

Trial	Epochs	Batch Size	Learning Rate	Weight Decay	Val F1
1	4	16	9.589965×10^{-5}	4.26051×10^{-3}	0.8684
2	4	8	8.048835×10^{-6}	3.16845×10^{-3}	0.8276
3	4	8	9.935125×10^{-5}	3.17931×10^{-3}	0.8993
4	**6**	**16**	$\mathbf{7.501154 \times 10^{-5}}$	$\mathbf{4.96535 \times 10^{-3}}$	**0.9132**
5	5	16	2.979683×10^{-6}	3.46196×10^{-3}	0.6861

4.8 Zero-Shot Domain Transfer

Following the fine-tuning process of both models, zero-shot domain transfer was conducted to evaluate and compare their generalization abilities. This evaluation aimed to explore how the colloquial style, cultural and linguistic diversity, and the prevalent use of English offensive and vulgar expressions in code-switched German rap lyrics impact the zero-shot performance of both FTMs. To achieve this, the rap lyrics dataset, which has not been presented to the models during fine-tuning, was employed exclusively as an evaluation set for both models.

4.9 Quantitative Analysis

The metrics F1 score, precision, and recall were used as the most common evaluation metrics in classification tasks. This provided a more balanced assessment compared to relying solely on accuracy. Additionally, duplicate-free gold standard sets were created for the colloquial language German and English validation and test datasets, alongside a gold standard test set for the code-switched rap data. These sets include profane words, i.e., those labeled B-B and B, and were used to evaluate the performance of the FTMs.

Excluding duplicates for evaluation ensures equal weighting for each token's presence or absence in the profanity classes, maintaining evaluation consistency and simplifying the assessment process, facilitating comparisons between different models or iterations. Precision, recall, and F1 score were computed for a given set of predictions and labels. Subsequently, the intersection between the extracted expressions and the gold standard set was computed, and precision, recall, and F1-score were determined based on this intersection and the sizes of the gold set and extracted expressions set.

4.10 Qualitative Analysis

In addition to the quantitative analysis using precision, recall, and F1-score, qualitative analysis was conducted in three ways. Firstly, the annotated datasets utilized in this experiment contain word-level annotations for both single-word and multi-word profane expressions. This comprehensive annotation enabled a thorough examination of detected profanities and facilitated an easy and effective manual comparison of both FTMs' performance in successfully identifying single-word and multi-word expressions, presenting a significant advantage over traditional text classification methods.

Secondly, a primary objective was to assess both FTMs' robustness in detecting newly coined words. The dataset derived from rap music provided a rich source of diverse creative words creations. The neologisms for evaluation were manually selected based on their rarity in current language usage and their absence from established dictionaries, highlighting the dynamic nature of language evolution. This aimed to best ensure that these words were not encountered by the FTMs during pre-training or fine-tuning, thereby assessing their ability to adapt to and handle innovative linguistic expressions.

Lastly, profane expressions were manually categorized into four language-specific groups, i.e., German, English, Both (used in both languages, such as *bastard*), and Mixed (both languages are mixed in the same word, such as *Muschibattle*), providing deeper insights into both FTMs' performance for each language individually.

5 Results

This chapter is dedicated to presenting the results of this experiment. It starts with the results of the fine-tuning process, followed by the results of both FTMs on the colloquial language test dataset, and then moves on to the zero-shot domain transfer results.

5.1 Fine-Tuning Results

The fine-tuning process for the monolingual FTM was carried out over five epochs with a batch size of 585. The performance metrics were evaluated at every 100 steps. Throughout this process, both training and validation losses were tracked. The validation loss exhibited a downward trend, decreasing from 0.139503 at step 100 to 0.069853 at step 500, indicating the model's improved performance and ability to generalize to unseen data. Additionally, precision, recall, and F1 score were measured on the validation set at each evaluation step to assess the model's classification capabilities.

Precision demonstrated a notable increase over the course of fine-tuning, reaching its highest value of 91.30% at step 500. This suggests the model's enhanced ability to accurately identify positive instances within the dataset. Recall, while showing some fluctuations, ultimately displayed an ascending trend as well, peaking in a value of 87.50% at step 500, indicating the model's improved capacity to capture relevant instances. Lastly, the F1 score consistently improved throughout fine-tuning, reaching 89.36% at step 500, indicating an overall enhancement in the model's classification performance. Table 19 provides a succinct tabular summary of the fine-tuning results of the monolingual FTM.

Table 19: Overview of the fine-tuning results of the monolingual FTM

Step	Validation Loss	Precision (%)	Recall (%)	F1 (%)
100	0.139503	84.85	58.33	69.14
200	0.097908	88.37	79.17	83.52
300	0.079122	89.13	85.42	87.23
400	0.095392	92.68	79.17	85.39
500	0.069853	91.30	87.50	89.36

The fine-tuning phase of the bilingual FTM involved six epochs and a batch size of 546. Similar to the monolingual FTM, the validation loss of the bilingual FTM showed a progressive decrease, starting from 0.093317 at step 100 reaching 0.05371 at step 500. Additionally, precision, recall, and F1 metrics showed an upward trend. Precision achieved its maximum value of 92.11% at step 500, recall stayed consistently high, peaking at 93.33% during step 500, and F1 score displayed consistent enhancement, culminating in 92.72% at step 500. Table 20 provides a tabular overview of the fine-tuning results of the bilingual FTM.

Table 20: Overview of the fine-tuning results of the bilingual FTM

Step	Validation Loss	Precision (%)	Recall (%)	F1 (%)
100	0.093317	72.11	91.33	80.59
200	0.063833	81.87	93.33	87.23
300	0.057581	85.63	91.33	88.39
400	0.049946	91.50	93.33	92.41
500	0.053715	92.11	93.33	92.72

These results underscore the effectiveness of the carefully chosen hyperparameters in improving both FTMs' performance and their ability to achieve high precision, recall, and F1 score values on the validation set.

5.2 Results on the Colloquial Language Test Set

In the evaluation of the models' performance on the test set, notable differences were observed between the monolingual and the bilingual FTM as depicted in Figure 12.

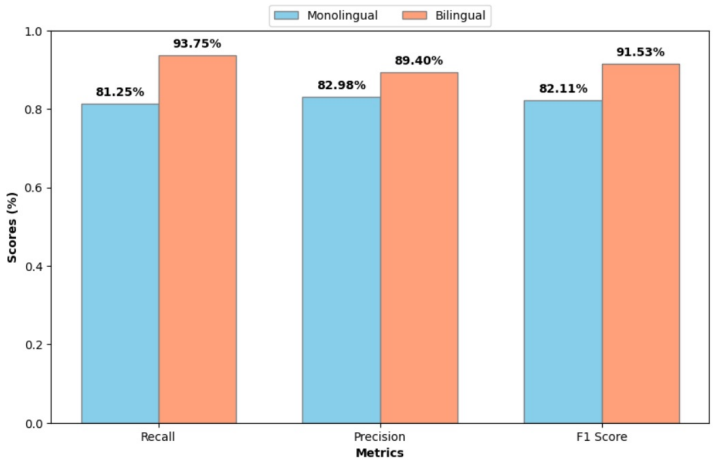

Figure 12: Comparative overview of both FTMs' performances

The monolingual FTM showed solid results, achieving a recall of 81.25%, a precision of 82.98%, and an F1 score of 82.11% indicating a balanced performance. The bilingual FTM, however, significantly outperformed the monolingual FTM, with a recall of 93.75%, precision of 89.40%, and an F1 score of 91.53%. The recall improved by 15.38% relative to the monolingual FTM, indicating a substantial improvement in the model's ability to correctly identify profane expressions, resulting in fewer false negatives. Precision increased by approximately 7.74%, reflecting a modest yet meaningful improvement in the accuracy of the model's profanity predictions. The F1 score improved by approximately 11.47%, underscoring the bilingual FTM's general superior performance. Overall, the bilingual FTM demonstrated notable performance improvements across all key metrics compared to the monolingual FTM. The most substantial gain was observed in recall, while the least was in precision.

5.3 Zero-Shot Domain Transfer Results

Comparing the monolingual and bilingual FTMs in the zero-shot domain transfer setting reveals a significant performance difference in favor of the bilingual FTM. The monolingual FTM extracted 86 expressions, matching 54 of the 66 gold standard expressions, resulting in a recall of 81.82%, a precision of 62.79%, and an F1 score of 71.05%. In contrast, the bilingual FTM extracted 84 expressions, with 62 matching the gold standard, leading to a recall of 93.94%, a precision of 73.81%, and an F1 score of 82.67%. Figure 13 illustrates the performance comparison between both models in the zero-shot domain transfer setting.

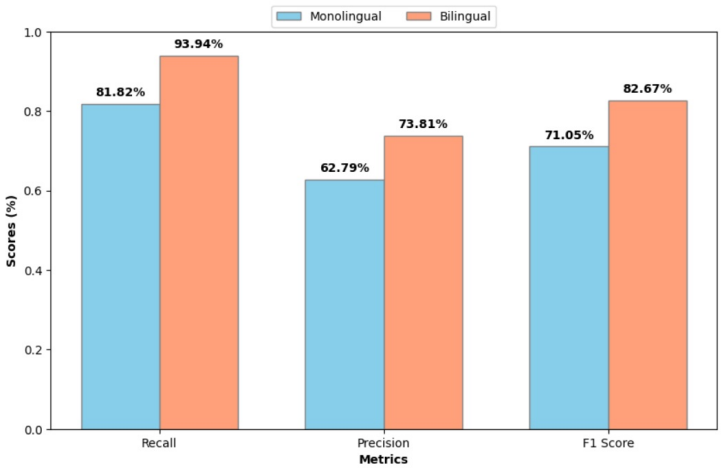

Figure 13: Performance comparison of the monolingual and bilingual FTM in the zero-shot domain transfer setting

The chart highlights the three key performance metrics: recall, precision, and F1 Score. Each metric is displayed as a percentage. In summary, the bilingual FTM achieved a higher performance than the monolingual model by 14.82% recall, 17.55% precision, and 16.35% F1-score relative to the monolingual FTM. These results highlight the superior overall performance of the bilingual FTM in identifying and extracting expressions accurately.

To further assess the models, all 117 total profane expressions including duplicates were classified into four categories. The first category, German, consists of 56 words, including expressions, such as *Hurentochter* and *Drecksbulle*. The second category, English, comprises 45 words and encompasses expressions, such as *bitchtits* and *motherfucker*. The third category, Both, contains 12 words used in both languages, such as *nigga* and *bastard*. The fourth category, Mixed, includes four words that combine elements of both languages, such as *Muschibattle* and *Dreckstoys*.

As illustrated in Figure 14, the monolingual FTM correctly identified 82.14% of the German expressions, 66.67% of the English expressions, 75.00% of Both, and 75.00% of Mixed. In contrast, the bilingual FTM demonstrates significantly higher performance across all categories, with values of 96.43% for German, 93.33% for English, and a perfect 100.00% for both the Both and Mixed categories.

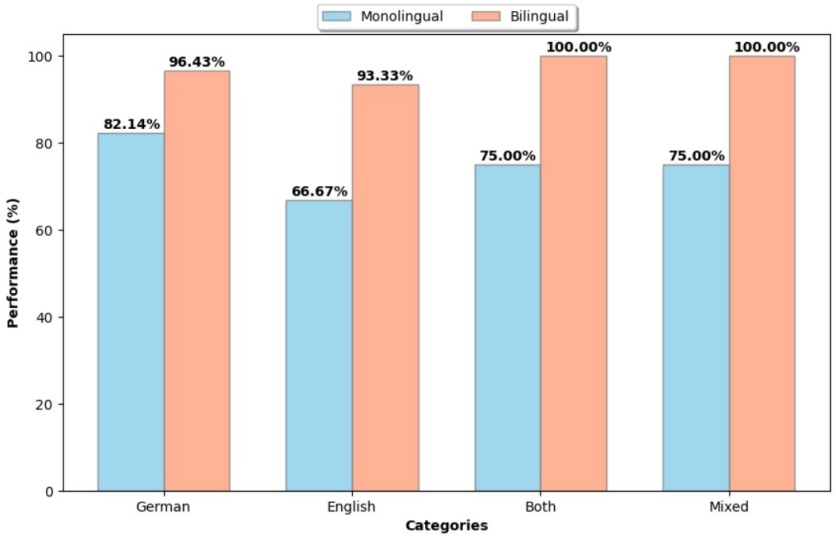

Figure 14: Performance comparison illustrating the accuracy of word identification in different language categories between the monolingual and the bilingual FTM

As a result, the bilingual FTM exhibits the most significant performance improvement, with a rise of 26.66% in the English category and the least improvement in the German category, at 14.29%. Moreover, with 96.43% of correctly identified

German expressions and 93.33%, the bilingual FTM performed slightly better in the German category compared to the English category. The improvement in performance within the German category is particularly noticeable when examining the German expression *fick*. Out of the nine occurrences of *fick* in the dataset, the German monolingual FTM failed to detect seven, whereas the bilingual FTM accurately identified all nine instances. Additionally, the monolingual FTM failed to recognize any multi-word expressions, leading to zero instances for class *B*. On the other hand, the bilingual FTM correctly identified both multi-word expressions in the rap dataset. Nevertheless, it also misclassified one expression as a multi-word expression when it was not.

Both models have demonstrated robustness in recognizing novel words, that have not present in their fine-tuning data. Through the identification of components of known expressions, both models successfully inferred new profanities. As illustrated in Table 21, the bilingual FTM marginally outperformed the monolingual one, detecting one more expression, i.e., *boss-cock*.

Table 21: Comparative performance in identifying neologisms

Neologism	**Monolingual FTM**	**Bilingual FTM**
`kanackenfreestyle`	✓	✓
`hurensohnköpfen`	✓	✓
`bitchbart`	✓	✓
`disco-hoes`	✗	✗
`hurentochter`	✓	✓
`boss-cock`	✗	✓
`bitchrap`	✓	✓
`muschibattle`	✓	✓
`facedrive`	✗	✗
`bitchtits`	✓	✓

6 Discussion

This thesis aimed to improve profanity detection by investigating the benefits of incorporating English colloquial data into the fine-tuning process of a multilingual PLM initially fine-tuned on German data, resulting in one monolingual and one bilingual FTM. The primary objectives were to assess the performance improvement in detecting profane expressions within colloquial language and to evaluate the zero-shot domain transfer capability of the FTMs on code-switched German rap lyrics. The findings indicated significant improvements in both scenarios when utilizing the bilingual FTM compared to the monolingual FTM, illustrating the effectiveness of the research setup in addressing the research questions. The results of this experiment highlight several important implications for the field of profanity detection.

Firstly, the consistent superior performance of the bilingual FTM over the monolingual FTM, evident in its higher precision, recall, and F1 score across both the colloquial language test set and the zero-shot domain transfer rap test set, highlights the importance of utilizing multilingual data for fine-tuning to enhance model robustness and performance. Additionally, despite both FTMs showcasing strong generalization abilities in identifying neologisms, the bilingual FTM slightly outperformed the monolingual fine-tuned one.

Secondly, while it was anticipated that incorporating English colloquial data would lead to a rise in the detection of English profanities, the most interesting spike occurred unexpectedly in the accuracy of identifying German swear words, with a notable improvement in identifying the German expression *fick*. This enhancement in performance is particularly remarkable considering that German is the dominant language in the German rap dataset and accounts for roughly 80% of the words.

Thirdly, the superior performance of the bilingual FTM in identifying profane expressions within code-switched German rap lyrics indicates that bilingual fine-tuning enables better handling of code-mixed content. This is particularly relevant given the prevalence of code-switching in modern social media and colloquial speech and the scarcity of explicitly code-switched labeled training data. The bilingual FTM's ability to accurately identify the majority of German and English profanities compared to the lower performance of the monolingual FTM, validates the assumption that bilingual fine-tuning enhances the model's adaptability to different linguistic contexts.

Fourthly, the significant improvement in recall indicates the bilingual FTM's

efficacy in identifying a higher proportion of the actual profanities within the text, thereby minimizing the occurrence of false negatives. This aspect holds significant importance in applications, such as profanity detection, where overlooking profane expressions, i.e., false negatives, can be more problematic than mistakenly flagging non-profane ones, i.e., false positives. Consequently, an advancement in recall signifies an enhanced capacity of the model to detect and identify all occurrences of profanities, making it more reliable for profanity detection tasks.

Lastly, the creation of manually annotated colloquial German and English datasets on word level is a notable contribution to the field. These datasets, which are crucial for fine-tuning and evaluating models in profanity detection, fill a significant gap in the availability of high-quality annotated data, particularly in the low-resource and code-switched domains. The datasets' uniqueness and the insights gained from their utilization demonstrate the potential for further research and application in multilingual and low-resource language settings. Moreover, annotating profanities at the word level enables precise and clear differentiation between single-word and multi-word expressions, a distinction that cannot be achieved with sentence-level annotations. This annotation approach also supports advanced technological solutions enabling more precise fine-tuning of classifiers.

The notable results for the profanity classification task utilizing a multilingual PLM for code-mixed content are consistent with recent research conducted by Sharma et al. (2024) and Ghosh et al. (2023). These studies have demonstrated that employing a multilingual BERT variant yields favorable outcomes in terms of precision, recall, and F1 scores for hate speech text classification in code-switched Hindi-English contexts. The findings of this thesis demonstrate that multilingual PLMs also achieve notable results for profanity detection in a code-switched environment at the word level, not just at the text level, with additional bilingual fine-tuning enhancing their performance even more. Since the bilingual FTM consistently outperformed the monolingual FTM, achieving significantly higher results in the German profanity category, and considering that the dataset was 80% German words, the findings suggest that the bilingual FTM excels in a code-switched but predominantly German environment.

The first research question was addressed by fine-tuning one XLM-R model on annotated data for colloquial German profanity detection at the word level, while a second XLM-R model was bilingual fine-tuned by incorporating English colloquial data along with the German data. The performances of these two fine-tuned models were compared to evaluate the extent to which the incorporation of English data enhances the efficiency of XLM-R in detecting profane expressions. For the second research question, zero-shot domain transfer experiments were conducted, and both models were compared within the domain of code-switched German rap lyrics to assess the extent to which the colloquial style, linguistic diversity, and heavy use of

English profanity words affect the efficiency of the FTMs.

Both experiments illustrate the significant benefits of integrating English data during the fine-tuning phase of XLM-R for profanity detection. The results underscore the bilingual FTM's enhanced performance in detecting profanity, evident across both the colloquial language and the code-switched German rap lyrics domain. This underscores the significance of employing bi- and multilingual fine-tuning methods to effectively manage the evolving nature of natural languages as well as the linguistic diversity in code-switched settings.

Despite the promising results, several challenges and limitations were encountered. The dynamic nature of natural language, including the emergence of new profane expressions, continually challenges the relevance and accuracy of hate speech and profanity detection models. As a result, the datasets created for the experiment of this thesis may need regular updates to reflect these linguistic changes. Furthermore, defining and annotating hate speech, offensive, and vulgar language inherently involves subjectivity, posing a significant challenge. Cultural and linguistic differences further complicate the annotation process, potentially leading to inconsistencies and biases in the data. Addressing these challenges requires developing more standardized and culturally sensitive annotation guidelines and incorporating diverse perspectives in the annotation process. Additionally, this reliance on manually annotated data underscores the need for more scalable and efficient annotation methods.

7 Conclusion

This thesis aimed to address the scarcity of labeled code-switched profanity detection data, alongside the challenge of the evolving nature of natural language and the resulting limitations of profanity detection using hard-coded profanity lists. These issues were addressed by utilizing the transfer learning capabilities of a multilingual PLM, as well as by taking advantage of the influence of English on German and the increasing prevalence of code-switching in everyday German language. Instead of relying on explicitly code-switched data, more readily available English colloquial data was used to enhance the German data and improve profanity detection performance in a code-switched low-resource domain. Moreover, the bilingual FTM exhibited strong performance in zero-shot domain transfer, a beneficial ability, as it allows the model to adapt and recognize newly coined words without needing specific prior fine-tuning or examples in that particular domain.

The findings of this thesis open a multitude of promising areas for future research. Based on the results for profanity detection in code-switched German and English data, extending the research to include additional code-switched language combinations will help to validate and generalize the findings. Future studies could explore the integration of more diverse code-switched datasets, encompassing various language pairs and contexts, to further enhance model performance and generalization capabilities. Especially the domain of German rap lyrics, influenced by various different minority languages, such as Russian or Turkish, presents an opportunity for exploring different language combinations and further investigating the efficacy of bilingual fine-tuning for profanity detection across diverse linguistic and cultural milieus. This could offer profound insights into the applicability and constraints of bilingual fine-tuning, enriching our understanding of its effectiveness.

Considering the impressive performance of the bilingual FTM trained on German and English in identifying German profanities within a predominantly German environment, further exploration in research could investigate the effectiveness of bi- or multilingual fine-tuning across various dialects and standard varieties of the German language. This could involve examining the diverse array of province-specific profanities in Austria. Investigation might encompass monolingual, bilingual, or multilingual fine-tuning involving languages spoken in the bordering regions of Austrian provinces, aiming to understand their effects on the detection of province-specific profanities and enhancing proficiency in identifying them.

The lack of high-quality, annotated datasets in low-resource languages and code-

switched contexts, particularly those with word-level annotations for profanity detection, remains a significant hurdle in advancing in this field. Although this thesis provides valuable data and insights, it is essential for the broader research community to continue investing in the development and dissemination of multilingual and code-switched datasets. To address the challenge of manual annotation, future research could focus on developing automated tools and techniques for annotating hate speech and profanity data. The use of crowd-sourcing methods could streamline the annotation process and improve scalability. Additionally, incorporating machine learning based approaches to refine and validate annotations could help mitigate subjectivity and bias in the data.

Bibliography

Charu C. Aggarwal (2018). *Neural Networks and Deep Learning: A Textbook*. Springer, Cham.

Jean Aitchison (2013). *Language Change: Progress or Decay?*, 4th edition. Cambridge University Press, Cambridge.

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama (2019). Optuna: A Next-generation Hyperparameter Optimization Framework. In *KDD '19: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, page 2623–2631, New York, USA. Association for Computing Machinery.

Saleh Al-Salman and Ahmad S. Haider (2021). COVID-19 Trending Neologisms and Word Formation Processes in English. *Russian Journal of Linguistics*, 25(1):24–42.

Iryna Aleksandruk, Oleksandra Palchevska, and Petro Hubych (2023). The Impact of Neologisms on the Development of the Modern Ukrainian Media Discourse of War. *Scientific Collection «InterConf+»*, 30(143):195–210.

Samy H. Alim (2006). *Roc the Mic Right: The Language of Hip Hop Culture*. Routledge, New York, London.

Nasser Alshammari and Saad Alanazi (2021). The Impact of Using Different Annotation Schemes on Named Entity Recognition. *Egyptian Informatics Journal*, 22(3):295–302.

Laith Alzubaidi, Jinshuai Bai, Aiman Al-Sabaawi, Jose Santamaría, A. S. Albahri, Bashar Sami Nayyef Al-dabbagh, Mohammed A. Fadhel, Mohamed Manoufali, Jinglan Zhang, Ali H. Al-Timemy, Ye Duan, Amjed Abdullah, Laith Farhan, Yi Lu, Ashish Gupta, Felix Albu, Amin Abbosh, and Yuantong Gu (2023). A Survey on Deep Learning Tools Dealing with Data Scarcity: Definitions, Challenges, Solutions, Tips, and Applications. *Journal of Big Data*, 10(1).

Jannis Androutsopoulos (2010). Multilingualism, Ethnicity and Genre in Germany's Migrant Hip Hop. In Marina Terkourafi, editor, *The Languages of Global Hip Hop*,

Advances in Sociolinguistics, pages 19–43. Continuum International Publishing Group, London.

Jannis Androutsopoulos and Arno Scholz (2002). On the Recontextualization of Hip-Hop in European Speech Communities: A Contrastive Analysis of Rap Lyrics. *Philologie im Netz*, 19(2002):1–42.

René Appel and Pieter C. Muysken (2006). *Language Contact and Bilingualism*. Amsterdam University Press, Amsterdam.

Stavros Assimakopoulos, Fabienne H. Baider, and Sharon Millar (2017). *Online Hate Speech in the European Union: A Discourse-Analytic Perspective*, 1st edition. Springer Open, Cham.

Taha Shangipour Ataei, Kamyar Darvishi, Soroush Javdan, Amin Pourdabiri, Behrouz Minaei-Bidgoli, and Mohammad Taher Pilehvar (2023). Pars-OFF: A Benchmark for Offensive Language Detection on Farsi Social Media. *IEEE Transactions on Affective Computing*, 14(4):2787–2795.

Peter Auer (1998). *Code-Switching in Conversation: Language, Interaction and Identity*. Routledge, London, New York.

Azad (2001). Samy De Bitch!! (7 Lektionen). In *Leben*. pelham power productions (3p).

Matthias Bachfischer, Uchenna Akujuobi, and Xiangliang Zhang (2018). KAUSTmine-Offensive Comment Classification on German Language Microposts. In *Proceedings of GermEval 2018, 14th Conference on Natural Language Processing (KONVENS 2018)*, pages 33–37. Verlag der Österreichischen Akademie der Wissenschaften.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio (2015). Neural Machine Translation by Jointly Learning to Align and Translate. In *3rd International Conference on Learning Representations, ICLR Conference Track Proceedings*, San Diego, CA, USA. ICLR.

Utsab Barman, Amitava Das, Joachim Wagner, and Jennifer Foster (2014). Code Mixing: A Challenge for Language Identification in the Language of Social Media. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 13–23, Doha, Qatar. Association for Computational Linguistics.

Eva Bartz, Thomas Bartz-Beielstein, Martin Zaefferer, and Olaf Mersmann (2023). *Hyperparameter Tuning for Machine and Deep Learning with R: A Practical Guide*, 1st edition. Springer Nature, Singapore.

Kristy Beers Fägersten (2012). *Who's Swearing Now? The Social Aspects of Conversational Swearing*. Cambridge Scholars Publishing, Newcastle upon Tyne.

Arie Ben-David (2007). A Lot of Randomness Is Hiding in Accuracy. *Engineering Applications of Artificial Intelligence*, 20(7):875–885.

Yoshu Bengio, Patrice Simard, and Paolo Frasconi (1994). Learning Long-Term Dependencies with Gradient Descent is Difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166.

James Bergstra and Yoshua Bengio (2012). Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research*, 13(10):281–305.

Michael Berry (2018). *Listening to Rap: An Introduction*, 1st edition. Routledge, Boca Raton, FL.

Kathrin Bower (2011). Minority Identity as German Identity in Conscious Rap and Gangsta Rap: Pushing the Margins, Redefining the Center. *German Studies Review*, 34(2):377–398.

Erman Boztepe (2003). Issues in Code-Switching: Competing Theories and Models. *Studies in Applied Linguistics and TESOL*, 3(2):1–27.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei (2020). Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Karl Bühler (1934). *Sprachtheorie: Die Darstellungsfunktion der Sprache*. Gustav Fischer, Jena.

Isabel Cachola, Eric Holgate, Daniel Preoţiuc-Pietro, and Junyi Jessy Li (2018). Expressively Vulgar: The Socio-dynamics of Vulgarity and Its Effects on Sentiment Snalysis in Social Media. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2927–2938, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Galo Castillo-López, Arij Riabi, and Djamé Seddah (2023). Analyzing Zero-Shot Transfer Scenarios across Spanish Variants for Hate Speech Detection. In *Tenth*

Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2023), pages 1–13, Dubrovnik, Croatia. Association for Computational Linguistics.

José Cañete, Gabriel Chaperon, Rodrigo Fuentes, Jou-Hui Ho, Hojin Kang, and Jorge Pérez (2020). Spanish pre-trained bert model and evaluation data. In *PML4DC at ICLR 2020*.

Bharathi Raja Chakravarthi, Ruba Priyadharshini, Shubanker Banerjee, Manoj Balaji Jagadeeshan, Prasanna Kumar Kumaresan, Rahul Ponnusamy, Sean Benhur, and John Philip McCrae (2023). Detecting Abusive Comments at a Fine-grained Level in a Low-Resource Language. *Natural Language Processing Journal*, 3.

Tannu Chauhan, Surbhi Rawat, Samrath Malik, and Pushpa Singh (2021). Supervised and Unsupervised Machine Learning based Review on Diabetes Care. In *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)*, volume 1, pages 581–585, Coimbatore, India. IEEE.

Siqi Chen, Yijie Pei, Zunwang Ke, and Wushour Silamu (2021). Low-Resource Named Entity Recognition via the Pre-Training Model. *Symmetry*, 13(5).

Yutian Chen, Xingyou Song, Chansoo Lee, Zi Wang, Richard Zhang, David Dohan, Kazuya Kawakami, Greg Kochanski, Arnaud Doucet, Marc' Aurelio Ranzato, Sagi Perel, and Nando de Freitas (2022). Towards Learning Universal Hyperparameter Optimizers with Transformers. In *Advances in Neural Information Processing Systems*, volume 35, pages 32053–32068. Curran Associates, Inc.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio (2014). On the Properties of Neural Machine Translation: Encoder–Decoder Approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov (2020). Unsupervised Cross-Lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample (2019). Cross-Lingual Language Model Pretraining. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Michele Corazza, Stefano Menini, Pinar Arslan, Rachele Sprugnoli, Elena Cabrio, Sara Tonelli, and Serena Villata (2018). Comparing Different Supervised Approaches to Hate Speech Detection. In *EVALITA Evaluation of NLP and Speech Tools for Italian*, pages 232–236.

Serafín M. Coronel-Molina and Beth L. Samuelson (2017). Language Contact and Translingual Literacies. *Journal of Multilingual and Multicultural Development*, 38(5):379–389.

Louis Alexander Cotgrove (2018). The Importance of Linguistic Markers of Identity and Authenticity in German Gangsta Rap. *Journal of Languages, Texts, and Society*, 2:67–98.

David Crystal (2010). *The Cambridge Encyclopedia of Language*, 3rd edition. Cambridge University Press, Cambridge [u.a.].

Andrea Dal Pozzolo, Olivier Caelen, and Gianluca Bontempi (2015). When is Undersampling Effective in Unbalanced Classification Tasks? In Annalisa Appice, Pedro Pereira Rodrigues, Vítor Santos Costa, Carlos Soares, João Gama, and Alípio Jorge, editors, *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2015 Porto, Portugal, September 7–11, 2015 Proceedings, Part I*, Lecture Notes in Artificial Intelligence, pages 200–215. Springer, Cham, Heidelberg, New York, Dordrecht, London.

Ofer Dekel and Ohad Shamir (2010). Multiclass-Multilabel Classification with More Classes than Examples. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9, pages 137–144. PMLR.

Margaret Deuchar (2020). Code-Switching in Linguistics: A Position Paper. *Languages*, 5(2).

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2019). BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Domo, Inc. (2023). Domo Resource - Data Never Sleeps 11.0. `https://www.domo.com/learn/infographic/data-never-sleeps-11`. Accessed: 09/05/2024.

Eduard Ștefan Dumitru and Virgil Tudor (2022). The Evolution of Hip Hop Culture. *Research & Science Today*, 24(2):223–238.

Carol M. Eastman (1992). Codeswitching as an Urban Language-Contact Phenomenon. In Carol M. Eastman, editor, *Codeswitching*, volume 89 of *Multilingual Matters*, pages 1–18. Channel View Publications Ltd., Bristol, Blue Ridge Summit.

Mai ElSherief, Caleb Ziems, David Muchlinski, Vaishnavi Anupindi, Jordyn Seybolt, Munmun De Choudhury, and Diyi Yang (2021). Latent Hatred: A Benchmark for Understanding Implicit Hate Speech. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 345–363. Association for Computational Linguistics.

Eminem (2002). Lose Yourself. In *8 Mile: Music from and Inspired by the Motion Picture*. Shady, Aftermath, Interscope.

Oliver Ettrich, Sven Stahlmann, Henrik Leopold, and Christian Barrot (2024). Automatically Identifying Customer Needs in User-Generated Content Using Token Classification. *Decision Support Systems*, 178:1–11.

Abolfazl Farahani, Behrouz Pourshojae, Khaled Rasheed, and Hamid R. Arabnia (2020). A Concise Review of Transfer Learning. In *2020 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 344–351, Las Vegas, NV, USA. IEEE.

Robert Faris, Amar Ashar, Urs Gasser, and Daisy Joo (2016). Understanding Harmful Speech Online. *Berkman Klein Center Research Publication*.

Matthias Feurer and Frank Hutter (2019). Hyperparameter Optimization. In Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren, editors, *Automated Machine Learning: Methods, Systems, Challenges*, The Springer Series on Challenges in Machine Learning, pages 3–33. Springer, Cham.

Csaba Földes (2010). Was ist Kontaktlinguistik? Notizen zu Standort, Inhalten und Methoden einer Wissenschaftskultur im Aufbruch. In Hubert Bergmann, Manfred Michael Glauninger, Evelyne Wandl-Vogt, and Stefan Winterstein, editors, *Fokus Dialekt. Analysieren–Dokumentieren–Kommunizieren. Festschrift für Ingeborg Geyer zum 60. Geburtstag (Germanistische Linguistik 199–201)*, pages 133–156. Georg Olms Verlag, Hildesheim, Zürich, New York.

Mikel L. Forcada (2017). Making Sense of Neural Machine Translation. *Translation Spaces*, 6(2):291–309.

Murray Forman (2002). *The 'Hood Comes First: Race, Space, and Place in Rap and Hip-Hop*. Wesleyan University. Press, Middletown, Connecticut.

Paula Fortuna and Sérgio Nunes (2018). A Survey on Automatic Detection of Hate Speech in Text. *ACM Computing Surveys*, 51(4):1–30.

Antigoni Founta, Constantinos Djouvas, Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Gianluca Stringhini, Athena Vakali, Michael Sirivianos, and Nicolas Kourtellis (2018). Large Scale Crowdsourcing and Characterization of Twitter Abusive Behavior. *Proceedings of the International AAAI Conference on Web and Social Media*, 12(1):491–500.

Robert Friedman (2023). Tokenization in the Theory of Knowledge. *Encyclopedia*, 3(1):380–386.

Penelope Gardner-Chloros (2009). *Code-Switching*. Cambridge University Press, Cambridge.

Penelope Gardner-Chloros (2020). Contact and Code-Switching. In Raymond Hickey, editor, *The Handbook of Language Contact*, 2nd edition, Blackwell Handbooks in Linguistics, pages 181–199. Wiley-Blackwell, Hoboken, NJ.

Koyel Ghosh, Apurbalal Senapati, Mwnthai Narzary, and Maharaj Brahma (2023). Hate Speech Detection in Low-Resource Bodo and Assamese Texts with ML-DL and BERT Models. *Scalable Computing. Practice and Experience*, 24(4):941–955.

Nuno Guimarães, Ricardo Campos, and Alípio Jorge (2024). Pre-Trained Language Models: What Do They Know? *WIREs Data Mining and Knowledge Discovery*, 14(1).

John Joseph Gumperz (1982). *Discourse Strategies*, 1st edition. Cambridge University Press, Cambridge [u.a.].

Vanessa Hahn, Dana Ruiter, Thomas Kleinbauer, and Dietrich Klakow (2021). Modeling Profanity and Hate Speech in Social Media with Semantic Subspaces. In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, pages 6–16. Association for Computational Linguistics.

Josiane F. Hamers and Michel H. A. Blanc (2000). *Bilinguality and Bilingualism*, 2nd edition. Cambridge University Press, Cambridge.

Einar Haugen (1956). *Bilingualism in the Americas: A Bibliography and Research Guide*. American Dialect Society [u.a.], University of Alabama.

Amir Hazem, Mérieme Bouhandi, Florian Boudin, and Beatrice Daille (2020). TermEval 2020: TALN-LS2N System for Automatic Term Extraction. In *Proceedings of the 6th International Workshop on Computational Terminology*, pages 95–100, Marseille, France. European Language Resources Association.

Dieter Herberg, Michael Kinne, and Doris Steffens (2012). *Neuer Wortschatz: Neologismen der 90er Jahre im Deutschen*. Walter De Gruyter GmbH, Berlin, Boston.

Herrad Heselhaus (2022). The Creative Use of Language in German Refugee Politics, 2015-2016. In Paul Iida, Timothy Reagan, John W. Schwieter, Cuhullan Tsuyoshi McGivern, and Jason Man-Bo Ho, editors, *Critical Perspectives on Teaching, Learning, and Society*, volume 8 of *Readings in Language Studies*, pages 281–301. Information Age Publishing, Inc., Charlotte, NC.

Sepp Hochreiter and Jürgen Schmidhuber (1997). Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Julia Hofweber, Theodoros Marinis, and Jeanine Treffers-Daller (2020). How Different Code-Switching Types Modulate Bilinguals' Executive Functions: A Dual Control Mode Perspective. *Bilingualism: Language and Cognition*, 23(4):909–925.

Eric Holgate, Isabel Cachola, Daniel Preoţiuc-Pietro, and Junyi Jessy Li (2018). Why Swear? Analyzing and Inferring the Intentions of Vulgar Expressions. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4405–4414, Brussels, Belgium. Association for Computational Linguistics.

Lichan Hong, Gregorio Convertino, and Ed Chi (2011). Language Matters In Twitter: A Large Scale Study. *Proceedings of the International AAAI Conference on Web and Social Media*, 5(1):518–521.

Jinpeng Hu, Yaling Shen, Yang Liu, Xiang Wan, and Tsung-Hui Chang (2022). Hero-Gang Neural Model For Named Entity Recognition. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1924–1936, Seattle, United States. Association for Computational Linguistics.

Nan Hu, Yike Wu, Guilin Qi, Dehai Min, Jiaoyan Chen, Jeff Z. Pan, and Zafar Ali (2023). An Empirical Study of Pre-Trained Language Models in Simple Knowledge Graph Question Answering. *World Wide Web*, 26(5):2855–2886.

Justus A. Ilemobayo, Olamide Durodola, Oreoluwa Alade, Opeyemi J. Awotunde, Adewumi T. Olanrewaju, Olumide Falana, Adedolapo Ogungbire, Abraham Osinuga, Dabira Ogunbiyi, Ark Ifeanyi, Ikenna E. Odezuligbo, and Oluwagbotemi E. Edu (2024). Hyperparameter Tuning in Machine Learning: A Comprehensive Review. *Journal of Engineering Research and Reports*, 26(6):388–395.

Md Saroar Jahan and Mourad Oussalah (2023). A Systematic Review of Hate Speech Automatic Detection Using Natural Language Processing. *Neurocomputing*, 546.

Roman Jakobson (1960). Closing Statement: Linguistics and Poetics. In Thomas A. Seboek, editor, *Style in Language*, pages 350–377. The Technology Press of Massacltusetts Institute of Techtology and John Wiley & Sons, Inc, London, New York.

Christian Janiesch, Patrick Zschech, and Kai Heinrich (2021). Machine Learning and Deep Learning. *Electronic Markets*, 31(3):685–695.

Timothy Jay (2000). *Why We Curse: A Neuro-Psycho-Social Theory of Speech*. John Benjamins Publishing Company, Philadelphia.

Vincent Williams Jonathan and Erwin Budi Setiawan (2023). Feature Expansion Using GloVe for Hate Speech Detection Using Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) Method in Twitter. In *2023 International Conference on Data Science and Its Applications (ICoDSA)*, pages 197–202. IEEE.

Navya Jose, Bharathi Raja Chakravarthi, Shardul Suryawanshi, Elizabeth Sherly, and John P. McCrae (2020). A Survey of Current Datasets for Code-Switching Research. In *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pages 136–141. IEEE.

Daniel Jurafsky and James H. Martin (2023). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Pearson Prentice Hall, Upper Saddle River, N.J.

Leslie Pack Kaelbling, Michael L. Littman, and Andrew W. Moore (1996). Reinforcement Learning: A Survey. *Journal of Artificial Intelligence Research*, 4:237–285.

Anila Kananaj and Rozana Rushiti (2024). Exploring the Linguistic Landscape of a Global Pandemic: Covid-19 Neologisms. *Revista de Gestão Social e Ambiental*, 18(8):1–13.

Sushim Kanchan and Abhay Gaidhane (2023). Social Media Role and Its Impact on Public Health: A Narrative Review. *Cureus*, 15(1).

Spyridon Kardakis, Isidoros Perikos, Foteini Grivokostopoulou, and Ioannis Hatzilygeroudis (2021). Examining Attention Mechanisms in Deep Learning Models for Sentiment Analysis. *Applied Sciences*, 11(9).

Simrat Kaur, Sarbjeet Singh, and Sakshi Kaushal (2021). Abusive Content Detection in Online User-Generated Data: A survey. *Procedia Computer Science*, 189:274–281.

Sugandha Kaur (2017). Word Naming in Bodo–Assamese Bilinguals: The Role of Semantic Context, Cognate Status, Second Language Age of Acquisition and Proficiency. *Journal of Psycholinguistic Research*, 46(5):1167–1186.

John D. Kelleher (2019). *Deep Learning*. MIT Press, Cambridge, Massachusetts.

Mohsin Khan (2013). Neologisms in Urdu A Linguistic Investigation of Urdu Media. *Language in India*, 13(6):818–826.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, Shruti Gupta, Subhash Chandra Bose Gali, Vish Subramanian, and Partha P. Talukdar (2021). MuRIL: Multilingual Representations for Indian Languages. *CoRR*, abs/2103.10730.

Annette Klosa-Kückelhaus and Ilan Kernerman (2022). *Lexicography of Coronavirus-Related Neologisms*. Walter de Gruyter, Berlin, Boston.

Annette Klosa-Kückelhaus (2022). German Corona-Related Neologisms and Their Lexicographic Representation. In Annette Klosa-Kückelhaus and Ilan Kernerman, editors, *Lexicography of Coronavirus-Related Neologisms*, volume 163 of *Lexicographica. Series Maior*, pages 27–42. Walter de Gruyter, Berlin, Boston.

Didem Koban (2013). Intra-Sentential and Inter-Sentential Code-Switching in Turkish-English Bilinguals in New York City, U.S. *Procedia - Social and Behavioral Sciences*, 70:1174–1179.

Philipp Koehn (2020). *Neural Machine Translation*. Cambridge University Press, Cambridge.

Evans Kotei and Ramkumar Thirunavukarasu (2023). A Systematic Review of Transformer-Based Pre-Trained Language Models through Self-Supervised Learning. *Information*, 14(3).

Natalie Kramar and Olga Ilchenko (2023). Neologisms in the Media Coverage of the Russia-Ukraine War in the Context of Information Warfare. *Studies about Languages/Kalbų studijos*, 43:14–28.

Max Kuhn and Kjell Johnson (2013). *Applied Predictive Modeling*, 1st edition. Springer Nature, New York, NY.

Gordana Lalic-Krstin and Nadezda Silaski (2018). From Brexit to Bregret: An Account of Some Brexit-Induced Neologisms in English. *English Today*, 34(2):3–8.

Sahinur Rahman Laskar, Bishwaraj Paul, Pankaj Dadure, Riyanka Manna, Partha Pakray, and Sivaji Bandyopadhyay (2023). English–Assamese Neural Machine Translation Using Prior Alignment and Pre-Trained Language Model. *Computer Speech & Language*, 82.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer (2020). BART: Denoising Sequence-to-Sequence Pre-Training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880. Association for Computational Linguistics.

John M. Lipski (2005). Code-Switching or Borrowing? No sé so no puedo decir, you know. In *Selected Proceedings of the Second Workshop on Spanish Sociolinguistics*, pages 1–15. Cascadilla Proceedings Project Somerville.

John T. Littlejohn and Michael T. Putnam (2010). Empowerment through Taboo: Probing the Sociolinguistic Parameters of German Gangsta Rap Lyrics. In Marina Terkourafi, editor, *The Languages of Global Hip Hop*, Advances in Sociolinguistics, pages 120–138. Continuum International Publishing Group, London.

Fangyu Liu, Qianchu Liu, Shruthi Bannur, Fernando Pérez-García, Naoto Usuyama, Sheng Zhang, Tristan Naumann, Aditya Nori, Hoifung Poon, Javier Alvarez-Valle, Ozan Oktay, and Stephanie L. Hyland (2023). Compositional Zero-Shot Domain Transfer with Text-to-Text Models. *Transactions of the Association for Computational Linguistics*, 11:1097–1113.

Xueqing Liu and Chi Wang (2021). An Empirical Study on Hyperparameter Optimization for Fine-Tuning Pre-Trained Language Models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2286–2300. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR*, abs/1907.11692.

Kelvin Luu, Daniel Khashabi, Suchin Gururangan, Karishma Mandyam, and Noah A. Smith (2022). Time Waits for No One! Analysis and Challenges of Temporal Misalignment. In *Proceedings of the 2022 Conference of the North*

American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 5944–5958, Seattle, United States. Association for Computational Linguistics.

Kosisochukwu Judith Madukwe and Xiaoying Gao (2019). The Thin Line Between Hate and Profanity. In *AI 2019: Advances in Artificial Intelligence: 32nd Australasian Joint Conference, Adelaide, SA, Australia, December 2-5, 2019, Proceedings*, Lecture Notes in Artificial Intelligence, pages 344–356, Cham. Springer International Publishing.

Christian Mair (2018). Stabilising Domains of English-language Use in Germany: Assessing the Interplay of Emancipation and Globalization of ESL Varieties. In Sandra C. Deshors, editor, *Modeling World Englishes: Assessing the interplay of emancipation and globalization of ESL varieties*, volume G61 of *Varieties of English Around the World*, pages 45–76. John Benjamins Publishing Company, The Netherlands.

Stefano Manfredi, Marie-Claude Simeone-Senelle, and Mauro Tosco (2015). Language Contact, Borrowing and Codeswitching. In Amina Mettouchi, Martine Vanhove, and Dominique Caubet, editors, *Corpus-based Studies of Lesser-described Languages: The CorpAfroAs Corpus of Spoken AfroAsiatic Languages*, volume 68 of *Studies in Corpus Linguistics*, pages 283–308. John Benjamins Publishing Company, Amsterdam, Philadelphia.

Yaron Matras and Evangelia Adamou (2020). Borrowing. In Evangelia Adamou and Yaron and Matras, editors, *The Routledge Handbook of Language Contact*, 1st edition, pages 237–251. Routledge, London.

Matthias R. Mehl and James W. Pennebaker (2003). The Sounds of Social Life: A Psychometric Analysis of Students' Daily Social Environments and Natural Conversations. *Journal of Personality and Social Psychology*, 84(4):857–870.

Rahul Mehta and Vasudeva Varma (2023). LLM-RM at SemEval-2023 Task 2: Multilingual Complex NER Using XLM-RoBERTa. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 453–456, Toronto, Canada. Association for Computational Linguistics.

Jürgen M. Meisel (1994). Coce-Switching in Young Bilingual Children: The Acquisition of Grammatical Constraints. *Studies in Second Language Acquisition*, 16(4):413–439.

Joo Er Meng, Rajasekar Venkatesan, and Wang Ning (2016). An Online Universal Classifier for Binary, Multi-class and Multi-label Classification. In *2016 IEEE*

International Conference on Systems, Man, and Cybernetics (SMC), Budapest, Hungary. IEEE.

Online Dictionary Merriam-Webster (2024). "lilliputian". `https://www.merriam-webster.com/dictionary/Lilliputian`. Accessed: 09/05/2024.

Carolina Mewengkang and Andi Hamzah Fansury (2021). Writing Daily Status on Social Media: Code-Mixing and Code-Switching Phenomena: A Literature Review. *Klasikal: Journal of Education, Language Teaching and Science*, 3(3):80–87.

Milica Mihaljević, Lana Hudeček, and Kristian Lewis (2022). Coronavirus-Related Neologisms: A Challenge for Croatian Standardology and Lexicography. In Annette Klosa-Kückelhaus and Ilan Kernerman, editors, *Lexicography of Coronavirus-Related Neologisms*, volume 163 of *Lexicographica. Series Maior*, pages 163–190. Walter de Gruyter, Berlin, Boston.

Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heintz, and Dan Roth (2024). Recent Advances in Natural Language Processing via Large Pre-Trained Language Models: A Survey. *ACM Computing Surveys*, 56(2).

Sudhanshu Mishra, Shivangi Prasad, and Shubhanshu Mishra (2021). Exploring Multi-Task Multi-Lingual Learning of Transformer Models for Hate Speech and Offensive Speech Identification in Social Media. *SN Computer Science*, 2(2).

Money Boy (2010). Dreh den Swag auf. In *Swagger Rap*. YouTube.

Money Boy (2014). *MC Fetti du Biersäufer*. YouTube.

Money Boy (2023). *Deppen die rappen*. YouTube.

Syrielle Montariol, Arij Riabi, and Djamé Seddah (2022). Multilingual Auxiliary Tasks Training: Bridging the Gap between Languages for Zero-Shot Transfer of Hate Speech Detection Models. In *Findings of the Association for Computational Linguistics: AACL-IJCNLP 2022*, pages 347–363. Association for Computational Linguistics.

Guido F. Montufar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio (2014). On the Number of Linear Regions of Deep Neural Networks. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Eduaro F. Morales and Hugo Jair Escalante (2021). A Brief Introduction to Supervised, Unsupervised, and Reinforcement Learning. In Alejandro A. Torres-García, Carlos Alberto Reyes Garcia, Luis Villasenor-Pineda, and Omar Mendoza-Montoya, editors, *Biosignal Processing and Classification Using Computational Learning and Intelligence: Principles, Algorithms, and Applications*, pages 111–129. Academic Press, London, UK.

Swapnanil Mukherjee and Sujit Das (2023). Application of Transformer-Based Language Models to Detect Hate Speech in Social Media. *Journal of Computational and Cognitive Engineering*, 2(4):278–286.

Natascha Müller, Laia Arnaus Gil, Nadine Eichler, Jasmin Geveler, Malin Hager, Veronika Jansen, Marisa Patuto, Valentina Repetto, and Anika Schmeißer (2015). *Code-Switching: Spanisch, Italienisch, Französisch. Eine Einführung*, 1st edition. Narr Francke Attempto Verlag, Tübingen.

Marissa Kristina Munderloh (2017). Rap in Germany – Multicultural Narratives of the Berlin Republic. In Uwe Schütte, editor, *German Pop Music*, volume 6 of *Companions to Contemporary German Culture*, pages 189–210. De Gruyter, Berlin, Boston.

Kevin P. Murphy (2012). *Machine Learning: A Probabilistic Perspective*. Adaptive Computation and Machine Learning Series. MIT Press, Cambridge, Massachusetts, London.

Pieter Muysken (2000). *Bilingual Speech: A Typology of Code-Mixing*. Cambridge University Press, Cambridge.

Carol Myers-Scotton (1997). *Duelling Languages: Grammatical Structure in Codeswitching*, 1st edition. Clarendon Press, Oxford.

Carol Myers-Scotton (2002). *Contact Linguistics: Bilingual Encounters and Grammatical Outcomes*, 1st edition. Oxford University Press, Oxford [u.a.].

Kensuke Nakamura and Byung-Woo Hong (2019). Adaptive Weight Decay for Deep Neural Networks. *IEEE Access*, 7:118857–118865.

Usman Naseem, Matloob Khushi, Vinay Reddy, Sakthivel Rajendran, Imran Razzak, and Jinman Kim (2021). BioALBERT: A Simple and Effective Pre-Trained Language Model for Biomedical Named Entity Recognition. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE.

OWID – Online Wortschatz Informationssystem Deutsch Leibniz-Institut für Deutsche Sprache Neologismenwörterbuch (2006ff.). "covidiot". `https://www.owid.de/docs/neo/listen/corona.jsp#covididiot`. Accessed: 09/05/2024.

Daniel Nkemelu, Harshil Shah, Irfan Essa, and Michael Best (2022). Tackling Hate Speech in Low-Resource Languages with Context Experts. In *Proceedings of the 2022 International Conference on Information and Communication Technologies and Development (ICTD2022)*, New York, USA. Association for Computing Machinery.

Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang (2016). Abusive Language Detection in Online User Content. In *Proceedings of the 25th International Conference on World Wide Web*, pages 145–153.

Debora Nozza and Dirk Hovy (2023). The State of Profanity Obfuscation in Natural Language Processing Scientific Publications. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3897–3909, Toronto, Canada. Association for Computational Linguistics.

Kriti Ohri and Mukesh Kumar (2021). Review on Self-Supervised Image Recognition Using Deep Neural Networks. *Knowledge-Based Systems*, 224.

Els Oksaar (1988). *Fachsprachliche Dimensionen.* Gunter Narr Verlag, Tübingen.

Gerhard Paaß and Sven Giesselbach (2023). *Foundation Models for Natural Language Processing: Pre-Trained Language Models Integrating Media*, 1st edition. Artificial Intelligence: Foundations, Theory, and Algorithms. Springer International Publishing, Cham.

Endang Wahyu Pamungkas, Valerio Basile, and Viviana Patti (2020). Do You Really Want to Hurt Me? Predicting Abusive Swearing in Social Media. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6237–6246, Marseille, France. European Language Resources Association.

Endang Wahyu Pamungkas, Valerio Basile, and Viviana Patti (2023). Investigating the Role of Swear Words in Abusive Language Detection Tasks. *Language Resources and Evaluation*, 57(1):155–188.

Mia Perlina and Mita Agustinah (2022). Code-mixing by a Content Creator Gita Savitri Devi: How and why? *Rainbow: Journal of Literature, Linguistics and Culture Studies*, 11(2):1–8.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer (2018). Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Carol W. Pfaff (1979). Constraints on Language Mixing: Intrasentential Code-Switching and Borrowing in Spanish/English. *Language: A Journal of the Linguistic Society of America*, 55(2):291–318.

Fabio Poletto, Valerio Basile, Manuela Sanguinetti, Cristina Bosco, and Viviana Patti (2021). Resources and Benchmark Corpora for Hate Dpeech Detection: A Systematic Review. *Language Resources and Evaluation*, 55:477–523.

Shana Poplack (1980). 'Sometimes I'll Start a Sentence in Spanish Y TERMINO EN ESPAÑOL': Toward a Typology of Code-Switching. *Linguistics*, 18(7):581–618.

Shana Poplack (1988). Contrasting Patterns of Codeswitching in Two Communities. In Monica Heller, editor, *Codeswitching: Anthropological and sociolinguistic perspectives*, volume 48 of *Contributions to the Sociology of Language*, pages 215–244. Mouton de Gruyter, Berlin, New York, Amsterdam.

Shana Poplack (2015). Code Switching: Linguistic. In James D. Wright, editor, *International Encyclopedia of the Social & Behavioral Sciences*, 2nd edition, pages 918–925. Elsevier, Oxford.

Shana Poplack and Marjory Meechan (1998). Introduction: How Languages Fit Together in Codemixing. *International Journal of Bilingualism*, 2(2):127–138.

Philipp Probst, Anne-Laure Boulesteix, and Bernd Bischl (2019). Tunability: Importance of Hyperparameters of Machine Learning Algorithms. *Journal of Machine Learning Research*, 20(53).

Elisabeth Putterer (2019). Die Neologismen der Flüchtlingskrise in der deutschen und ungarischen Presse. *Initium*, 1:189–222.

XiPeng Qiu, TianXiang Sun, YiGe Xu, YunFan Shao, Ning Dai, and XuanJing Huang (2020). Pre-Trained Models for Natural Language Processing: A Survey. *Science China. Technological Sciences*, 63(10):1872–1897.

Veenu Rani, Syed Tufael Nabi, Munish Kumar, Ajay Mittal, and Krishan Kumar (2023). Self-supervised Learning: A Succinct Review. *Archives of Computational Methods in Engineering*, 30(4):2761–2775.

Claudia Maria Riehl (2014). *Sprachkontaktforschung: Eine Einführung*, 3rd edition. Narr Verlag, Tübingen.

Julian Risch, Eva Krebs, Alexander Löser, Alexander Riese, and Ralf Krestel (2018). Fine-Grained Classification of Offensive Language. In *Proceedings of GermEval 2018, 14th Conference on Natural Language Processing (KONVENS*

2018), pages 38–44, Vienna, Austria. Verlag der Österreichischen Akademie der Wissenschaften.

Ethan Roberts (2024). Automated hate speech detection in a low-resource environment. *Journal of the Digital Humanities Association of Southern Africa*, 5(1).

Thomas Frank Rückstieß (2016). *Reinforcement Learning in Supervised Problem Domains*. Ph.D. thesis, Technische Universität München.

Andre Rusli, Alethea Suryadibrata, Samiaji Bintang Nusantara, and Julio Christian Young (2020). A Comparison of Traditional Machine Learning Approaches for Supervised Feedback Classification in Bahasa Indonesia. *International Journal of New Media Technology (IJNMT)*, 7(1):28–32.

Raied Salman and Vojislav Kecman (2012). Regression as Classification. In *2012 Proceedings of IEEE Southeastcon*, pages 1–6, Orlando, FL, USA. IEEE.

Joe Salmons (1990). Bilingual Discourse Marking: Code Switching, Borrowing, and Convergence in Some German-American Dialects. *Linguistics*, 28(3):453–480.

Gillian Sankoff (2002). Contact and Code-Switching. In J. K. Chambers, Peter Trudgill, and Natalie Schilling-Estes, editors, *The Handbook of Language Variation and Change*, Blackwell Handbooks in Linguistics, pages 638–668. Wiley-Blackwell.

Muriel Saville-Troike (2003). *The Ethnography of Communication: An Introduction*, 3rd edition. Blackwell Publishing, Malden.

Anna Schmidt and Michael Wiegand (2017). A Survey on Hate Speech Detection Using Natural Language Processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, pages 1–10, Valencia, Spain. Association for Computational Linguistics.

Monika Schwarz-Friesel (2013). *Sprache und Emotion*, 2nd edition. UTB GmbH A. Francke, Stuttgart Tübingen Basel.

Carol Myers Scotton and William Ury (1977). Bilingual Strategies: The Social Functions of Code-Switching. *Linguistics*, 15(193):5–20.

Deepawali Sharma, Aakash Singh, and Vivek Kumar Singh (2024). THAR- Targeted Hate Speech Against Religion: A High-Quality Hindi-English Code-Mixed Dataset with the Application of Deep Learning Models for Automatic Detection. *ACM Transactions on Asian and Low-Resource Language Information Processing*.

Shindy (2023). *Free Spirit*. Sony Music Entertainment.

Melanie Siegel and Markus Meyer (2018). h da Submission for the Germeval Shared Task on the Identification of Offensive Language. In *Proceedings of GermEval 2018, 14th Conference on Natural Language Processing (KONVENS 2018)*, pages 16–20, Vienna, Austria. Verlag der Österreichischen Akademie der Wissenschaften.

Leandro Silva, Mainack Mondal, Denzil Correa, Fabrício Benevenuto, and Ingmar Weber (2016). Analyzing the Targets of Hate in Online Social Media. *Proceedings of the International AAAI Conference on Web and Social Media*, 10(1):687–690.

Sonish Sivarajkumar and Yanshan Wang (2022). HealthPrompt: A Zero-Shot Learning Paradigm for Clinical Natural Language Processing. In *AMIA Annual Symposium Proceedings*, volume 2022, pages 972–981. American Medical Informatics Association.

Jasper Snoek, Hugo Larochelle, and Ryan P. Adams (2012). Practical Bayesian Optimization of Machine Learning Algorithms. In *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

Ivan Srba, Gabriele Lenzini, Matus Pikuliak, and Samuel Pecar (2021). Addressing Hate Speech with Data Science: An Overview from Computer Science Perspective. In Sebastian Wachs, Barbara Koch-Priewe, and Andreas Zick, editors, *Hate Speech-Multidisziplinäre Analysen und Handlungsoptionen: Theoretische und empirische Annäherungen an ein interdisziplinäres Phänomen*, pages 317–336. Springer Fachmedien, Wiesbaden.

Felix Stahlberg (2020). Neural Machine Translation: A Review. *The Journal of Artificial Intelligence Research*, 69:343–418.

Phoey Lee Teh and Chi-Bin Cheng (2020). Profanity and Hate Speech Detection. *International Journal of Information and Management Sciences*, 31(3):227–246.

Phoey Lee Teh, Chi-Bin Cheng, and Weng Mun Chee (2018). Identifying and Categorising Profane Words in Hate Speech. In *Proceedings of the 2nd International Conference on Computing and Data Analysis*, pages 65–69.

Marina Terkourafi (2010). Introduction: A Fresh Look at Some Old Questions. In Marina Terkourafi, editor, *The Languages of Global Hip Hop*, Advances in Sociolinguistics, pages 1–18. Continuum International Publishing Group, London.

Flor Miriam Plaza-del-arco, Debora Nozza, and Dirk Hovy (2023). Respectful or Toxic? Using Zero-Shot Learning with Language Models to Detect Hate

Speech. In *The 7th Workshop on Online Abuse and Harms (WOAH)*, pages 60–68, Toronto, Canada. Association for Computational Linguistics.

Sergios Theodoridis (2015). *Machine Learning: A Bayesian and Optimization Perspective*, 1st edition. Elsevier Science & Technology, San Diego, USA.

Sarah G. Thomason (2001). *Language Contact: An Introduction*. Edinburgh University Press, Edinburgh.

Aleksej Tikhonov (2020). Multilingualism and Identity: Polish and Russian Influences in German Rap. *Multiethnica: Journal of the Hugo Valentin Centre*, 40:55–66.

Peter Trudgill (2003). *A Glossary of Sociolinguistics*. Edinburgh University Press, Edinburgh.

Joseph P. Turian, Luke Shen, and I. Dan Melamed (2003). Evaluation of Machine Translation and Its Evaluation. In *Proceedings of Machine Translation Summit IX: Papers*, New Orleans, USA.

Robert Vacareanu, Enrique Noriega-Atala, Gus Hahn-Powell, Marco A. Valenzuela-Escarcega, and Mihai Surdeanu (2024). Active Learning Design Choices for NER with Transformers. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 321–334, Torino, Italia. ELRA and ICCL.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin (2017). Attention is All You Need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008.

Peter Vickers, Loic Barrault, Emilio Monti, and Nikolaos Aletras (2023). We Need to Talk About Classification Evaluation Metrics in NLP. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 498–510, Nusa Dua, Bali. Association for Computational Linguistics.

Zeljko Vujovic (2021). Classification Model Evaluation Metrics. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 12(6):599–606.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman (2018). GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *Proceedings of the 2018 EMNLP Workshop*

BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Benyou Wang, Qianqian Xie, Jiahuan Pei, Zhihong Chen, Prayag Tiwari, Zhao Li, and Jie Fu (2024). Pre-Trained Language Models in Biomedical Domain: A Systematic Survey. *ACM Computing Surveys*, 56(3).

Haifeng Wang, Jiwei Li, Hua Wu, Eduard Hovy, and Yu Sun (2023). Pre-Trained Language Models and Their Applications. *Engineering*, 25(6):51–65.

Shuohuan Wang, Jiaxiang Liu, Xuan Ouyang, and Yu Sun (2020). Galileo at SemEval-2020 Task 12: Multi-lingual Learning for Offensive Language Identification Using Pre-Trained Language Models. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1448–1455, Barcelona (online). International Committee for Computational Linguistics.

Wenbo Wang, Lu Chen, Krishnaprasad Thirunarayan, and Amit Sheth (2014). Cursing in English on Twitter. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing*, pages 415–425. ACM.

Zeerak Waseem, Thomas Davidson, Dana Warmsley, and Ingmar Weber (2017). Understanding Abuse: A Typology of Abusive Language Detection Subtasks. In *Proceedings of the First Workshop on Abusive Language Online*, pages 78–84, Vancouver, BC, Canada. Association for Computational Linguistics.

Rongxiang Weng, Heng Yu, Shujian Huang, Shanbo Cheng, and Weihua Luo (2020). Acquiring Knowledge from Pre-Trained Model to Neural Machine Translation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(5):9266–9273.

Gregor Wiedemann, Eugen Ruppert, Raghav Jindal, and Chris Biemann (2018). Transfer Learning from LDA to BiLSTM-CNN for Offensive Language Detection in Twitter. In *Proceedings of GermEval 2018, 14th Conference on Natural Language Processing (KONVENS 2018)*, pages 85–94, Vienna, Austria. Verlag der Österreichischen Akademie der Wissenschaften.

Michael Wiegand, Anastasija Amann, Tatiana Anikina, Aikaterini Azoidou, Anastasia Borisenkov, Kirstin Kolmorgen, Insa Kröger, and Christine Schäfer (2018a). Saarland University's Participation in the GermEval Task 2018 (UdSW) – Examining Different Types of Classifiers and Features. In *Proceedings of GermEval 2018, 14th Conference on Natural Language Processing (KONVENS 2018)*, pages 21–27, Vienna, Austria. Verlag der Österreichischen Akademie der Wissenschaften.

Michael Wiegand, Josef Ruppenhofer, Anna Schmidt, and Clayton Greenberg (2018b). Inducing a Lexicon of Abusive Words – a Feature-Based Approach. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1046–1056, New Orleans, Louisiana. Association for Computational Linguistics.

Michael Wiegand, Melanie Siegel, and Josef Ruppenhofer (2018). Overview of the GermEval 2018 Shared Task on the Identification of Offensive Language. In *Proceedings of GermEval 2018, 14th Conference on Natural Language Processing (KONVENS 2018)*, pages 1–10, Vienna, Austria. Verlag der Österreichischen Akademie der Wissenschaften.

Taylor L. Wilson (1953). "Cloze Procedure": A New Tool for Measuring Readability. *Journalism & Mass Communication Quarterly*, 30(4):415–433.

Shang Cheong Wong, Phoey Lee Teh, and Chi-Bin Cheng (2020). How Different Genders Use Profanity on Twitter? In *Proceedings of the 4th International Conference on Computing and Data Analysis*, pages 1–9.

Quirin Würschinger, Mohammad Fazleh Elahi, Desislava Zhekova, and Hans-Jörg Schmid (2016). Using the Web and Social Media as Corpora for Monitoring the Spread of Neologisms. The case of 'rapefugee', 'rapeugee', and 'rapugee'. In *Proceedings of the 10th Web as Corpus Workshop*, pages 35–43, Berlin. Association for Computational Linguistics.

Zhengzheng Xing, Jian Pei, and Eamonn Keogh (2010). A Brief Survey on Sequence Classification. *SIGKDD Explorations*, 12(1):40–48.

Sargam Yadav and Abhishek Kaushik (2023). Comparative Study of Pre-Trained Language Models for Text Classification in Smart Agriculture Domain. In Swagatam Das, Snehanshu Saha, Carlos A. Coello Coello, and Jagdish Chand Bansal, editors, *Advances in Data-driven Computing and Intelligent Systems: Selected Papers from ADCIS 2022*, Lecture Notes in Networks and Systems, pages 267–279. Springer.

Li Yang and Abdallah Shami (2020). On Hyperparameter Optimization of Machine Learning Algorithms: Theory and Practice. *Neurocomputing*, 415:295–316.

Qiang Yang, Yu Zhang, Wenyuan Dai, and Sinno Jialin Pan (2020). *Transfer Learning*. Cambridge University Press, Cambridge.

Maria Arshalouis Yapondjian (2005). *Using the Four Elements of Hip-Hop as a Form of Self-Expression in Urban Adolescents*. Ph.D. thesis, University of Hartford.

Wonjin Yoon, Jinhyuk Lee, Donghyeon Kim, Minbyul Jeong, and Jaewoo Kang (2020). Pre-Trained Language Model for Biomedical Question Answering. In Peggy Cellier and Kurt Driessens, editors, *Machine Learning and Knowledge Discovery in Databases: International Workshops of ECML PKDD 2019, Würzburg, Germany, September 16–20, 2019, Proceedings, Part II*, Communications in Computer and Information Science, pages 727–740. Springer, Cham.

George Yule (2023). *The Study of Language*, 8th edition. Cambridge University Press, Cambridge.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar (2019a). Predicting the Type and Target of Offensive Posts in Social Media. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pages 1415–1420, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar (2019b). SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media (OffensEval). In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 75–86, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Xuan Zhang and Kevin Duh (2020). Reproducible and Efficient Benchmarks for Hyperparameter Optimization of Neural Machine Translation Systems. *Transactions of the Association for Computational Linguistics*, 8:393–408.

Zhixue Zhao, Ziqi Zhang, and Frank Hopfgartner (2021). A Comparative Study of Using Pre-Trained Language Models for Toxic Comment Classification. In *Companion Proceedings of the Web Conference 2021*, page 500–507, New York, USA. Association for Computing Machinery.

Yifan Zhou (2020). A Review of Text Classification Based on Deep Learning. In *Proceedings of the 2020 3rd International Conference on Geoinformatics and Data Analysis*, pages 132–136, New York, USA. Association for Computing Machinery.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler (2015). Aligning Books and Movies: Towards Story-Like Visual Explanations by Watching Movies and Reading Books. In

2015 IEEE International Conference on Computer Vision (ICCV), pages 19–27. IEEE.

Tena Šinjori (2019). Neologisms Concerning Migration Processes: A Czech Example. *Colloquium: New Philologies*, 4(3):119–128.

Acronyms

AFN American Forces Network. 25

AI Artificial Intelligence. 35

ANN Artificial Neural Network. 31

ANNs Artificial Neural Networks. 5, 32

AUC Area Under the Curve. 48

BERT Bidirectional Encoder Representations from Transformers. 2, 31, 38–40, 49, 50, 70

BO Bayesian Optimization. 43

CLM Causal Language Modeling. 38

CNNs Convolutional Neural Networks. 1, 38, 49

DL Deep Learning. 1, 5, 30, 31, 43, 44, 46, 49

DNNs Deep Neural Networks. 11, 32, 33, 42

ELMo Embeddings from Language Models. 38

EU European Union. 10, 22, 23

FFNN Feed-Forward Neural Network. 13, 32, 36

FFNNs Feed-Forward Neural Networks. 33

FN False Negatives. 47, 48

FP False Positives. 47, 48

FSL Few-Shot Learning. 43, 44

FTM Fine-Tuned Model. 11, 13, 53, 59, 60, 63–67, 69–71, 73

FTMs Fine-Tuned Models. 3, 4, 13, 59–61, 63–65, 69, 71

GLUE General Language Understanding Evaluation. 46, 48

GPT Generative Pre-Trained Transformer. 38

GRU Gated Recurrent Unit. 34

HHN Hip Hop Nation. 26–28

HHNL Hip Hop Nation Language. 5, 13, 26–28

HPO Hyperparameter Optimization. 43

LSTM Long Short-Term Memory. 34, 35, 38, 49

mBERT Multilingual BERT. 40, 49, 50

MC Master of Ceremonies. 24

MCC Matthews Correlation Coefficient. 48

ML Machine Learning. 1, 5, 23, 29–31, 42–44, 47, 50

MLM Masked Language Modeling. 13, 38–40

MT Machine Translation. 34, 35, 38, 46

MuRIL Multilingual Representations for Indian Languages. 49, 50

NEIHS North-East Indian Hate Speech. 50

NER Named Entity Recognition. 13, 38, 46

NLP Natural Language Processing. 1, 4, 5, 7, 9–11, 31, 37, 38, 42, 47, 48, 50

NMT Neural Machine Translation. 43

NN Neural Network. 34

NNs Neural Networks. 33, 35, 45

OSL One-Shot Learning. 43, 44

PLM Pre-Trained Language Model. 3, 4, 42, 45, 58, 69, 70, 73

PLMs Pre-Trained Language Models. 1, 2, 5, 30, 31, 37, 38, 40–43, 70

RNNs Recurrent Neural Networks. 1, 33, 34, 38

RoBERTa Robustly optimized BERT approach. 31, 39

ROC Receiver Operating Characteristic. 48

RS Random Search. 43

SALT Social and Language Technologies. 8

Seq2Seq Sequence-to-Sequence. 34

SOTA state-of-the-art. 1, 3, 5, 38

SVMs Support Vector Machines. 50

THAR Targeted Hate Speech Against Religion. 49

TN True Negatives. 47

TP True Positives. 47, 48

U.S. United States. 25–28

UK United Kingdom. 10, 22, 23

XLM-R Cross-Lingual Language Model-RoBERTa. 2, 3, 5, 31, 39, 40, 53, 57–60, 70, 71

ZSL Zero-Shot Learning. 43, 44