Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

FAKULTÄT
FÜR MATHEMATIK, INFORMATIK
UND NATURWISSENSCHAFTEN

Language Technology

MASTERTHESIS

Enhancing Sentiment Analysis: Model Comparison, Domain Adaptation, and Lexicon Evolution in German Data

Robert Georg Geislinger

Language Technology
Department of Informatics
Faculty of Mathematics, Informatics and Natural Sciences

Universität Hamburg
Hamburg, Germany

A thesis submitted for the degree of

Master of Science (M. Sc.)
Matrikelnummer: 6947836
1st Examiner: Prof. Dr. Chris Biemann
2nd Examiner: Dr. Steffen Remus
Supervisor: Tim Fischer

Enhancing Sentiment Analysis: Model Comparison, Domain Adaptation, and Lexicon Evolution in German Data

Masters's Thesis submitted by: Robert Georg Geislinger

Date of Submission: 25.7.2024

Supervisor: Tim Fischer, M. Sc., Universität Hamburg

Committee:
1st Examiner: Prof. Dr. Chris Biemann, Universität Hamburg
2nd Examiner: Dr. Steffen Remus, Universität Hamburg

Universität Hamburg, Hamburg, Germany
Faculty of Mathematics, Informatics and Natural Sciences
Department of Informatics

Language Technology

Affidavit

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Arbeit selbst verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

I hereby declare upon oath that I have written the present thesis independently and have not used further resources and aids than those stated in the dissertation.

25.7.24

Date

R. Geisinger

Signature

(Robert Georg Geislinger)

Abstract

This study investigates multiple aspects of sentiment analysis on German datasets, comparison of various models, domain adaptation strategies with sparse labeled data, and automatic lexicon adaptation using advanced learning techniques. The initial findings demonstrate wide variation in model performance based on the dataset and domain, with models optimized on specific domains generally outperforming others. Progress between model generations shows promising enhancements, but no single model consistently outperforms across all domains, highlighting the need for model adaptation and fine-tuning.

Effective strategies for domain adaptation with sparse labeled data are analyzed. In-context learning demonstrates benefits, though its success varies across different models and datasets. Fine-tuning techniques like LoRA adapters show significant promise, especially in low-data situations. Training with the SetFit approach also produces positive outcomes but requires substantial computational resources. Both LoRA and SetFit techniques have potential for enhancing model performance in data-scarce environments; however, no single strategy is universally applicable to all models and datasets.

Large language models demonstrate promising results in creating domain-specific lexicons with prompting. The generated lexicons can outperform existing ones, increasing the possibility of creating domain-specific lexicons without the need for annotation. While embedding-based lexicon extension produces mixed outcomes, it can outperform traditional lexicons like GerVADER in certain datasets, indicating potential for leveraging the capabilities of large language models. Detailed analysis is made to gain insights into the challenges and benefits of this approach.

Contents

1 Introduction

In the digital age, user-generated content is widespread across social media platforms, online forums and digital communication channels. This has revolutionized the way individuals and organizations engage with the world. Authors can express their opinions on any public topic, including individuals, products, events, or ideas. This immense amount of data provides valuable opportunities for insights and analysis, particularly in understanding human emotions and opinions through sentiment analysis.

Research on sentiment analysis is motivated by many use cases for various domains, including social science, digital humanities, customer feedback, crisis detection and public health monitoring (Tan et al., 2022; Wankhade et al., 2022). Sentiment analysis in news articles helps researchers and policymakers understand how societal perspectives in the media evolve (Balahur et al., 2010). Sentiments on various topics can change quickly, especially in current political cases, or gradually shift over years and decades. For example, analyzing social media posts can help detect shifts in societal opinions during ongoing events (Thelwall et al., 2011; Rauh, 2018). In product reviews, aspect-based sentiment analysis provides detailed insights at a granular level. This detailed insight is crucial for product enhancements, development and for new marketing plans based on customer feedback (Zhang et al., 2023; van Kleef et al., 2015).

Sentiment analysis, also known as opinion mining, involves using computational techniques to identify and extract subjective information from textual data. It is a text classification task that belongs to the domain of natural language processing (NLP). It includes the automatic identification, aggregation, and classification of sentiments expressed in texts. The sentiment polarity of words, phrases, sentences, or entire texts can be classified into predefined categories, which can range from binary (positive, negative) to including a neutral category or even more nuanced classifications (Tsytsarau and Palpanas, 2012).

Sentiment analysis in written communication presents unique challenges due to the complexity and variability of language. Often contextual cues, linguistic patterns, and advanced techniques such as deep learning are utilized to categorize the expressed

sentiment. The written language lacks acoustic information, but other layers such as slang, sarcasm, opinions or emotions, can still be part of written communication. The text must not only be read word by word; the structure, order and both explicit and implicit meaning, are also important in understanding language (Norvig, 1987; X Hu and H Liu, 2012). Documents can differ in writing style and may contain spelling errors or ungrammatical sentences. When slang is used, it is not always possible for humans to determine the sentiment of a text at first glance (e.g., "This steak is the bee's knees!" which means the steak is excellent or outstanding). Normalization and error correction can be a preprocessing step to handle mistakes and convert slang to more clear sentences to create robust systems (Schouten and Frasincar, 2016). Sarcasm makes it difficult to determine whether a statement is meant seriously or not. This has led to its direction of research focusing on sarcasm (Ghosh et al., 2015). Sentiment is sometimes concealed within the structure of text (Polanyi and Zaenen, 2006). The representation of sentiment, or the way the author expresses their opinion, is closely tied to the domain and can depend on subtle nuances. Due to the variety of ways to express oneself, there are countless possibilities to convey sentiment. Utterances can hold sentiment explicitly stated (e.g., "I don't like dogs!"), subtly expressed (e.g., "Considering the price, the hotel was fine"), not necessarily limited to a single word (e.g., "The hotel was bad, but way better than I expected") or about various aspects (e.g., "The weather was bad, but the food was delicious"). The inherent flexibility and ambiguity in natural language can make it challenging to extract sentiment from a text, making it a field of ongoing research (Schouten and Frasincar, 2016).

Sentiment analysis employs various methodologies and techniques to interpret and classify sentiment within textual data. These approaches primarily fall into three categories: rule-based lexical methods, machine learning and large language models.

Lexicon-based methods utilize dictionaries containing positive and negative words. These dictionaries can also contain values representing the strength of polarity. The scale typically ranges from a negative value for the most negative sentiment to a positive value for the most positive. Dictionaries can be human-created, semi-automatically generated or automatically generated. The term semi-automatically refers to the creation of lexicons by humans supported by NLP methodologies, whereas the term automatically refers to methods that generate lexicons without human involvement (Remus et al., 2010). When calculating the sentiment of a text, each word is assigned a value from the dictionary or zero if the word is not present in the lexicon. Rule-based systems use algorithms to determine common linguistic expressions connected to sentiment, such as negation or intensifiers. If the total score after summing the values is above a certain threshold, the text is categorized as positive; if the score is below a certain threshold, it is categorized as negative. Otherwise, it is categorized as neutral. Lexicons require only a small set of data to yield good results, but creating domain-specific lexicons requires extensive domain knowledge or substantial computational resources. Creating rules to capture linguistic details is time-consuming, but the results are transparent, comprehensible, and computationally inexpensive. The sentiment of words can vary or even reverse across different domains. Without considering context, this technique can fail to determine the sentiment accurately, not handling linguistic patterns such as phrases or nuances due to the inflexibility to unknown data and structure (Tymann et al., 2019).

Machine learning models are statistical models and encompass a wide range of techniques and algorithms. This includes classical approaches such as Naïve Bayes Classifier (NBC) and more sophisticated approaches like Neural Networks (NN). Both are trained on large amounts of data with the goal of creating a generalized model (Witten et al., 2011). While NBC, based on the Bayes rules, learns conditional probabilities out of the data, NN learns patterns from the training data. Both, NBC and NN, heavily depend on training data. Like lexicons, NBC may not require extensive computational training but may struggle with unknown words or linguistic features (Jung et al., 2016). Achieving good results with NNs requires significant computational resources. This makes them time-consuming and costly to create, but unlike lexicons, they do not heavily depend on domain knowledge. This helps to generalize patterns in the given data and also classify unseen data accurately. While this makes them effective for the specific topics they are trained on, they yield lower results when faced with domain shifts (Guhr et al., 2020). Furthermore, they can be utilized to create or adapt existing lexicons (Li and Shah, 2017). Because machine learning models are not transparent in their decision-making processes, the underlying reasons for the results are difficult to observe. Due to the drastic increase in the usage of machine learning models lately, gaining better insight into the decision-making processes has become more crucial. The decision-making process of machine learning models is mostly opaque, producing results without a comprehensible explanation. Since the result can have a significant impact on decision-making in some domains, explainability is an ongoing field of research (Zielinski et al., 2023).

Large language models (LLMs), based on the transformer architecture, represent the state-of-the-art (SOTA) in NLP. They have gained prominence over the past few years. Transformers are built on attention mechanisms, which allow them to incorporate contextual information like slang, hashtags, and sarcasm (Kheiri and Karimi, 2023). Their large number of parameters allows them to learn more intricate structures from the provided data. Achieving satisfactory results requires a substantial amount of data, enabling LLMs to learn diverse patterns. In addition to classification, they support natural language queries, expressed in human language and offer human-readable reasoning for their decisions. Pre-trained LLMs can be fine-tuned for new tasks or adapted using few-shot learning techniques to accommodate domain shifts. Training LLMs requires high-end computational hardware, which is challenging for small research groups or companies seeking to adapt them (Hu et al., 2022). Some models are not freely available and can only be accessed through a proprietary programming interface. This incurs costs for each use. Processing data in the provider's data center raises concerns about the privacy of personal and business information (Sebastian, 2023).

Adapting sentiment analysis to the diverse and dynamic nature of modern communication presents unique challenges. In modern communication, elements such as emojis, abbreviations and hashtags convey valuable information beyond solely negotiation or sarcasm. Sentiment varies across domains; for instance, movie reviews use different linguistic characteristics than news articles. Furthermore, accounting for the cultural context is essential when analyzing texts. The expression of opinions varies among cultural groups, countries, continents, and between native and non-native speakers. The mentioned methods face challenges when adapting to new domains. Creating dictionaries for lexical-based approaches is tedious and requires domain knowledge. Training a machine learning model for a specific domain requires extensive domain data and significant computational power, even if the model is pre-trained. Data must

be well-labeled or preprocessed, a process that can be tedious and time-consuming. LLMs are trained on billions of words and can handle a many tasks out of the box, but they need fine-tuning to achieve decent results. LLMs initially require significant computational power to run, and even more for fine-tuning (Wankhade et al., 2022). In conclusion, there is no one model that fits all application scenarios.

1.1 Research Questions

The hypothesis of this thesis is that LLMs can outperform traditional sentiment analysis approaches that rely on lexicons or machine learning. Although there are available out-of-the-box (OOTB) models, they often are not well-suited for specific domains. There is no universal state-of-the-art solution that brings satisfying results for every domain, highlighting the need for domain-specific adaptation. This leads to the following research questions, which will be addressed in this thesis:

RQ1: Which model achieves the highest performance in sentiment analysis on German datasets? Models are often trained on freely available data such as social media posts, leveraging web crawling for its diverse and cost-effective nature without incurring license expenses. Since most texts on the internet are in English, the highest-performing models also are typically designed for the English language. Understanding common errors made by available models is necessary because they often struggle with domain and language switching. This approach helps to identify the best-performing OOTB models and provides insights into common error sources. To address this question, various models are evaluated and compared on German datasets. A detailed error analysis is conducted to understand common errors among the models, identify similarities and interpret the causes of these errors.

RQ2: Which strategies prove effective for domain adaptation in sentiment analysis, particularly when there is a scarcity of labeled data in the target domain? Domain adaptation is crucial because no one-size-fits-all model exists. Since many domains have limited labeled data for sentiment analysis, achieving satisfactory results through domain adaptation can be challenging. What is the best strategy for adapting an existing model without extensive data or computational resources? Ultimately, an effective technique should facilitate rapid model adaptation. To answer this question, different approaches of fine-tuning the models are compared and tested. This involves selecting promising models from *RQ1* and fine-tuning it on a domain-specific dataset, while evaluating multiple adaptation approaches.

RQ3: How can lexicons be automatically adapted, generated or updated using weak or unsupervised learning techniques? The hypothesis is that machine learning-based approaches outperform lexical-based methods according to common metrics in the field of NLP. Lexicons on the other hand are computationally inexpensive, transparent and can be easily shared for proper repeatability. Is there a way to generate or adapt a domain-specific lexicon with weak or unsupervised learning? To accomplish this, the best-known models are employed and unsupervised or weak learning techniques are applied to generate lexicons for specific domains.

2
Background

2.1 Sentiment analysis

The study of people's expressions of emotions, opinions, or stances about a topic, known as sentiment analysis, is part of Natural Language Processing (NLP). The words do not only have polarity — most often positive or negative — but also a valence, which is the intensity of the word or emotion. Polarity and valence are combined in sentiment analysis to classify text as positive, neutral, or negative. Sentiment analysis is used in various domains, such as in the analysis of political views or communities, product or movie reviews, news article categorization, psychological or social sciences, and the analysis of natural disasters or events (Schmidt et al., 2022; Shalunts et al., 2014; Jung et al., 2016). The domain is described as the context in which the writer expresses their opinion. This can, for example, include social media posts, which are mostly short, opinionated, and written from a personal perspective; news articles, which are more formal with longer sentences and describe situations or events in greater detail; movie or article reviews, which are more descriptive and opinionated; or statements from political parties or companies in advertisements, which are concise to influence public opinion or consumer behavior.

Sentiment analysis is a text classification task with different levels of detail: aspect, phrase, sentence, or document. Document-level sentiment analysis is used to analyze the sentiment of entire documents. This can be useful for analyzing book chapters. Sentence-level sentiment analysis is used to determine the sentiment for each sentence independently. Texts with a wide range of sentiments can benefit from this. Sentences can contain multiple phrases, each of which can include one or more aspects. Classification is done at the phrase level. It is often utilized for analyzing longer product reviews. Aspect-level sentiment analysis is the most detailed method and is used to classify individual aspects of a sentence, where each sentence can contain multiple aspects. A single sentence can contain multiple aspects with different sentiments (Wankhade et al., 2022).

How polarity is expressed can vary between writers, topics, and domains. Words that are used to express positive sentiment in one domain can be used to express negative sentiment in another domain. For example, the German word "scharf" can

mean “seared” as a cooking technique for meat or ’sharp’ in the context of edges. “Das Steak ist scharf angebraten” means “The steak is seared”, which would be considered a good cooking technique. “Die Ecken sind sehr scharf!” means “The corners are very sharp!” which is intended as a caution.

Additionally, elements such as sarcasm, slang, punctuation, capitalization, negation, or, especially for social media, emoticons, tags, or abbreviations can significantly affect sentiment and even reverse it (Kheiri and Karimi, 2023). The most common methods for sentiment analysis include rule-based approaches with word lists, classical machine learning techniques such as the Naïve Bayes Classifier, neural network methods like BERT, or the use of large language models.

2.2 Lexicon

The baseline for sentiment analysis is often the lexicon-based method, which is based on affect lexicons. Affect lexicons are word lists consisting of words, their inflections and are labeled with their prior polarity with a value. Inflection is the technique of changing a word to fit different grammatical categories, such as tense, number, gender, and case, without changing its main meaning or part of speech. Each word is assigned with a value between -1 and 1, representing the polarity (negative values represent negative sentiment and vice versa). The higher the positive value, the more positively the word is marked; the lower the value, the more negatively the word is marked. The most straightforward implementation is to check every word in the document for the corresponding value in the lexicon. Assign the word the value if it exists in the lexicon, and afterward sum the values up. It is computationally inexpensive because, for classification, two word lists are needed: one with positive words and the other with negative words, both with a value representing their sentiment strength. They have the advantage of transparent and reproducible results, but one of the main disadvantages is their inflexibility. As lexicon-based methods are rule-based, the outcomes are fully comprehensible. Since these methods lack context when implemented directly, the lexicons need to fit the domain of the text; otherwise, the classification can not only be inaccurate but also false (Polanyi and Zaenen, 2006). Thus, phrases cannot be detected. The task of creating lexicons can be tedious and requires domain knowledge. In English, there are solutions like General Inquirer (GI), which has one of the oldest lexicons, extended several times. For classification, the document is split into single words and compared to the lexicon. If the word is listed in one of the lexicons, a value is assigned to it. Afterward, the values are summed together, and if the overall score is negative, the document is classified as negative, and vice versa (Khoo and Johnkhan, 2018).

To take context into account, rule-based algorithms are used. This does not help for phrases (“Alles in Butter” which means “all right”), specific sentence orders, or unseen phenomena like slang or irony. Every language construct needs to be caught by rules. This can make it difficult for lexicons to take context into account, since grammar structures differ from those of other languages. Negations can occur after the subject to which they are connected, or there can be multiple words between the negation and the subject it refers to, which makes them difficult to cover with rules (Tymann et al., 2019). Since words can have another sentiment regarding their context, this can also be only addressed by handcrafted lexicons or rules (Shaukat et al., 2020).

SentiWS[1] is short for "Sentiment Wortschatz" (Sentiment vocabulary) and is a publicly available resource for sentiment analysis for the German language. It consists of adjectives, adverbs, and nouns explicitly and implicitly containing a sentiment. It contains about 1,650 negative and 1,800 positive words and their inflections, which makes about 16,000 positive and 16,000 negative forms. It is crafted by collecting sentiment bearing words from different sources. Sources were the General Inquirer lexicon automatically translated into German, words which often occur in positive or negative documents, the German Collocation Dictionary, and manually added words from the financial domain (Remus et al., 2010).

GerVADER[2] is the German adaption of VADER (for Valence Aware Dictionary for sEntiment Reasoning), an implementation of the lexicon approach. GerVADER, like VADER, also adapts heuristics and multiple linguistic features common in the domain of social media. Abbreviations or emoticons were copied from VADER because both are common and identical in both languages (like "lol" or "rotfl"). GerVADER is a rule-based system that relies on SentiWS, which has been enhanced with additional words and rules for capitalized words (Hutto and Gilbert, 2014; Tymann et al., 2019).

2.2.1 TF-IDF

TF-IDF, short for term frequency-inverse document frequency, is a statistical measure that assesses the importance of a term in a document relative to a corpus of documents. The term frequency component quantifies how often a term appears in a specific document and helps to understand its relevance in the context of the document. Words that frequently appear within a document are considered more significant in conveying information about that document's content. The logarithm is often used to normalize the results and account for varying weights. The calculation for term frequency tf of a term t within a document d is typically computed as follows:

$$tf_{t,d} = log(count(t, d) + 1)$$

Document frequency df_t reflects the occurrences of a term t across multiple documents, measuring how frequently used the term is within the corpus. Terms that are present in numerous documents may not contribute significantly to categorizing individual documents. Inverse document frequency idf is utilized to assign weight to terms based on their presence across the corpus. Terms occurring less frequently across documents receive higher weight. The formula for calculating the inverse document frequency idf_t for the term t using the weight N (representing the total number of documents) is typically expressed as:

$$idf_t = log\frac{N}{df_t}$$

Common words in the document collection receive a low score, as their information about the particular document is likely to be low. This technique is commonly employed in various areas of NLP, including information retrieval and classification tasks (Jurafsky and Martin, 2024).

1. https://wortschatz.uni-leipzig.de/de/download/#sentiWSDownload
2. https://github.com/KarstenAMF/GerVADER

c-TF-IDF

To analyze topic representations within cluster documents in a collection, the cluster-based TF-IDF (c-TF-IDF) approach is utilized. When applying the c-TF-IDF technique, documents within a cluster are concatenated to form a single unified document representation of the entire cluster. The TF-IDF formula is adjusted when implementing c-TF-IDF to accurately capture the importance of terms within the cluster as a whole. The modification involves incorporating the average number of words per class A, in the calculation to normalize the term's importance relative to the total word distribution across all classes. To represent the importance of a term t in a class c, the average number of words per class A is divided by the term frequency tf_t as follows:

$$tf_{t,c} = log(1 + \frac{A}{tf_t})$$

By adjusting the TF-IDF formula and considering the cluster-level representations, it becomes possible to generate topic word distributions for each cluster in the collection. This is useful in identifying the most informative words specific to each cluster (Grootendorst, 2022).

2.3 Machine learning

While traditional lexicon-based methods have been standard, advances in technology have introduced the benefits of machine learning techniques. Machine learning (ML) covers several methods (Witten et al., 2011). A classical ML approach is the Naïve Bayes Classifier, a statistical classifier based on Bayes' theorem, which classifies the most likely class, assuming each feature is independent (Rish, 2001). Nowadays, modern machine learning models are mostly based on Artificial Neural Networks (ANN) which are inspired by the biological nervous system. They are built with neurons, which are structured layerwise, with interconnections between the layers. The input data is initially loaded into the input layer and sequentially passed through the internal hidden layers until it reaches the output layer. Each internal hidden layer processes the input it receives from the previous layer, using assigned weights to generate its own output, which is then passed to the next layer. The weights, which are the trainable parameters of the model, are adjusted during the training process to minimize error and maximize the model's performance. The model is trained using either supervised or unsupervised learning techniques to learn based on given input and output data. In supervised learning, labeled data is used, where each input has an assigned label that the model is supposed to predict. The model receives the input, produces an output, and utilizes the training label to correct itself and minimize the prediction error. In unsupervised learning, the model learns to identify patterns and structures from unlabeled data. The tasks can include, but are not limited to, identifying objects in images or generating answers to input questions (O'Shea and Nash, 2015; Brown et al., 2020). The network architecture can vary, depending on the use case. Architectures like Long Short-Term Memory (LSTM) have interconnections to the next layer and also to the current layer, giving the possibility of traversing input data through multiple steps. When analyzing sentences, LSTM can consider previous input tokens, while Bi-LSTM lets it consider both before and after the actual token (Hochreiter and Schmidhuber, 1997; Tabinda Kokab et al., 2022).

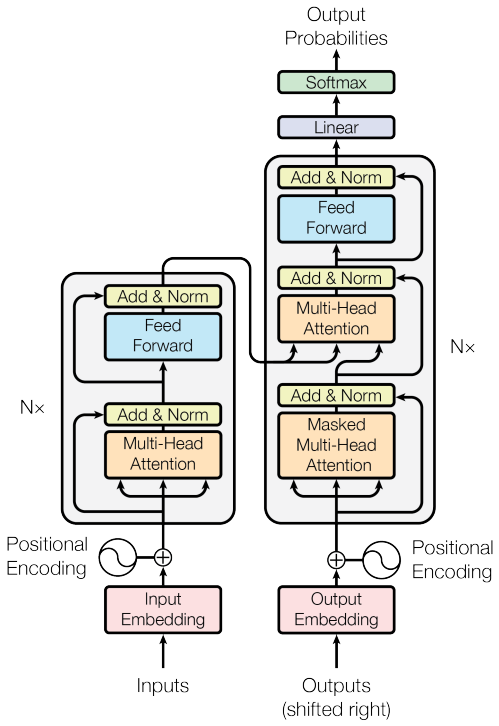


Figure 2.1: Standard structure of the transformer architecture (Vaswani et al., 2017)

2.3.1 Transformer

Language models nowadays are most commonly based on the transformer architecture and trained on a huge number of sentences, learning diverse patterns in the data, which makes them flexible and enables them to outperform previous approaches (Devlin et al., 2019). While transformers share the attention mechanism with earlier architectures, they are generally considered a fundamentally new architecture. Context-sensitive architectures like LSTM or Bi-LSTM calculate a sequence of hidden states at each computation step, considering the previous states (and future states for Bi-LSTM). The transformer architecture avoids sequential processing and instead relies entirely on an attention mechanism, enabling it to learn global dependencies between input and output sequence. The attention mechanism involves parameters that enable the calculation of attention scores, determining the relative importance of different parts of the input sequence. The structure of the transformer model is illustrated in Figure 2.1 with the encoder on the left side and the decoder on the right side (Vaswani et al., 2017).

The standard transformer architecture consists of a decoder and an encoder, both of which can also be utilized independently. An encoder maps input symbols into an internal numerical representation and sequentially transforms the input data into higher-level representations of entire sequences.

In contrast, decoder blocks operate in an autoregressive manner, generating each token one at a time based on previous tokens. Autoregression refers to generating the next token by depending on all previous generated symbols and the input sequence, meaning token n depends on the input and the previously generated tokens 1 to $n-1$. In contrast, encoder-decoder architectures can utilize the full context of the input data in both encoding and decoding phases (Vaswani et al., 2017). This autoregressive architecture is often referred to as Causal Language Models (Causal LMs) and is particularly

beneficial for text generation tasks. Causal LLMs are trained by predicting the next token in a sequence based on all preceding tokens. Large language models are mostly based on a decoder-only architecture due to generative performance (Gemma Team et al., 2024; AI@Meta, 2024; Brown et al., 2020). The BERT architecture is based on encoder blocks (Devlin et al., 2019).

BERT

BERT (short for Bidirectional Encoder Representations from Transformers) is a neural network architecture based on the transformer architecture, specifically utilizing the encoder blocks within that structure. BERT is trained on extensive text corpora to understand and learn the contextual relationships within the training data. This enables the model to create context-aware word embeddings, unlike models like GloVe, which always generate the same embedding for a word regardless of its meaning in the sentence (Devlin et al., 2019; Pennington et al., 2014). The encoder structure of BERT is particularly beneficial for classification tasks, as it allows the model to analyze and classify the text after fully processing it. The BERT architecture includes training for next sentence prediction (NSP) to learn the context between two sentences and determine if one follows the other. BERT models undergo pretraining on a large corpus of unlabeled data, using a technique called masking. Masking replaces certain tokens in the input with a mask token. The network is trained to predict the original tokens that were masked. After this pretraining phase, BERT models are fine-tuned on labeled data for specific downstream tasks (Devlin et al., 2019) There are several variations of the BERT architecture available, including RoBERTa and DistilBERT. RoBERTa (A Robustly Optimized BERT Pretraining Approach) does not include NSP training and utilizes different masking strategies to improve robustness (Liu et al., 2019). DistilBERT is a compact model trained using knowledge distillation, which enables it to replicate the behavior of a larger teacher model (Sanh et al., 2019).

The BERT architecture is utilized by the model trained by Guhr et al. (2020). The pre-trained model, called “German BERT Cased small” is trained on a diverse range of texts including Wikipedia, German law documents, and news articles. The model is fine-tuned for the classification task on different datasets in the domain of customer reviews (hotel, apps, movies), GermEval, social media (political, general), and Wikipedia (Guhr et al., 2020).

Another frequently used sentiment analysis model, based on the monthly downloads on Huggingface, is trained by the user Lxyuan[3]. The model is based on the DistilBERT architecture. The teacher model employed in the distillation phase is trained using transfer learning techniques (Laurer et al., 2024; Lxyuan, 2023).

SBERT

SBERT, short for Sentence-BERT, modifies the BERT architecture to enable faster sentence comparisons based on semantic similarity while maintaining nearly the same accuracy. This architecture is also known as a sentence transformer. For this, the SBERT architectures use a siamese network architecture, creating the embeddings for two sentences by two identical networks. Embeddings are internal vector representations created by the model, encapsulating the information extracted from the input. The

3. https://huggingface.co/models?language=de&sort=trending&search=sentiment accessed on 1.5.2024

embeddings are compared by calculating their cosine similarity to measure their relative distance. Since BERT creates embeddings for individual tokens, SBERT adds a pooling operation, such as mean pooling, to the output of BERT or RoBERTa to provide fixed-size sentence embeddings. The model is trained so that semantically similar inputs produce more similar embeddings, whereas semantically different inputs produce more distinct embeddings (Reimers and Gurevych, 2019).

Google Gemma

Google Gemma is a model family of LLMs developed by Google. The models are available in two parameter sizes: 2 billion (referred to as 2B) and 7 billion (7B). The 2B model is trained on 3 trillion tokens and the 7B model is trained on 6 trillion tokens. The primary training data language is English. Instruct fine-tuned models are also available, referred to as 2B Instruct and 7B Instruct. The context length for all models is 8,192 tokens. For positional encoding, Rotary Positional Encoding (RoPE) is used to calculate a vector of the absolute position of a token and the relative distance between a second token to calculate the attention. RMSNorm is used to stabilize the training by normalizing different layers (Gemma Team et al., 2024).

Meta AI Llama

Llama is a LLM family introduced by Meta AI. The models are based on the Llama architecture, introduced with Llama 1. The architecture is based on the decoder blocks of the transformer architecture. The Llama 2 models are available in four parameters sizes: 7 billion (referred to as 7B), 13 billion (13B), 34 billion (34B) and 70 billion (70B). There are also available as an instruct fine-tuned version, referred to as 7B Chat, 13B Chat, 34B Chat and 70B Chat. The Llama 2 architecture has an extended context length of 4,096 tokens and Grouped Query Attention (GQA) for the 34B and 70B model for lower memory usage. The Llama 2 architecture also uses RoPE (Touvron et al., 2023). In Llama 3, models are available with a parameter size of 8 billion (8B) and 70 billion (70B), and the context length extended from 4,096 tokens to 8,192 tokens. Furthermore, GQA is now used in all sizes compared to Llama 2. Llama 3 architecture also uses RMSNorm. The Llama 3 models are available in two parameter sizes of 8 billion (8B) and 70 billion (70B). The models are fine-tuned for instruction-following tasks and are referred to as 8B Instruct and 70B Instruct. The models are trained on 15 trillion tokens, of which are 5% other languages than English (AI@Meta, 2024).

MistralAI Mistral

Mistral is a LLM model family introduced by MistralAI. The models are available in parameter sizes of 7 billion (referred to as 7B) and 22 billion parameters (22B). Both parameter sizes are also available as instruct fine-tuned versions, referred to as 7B Instruct and 22B Instruct. All models are also available in a mixture-of-experts version, containing 8 distinct groups of parameters, called experts, on which the model decides which specific block to use. These models are referred to as 8x7B for the 7 billion parameters version and 8x22B for the 22 billion parameters version. The model architecture uses GQA and Sliding Window attention (SWA). SWA handles longer input with reduced computational cost (Jiang et al., 2023).

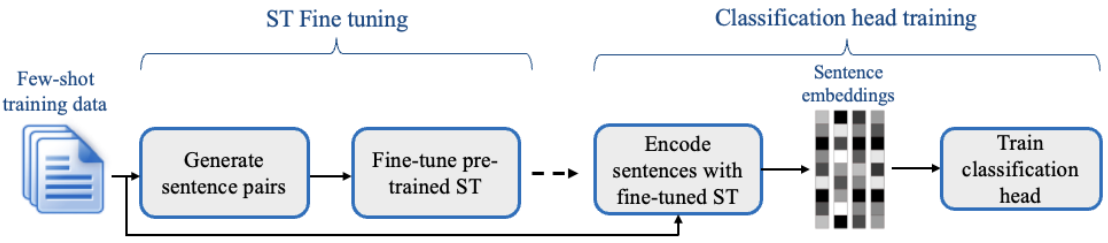


Figure 2.2: SetFit training pipeline for sentence classification. The left side represents the first step and the right side the second step (Tunstall et al., 2022).

2.3.2 Fine-tuning

Fine-tuning involves adapting pre-trained models by training them with a comparatively small amount of data for a specific downstream task. This technique requires less training time and data compared to creating a model from scratch (Devlin et al., 2019). Various techniques exist for fine-tuning pre-trained models.

In-context learning

LLMs have the ability to handle tasks even when they are not specifically trained for them. Tasks for which the model is not explicitly trained are called zero-shot tasks. When given examples, these are called few-shot tasks. In-context learning describes the ability of the LLM to learn from additional information or examples provided in the input, which are not part of the primary task but serve as additional context. The model retrieves information from the input and performs the task without any additional training. Since no additional training is necessary, this is a cost-effective method to improve results (Brown et al., 2020).

Sentence Transformer Fine-tuning

SetFit (Sentence Transformer Fine-tuning) is an efficient fine-tuning method for sentence transformers. SetFit trains sentence transformer models in two steps: firstly, fine-tuning the siamese network on sentence pairs, and then training a classifier head on the newly trained embeddings.

In the first step, the model's embeddings are trained on sentence pairs. For each pair, two sentences from the labeled training data are combined to form positive and negative pairs. Positive pairs are two sentences from the same class, while negative pairs are sentences from separate classes. The model is trained using contrastive learning to ensure that embeddings of positive pairs are brought closer together while being pushed further apart for negative pairs. Combining sentences into pairs expands the available training data, which is beneficial for smaller training data but increases the training time for bigger training datasets drastically. In the second step, the classification head is trained on the fine-tuned embeddings to perform classification. The multiple steps of the training are illustrated in Figure 2.2, showing the first step of training the embeddings on the left side and the second step of training the classification head on the right side (Tunstall et al., 2022).

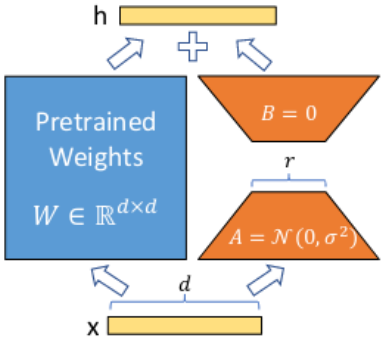


Figure 2.3: LoRA adapter matrices A and B processing the input x simultaneously with the model weights, combined to the output h (Hu et al., 2022).

LoRA adapter

Adapters are a fine-tuning technique that involves adding parameter-efficient components, often in the form of small modules or sublayers, into an existing model to adapt it for specific tasks. Compared to the original model, adapters require only a minimal number of additional parameters, leveraging the pre-existing knowledge in the model but still achieve good results. The parameters of the original model remain unchanged during the fine-tuning process. During training, the additional parameters introduced by the adapter are often trained on a smaller, task-specific dataset compared to the original pre-training dataset. This approach is particularly beneficial when computational resources are limited, when only a small amount of training data is available, or when flexibility and scalability across multiple tasks are desired. A single pre-trained model can be used with different adapters for each downstream task (Houlsby et al., 2019).

LoRA (Low-Rank Adaptation) adapters are a cost-effective fine-tuning technique that extends the adapter method for efficiency and performance. The LoRA adapter is represented by two smaller matrices that reduce the rank of the weight matrices within the model while maintaining the overall computational dimensionality. As illustrated in Figure 2.3, the LoRA adapter matrices (right), each matrices A and B with the dimension r, process the input similarly to the pretrained model (left). The outputs of both the pretrained layer and the adapter are linearly combined and further processed (Hu et al., 2022).

3
Related Work

3.1 Lexicon

Lexical-based methods, among the oldest approaches in sentiment analysis, are still utilized today. A rule-based approach to recommending movies based on sentiment in reviews was proposed by Turney (2002). The pipeline involves the use of a part-of-speech tagger to identify phrases containing adjectives and adverbs. The semantic orientation of phrases was estimated by association. A positive semantic orientation was estimated for phrases with good associations (e.g., "romantic ambience") and a negative orientation for phrases with bad associations (e.g., "terrifying events"). If the average semantic orientation of the review is positive, the movie was classified as recommended. Otherwise, the movie was not recommended. The sole focus on these features resulted in moderate outcomes for movie reviews. Focusing on specific aspects of a review may not necessarily reflect the overall sentiment.

To create or expand a lexicon with technical approaches, there are mainly two methods available: automatic or semi-automatic approaches. In the work of Taboada et al. (2011), , lexicons were automatically collected from various sources. Sentiment-bearing parts like adjectives, nouns, verbs, and adverbs were collected and weighted. Human annotators evaluated the dictionary. Algorithms were developed to take intensifiers into account. Intensifiers can be amplifiers (e.g., very, incredibly) that increase the sentiment of neighboring words and downtoners (e.g., somewhat, a bit) that decrease it. The development of rules for identifying negation requires linguistic expertise. Identifying intensifiers and negation helps to avoid classification errors. This gave strong performance across several domains.

Furthermore, for languages with limited training data, one option is to translate pre-existing sentiment dictionaries. In the work of Ali et al. (2021), this was done for Sindhi, an Indo-Aryan language spoken by more than 75 million people, by using English dictionaries. The English sentiment lexicons EmoLex and Bing Liu's were combined and given a polarity score from the SentiWordNet synset (Mohammad and Turney, 2010; M Hu and B Liu, 2004; Sebastiani and Esuli, 2006). During preprocessing, the

duplicate words were removed and the remaining words were translated into Sindhi using a bilingual dictionary. Modifiers for Sindhi, such as intensifiers and negations, were collected, automatically translated into English, and assigned a polarity score from the SentiWordNet synset with human annotation.

In predicting literary quality, the work by Bizzoni et al. (2023) included sentiment as a measurement alongside stylometric properties. Stylometric measures can include lexical diversity, redundancy, readability, or adverb percentage. The quality of the literature was measured by using the average ratings in an online database with a scale from 1 (worst) to 5 (best). Lexical sentiment analysis gave good performance across various domains and genres. Global parameters such as the average sentiment across the entire document were included as additional features in the predictions. Dynamic features such as variation in sentiment, i.e., how the sentiment changes over time, and sentiment intensity, which measures the intensity of the sentiment, also improved the results. Books with unpredictable arcs received lower scores, while higher average sentiment and more positive endings received higher scores, provided the sentiment was not too flat or repetitive.

Because lexical methods cannot take context like negation into account, researchers developed ways to improve this method with static rules. Jurek et al. (2015) demonstrated a more sophisticated approach, incorporating intensifiers and negation. They designed algorithms and formulas to consider context by effectively handling negations and intensifiers. The formula not only inverts the sentiment of a negated phrase but also calculates a value based on both negation and the lexical value of the word. A similar approach is employed for intensifiers. These approaches performed well on social media posts discussing certain events and on movie reviews.

3.2 Machine Learning

The Naïve Bayes Classifier is a machine learning-based approach that relies on the assumption that every word is independent of each other. They are often employed as baselines for machine learning text classification and sentiment analysis tasks. Various types of Naïve Bayes Classifiers exist, such as the Multinomial Naïve Bayes Classifier (MNBC) and the Binarized Naïve Bayes Classifier (BNBC). Since the probability of an NBC depends on word frequency, it assigns a zero probability to words that are not in the training data. The difficulty of predicting sentiment for a document lies in identifying the sentiment-bearing phrases within the text. BNBC is often the choice because it focuses on presence instead of frequency. Naïve Bayes classifiers learn the sentiment probabilities of words within the data to determine whether a given text is positive or negative. NBC has the advantage of being inexpensive and quickly trained compared to other machine learning techniques, but cannot handle unknown words well.

The work by Jung et al. (2016) utilized BNBC to analyze texts sourced from social media. To effectively handle out-of-vocabulary (OOV) words, the Laplace smoothing technique was employed. SparkR, a framework to execute data science tasks in a data cluster environment, was used for parallelization.

Instead of training a model on text as input, Tan et al. (2022) trained an LSTM model on the embeddings of a pre-trained RoBERTa model as input. This enables the LSTM-based model to extract semantic and syntactic information efficiently, even with

limited training data. Thus, the new LSTM-based model can rely on the learned data from the RoBERTa model without needing to be trained on the entire data set. As LSTM is good at handling long-distance dependencies, it provides a suitable approach for detecting context over a longer span of words. Long-distance dependencies in text can be negations or intensifiers connected to their origin over the sentence, taking previously seen words in the text into account. To augment the data, GloVe model embeddings were utilized, enhancing the new model's performance. This newly trained model outperforms all previous models on the benchmarked data sets.

To consider context from both directions, before and after the word in the document, Tabinda Kokab et al. (2022) employed a bidirectional LSTM (Bi-LSTM) architecture. In addition to LSTM, Bi-LSTM can take context from both forward and backward directions into account. To account for short dependencies without using standard CNN, a different approach was employed. Due to parameter explosion from expanding standard CNNs, Dilated CNNs (DCNNs) are utilized. The trained model produces strong results across diverse data sets, including movie reviews, social media posts, and election reviews.

In the work of Barbieri et al. (2022), a multilingual BERT approach was trained for the domain of microblogging. The model performed better than general-domain multilingual approaches in the domain of microblogging. A multilingual dataset of social media posts in eight languages was released in addition to the pre-trained models. Models trained on multiple languages, compared to monolingual models, showed performance gains for multiple languages. When language-specific data is unavailable, cross-lingual zero-shot outcomes show promise.

3.3 Large Language Models

Lately, Large Language Models (LLMs) with their extensive parameter space have shown impressive performance on various NLP tasks (Radford et al., 2018; OpenAI, 2023). They can generate human-like responses and simulate thought reflection in decision-making processes. Mixed sentiments and linguistic nuances, such as emojis, slang, hashtags, sarcasm, cultural context, or abbreviations, can be handled by LLMs due to their extensive training corpora. The interpretation focuses on these nuances and consistently handles them within the context of the text. LLMs can handle zero-shot tasks, which the model is not specifically trained on, as well as out-of-vocabulary words. LLMs make it possible to infer them through instruct-based tasks. This enables the generation of response texts that reflect decisions in natural language.

The research by Kheiri and Karimi (2023) compared the results of instruct-based approaches with those of fine-tuned models and embedding classification across various GPT-3.5 Turbo models (Brown et al., 2020). Given that GPT embeddings effectively capture language context and nuances, they were also used to train a model based on Random Forest, an ensemble learning method. These approaches were tested on the SemEval2017 Task 4 dataset, which consists of social media posts (Bethard et al., 2017).

Context is often provided to help LLMs understand the downstream task and improve their results. In the work of Min et al. (2022), the crucial parts for the performance gain were evaluated. In classification tasks, such as sentiment analysis, this is achieved by providing the task description and the gold label as additional input. The context with accurate gold labels was compared to experiments done with the identical context but

with incorrect labels and also with out-of-domain examples. The performance generally increased with any given context and performed best with the domain context and correct label. The context had the biggest impact on results. The domain context and the correct label further enhanced the outcomes, but not with the same impact. This indicates that the most influence comes from context, which can be further improved by correctly labeled domain data.

To extend the idea of in-context learning, the context is enriched with domain-specific keywords or examples, as demonstrated in the work by Aycock and Bawden (2024). This has been done in the context of machine translation with LLMs by additionally providing three examples for few-shot learning, as well as keywords and labels about the domain. The few-shot examples helped the model produce the desired output format and improve the results.

Since LLMs are capable of understanding and generating text, Sun et al. (2023) used LLMs to check the work of other LLMs. In this iterative process, the first model works as a generator, predicting the label and giving reasoning for the decision. The second model works as a discriminator, deciding if the prediction and reasoning on the text is correct. If both models agree, the round is over; if they do not agree, the first model makes a new prediction based on the output of the second model. One round is finished if both models finally agree or after a fixed number of turns. The models then switch and predict the labels a second time. If the decisions of both rounds do not align, a third LLM is used to reverse the of the generator and discriminator and decide on the most voted decision. This improves the result in every combination, no matter of whether the generator and discriminator are the same model or different models.

Comparisons of different approaches, namely lexical, machine learning, and LLMs, were conducted by Barnes (2023). The experiments were conducted with multiple languages under different settings. Low-resource training situations were compared to fully supervised situations. Fully supervised models outperformed low-resource solutions on ML and LLM-based models. The out-of-domain loss for fully supervised models was lower compared to few shot models. Lexical approaches performed well compared to domain-shifted and low-resource environments in English. In cross-lingual scenarios, the smaller RoBERTa models achieved better results compared to larger RoBERTa models and dictionary-based approaches. For low-resource languages, the larger models achieved better results, on the other hand.

4 Methodology

This chapter introduces the experiments in detail. The datasets and their domains, the different approaches, and the corresponding models are covered. The metrics used to evaluate the methods are also introduced and covered. The training methods for the research questions RQ2 and RQ3 and their specifics are discussed.

4.1 Datasets

The focus of this thesis is on German sentiment analysis across various domains. Since the way sentiment is encoded into text can vary between domains, multiple datasets from different domains are considered to provide a broader view of the topic. These diverse datasets should provide an overview of how different models perform across various domains, even if they were not specifically trained or tuned for these specific areas. Due to copyright restrictions, caution, or legal difficulties, some datasets based on crawled data from social media or microblogging sites often lack the actual text, including only the ID and the label. As original sources can change over time (e.g., posts can be modified or deleted, users can become invisible, accounts can be suspended or removed), the resulting dataset can undergo significant changes, making it difficult or unfeasible to compare results. Over time, the dataset can become sparse, making it difficult to compare results between publications if a significant amount of data is missing. To address these problems, only datasets with available annotations and texts are considered for benchmarking. Additionally, information about the quality of the annotations, such as annotation guidelines or Cohen's kappa/Fleiss' kappa agreement, is also considered. An overview of the datasets, including their classes and class distributions, is provided in Table 4.1. This comprehensive analysis aims to provide insights into the performance of sentiment analysis models on different datasets and thus promote progress in this field. The section headings of the datasets are used as names in the thesis to reference the datasets.

```
ID: http://twitter.com/tomvomizh/statuses/815662187217829890
Text: @aLienMAstA Nicht ganz. DB Regio und GVH, soviel ich weiß.
Relevant: true
Sentiment: neutral
Aspect:Polarity: Allgemein:neutral
```

Figure 4.1: Example of a document from the GermEval dataset with all metadata. Translation: "@aLienMAstA Not quite. DB Regio and GVH, as far as I know"

4.1.1 GermEval

The GermEval dataset consists of texts in German from social media, microblogs, and news articles about the state-owned railway provider Deutsche Bahn (DB). It was created as a shared task with four subtasks, one of which—the sentiment analysis at the document level—is utilized here. The dataset was collected from various sources, including social media, microblogs, news articles, and question-and-answer forums. The dataset is divided into four subsets: train, dev, test synchronic, and test diachronic. The documents are labeled as 65% to 68% neutral, 25% to 30% negative, and 4% to 6% positive over the subsets. All subsets, except for the diachronic test set, were collected between May 2015 and June 2016. The diachronic test set was collected between November 2016 and January 2017. The synchronic subset captures a snapshot in time without considering historical development, while the diachronic subset tracks the development of phenomena over the collection timespan. To cover all seasonal problems (e.g., air conditioning failure in the summer heat or slippery tracks due to wet leaves in autumn), the collection period was set to one year. Only entries marked as "relevant" in the dataset are evaluated. In total, the dataset consists of 26,198 documents, with lengths ranging from 11 to 32,818 characters. Each document is annotated with one sentiment label (positive, negative, or neutral) at the document level and an aspect polarity at the token or span level. An example document is shown in Listing 4.1. Each sample was annotated by two annotators and checked by a supervisor if there was a disagreement between the annotators. The inter-annotator agreement for polarity started between 0.35 and 0.79 and improved to a range of 0.90 to 1.00 in the last iteration (Wojatzki et al., 2017). The annotation guidelines are available in German[1].

4.1.2 OMP

To cover another part of social media, in this case news comments made by users, the One Million Post (OMP) dataset is utilized. This dataset contains one million user comments posted below news articles on the German-language Austrian news website Der Standard. The comments are linked to the corresponding news articles. Additionally, responses to earlier user comments are traceable. The comments are categorized into labels such as off-topic, feedback, and the three sentiment classes: positive, negative, and neutral. The posts may consist of text only or both a headline and text. For this study, only posts labeled as positive, negative, or neutral are considered. This subset consists of 3,599 posts, each ranging in length from 6 to 998 characters, including both

1. http://ltdata1.informatik.uni-hamburg.de/germeval2017/Guidelines_DB_v4.pdf

Datasets	Domain	Size	Positive	Negative	Neutral
GermEval	SM, Review, Company	2095	105	780	1670
OMP	News commentaries	3,599	43	1,691	1,865
Schmidt	SM, politics	357	97	108	152
Wikipedia	Online Encyclopedia	10,000	0	0	10,000
total		16,051	245	2579	13,687

Table 4.1: An overview of the datasets used in this work, including the subsets of the datasets and the sizes of their classes used for evaluation. The size belongs to the test dataset for evaluation (GermEval 2017 test synchronic, Schmidt test) or the size of the full dataset (OMP, Wikipedia).

headline and text where applicable. The classes are 1% positive, 47% negative and 52% labeled neutral (Schabus et al., 2017). Since the dataset does not come with subsets of dev or train for later experiments involving training, the first half of the dataset is used for training and the second half is used for testing.

4.1.3 Schmidt

Politics on social media has garnered significant interest from the research community due to its influence on public opinion and political discourse (Highfield, 2017; Trottier, Fuchs, et al., 2015; Kruse et al., 2018). The dataset represents the political domain on social media due to its comprehensive coverage of political discourse. The dataset comprises microblogging posts from politicians and political parties who were part of the 19th Bundestag, which existed from 24 October 2017 to 26 October 2021. The data were collected between January 2021 and December 2021, encompassing the period before and after the 20th Bundestag election on September 26, 2021. A total of 1,785 social media posts from 89 accounts were collected. These accounts include the ten largest personal accounts of party members and the three largest main accounts for each party, as measured by follower count. The documents range in length from one to 762 characters. Of all documents, approximately 27% are labeled positive, 30% negative, and 42% neutral. Three annotators labeled each sample, achieving a Fleiss' Kappa agreement score of 0.53. The data collection surrounding the election provides insights into sentiment changes before and after the election. This dataset provides insights into sentiment changes before and after the election, helping to understand the sentimental shifts of political parties and politicians on social media during the election (Schmidt et al., 2022).

4.1.4 Wikipedia

To provide a source of neutral text, Wikipedia, an online encyclopedia, is utilized. Since Wikipedia, as an encyclopedia, is viewed as having a neutral sentiment, every sentence in the dataset is labeled as neutral. The Leipzig Corpora Collection serves as a repository for various monolingual datasets from different sources. For this study, the most recent German Wikipedia dataset from 2021 is utilized. The subset is composed of 10,000 randomly sampled sentences (Goldhahn et al., 2012). The sentences vary between 16 and 255 characters in length. The purpose of using the Wikipedia dataset is to determine whether sentiment classification works not only for domains frequently containing

sentiment, such as social media or politics, but also for neutral texts. The dataset is used solely for evaluation purposes for the RQ1 experiments and not for training.

4.2 Metrics

To evaluate the results, different metrics can be utilized for comparison. The most common metrics in the task of sentiment analysis are accuracy, precision, recall, and F1-score (Schabus et al., 2017; Schmidt et al., 2022; Guhr et al., 2020; Laurer et al., 2024). Every dataset, except Wikipedia, is labeled at the document level into one of the three classes: negative, positive, and neutral. This makes the evaluation a multiclass classification.

Accuracy

The accuracy is a measure of how many labels are predicted correctly, named as True Positive (TP) and True Negative (TN) over all predictions, including False Positive (FP) and False Negative (FN) labels. The calculation is made by dividing all correct predicted labels by the total number of predictions.

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

Measures such as recall, precision, and the F1-score is traditionally designed for binary classification. The measures can be adapted to multiclass problems using various methods. The most common strategies are micro, macro, and weighted averaging. Micro averaging is calculated at the label level, where each label is assigned the same weight. It is calculated by taking True Positives (TP), False Positives (FP), and False Negatives (FN) into account, while excluding True Negatives (TN). This method yields results that are equivalent to accuracy for precision, recall, and F1-score. Macro averaging computes the measures at the class level, without considering class imbalance. This method is suitable for cases where each class is equally important or when dealing with balanced datasets. Weighted averaging also calculates the measures at the class level, as in macro averaging, but assigns weights to each measure based on class size. This adaption is applicable when each label is equally important, but the classes are not equally distributed in the dataset (Pedregosa et al., 2011). The weighted average technique is employed in this thesis to calculate precision, recall, and the F1-score, denoted as $_w$.

Precision

Precision is a measure to capture how many positive predictions made by the model are correct. The measure therefore divides the True Positive labels by all positive predicted labels, including False Positives:

$$Precision = \frac{TP}{TP + FP}$$

To account for multiclass and class imbalance, weighted averaging is applied at the class level, with weights proportional to class size, and the results are summed across all classes.

$$Precision_w = \sum_{c}^{classes} \frac{TP}{TP + FP} * c_{size}$$

Recall

Recall indicates the proportion of how many positive labels the model predicts out of all positive labels. A higher recall value indicates more True Positive labels predicted.

$$Recall = \frac{TP}{TP + FN}$$

When using weighted averaging, recall results in mathematical equivalence to accuracy.

$$Recall_w = \sum_{c}^{classes} \frac{TP}{TP + FN} * c_{size}$$

F1-score

The F1-score is calculated using precision and recall and gives a harmonic mean of both measures.

$$F1 = 2 * \frac{Precision + Recall}{Precision * Recall}$$

While the F1-score in binary classification is between recall and precision, this is no longer the case with the weighting of the classes.

$$F1_w = \sum_{c}^{classes} F1_c * size_c$$

```
<|begin_of_text|>
<|start_header_id|>user<|end_header_id|>
    Classify the sentiment of the text into ONE of the three classes:
    neutral, negative or positive.
    Split the answer in two parts: Label and Reasoning. Text:
    Sollte das WLan bei der Bahn aus Versehen einmal funktionieren,
   wird halt nach 200MB gedrosselt. Lächerlich. https://t.co/pLOuRTgMTs
<|eot_id|>
<|start_header_id|>assistant<|end_header_id|>
```

Figure 4.2: Prompt converted into the chat template for the Llama-based instruct models. The document is taken from the GermEval dataset.

4.3 Chat Template for Instruct-based models

All evaluated LLMs are trained on a chat template. A chat template is a predefined format guiding the interaction and giving the model additional information about the input. This information can be the role or the order. The template differentiates between the user and system role. The model is trained to interpret the user data as input and the system data as previously generated data. Custom context, denoted with the system, can be added to the context to simulate previous interactions, which the model interprets as generated by its actions. This structure guides the conversation flow, ensuring that each input is clearly addressed in a structured manner. As illustrated in Listing 4.2, a task and a document of the GermEval dataset is converted into the chat-template for Llama-based models (Touvron et al., 2023; AI@Meta, 2024).

5

RQ1: Identifying the Best Model for Sentiment Analysis on German Datasets

The first research question demands comparing existing models across various domains using different approaches. This comparison offers an initial understanding of the outcomes of these approaches and their potential. The comparison provides insights into how each approach performs on the investigated datasets and their relative effectiveness.

All methods, including rule/lexicon-based, machine learning, and large language models, are evaluated and compared. Given the differences in training data, parameters, and the expected capabilities of the models, the results are anticipated to provide insights into the possibilities and limitations of these methods. The Wikipedia dataset offers additional insights into model performance, specifically on the neutral class.

5.1 Models

The GerVADER model is selected for the rule- and lexicon-based approach. The documents are converted into a CSV (comma-separated values) file and processed for further analysis. As ML-based models, the models by Lxyuan and Guhr are used (Guhr et al., 2020; Lxyuan, 2023). For the Guhr model, the data is preprocessed to convert digits to text and remove special characters. For the Lxyuan model, no preprocessing steps were stated. For LLMs, foundation models fine-tuned for instruction-based inference were selected. The selected models include Gemma 1.1 7B Instruct (shortened to Gemma), Llama 2 13B Chat (Llama 2), Llama 3 8B Instruct (Llama 3), Mistral 7B Instruct v0.2 (Mistral), and Mistral 8x7B Instruct v0.1 (Mistral 8x). Since the models run on a local server, they need to fit on an A100 GPU with 80 GB of video memory. Although most models generally fit on this hardware, the Mistral 8x and Llama 2 models ran out of memory. The Llama 2 model only succeeded on the OMP, Schmidt and Wikipedia dataset, Mistral 8x only succeeded on the Schmidt dataset. Both models are excluded from further analysis after the first research question.

5.1.1 Prompt

Since the evaluated LLMs are instruction-fine-tuned, the task must be described using a human language prompt. The task is to read the given text and return only the prediction without explanations (Zhang et al., 2023).
The first draft of the prompt was accordingly short and precise:

```
Classify the sentiment of the text into ONE of three classes:
neutral, negative, or positive.
Provide only the correct category without explanation. Text:
```

After some pre-experiments, not all answers generated by the LLMs followed the described task precisely. While some models only provided the answer as requested, others added reasoning, despite it being explicitly forbidden by the prompt. The responses of the models are in human language, unlike lexical or machine learning approaches, making post-processing necessary for evaluation. Simply checking for the class labels "positive', "negative", or "neutral" was not successful, as some models provided extensive, off-topic text. Even if the prediction was clear, the unwanted reasoning could still contain the class labels, making the result unusable for processing. The prompt was extended to address the issue of unsuppressible reasoning. The prompt was modified to include reasoning in the answer, explicitly placed after the prediction. This is meant to handle the output of the model better and facilitate post-processing (Buscemi and Proverbio, 2024). This splits the answer into two parts: the predicted label and the reasoning. After further testing, the final prompt is:

```
Classify the sentiment of the text into ONE of the three classes:
neutral, negative, or positive.
Split the answer into two parts: Label and reasoning. Text:
```

Since the LLMs are autoregressive and create the answer token by token, the reasoning at the end has the least impact on the result. Only the prompt as part of the input affects the results; the order of classification and reasoning does not. This approach resulted in unambiguous answers for 99.7% of all documents (e.g., "This text contains hate speech. I'm unable to categorize it."). The prompt is converted into the model's chat-template format and processed by the model.

5.2 Results

The results of this research question are divided by datasets. For each dataset, the results are analyzed, prepared, and compared. The absolute number of results can vary slightly, as some documents produced no clear answers on certain models.

5.2.1 GermEval

RQ1 GermEval

Model	Architecture	Accuracy Recall_w	Precision_w	F1_w
GerVADER	Rule/Lexicon	0.37	0.62	0.44
Guhr	BERT	<u>0.73</u>	<u>0.72</u>	<u>0.72</u>
Lxyuan	DistilBERT	0.33	0.50	0.24
Gemma	Gemma	0.49	0.67	0.49
Llama 3	Llama	0.56	0.65	0.56
Mistral	Mistral	0.65	0.69	0.66

Table 5.1: Performance metrics of models on the GermEval dataset. Underlined values indicate the best results. Recall_w equals Accuracy and is not elaborated further.

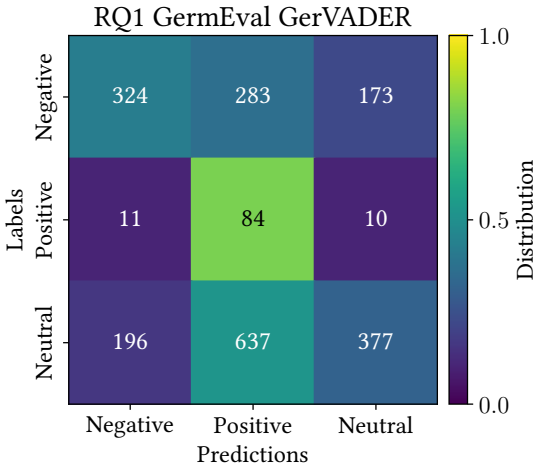


Figure 5.1: Confusion matrix for GerVADER on the GermEval dataset. Rows represent labels, columns represent predictions. Colors indicate prediction distribution in each row, summing to one.

The results of the evaluation for all approaches are listed in Table 5.1.

The accuracy of the GerVADER model on this dataset is slightly better than random guessing, which is 0.33 or 33% with three classes. The confusion matrix in Figure 5.1 presents detailed results for each class. The rows represent the actual labels, whereas the columns represent the predicted labels. The colors indicate the distribution of predictions for each class label. The optimal result would display a diagonal line from the upper-left to the lower-right corner in yellow, indicating high accuracy across all classes.

The GerVADER model predicts negative labels with an accuracy of 0.41, positive labels with 0.80, and neutral labels with 0.31. The green box in the middle indicates high accuracy for positive labels, while the blue color in the lower-right corner indicates low accuracy for neutral labels. Because the neutral class is the largest in the dataset, the overall accuracy is only 0.37. Reports indicate that GerVADER has issues with accurately classifying negative sentiment. The authors of GerVADER state that long sentence structures in German are difficult to predict because negation is sometimes separated from its context (Tymann et al., 2019). This issue may also affect documents from news sources, which typically contain longer sentences compared to social media comments. On the GermEval dataset, the model often incorrectly predicts documents as positive, which supports the statement.

The Guhr model has the best performance on all metrics on the dataset. It achieves the best results, outperforming even the LLM models. The GermEval dataset is part of the training data (Guhr et al., 2020). This may be the primary reason for the high performance of the model, surpassing all other models and approaches, including LLMs. The Lxyuan model is as accurate as random guessing, correctly predicting the label with an accuracy of 0.33. The Lxyuan model is trained on over 3000 German microblogging posts; however, it seems unable to generalize to the broader domain of German social media. News articles, which are part of the GermEval dataset, can make it difficult for the Lxyuan model to perform well, as such content is not included in its training data.

The LLM models perform better than the GerVADER model, but not as well as the Guhr model. The Gemma model has a lower accuracy of 0.49 compared to the Llama 3 model, which has an accuracy of 0.56. The precision of the Gemma model is higher at 0.67 compared to Llama 3 with 0.65. The best-performing LLM-based model is Mistral, with an accuracy of 0.65. The Guhr model has a 12.8% higher accuracy when compared to the Mistral model. The Gemma and Llama 3 models are behind the Mistral model. The Mistral model, as the best-performing LLM, still does not perform as well as the BERT-based Guhr model.

5.2.2 OMP

RQ1 OMP

Model	Architecture	Accuracy	Precision_w	F1_w
GerVADER	Rule/Lexical	0.32	0.56	0.40
Guhr	BERT	0.48	0.50	0.48
Lxyuan	DistilBERT	0.43	0.54	0.32
Gemma	Gemma	0.48	0.62	0.46
Llama 2	Llama	0.46	<u>0.63</u>	0.48
Llama 3	Llama	0.55	<u>0.63</u>	0.53
Mistral	Mistral	<u>0.58</u>	0.58	<u>0.57</u>

Table 5.2: Performance metrics of models on the OMP dataset.

Table 5.2 lists the results on the OMP dataset. The GerVADER model only achieves an accuracy of 0.32, which is not better than random guessing. This could be an out-of-domain problem for the approach, as the documents are discussions referring to other commentators or news articles. An explanation could be the presence of typing errors

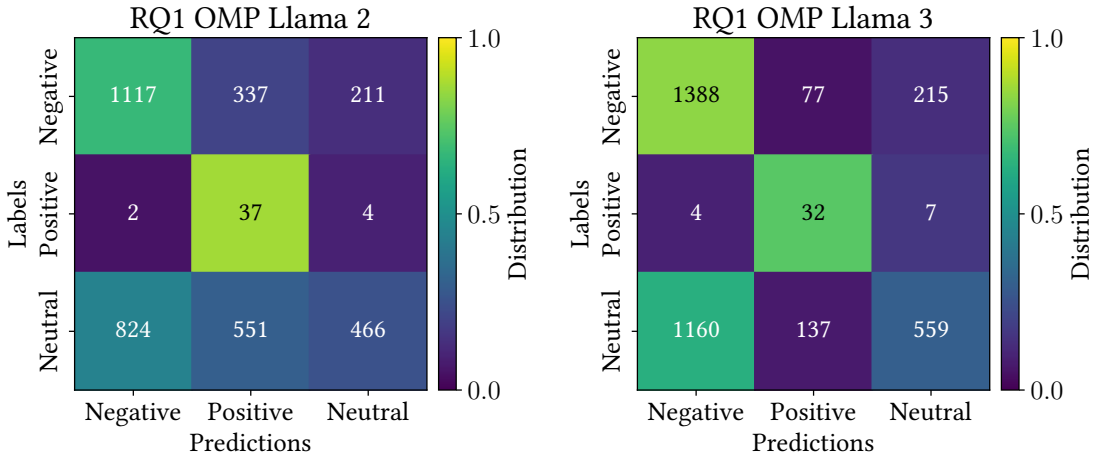


Figure 5.2: Confusion matrix for Llama 2 (left) and Llama 3 (right) on the OMP dataset.

in the comments, which pose a challenge for lexical methods in achieving accurate outcomes. Unlike GermEval, which also includes newspaper articles, the OMP dataset consists solely of comments written by individuals on news articles.

Although the Guhr model has limited performance, it is comparable to the Gemma and Llama 2 models in terms of accuracy. The Guhr model is 4% more accurate than Llama 2, the lowest-performing LLM in measures of accuracy. The similarity between the domain of the training data and the OMP dataset may explain the good performance of the Guhr model. The Lxyuan model, which is also trained on social media posts, is 10% lower in accuracy than the Guhr model but achieves an 8% higher precision.

On this dataset, LLM-based models are closer in performance to each other and to BERT-based models. The Gemma, Llama 2, and Llama 3 models have similar accuracy to the Guhr model. The Llama 3 and Mistral models perform better on the dataset. The Mistral model has 26% higher accuracy compared to the Llama 2 model. The Llama 3 model performs better than the Llama 2 model on all measures, despite having 40% fewer trainable parameters. The detailed results of both Llama-based models, shown in a confusion matrix, are presented in Figure 5.2. The Llama 3 model predicts the negative class with an accuracy of 0.82, which is 23% higher than that of the Llama 2 model. The Llama 3 model also demonstrates better accuracy for neutral documents. Although the Llama 2 model frequently misclassifies neutral documents as either negative or positive, the Llama 3 model almost exclusively classifies them as negative.

5.2.3 Schmidt

The dataset, like the OMP dataset, is not explicitly named as a training dataset for any of the tested approaches, but this cannot be ruled out since it has been publicly available since 2022. The results in Table 5.3 indicate that the GerVADER method achieves its best results so far in absolute terms. The GerVADER lexical method has similar accuracy to the BERT-based models, but outperforms both in the F1-score. In terms of accuracy, it is approximately 16% lower than Llama 2, which has the lowest score of all LLMs tested. This may be attributed to the fact that SentiWS, the sentiment

RQ1 Schmidt

Model	Architecture	Accuracy	Precision_w	F1_w
GerVADER	Rule/Lexical	0.48	0.60	0.45
Guhr	BERT	0.50	0.59	0.44
Lxyuan	DistilBERT	0.52	0.73	0.39
Schmidt	BERT	0.93*		
Gemma	Gemma	0.60	0.72	0.54
Llama 2	Llama	0.57	0.68	0.48
Llama 3	Llama	0.61	0.69	0.61
Mistral	Mistral	<u>0.68</u>	0.70	<u>0.67</u>
Mistral 8x	Mistral	<u>0.68</u>	<u>0.75</u>	0.64

Table 5.3: Performance metrics of models on the Schmidt dataset. Result with an asterisk is from the original paper.

lexicon on which GerVADER is based, is also collected from the financial news domain. Financial regulations are a typical election campaign topic. Since social media posts are generally shorter than other types of documents, such as news articles, the restricted context window of this rule-based method is less limiting for detecting negations and intensifiers. The grammatical quality can be considered higher, since only statements from official politician accounts and political party accounts are included. This dataset is likely to contain documents written by professional adversarial and advertising experts, unlike the OMP dataset. This could result in fewer grammatical errors. Since politicians use social media primarily to set talking points, the documents are more likely to contain sentimental words. However, further investigation is required to confirm this.

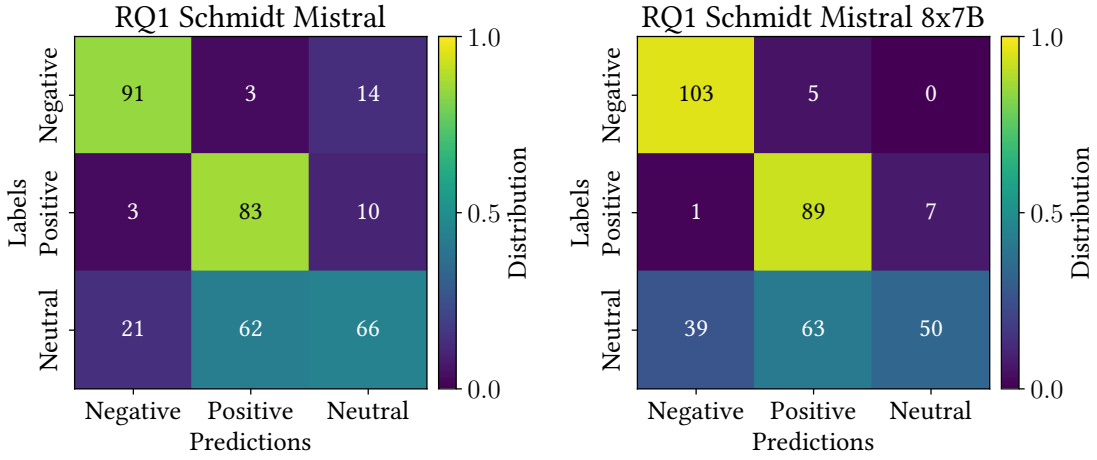


Figure 5.3: Confusion matrices of class-wise results on the Schmidt dataset: Mistral (left) and Mistral 8x (right).

The dataset domain is politics in social media, which is also the training domain for the Guhr model. Both BERT-based models are also trained on the SB10k dataset, which consists of social media postings (Cieliebak et al., 2017). Since both models were trained on domains identical to the dataset, it appears they do not significantly benefit from it. Official political accounts may have a different writing style compared to

private social media comments. GerVADER demonstrates the same level of performance as both BERT-based models. The results stated by Schmidt et al. (2022) outperform all other approaches. It is BERT-based and has been fine-tuned on GermEval and Schmidt to achieve an accuracy of 0.93. Since the model is not publicly accessible, the outcomes are not further considered.

All LLMs, including Llama 2 and Mistral 8x, performed significantly better than the other tested methods. Both Mistral-based models performed best on this dataset. In precision metrics, the Mistral 8x model outperforms the Mistral model; however, the weighted F1-score is lower. Since the F1-score is dependent on both recall and precision due to the weighting of class results, it can vary. The detailed results from both Mistral-based models in 5.3 show the distribution of predictions side-by-side. While the Mistral model (left) has higher accuracy in the neutral class, the Mistral 8x model (right) does better in the negative and positive classes. Although the Mistral model often categorizes neutral documents as positive, the Mistral 8x model predicts them almost twice as frequently as negative. The GerVADER model is surprisingly strong on this dataset, which could be due to the estimated quality of the data. This appears to be beneficial for the LLMs as well. One reason could be the better cross-lingual performance on well-written and structured texts.

5.2.4 Wikipedia

RQ1 Wikipedia

Model	Architecture	Accuracy	Precision_w	F1_w
GerVADER	Rule/Lexical	0.44	1.0	0.61
Guhr	BERT	0.99	1.0	0.99
Lxyuan	DistilBERT	0.03	1.0	0.05
Gemma	Gemma	0.57	1.0	0.73
Llama 2	Llama	0.45	1.0	0.62
Llama 3	Llama	0.76	1.0	0.86
Mistral	Mistral	0.87	1.0	0.93

Table 5.4: Performance metrics of models on the Wikipedia dataset.

The Wikipedia dataset can serve as a valuable indicator of how the approaches handle the neutral class. Since the dataset has only one class, the precision stated in Table 5.4 is always 1. The GerVADER approach is performing very well on this dataset, almost as well as the Llama 2 model in terms of accuracy. This could be due to the lexical and rule-based approach, predominantly working on a bag-of-words model that lacks context. Since sentiment-bearing words can be part of encyclopedic texts, this does not necessarily make them sentiment texts.

The Guhr model was trained using the Wikipedia 2016 dataset, which contains 1 million sentences. Wikipedia data accounts for 91.71% of all training examples for the neutral class. Therefore, Wikipedia data constitutes the main part of the neutral training data. The 2016 dataset does not differ significantly from the 2021 dataset, as length and structure are likely similar. Online encyclopedias are likely to have different sentence structures compared to news articles or social media posts. This could explain

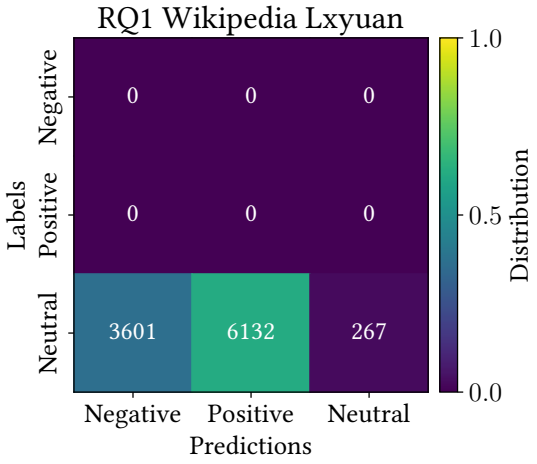


Figure 5.4: Confusion matrix of Lxyuan model results on the Wikipedia dataset. Only the neutral label is present; all other rows are zero.

the outstanding results of the Guhr model, as the length and structure of the input data make it easier for the model to predict the dataset correctly. The neutral class, which accounts for 18.7% of the model's training data, is still underrepresented. The model seems to overfit on Wikipedia documents. Even though the Lxyuan model is trained on all three classes, only 2.7% of all predictions are correct. As stated in the confusion matrix in Figure 5.4, the model predicts most of the data as positive, with more than 61% positive predictions and 36% negative predictions. The model cannot discriminate well between positive and neutral classes on this dataset.

The Llama 2 model's results are very similar to those of the rule-based GerVADER model. When compared to the Llama 2 model, the Mistral model has predicted the documents correctly almost twice as often. One reason could be that the Llama 2 model usually predicts either positive or negative outcomes more frequently. Compared to the detailed results for the OMP dataset shown in Figure 5.2 (left), the accuracy for all classes is the lowest for neutral. The low accuracy for the neutral class is indicated by the blue color, in contrast to the high accuracy for the negative and positive classes. The Llama 3 model is performing very well and is second best to the Mistral model. The Gemma model performance in terms of accuracy is higher compared to the Llama 2 model, but is 33% below the Llama 3 accuracy. The accuracy of the Llama 3 model is 69% higher compared to the Llama 2 model. The Mistral model clearly outperforms all other models, especially considering that the Guhr model may be overfitting.

5.3 Conclusion

The objective of this research was to evaluate various models for sentiment analysis on German datasets, comparing both monolingual and multilingual approaches across different domains. The Guhr model achieved the best performance on the GermEval dataset, likely due to being specifically trained on this data. At the same time, the Lxyuan model underperformed, even when compared to the rule-based GerVADER approach. Detailed analysis indicates that GerVADER performed well in classifying positive sentiments but struggled with neutral ones. The performance of the large language models varied in terms of accuracy, with the lowest accuracy at 0.49 for the Gemma model and the highest at 0.65 for the Mistral model. On the OMP dataset, the results were overall lower compared to GermEval. The performance of the Guhr model was moderate, notably surpassed by the Llama 3 and Mistral model. Llama 3 outperformed others in classifying positive and neutral sentiments, but often misclassified neutral documents as negative. All models performed well on the Schmidt dataset. The Lxyuan model slightly outperformed the Guhr model, while GerVADER matched the performance of BERT-based models. Interestingly, the Mistral 8x model showed better accuracy in classifying negative and positive sentiments compared to the standard Mistral model. The Wikipedia dataset highlighted overfitting of the Guhr model on Wikipedia data. The Lxyuan model performed not well, correctly predicting only 3% of data points, with a strong bias towards positive classifications. A significant performance improvement was evident between model generations for the Llama-based models.

The performance of LLMs varied widely, indicating potential issues with the zero-shot environment or domain applicability. Notably, successive generations of Llama-based models showed increasing accuracy. This variability in performance suggests a need for further investigation into domain-specific improvements.

6

RQ2: Effective Strategies for Domain Adaptation in Sentiment Analysis with Limited Labeled Data

The objective of RQ1 was to provide an overview of various approaches and their performance across different domains in comparison to one another. The hypothesis postulated that LLMs would outperform both lexical/rule-based and machine learning-based methods. While this was not true for every combination of domain and model, the overall performance of LLMs was generally superior. It is hypothesized that even a small amount of data can significantly enhance model adaptation. Since LLMs demonstrated superior overall performance and offer additional possibilities for enhancing their capabilities, they were subjected to further testing. Given that LLMs are trained on billions of tokens, it is reasonable to assume that their zero-shot performance can be improved. Additional methods like few-shot learning and adapter training could potentially yield further performance improvements.

For this research question, various models were further trained and evaluated. Three distinct training approaches were used to address the research questions: in-context learning (RQ2.1), LoRA adapters (RQ2.2), and SetFit (RQ2.3) All methods are briefly described in Chapter 2. The experimental details are provided in each subsection pertaining to the method.

6.1 RQ2.1: In-context learning

The first training method tested is in-context learning. As introduced in Section 2.3.2, LLMs can acquire additional information through their input without requiring further training. Hypothesis: Does the source of the examples affect the model's performance? To test this hypothesis, the prompt is extended with documents from the dataset being tested, a different dataset, or multiple datasets. The context varies among the Schmidt, GermEval, and OMP datasets, or a combination of all previously mentioned datasets.

6.1.1 Models

The models evaluated in the research questions are the instruct-fine-tuned models Gemma 1.1 7B Instruct (shortened to Gemma), Llama 3 8B Instruct (Llama 3), and Mistral 7B Instruct v0.2 (Mistral).

6.1.2 Prompt

For this experiment, the input is enhanced to provide context, guiding the model on the domain and the desired answer format. This also involves adapting the prompt. Since the model now receives guidance on the desired output format, the prompt is shortened to:

```
Classify the sentiment of the text into ONE of the three classes:
neutral, negative or positive. Text:
```

The examples are included as part of the input history, making it appear as if the model generated them previously. In this template, the answer is limited to the class name. Given that the datasets in these experiments contain three classes, the context is augmented with three examples, one for each class. The input is converted into the chat template for the model as described in Section 4.3, integrating three previously given examples and their corresponding answers. This extended input, incorporating documents from the GermEval dataset in the chat template for Llama-based models, can be seen in Appendix A.1. The examples with the correct labels are incorporated as history for the model. For this experiment, only three examples are necessary to provide the model with domain context and reference for output formatting. The document in the context was taken from the dataset being tested and was not removed from the test dataset. This should not significantly affect the results, as it accounts for only a small fraction of the entire dataset. For each dataset, three random examples were selected. Additionally, context from all datasets was combined and tested on the Schmidt dataset.

6.1.3 Results

The results of this research question are segmented by dataset. For each dataset, the results are analyzed, prepared, and compared. The absolute number of results may vary slightly, as some documents did not produce clear answers on certain models.

RQ2.1 GermEval			
Model	Accuracy	Precision_w	F1_w
Gemma	0.54↑	0.64↓	0.52↑
Llama 3	0.56•	0.62↓	0.58↑
Mistral	0.67↑	0.68↓	0.67↑

Table 6.1: Performance evaluation on the GermEval dataset. Compared to RQ1, arrows indicate improved ↑ or declined ↓ results, while • represents no change. The values are compared to Table 5.1.

GermEval

The GermEval dataset posed significant challenges for LLMs and other approaches, with the notable exception of Guhr's model, as evidenced in RQ1. Table 6.1 presents the evaluation results, with arrows indicating whether the outcomes, compared to RQ1, improved, remained stable, or declined.

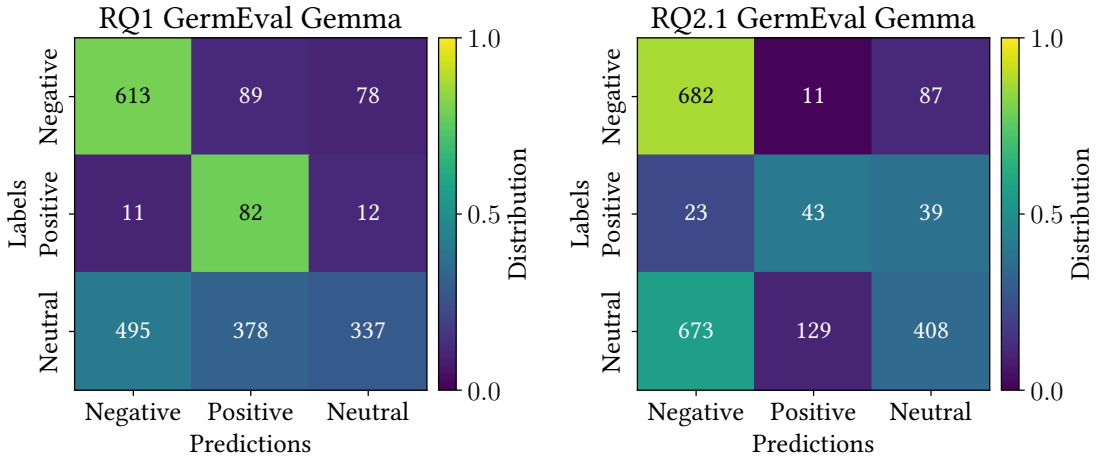


Figure 6.1: Performance comparison of Gemma on the GermEval dataset between RQ1 (left) and RQ2.1 (right).

Compared to the results from RQ1, the accuracy of the Gemma model increased by 10%, from 0.49 to 0.54, thereby narrowing the gap with other models. The model appeared to struggle with executing tasks in a zero-shot environment. The Gemma model achieved an accuracy of 0.49, outperforming the Llama 3 model but not the Mistral model. Figure 6.1, depicts the detailed results, including the confusion matrix for RQ1 (left) and RQ2.1 (right). The accuracy for the negative class increased from 0.79 to 0.87, which represents a 10% improvement. For the positive class, accuracy declined from 0.78 to 0.41, representing a 47% decrease. The model misclassified neutral documents as positive less frequently in the context-based results, improving the performance for the neutral class overall. Utilizing examples and context, the model enhanced performance for the neutral class. Although the positive class is the smallest in the dataset, the overall accuracy improved despite the decline in positive class performance.

The accuracy of the Llama 3 model did not improve, remaining below the Mistral model but slightly above the Gemma model. Performance in terms of precision slightly decreased, from 0.65 to 0.62.

The Mistral model excelled in in-context learning, improving its accuracy slightly from 0.65 to 0.67, a 3% increase. The model appeared to have a good understanding of the task in a zero-shot environment, therefore the slight improvement. At the same time, the Llama 3 model showed no overall improvement in accuracy. The dataset is diverse, comprising various authors and sources, including news articles and social media posts within the domain of Deutsche Bahn. This context may be appropriate for one source while being unhelpful for another. With only three examples in this setting, this mismatch could pose a problem. Similarly, when the context sources and the text to be predicted do not align, it becomes more challenging for the model to accurately predict the correct class.

OMP

RQ2.1 OMP

Model	Accuracy	Precision_w	F1_w
Gemma	0.61↑	0.64↑	0.59↑
Llama 3	0.60↑	0.65↑	0.59↑
Mistral	0.59↑	0.60↑	0.56↓

Table 6.2: Performance metrics of models on the OMP dataset, compared to Table 5.2.

As indicated in Table 6.2, all models showed improvements in accuracy and precision compared to the zero-shot experiment described in RQ1.

The performance by the Gemma model significantly improved for the dataset, achieving the highest scores in both accuracy and F1-score. The accuracy increased from 0.48 to 0.61, representing an improvement of over 27%.

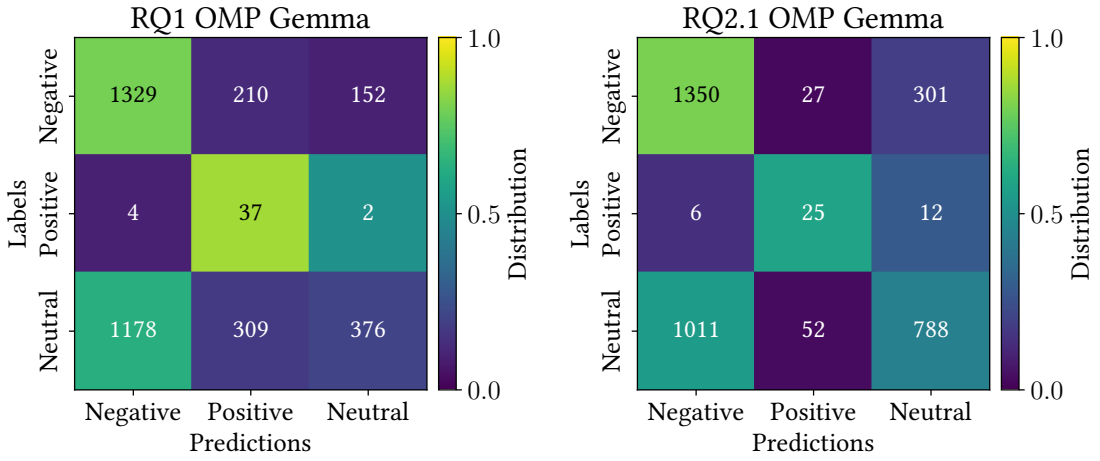


Figure 6.2: Detailed results comparison on the OMP dataset between RQ1 (left) and RQ2.1 (right) by the Gemma model.

The context helped the model to better predict the neutral class, doubling the number of correct predictions, as illustrated in Figure 6.2. It appears that the model was trained exclusively on binary sentiment classification for positive and negative sentiments. The performance for the negative class slightly improved compared to RQ1, but the accuracy for the positive class declined from 0.86 to 0.58, a decrease of 33%. A similar decline in accuracy for the positive class is observed in the GermEval dataset.

The Llama 3 model increased its accuracy from 0.55 to 0.60, a gain of 9%, and demonstrated the best precision among all models.

The accuracy of the Mistral model improved slightly from 0.58 to 0.59 and precision from 0.58 to 0.60 while the F1-score declined slightly from 0.57 to 0.56.

The accuracy of the Mistral model slightly increased from 0.58 to 0.59, and its precision improved from 0.58 to 0.60. The F1-score slightly declined from 0.57 to 0.56. This behavior is consistent with the trends observed in the GermEval dataset. The model performed best in RQ1 using the zero-shot environment but did not show improvement with in-context learning.

Schmidt

RQ2.1 Schmidt

Model	Accuracy	Precision_w	F1_w
Gemma	0.64↑	0.64↓	0.62↑
Llama 3	0.70↑	0.73↑	0.68↑
Mistral	0.61↓	0.68↓	0.60↓

Table 6.3: Performance metrics of models on the Schmidt dataset compared to Table 5.3.

The Gemma model shows improvements in few-shot experiments regarding accuracy and F1-score compared to RQ1, as presented in Table 6.3. The accuracy increased from 0.60 to 0.64, showing an approximate 7% improvement. Although the Gemma model showed an improvement compared to RQ1, other models demonstrated even greater enhancements.

The Llama 3 model achieved increases in accuracy, precision, and F1-score by 14.8%, 5.8%, and 11.5%, respectively, compared to the RQ1 results, representing the highest improvement among all tested models. As illustrated in Figure 6.3, the accuracy for the negative class decreased, while the accuracy for the positive class remained nearly unchanged compared to RQ1. The accuracy for the neutral class increased from 0.20 to 0.45, representing a 125% improvement. The detailed results of the Gemma model on the GermEval and OMP datasets showed that its improvement primarily results from better predictions for the neutral class. Similarly, the Llama 3 model appears to be primarily trained for binary sentiment classification.

The Mistral model achieved the best results on this dataset, with gains of 5.9% in accuracy, 7% in precision, and 7.5% in F1-score.

To determine whether the context itself or domain-specific guidance contributes more significantly to performance improvement, the models were tested on the Schmidt dataset with various contexts and also all contexts combined. Since all previous experiments related to this research question were conducted using one example per

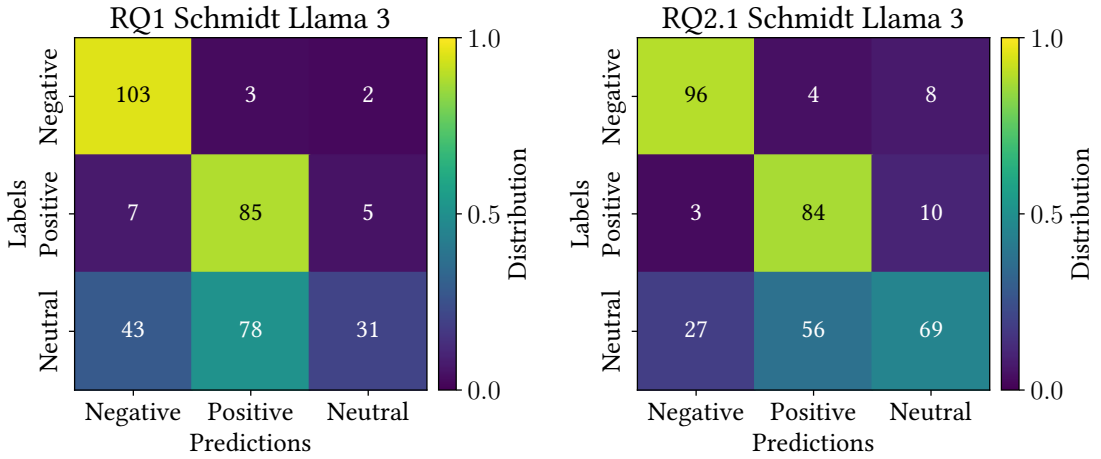


Figure 6.3: Comparison of Llama 3 results on the Schmidt dataset between RQ1 (left) and RQ2.1 (right).

RQ2.1 Schmidt

Model	Context	Accuracy	Precision_w	F1_w
Gemma	Schmidt	0.64↑	0.64↓	0.62↑
Gemma	GermEeval	0.63↑	0.64↓	0.61↑
Gemma	OMP	0.62↑	0.62↓	0.60↑
Gemma	All	0.64↑	0.66↓	0.62↑
Llama 3	Schmidt	0.70↑	0.73↑	0.68↑
Llama 3	GermEval	0.71↑	0.73↑	0.71↑
Llama 3	OMP	0.68↑	0.70↑	0.67↑
Llama 3	All	0.66↑	0.71↑	0.65↑
Mistral	Schmidt	0.61↓	0.68↓	0.60↓
Mistral	GermEval	0.67↓	0.70•	0.67•
Mistral	OMP	0.64↓	0.69↓	0.64↓
Mistral	All	0.66↓	0.69↓	0.65↓

Table 6.4: Performance of various contexts on the Schmidt dataset. Underlined values denote the best performance achieved by each model across all contexts.

class and the context provided by the dataset, the dataset was tested across all three contexts, including the combination of all contexts. Table 6.4 provides an overview of the results obtained from the Schmidt dataset under the different tested configurations.

The Gemma model achieved an improvement in accuracy and F1-score compared to the zero-shot experiments. The model could not achieve further improvement by the specific domain-related context. It appears that the model did not interpret the task sufficiently during the zero-shot experiments, and the observed improvement was primarily due to the explanation of the task rather than the addition of domain-specific knowledge. As illustrated in Figure 6.4, the accuracy for negative labels remained nearly identical, while the accuracy for positive labels declined, and the accuracy for neutral labels improved significantly. It appears the model has difficulties predicting the neutral

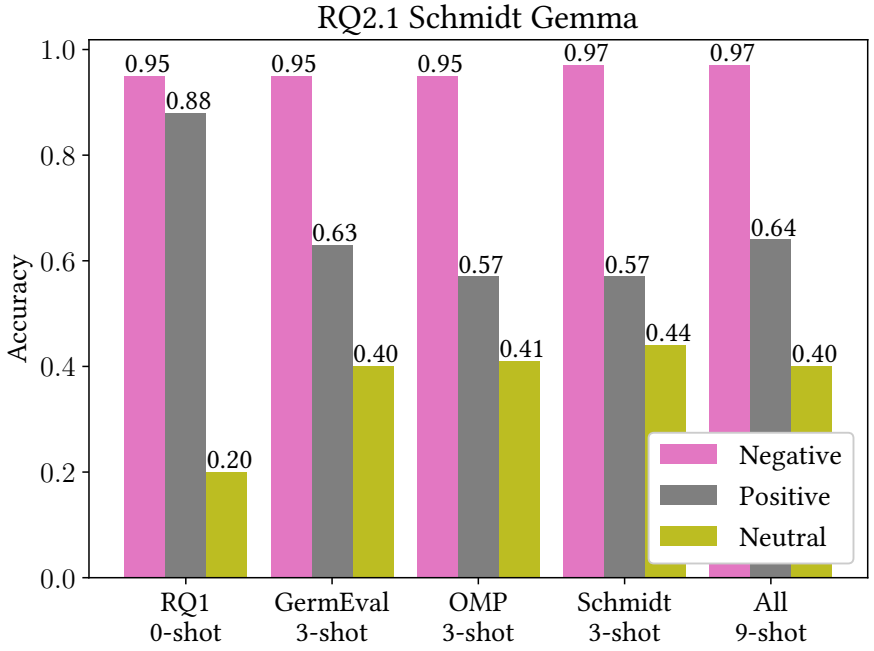


Figure 6.4: Side-by-side comparison of class accuracy by the Gemma model across different contexts and RQ1 on the Schmidt dataset.

class. With context, the accuracy for the neutral class increases, but it leads to more false predictions for the positive class. One reason could be, that if the sentiment is only slightly positive, the model, without in-context learning, tends to favor the positive class over the neutral class. When provided with context, the model is more likely to predict these documents as positive. The Llama 3 model demonstrated improvement across all measures compared to the zero-shot classification. This improvement appears to be due to the guidance of the task primarily, rather than the domain-specific context, similar to the Gemma model. The Mistral model performed best in the zero-shot classification task but declined across all contexts on the dataset. The model did not benefit from the examples and was misled by them. Some random samples generated by the model indicate that the model increasingly predicted the neutral class and exhibited greater uncertainty compared to other datasets and experiments.

6.1.4 Conclusion

The results of in-context learning varied across all models and datasets. Although Gemma initially did not perform well in the setting of the previous research question, it performed significantly better with additional information. The results indicate an increase in the neutral class across all datasets, including all contexts in the Schmidt dataset.

The Llama 3 model showed some improvement on certain datasets but did not match the progress of the Gemma model. The identical accuracy on the GermEval dataset compared for zero-shot might be due to the diverse structure of the dataset and the few examples provided, which can make it difficult to find guidance. Since the results for all contexts on the Schmidt dataset show the least favorable outcomes for Llama 3, this statement must be verified with further experiments.

The Mistral model had mixed results with the provided context on the dataset. It improved on the GermEval and OMP datasets but declined in all contexts on the Schmidt dataset. It seems the model understood the task well in the zero-shot setting and was misguided by the given context. The random selection of examples might be the cause. Overall, the results of the Mistral model are still good in the few-shot environment, but they are not as outstanding compared to the zero-shot results.

In conclusion, in-context learning can help guide LLMs to understand the task better, but this process needs adaptation to the model and the domain. The results by Min et al. (2022) have been confirmed by experiments with different contexts for Gemma and Llama 3, while the performance of the Mistral model declined. The right context can help, as Aycock and Bawden (2024) stated, but the results are not so clearly comprehensible here.

6.2 RQ2.2: Fine-tuning with LoRA adapters and classification head

Fine-tuning large language models by adjusting the model weights is a computationally expensive task. Sometimes, labeled training data is sparse and expensive to obtain, as labeling can be a tedious task that requires human annotators, significant time, and domain-specific knowledge. When the availability of training data or computational resources is limited, adapters can be a suitable approach. This research question aims to determine the necessary size of training data required to enhance the results. This should provide insight into the optimal size of training data for adapters when comparing different models.

6.2.1 Training details

Unlike the models in previous research questions, the models evaluated in this experiment are not instruction fine-tuned. The evaluated models include Gemma 7B (shortened to Gemma), Llama 3 8B (Llama 3) and Mistral 7B v0.1 (Mistral). These models do not require a prompt as part of their input. Additionally, an untrained classification head is initialized and set atop the model architecture. The model weights are frozen, indicating they remain untrained. Only the additional LoRA adapter layers undergo training. The adapter layers are trained using the following parameters, adapted from the approach proposed by Hu et al. (2022):

```
Alpha 128
Batch Size 8
Dropout 0.05
Epochs 2
Learning Rate -2e5
max grad norm 0.3
Rank 128
Target modules Q_proj, V_proj, all-linear
Warmup ratio 0.1
Weight decay 0.01
```

To compare various sizes of available training data, the datasets are divided into subsets of different sizes. Since the OMP dataset does not come with a predefined training set, it is divided into two parts. The first half of the samples is designated for training, while the second half is reserved for evaluation. The training subsets for OMP and Schmidt consist of 32, 64, 128, 256, 512, and 1,024 samples. Additional subsets of 2,048, 4,096, and 8,192 samples are created for the GermEval dataset due to its larger training data size. The class distribution remains unchanged across all subsets. Additionally, all models are trained on the complete training datasets: 16,201 samples for GermEval, 1,799 for OMP, and 1,428 for Schmidt. Each combination of model and subset is trained three times. For evaluation, the mean accuracy of all runs is used. The results of all experiments are provided in the Appendix in Section A.2.

6.2.2 Results

GermEval

RQ2.2 GermEval

Model	32	64	128	256	512	1024	2048	4096	8192	16201	OOTB
Gemma	0.44	0.40	0.43	0.43	0.51	0.57	0.62	0.68	0.78	0.80	0.49
Llama 3	0.43	0.40	0.41	0.43	0.44	0.56	0.61	0.74	0.78	0.81	0.56
Mistral	0.44	0.43	0.47	0.47	0.48	0.53	0.60	0.69	0.76	0.80	0.65

Table 6.5: Performance metrics of LoRA adapters with classification head on GermEval, compared to Table 5.1. Accuracy is the average of three passes. The training size at which the accuracy exceeds the OOTB model for the first time is underscored. The OOTB values are taken from Table 5.1.

The results are presented in Table 6.5, including findings from RQ1, marked as OOTB. The values represent the mean accuracy after three training runs. Detailed accuracy for each training run is provided in the Appendix in Table A.1. For training sizes up to and including 256 samples, none of the models show an improvement in their results. This may be due to randomness and the quality of the training data for the model. With smaller training sizes, individual documents can have a more significant impact compared to when larger training sets are used.

The Gemma model requires the least amount of training data to enhance its performance compared to the zero-shot results. Starting with 512 training samples, the accuracy improves incrementally. The highest accuracy of 0.80 is achieved by training the adapters with the full dataset of 16,201 samples. This represents a 60% increase in terms of accuracy over the OOTB results and is comparable to the Llama 3 and Mistral model. As observed in the previous evaluation, the model can improve its performance with limited data. Although the model is performing weakly in zero-shot evaluation, it can adapt quickly.

The Llama 3 model is unable to improve with smaller training sizes. With a training size of 2,048 samples, the model surpasses the OOTB results. The model is performing similarly to the Gemma and Mistral model starting at 1,024 samples, with some deviations. As the training data is selected randomly, this variability could be a relevant factor. With the full dataset, the Llama 3 model achieves an accuracy of 0.81, the highest among all

models on this dataset and 44% higher than the Llama 3 OOTB model. The model adapts to the dataset, but requires more training data to achieve optimal results. This may be due to its better initial performance in the OOTB setting compared to the Gemma model.

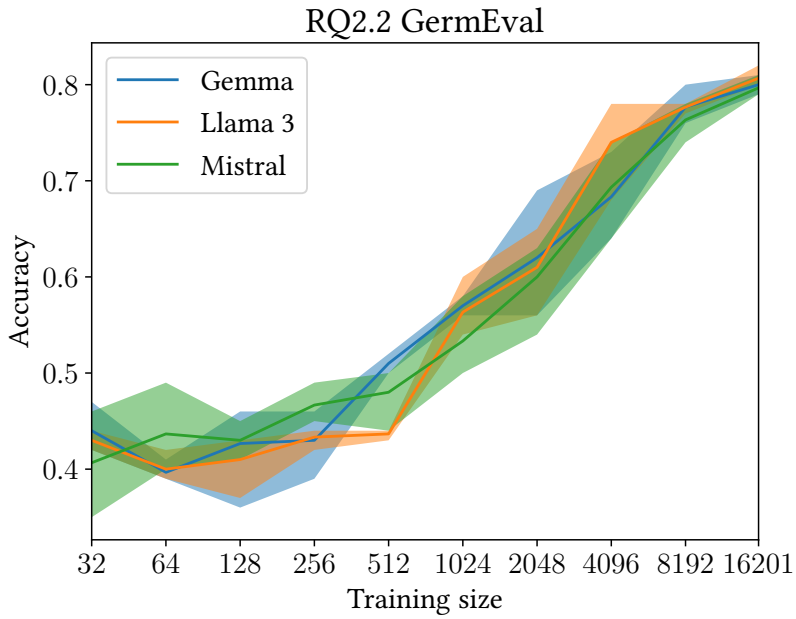


Figure 6.5: Performance metrics of models on the GermEval dataset. The y-axis represents the accuracy, while the x-axis is logarithmically scaled for training size. The lines indicate mean values, and the shaded areas represent the deviation across three runs.

As the best-performing model in the zero-shot setting in terms of accuracy, the Mistral model shows improvement during training. Starting with 4,096 samples, the model surpasses the OOTB model, reaching an accuracy of 0.80, which is 16% higher. The performance of the model improves with additional training data, matching the full-training-size performance of the other models.

The graph in Figure 6.5 illustrates the mean results for the models as lines and the deviations as shaded areas. All models show similar accuracy across different training sizes but struggle with sparse data below 512 samples. Due to random sampling, results for training sizes between 32 and 256 vary widely. The result scatter, shown as colored areas, decreases with increasing training size. The accuracy of all models showed minimal improvement when the training data nearly doubled from 8,192 to 16,201 samples. With training data of 8,192 samples or more, no model stands out.

OMP

The results presented in Table 6.6 indicate a more varied outcome compared to the evaluation on the GermEval dataset. While the GermEval dataset is the largest among the three evaluated, the OMP training dataset contains only 1,799 samples, which is just one-ninth of the size. When comparing the results between OOTB and this evaluation, it is important to note that RQ1 was assessed using all samples, whereas the dataset was split equally into training and test sets for this analysis. As shown in Figure 6.6, the model adaptation differs.

The Gemma model surpasses the OOTB performance, starting at a sample size of 64. For smaller training sizes, the model runs better than the other models, but still

RQ2.2 OMP

Model	32	64	128	256	512	1024	1799	OOTB
Gemma	0.45	0.50	0.50	0.49	0.56	0.63	0.67	0.48
Llama 3	0.31	0.46	0.51	0.54	0.55	0.63	0.66	0.55
Mistral	0.42	0.49	0.51	0.53	0.54	0.59	0.57	0.57

Table 6.6: Accuracy of LoRA adapters with classification head evaluated on the OMP dataset, compared to Table 5.2.

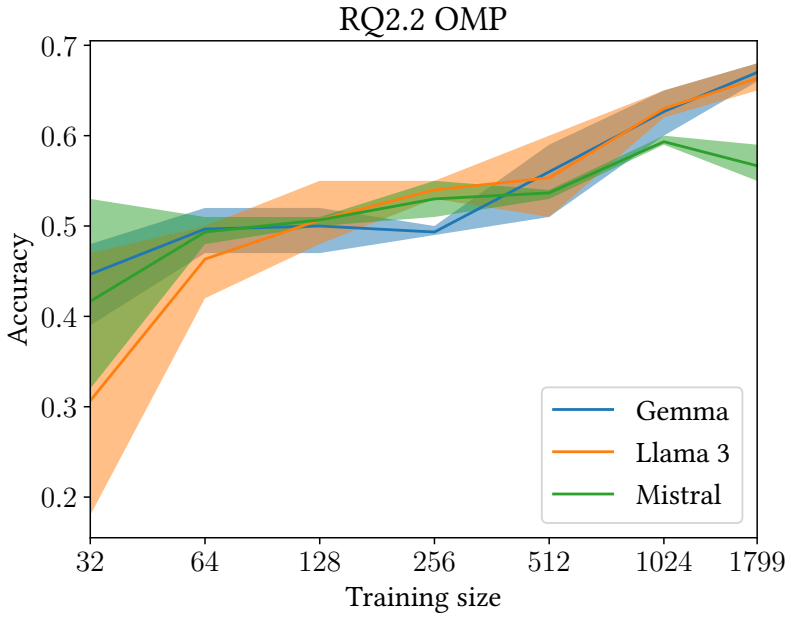


Figure 6.6: Accuracy of all training sizes on the OMP dataset.

falls short as compared to the OOTB model in terms of accuracy. At 512 samples, the Gemma model is superior to both the Llama 3 and Mistral model with the same training size. The Gemma model begins to surpass the Mistral OOTB model, which performed the best in RQ1. With larger training sizes, the Gemma model achieves an accuracy of 0.67, which is 39% higher as compared to the OOTB results. The additional features that have been learned help the model to achieve the best performance. This was also evident when evaluating the Gemma model with in-context learning on the OMP dataset. This suggests, as previously observed with on the GermEval dataset, the potential for adapting the Gemma model even with sparse data. Although the performance of the model in a zero-shot setting was lower compared to other models, it can be efficiently adapted to the domain using adapters.

The Llama 3 model also begins to improve its accuracy, starting at 1,024 training samples. The When trained on all available data, the model reaches an accuracy of 0.66, which is 20% higher compared to the OOTB performance. The model has the same accuracy as the Gemma model, both being well-adapted to the dataset and capable of improvement with trained adapters.

In contrast to the Gemma and Llama 3 models, the Mistral model cannot provide significant performance improvements. Although it reaches a higher accuracy at 1,024 training samples, the performance of the model decreases at 1,799 samples, whereas

the other models continue to improve with additional data. As shown in Figure 6.6, the accuracy of the Mistral model fluctuates a lot at 32 samples. For 1,024 samples, all training runs achieved an accuracy ranging from 0.59 to 0.60, as detailed in the Appendix, Table A.2. The model appears to require more training samples to surpass its OOTB performance and fully adapt to this domain in terms of accuracy.

Schmidt

RQ2.2 Schmidt

Model	32	64	128	256	512	1024	1428	OOTB
Gemma	0.38	0.33	0.41	0.41	0.55	0.64	0.75	0.60
Llama 3	0.31	0.32	0.41	0.39	0.52	0.66	0.75	0.61
Mistral	0.34	0.36	0.40	0.44	0.54	0.62	0.65	0.68

Table 6.7: Accuracy of models, trained on the Schmidt dataset, compared to Table 5.3.

The results presented in Table 6.7 indicate that only the Gemma and Llama 3 models are showing improvements in performance in terms of accuracy.

The performance of the Gemma model surpasses the OOTB results in terms of accuracy, starting at 1024 training samples, achieving an accuracy of 0.75, which is 25% higher as compared to the OOTB model. For training sizes below 512 samples, the Gemma model shows no substantial improvement. As previously discussed for other datasets, the Gemma model can improve even with sparse data, surpassing models like Mistral.

With 1,024 training samples, the Llama 3 model achieves an accuracy of 0.75, which is 23% higher as compared to the OOTB model. The Llama 3 model adapts well to sparse datasets, achieving great results when compared to the Gemma model. The performance of the model in terms of accuracy improves with in-context learning, and further enhances when using adapters and low-data fine-tuning.

The Mistral model is the only one that did not improve as compared to the OOTB in terms of accuracy. Even with all available training data, the Mistral model remains slightly behind the OOTB model. As shown in Figure 6.7, the resulting distribution is relatively high for 1,428 training examples compared to other models. The results, as visualized, improve with more training data, indicating the models learn effectively from given data. However, the Mistral model appears to need more data to adapt to specific domains. The accuracy of each training run is listed in the Appendix in Table A.3.

6.2.3 Conclusion

This research investigated how different models perform with varying training sizes, using LoRA adapters for efficient fine-tuning. The effectiveness of LoRA adapters for sparse data scenarios was evaluated by comparing models trained on different sample sizes. The hypothesis that LoRA adapters could be effective in such environments was partially confirmed.

On the GermEval dataset, which had the largest training set, all models showed improved performance. They reached similar accuracies with full training data, indicating the advantages of using adapters. Models with initially lower accuracy in the

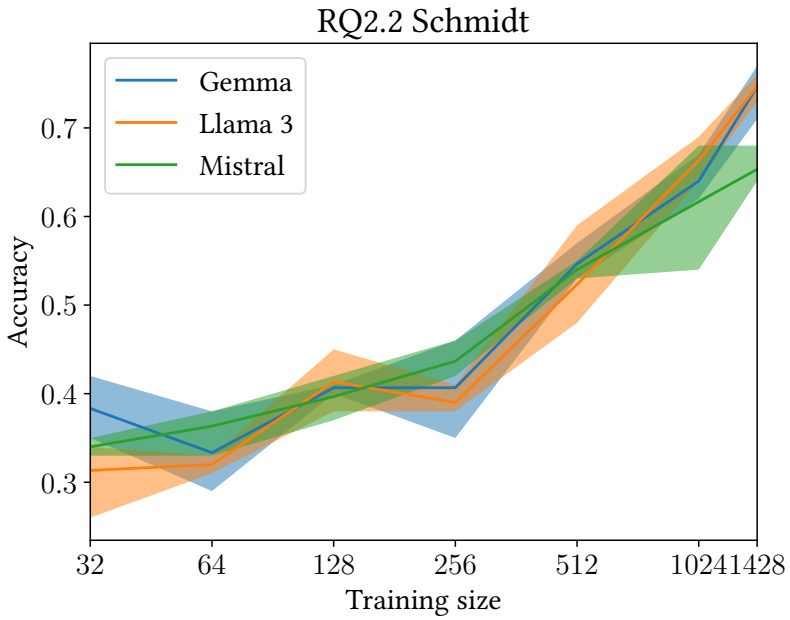


Figure 6.7: Performance metrics of all three models on the Schmidt dataset.

OOTB experiments required less data to improve, while those with better OOTB results needed more data, ultimately achieving comparable performance.

On the smaller OMP dataset, only the Gemma and Llama 3 models showed performance improvements, while the Mistral model did not. The limited training data was sufficient to fine-tune the Gemma and Llama 3 models, but Mistral appears to require more data.

Evaluated on the Schmidt dataset, similar trends were observed. The Gemma and Llama 3 models adapted well to sparse data, whereas the Mistral model improved but did not surpass the OOTB results. Given difficulties for the Mistral model to adapt with in-context learning, further research is needed to enhance its performance with limited data.

In summary, fine-tuning with LoRA adapters is feasible for sparse data, although not universally effective across all models. Some models adapt quickly with less data, whereas others require significantly more data. Enhancing results while reducing the required data should be a focus for future work, as the current findings show promise for low data environments. LoRA adapters appear effective with training data containing at least 512 samples, with performance improving further as more data becomes available. Future experiments should verify whether adjusting hyperparameters can yield even better results.

6.3 RQ2.3: Sentence Transformers Fine-tuning (SetFit)

Another effective technique for fine-tuning models, particularly Sentence Transformers, is referred to as SetFit. In contrast to adapters, SetFit does not introduce any additional parameters into the model. Instead, the model weights are refined utilizing a contrastive learning approach. This attribute makes SetFit exceptionally useful for situations where data is sparse. SetFit is a good third strategy for adapting a model to a domain.

6.3.1 Training details

In this experiment, the weights of a sentence transformer model are trained using the SetFit approach. Despite the availability of the sentence transformer-based Mistral 7B Instruct, it cannot be utilized due to memory limitations that hinder its training. To address this research question, it is necessary to adapt a smaller model. The top-performing embedding models undergo regular evaluation in the Massive Text Embedding Benchmark (MTEB). A part of this benchmark is the Amazon Review classification task, which involves categorizing Amazon reviews into one to five stars based on their sentiment. As models are also evaluated on the German subset, this yields valuable insights into the most effective OOTB sentiment classification model for German(Muennighoff et al., 2023).

As of June 1, 2024, the model "intfloat/multilingual-e5-large-instruct" was selected due to its exemplary performance and relatively compact size. This model, built on the RoBERTa architecture, benefits from training on multilingual data (Wang et al., 2024). The parameter selection is based on the work by Tunstall et al. (2022) with a few adjustments:

```
Batch size: 32
Epochs: 1
Sampling Strategy: Oversampling
Warmup proportion: 0.01
```

The SetFit method is not originally designed to use prompts; however, the model is fine-tuned using prompts. In the context of the Amazon Review Classification task, the fine-tuning process leverages a specific prompt. The adapted prompt is as follows:

```
Instruct: Classify the sentiment of a given text
as either positive, negative, or neutral.
Query:
```

To measure the performance on different training sizes, the data is divided into subsets containing 32, 64, 128, 256, 512, and 1024 samples for the OMP and Schmidt dataset. Additionally, an extra subset containing 2048 samples is included for the GermEval task. Considering the exponential growth in training time associated with larger datasets, fine-tuning for data sizes exceeding 2048 samples has not been explored.

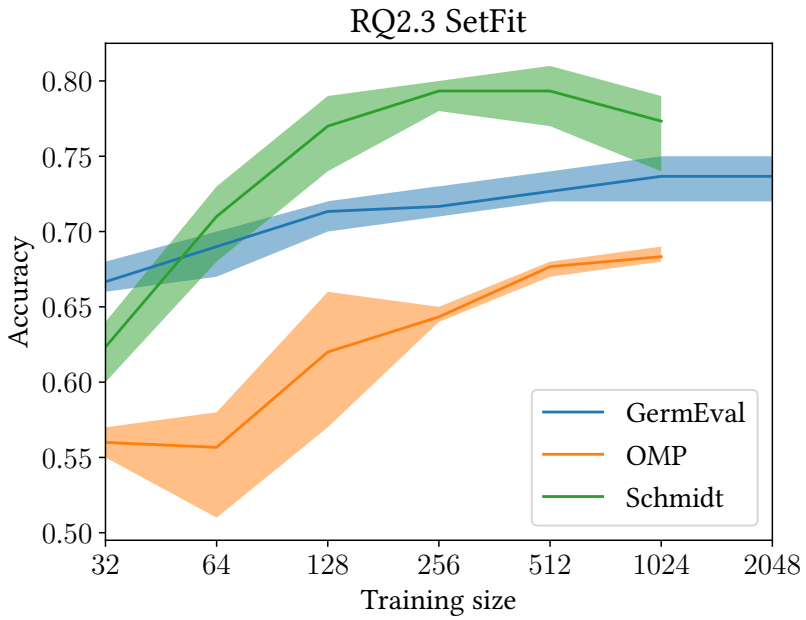


Figure 6.8: The y-axis represents accuracy, while the x-axis denotes training size. The lines indicate mean values, and the shaded areas represent the deviation across three runs.

6.3.2 Results

RQ2.3							
Dataset	32	64	128	256	512	1024	2048
GermEval	0.67	0.69	0.71	0.72	0.73	0.74	0.74
OMP	0.56	0.56	0.62	0.64	0.67	0.68	
Schmidt	0.62	0.71	0.77	0.79	0.79	0.77	
Training time (minutes)	2	3.5	9.5	30	112	483	1901

Table 6.8: SetFit training performance metrics, averaged over three runs.

The results are presented in Table 6.8, which shows the mean accuracy across different training sizes over three runs. Detailed data is available in the Appendix, Table A.4. The model shows good performance on the GermEval dataset, even with smaller training sizes. With additional data, the model achieves optimal performance at 1,024 samples and above in terms of accuracy. The model is performing slight improvements as the training data size increases. In comparison to training with LoRA adapters, the training time increases exponentially, rather than linearly, due to the integration of contrastive learning. Despite achieving the highest accuracy at 1,024 training samples, using half the training data yields almost the same accuracy with only a quarter of the training time. Since the improvement in terms of accuracy between 128 and 1,024 samples is marginal, the method proves effective when adapting with limited training data. Training with 4,096 samples was evaluated in one run, as detailed in the Appendix, Table A.4, but results showed only slight improvement. Further evaluation of training on 4,096 samples was not conducted due to the high training time of over 7,600 minutes (127 hours).

The results on the OMP dataset are lower for the trained models, and the deviation in training runs with fewer than 512 samples is significant, as shown in Figure 6.8.

The model adapts to the dataset with an overall accuracy of 0.68, compared to 0.56 with only 32 training samples. This is an increase of 21%. On the Schmidt dataset, the model achieves an accuracy of 0.79 after using 256 training samples. This is the smallest training size required to achieve the highest accuracy compared to the results on other datasets. The visualized results in Figure 6.8 indicate that training with more than 256 examples results in greater scattering. It appears the model begins to overfit, as the accuracy declines with increased data. Comparing the datasets, the model is performing well early and rapidly reaches its peak performance. One potential reason could be the diverse data sources in the GermEval dataset, in contrast to the Schmidt dataset, which consists solely of microblogging posts. The approach appears to deliver satisfactory results quickly, but also begins to overfit early.

6.3.3 Conclusion

The contrastive learning method demonstrated promising results from the early stages on the GermEval and Schmidt datasets. This training method can be particularly beneficial for small datasets, owing to its low time requirements. The requirement to train all weights makes it a computationally expensive method as well. In this case, the results are not directly comparable with those from prior research questions because the models differ significantly in architecture and size. Nevertheless, the findings illustrate the capability of the SetFit training approach in enhancing existing models with sparse data for sentiment analysis. The model adapts rapidly to the data, even with small training sets. With additional computational power, further evaluations using larger models are feasible, which could be particularly interesting given the promising results of this approach.

7

RQ3: Adapting and Generating Lexicons with Weak and Unsupervised Learning

In RQ1, the rule-based method using lexicons generally yielded lower results compared to machine learning models, including large language models. However, lexicons continue to be used and developed due to their extremely low computational requirements. One reason for this is the complete transparency of the results, as well as the ease and cost-effectiveness of reproducibility. Additionally, there is no need for extensive data engineering for their creation, adaptation, or prediction, making them easily usable. Unlike supervised machine learning models that demand large annotated datasets, the process of developing and maintaining lexicons can often be accomplished with minimal resources and infrastructure.

Since LLMs offer the advantage of performing tasks they are trained on, as well as tasks they are not explicitly trained for, this flexibility could be leveraged. LLMs could be utilized not only for analyzing texts for classification, but also for creating lexicons containing sentiment-bearing words. Given that LLMs like BERT or GPT have shown proficiency in understanding context and semantics, they can assist in generating contextually relevant lexicon entries. This could bridge the gap between traditional lexicon methods and machine learning approaches, creating a hybrid model that uses the strengths of both. Such an approach not only minimizes the labor-intensive process of manual lexicon creation but also allows adaptations as language changes, ensuring that the lexicons remain relevant and effective over time.

7.1 RQ3.1: Creating lexicons through LLM prompting

Large language models offer the advantage of understanding tasks, even those they were not explicitly trained on. Because LLMs are trained on extensive datasets, their learned information can be utilized in new downstream tasks through prompts. The idea is to create lexicon entries with relevant values, similar to those used in SentiWS or GerVADER. This unsupervised approach can aid in generating domain-specific lexicons without requiring human annotation.

7.1.1 Technical details

Models

To evaluate the approach, different Instruct-based models are utilized, specifically Gemma 7B Instruct (referred to as Gemma), Llama 3 8B Instruct (Llama 3), and Mistral 7B Instruct v0.2 (Mistral). The approach is tested on three datasets: GermEval, OMP, and Schmidt. The model is presented with the documents from the training set, without the labels. This ensures that the approach is tested in an unsupervised manner, highlighting the capability to work without labeled data.

Prompt

The task is divided into multiple parts, each providing guidance for the task. Initially, the task involves analyzing the sentiment of a text. Additionally, each sentiment-bearing word should be assigned a value between 1 (most positive) and -1 (most negative) to represent sentiment strength. Since the results of RQ2.1 show better performance for when LLMs are provided with contextual task information in most situations, in-context learning is employed. This step is crucial because the difficulty of the task is higher compared to previous tasks. Since models tend to justify their decisions, allowing them to provide additional reasoning helps maintain the desired format. The following prompt is used for creating the lexicon:

```
You are provided with a text, and your task is
to analyze the sentiment. Analyze each sentiment-bearing
word and emotion in the context of the text.
For each sentiment-bearing word, assign a sentiment value
between -1 (most negative) and 1 (most positive).
Neutral words or those not bearing sentiment should
not be assigned any value.
Format: word - value - reasoning.
```

This is a more detailed prompt compared to those used in RQ1 and RQ2, showing the increased complexity of the task. For this experiment, the model is provided with three examples of text and their corresponding answers to guide the process. The resulting text is divided by the hyphen, and sentiment-bearing words with their assigned values are added to a list. The results are collected, post-processed, and used as the lexicon for GerVADER.

Post-processing

For post-processing, several steps are applied to filter the data. Firstly, entries containing single characters, special characters, or stop words are removed. As each document is processed independently, single words may sometimes be assigned both positive and negative values in the process. Moreover, the assigned sentiment strength can differ between sentences. The average of these values is taken as the final value. Each observed inflection is assigned the same value, assuming that words generally maintain consistent sentiment regardless of their inflection. This approach also effectively handles any typos. Therefore, the lemma of each word is used for calculation. Values with a sentiment strength below 0.30 for positive words and above -0.30 for negative words are filtered out to reduce slightly positive or negative words, which can affect the accurate prediction of the neutral class.

7.1.2 Results

RQ3.1				
Dataset	GerVADER	Gemma	Llama 3	Mistral
GermEval	0.37	0.54	0.40	0.51
OMP	0.32	0.50	0.49	0.49
Schmidt	0.48	0.54	0.49	0.50

Table 7.1: Comparison of accuracy results between the GerVADER lexicon and LLM-generated lexicons.

The results presented in Table 7.1 offer a side-by-side comparison of the accuracy of lexicons generated by prompting the models. All lexicons created by prompting LLMs outperformed the OOTB GerVADER lexicon in terms of accuracy. The accuracy of the lexicon generated by the Gemma model is 46% higher as compared to the OOTB model, and even surpasses the Gemma zero-shot results from RQ1.

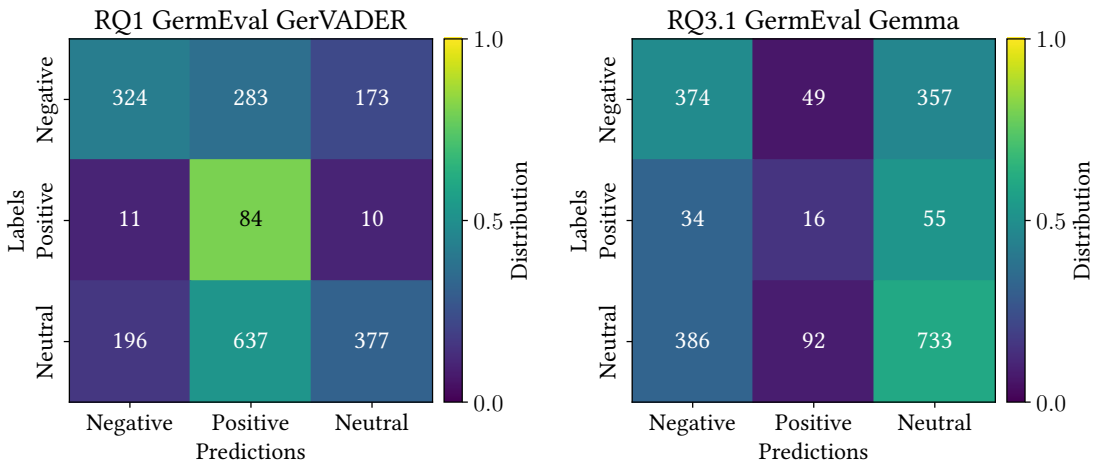


Figure 7.1: Confusion matrix for the GerVADER lexicon (left) and the generated lexicon with Gemma (right) on the GermEval dataset.

The class-wise results shown in Figure 7.1 point out significant improvements for the neutral class when comparing the GerVADER lexicon with the Gemma-generated lexicon for the GermEval dataset. The accuracy for the positive class is lower for the generated model when compared to the GerVADER lexicons results. The low accuracy for the positive class might be due to averaging the sentiment values of words across all occurrences. Only 6% of the GermEval dataset is labeled positive. Therefore, if a word is classified both positively and negatively within different sentences, it is more likely to be assigned a negative value due to the greater number of negative documents. The accuracy for the negative class is 0.48 for the Gemma generated lexicon, which is 14% higher than the GerVADER lexicons accuracy of 0.42.

The LLM-generated lexicons for the OMP dataset all outperformed the GerVADER OOTB lexicon. Notably, the OMP dataset does not have a train split. Therefore, RQ1 is evaluated on the complete dataset, while for this experiment, the lexicon is generated from the first half of the dataset and evaluated on the second half. The lexicon generated from the first half of the OMP dataset by the Gemma model performs best, achieving an accuracy of 0.50. This is 56% higher compared to the GerVADER lexicon in terms of accuracy. On the Schmidt dataset, the lexicon created with the Gemma model outperforms the GerVADER lexicon, with an accuracy of 0.54, which is 12% higher.

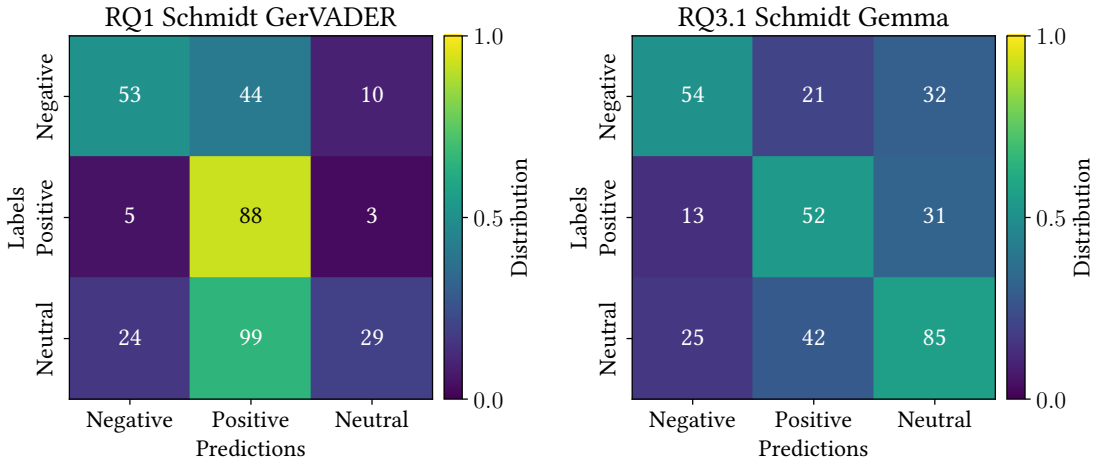
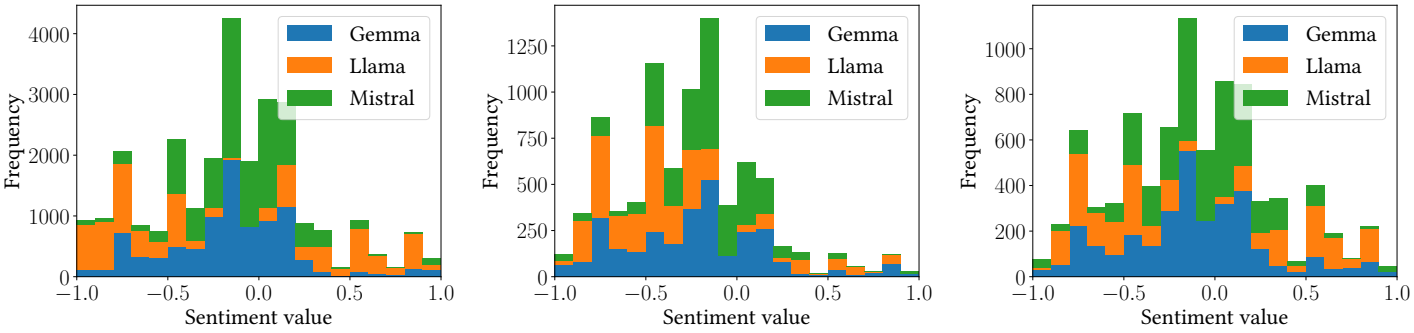


Figure 7.2: Confusion matrix for the GerVADER lexicon (left) and the generated lexicon with Gemma (right) on the Schmidt dataset.

The detailed results in Figure 7.2 compare the class-wise accuracy between RQ1 and the lexicon generated by the Gemma model. The generated lexicon performs equally well for the negative class, while the accuracy for the positive class declines. This decline may be due to the imbalanced training data in the dataset. The relatively small training dataset, with around 1400 examples, could also have an impact on this problem. Given that the training was unsupervised, additional data could still be beneficial without the need for labeling. Since the training was done unsupervised, additional data could be beneficial still without the need of labeling. The improved overall results are mainly related to the neutral class, achieving an accuracy of 0.56 for the generated lexicon by Gemma compared to 0.19 for the GerVADER lexicon.

The histograms in Figure 7.3 illustrate the distribution of sentiment between the lexicons for the datasets. The histograms show the lexicons after post-processing, but



(a) GermEval (b) OMP (c) Schmidt

Figure 7.3: Distribution of results for all models across the datasets.

without filtering out values below 0.30 for positive words and above -0.30 for negative words. The Llama 3 model tends to select higher polarities compared to the other models, particularly in the positive spectrum for the GermEval and OMP datasets. Neither the frequency nor the sentiment value can be directly considered; rather, they serve as guidance to help interpret the results.

All the models have included significantly more negative words than positive ones in the lexicons. This pattern is consistent with the initial general distribution observed in the dataset. However, at first glance, the number of positive words appears significantly smaller than the number of negative words.

This finding should be interpreted with caution due to certain unknown factors. It is unclear whether positive documents contain significantly more positive words compared to negative documents. Furthermore, it is unclear if positive documents tend to have more diverse sentiment descriptions, whereas negative documents may contain more repetitive negative sentiment. Additionally, it is uncertain whether positive words appear in documents with an overall negative sentiment.

7.2 RQ3.2: LLM Embeddings for lexicon extension

Since embeddings can convey a significant amount of information, the idea is to directly utilize them to expand an existing lexicon. The approach leverages embeddings by starting with a small lexicon and expanding it by searching for semantically similar words in an embedding database. Within the same domain, semantically similar words tend to produce similar word embeddings due to their contextual relevance. This methodology allows for the incremental and automated growth of the lexicon, minimizing the need for manual intervention. This can help to capture subtle semantic variations, enhancing the effectiveness and accuracy of the lexicon-based methods.

7.2.1 Technical Details

To create a small seed lexicon, class-based TF-IDF (c-TF-IDF) is used, finding the most significant terms in each class. To achieve this, the algorithm analyzes the term frequencies and their distribution in the training set. The 500 most frequently used words for both the positive and negative classes from the training dataset are selected, forming a small lexicon of up to 500 words for each category. Furthermore, a full

lexicon is generated using the c-TF-IDF approach to serve as a baseline for comparison against the extended seed lexicon.

Due to technical difficulties with the Mistral model, the tokenizer cannot create a mapping from tokens back to words. Consequently, the word embeddings are created only using the Gemma 7B (Gemma) and Llama 3 8B (Llama 3) models. The models generate outputs for each document, which are then segmented into individual words. The words are stored in the vector database without the label of the sentence. The weights of the last four layers are averaged to combine them, as this conveys the most useful features. The resulting words and their embeddings are stored in the vector database Weaviate ("Weaviate," n.d.).

Once the entire dataset has been processed in an unsupervised manner, words from the seed lexicons are used to query the vector database. For each seed word in the vector database, up to five embeddings are used to search for the five most similar embeddings, resulting in a maximum of 25 additional words. The results are filtered to ensure embeddings with the same word are not chosen. Additionally, each embedding ID is verified to ensure it is used only once for each seed word. When a word from the seed lexicon is found in the vector database, the most similar word embeddings are collected and the words are merged into the expanding lexicon. The sentiment values assigned to the seed lexicon words are then applied to these newly identified similar words. These new similar words, along with their assigned sentiment values, are added to the lexicon, extending it incrementally. In the post-processing, stop-words are removed, and the values are recalculated by taking the mean of the lemmas.

7.2.2 Results

RQ3.2				
Dataset	GerVADER	c-TF-IDF	Gemma	Llama 3
GermEval	0.37	0.54	0.44	0.44
OMP	0.32	0.49	0.57	0.53
Schmidt	0.48	0.48	0.45	0.41

Table 7.2: Performance comparison of the GerVADER, c-TF-IDF-based and LLM-extended lexicons.

As shown in Table 7.2, the extension of seed lexicons produced different levels of success across various datasets. The lexicon created using the c-TF-IDF method outperforms the OOTB GerVADER and both LLM-extended lexicons on the GermEval dataset in terms of accuracy. It shows the best performance among all methods tested. The LLM-extended lexicons created using Gemma and Llama 3 both improve results compared to the GerVADER lexicon but perform lower in accuracy compared to the full c-TF-IDF lexicon. Both extended lexicons achieve an accuracy of 0.44, which is 19% higher compared to the GerVADER lexicon.

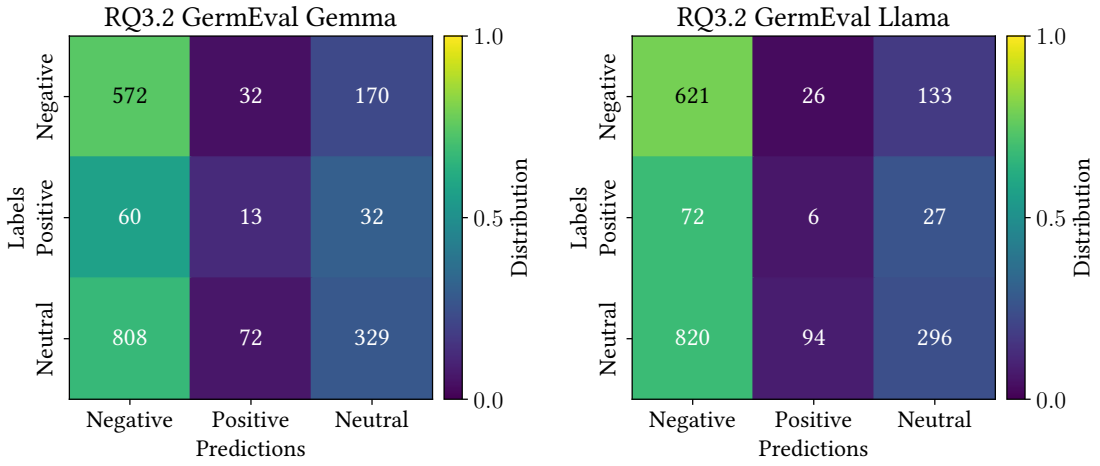


Figure 7.4: Confusion matrix for the results of the generated lexicon utilizing Gemma (left) and Llama 3 (right) on the Schmidt dataset.

As shown in Figure 7.4, both the Gemma and Llama models mostly predict the documents as negative. Both lexicons have a very low likelihood of predicting the positive class, which is visualized by the blue column in the middle. For the OMP dataset, the lexicon extended utilizing Gemma achieves the highest accuracy, which is 0.57. This accuracy is 78% higher compared to the accuracy of the GerVADER lexicon. Moreover, the c-TF-IDF-generated lexicon also outperforms the GerVADER lexicon in terms of accuracy. Comparing the results of the GerVADER lexicon with the other lexicons, none show a significant improvement in accuracy, except for the c-TF-IDF and Gemma embeddings methods.

7.3 Conclusion

Applying LLMs for lexicon generation can produce promising results. Creating lexicons by prompting leverages the robust capabilities of LLMs, enabling the creation of domain-specific lexicons without the need for extensive manual annotation. All models generate lexicons by analyzing the given documents and outperform the GerVADER lexicon, with the Gemma model producing the best results in terms of accuracy. A detailed analysis indicates that the greatest accuracy gains are achieved in correctly predicting the neutral class. These findings suggest that while traditional lexicons remain valuable, integrating LLM capabilities can significantly advance lexicon generation, offering more flexible and precise results. Creating lexicons by prompting LLMs is a promising approach that can notably enhance the lexicon generation process.

However, experimenting with LLM embeddings has been less successful compared to the previous approach. The primary goal of leveraging word embeddings to expand an existing lexicon and potentially enhance its performance was achieved only partially. The c-TF-IDF method, serving as a baseline, achieves comparable or superior accuracy to the GerVADER lexicon. Despite higher accuracy for the GermEval and OMP datasets, the class-wise accuracy on the GermEval dataset exhibits a considerable bias towards the negative class.

Thus, while employing additional information within embeddings presents an interesting avenue for future research, more work is necessary to manage and improve these results. Future efforts should concentrate on fine-tuning models and exploring advanced adapters to harness the full potential of this approach for even better outcomes. Additionally, further refining the calculation of sentiment values, for instance by considering the distance between a negative word and its opposing class, could contribute to achieving more accurate and nuanced performance.

8 Conclusion

The first research question compared the performance of various models for sentiment analysis on German datasets, both monolingual and multilingual approaches. The findings highlight that model performance can vary significantly depending on the dataset and domain. Generally, models specifically trained on particular datasets tended to perform better. However, some models have encountered difficulties, particularly with neutral sentiments or when applied to datasets outside their training domain. Overall, advancements between model generations offered promising performance enhancements. Across all domains, no single model consistently outperformed the others. This highlights the critical need for model adaptation and domain-specific fine-tuning to effectively handle diverse German-language datasets.

In the evaluation of effective strategies for domain adaptation with sparse labeled data, several key findings were observed. In-context learning shows good results to guide LLMs in understanding the task better, though its effectiveness varies depending on the model and dataset. While additional context can lead to significant performance improvements, the outcome is not always positive across all scenarios. For instance, the Gemma model demonstrated significant improvement with additional context, while the Mistral model showed mixed results. This indicates that in-context learning requires careful adaptation to both the model and the specific domain to achieve optimal results.

Fine-tuning techniques like LoRA adapters demonstrate potential, especially with limited data. Experiments demonstrated that on the GermEval dataset, which had the largest training set, all models showed improved performance, achieving similar accuracy levels with full training data. This highlights the advantages of employing LoRA adapters, especially for models that initially showed lower accuracy in out-of-the-box (OOTB) experiments. Models such as Gemma and Llama 3 adapted well to sparse datasets, although some models like Mistral require more data to improve. This points to the need for further research into why certain models like Mistral require more data to achieve significant performance gains.

The findings illustrate the capability of the SetFit training approach in enhancing models with sparse data. Contrastive learning shows promising results, especial for small datasets. However, the computational expense due to the need to train all weights

is a notable consideration. Overall, while no single strategy universally fits all models and datasets, both LoRA adapters and SetFit present promising approaches for enhancing model performance in low-data environments. Future work should focus on refining these techniques, adjusting hyperparameters, and exploring the impact of larger datasets and more computational resources. Additionally, the integration of hybrid approaches that combine the strengths of in-context learning, LoRA adapters, and contrastive learning could be considered to identify optimal combinations. Moreover, newer models with higher parameter sizes could be beneficial in future work or also LLMs trained on German texts.

The third research question aims to explore how lexicons can be automatically adapted, generated, or updated using weak or unsupervised learning techniques. While lexicons are computationally inexpensive, transparent, and easily shareable, the objective was to determine if domain-specific lexicons could be effectively generated or adapted using advanced learning techniques. The findings from this study suggest that the application of large language models for lexicon generation indeed presents promising results. The approach harnesses the capabilities of LLMs to create domain-specific lexicons without the need for extensive manual annotation. Considering the complexity of the task, the models have performed quite well. Furthermore, experimentation with LLM embeddings for lexicon extension showed mixed results. The aim of using word embeddings to expand existing lexicons and potentially enhance their performance was partially achieved. Specifically, for the GermEval and OMP datasets, LLM-extended lexicons outperformed GerVADER results, indicating promising directions for future research. This suggests that LLM-extended lexicons could outperform traditional ones in certain contexts. While traditional lexicons remain useful, integrating LLM capabilities can significantly enhance lexicon generation and adaptation. Future work should focus on further refinement of these approaches to achieve even better results. Furthermore, the implementation of more sophisticated techniques for calculating sentiment values could potentially enhance the performance.

Due to the rapid evolution and extension of languages, sentiment analysis is still a topic of research, even in the era of large language models. The presented work compares approaches, trying to help this direction of research.

The code to reproduce the results is freely available under an open license[1].

1. https://github.com/Alienmaster/MasterThesis

Appendices

A
Appendix

A.1 RQ2.1

```
<|begin_of_text|>
<|start_header_id|>user<|end_header_id|>
    Classify the sentiment of the text into ONE of the three classes:
    neutral, negative or positive.
   Text: köln: wo sich in der bahn ein mittfünfziger im trikot neben dich
    setzt und dir lebenstipps gibt. ich würde nicht tauschen wollen.
<|eot_id|>
<|start_header_id|>assistant<|end_header_id|>neutral<|eot_id|>
<|start_header_id|>user<|end_header_id|>
    Classify the sentiment of the text into ONE of the three classes:
    neutral, negative or positive.
    Text: RT @holgi: Hui, die neuen QR-Lesegeräte der Bahn
    sind mal sauschnell... Huiuiui
<|eot_id|>
<|start_header_id|>assistant<|end_header_id|>positive<|eot_id|>
<|start_header_id|>user<|end_header_id|>
    Classify the sentiment of the text into ONE of the three classes:
    neutral, negative or positive.
    Text: @DB_Bahn manchmal fragt man sich,
    warum es euch überhaupt noch gibt!!
<|eot_id|>
<|start_header_id|>assistant<|end_header_id|>negative<|eot_id|>
<|start_header_id|>user<|end_header_id|>
    Classify the sentiment of the text into ONE of the three classes:
    neutral, negative or positive.
    Text:  screams @ deutsche bahn.
<|eot_id|>
<|start_header_id|>assistant<|end_header_id|>
```

Figure A.1: In context learning prompt for a Llama-based model with three documents from GermanEval

A.2 RQ2.2

Model	Run	32	64	128	256	512	1024	2048	4096	8192	16201
Gemma	0	0.42	0.41	0.46	0.44	0.51	0.56	0.56	0.68	0.77	0.79
Gemma	1	0.43	0.39	0.46	0.46	0.50	0.58	0.69	0.64	0.76	0.81
Gemma	2	0.47	0.39	0.36	0.39	0.52	0.57	0.61	0.73	0.80	0.80
Llama 3	0	0.48	0.24	0.43	0.44	0.44	0.54	0.56	0.68	0.78	0.80
Llama 3	1	0.48	0.42	0.37	0.44	0.44	0.60	0.65	0.75	0.78	0.82
Llama 3	2	0.29	0.39	0.43	0.42	0.43	0.55	0.62	0.64	0.77	0.80
Mistral	0	0.46	0.49	0.45	0.45	0.50	0.50	0.63	0.70	0.77	0.79
Mistral	1	0.41	0.42	0.43	0.49	0.50	0.52	0.63	0.74	0.74	0.79
Mistral	2	0.35	0.40	0.41	0.46	0.44	0.58	0.54	0.64	0.78	0.81

Table A.1: RQ2.2: Results with LoRA adapters on GermEval.

Model	Run	32	64	128	256	512	1024	1796
Gemma	0	0.39	0.47	0.47	0.49	0.60	0.65	0.67
Gemma	1	0.48	0.50	0.52	0.50	0.51	0.63	0.66
Gemma	2	0.47	0.52	0.51	0.49	0.58	0.60	0.68
Llama 3	0	0.27	0.42	0.49	0.54	0.51	0.62	0.66
Llama 3	1	0.47	0.50	0.55	0.55	0.60	0.62	0.65
Llama 3	2	0.18	0.47	0.48	0.53	0.55	0.65	0.68
Mistral	0	0.32	0.51	0.51	0.53	0.54	0.59	0.55
Mistral	1	0.40	0.48	0.50	0.55	0.53	0.59	0.56
Mistral	2	0.53	0.49	0.51	0.51	0.54	0.60	0.59

Table A.2: RQ2.2: Results with LoRA adapters on OMP

Model	Run	32	64	128	256	512	1024	1425
Gemma	0	0.35	0.29	0.41	0.46	0.57	0.62	0.71
Gemma	1	0.38	0.33	0.40	0.41	0.54	0.67	0.76
Gemma	2	0.42	0.38	0.41	0.35	0.53	0.63	0.77
Llama 3	0	0.34	0.32	0.41	0.38	0.59	0.69	0.75
Llama 3	1	0.34	0.33	0.38	0.41	0.50	0.66	0.76
Llama 3	2	0.26	0.31	0.45	0.38	0.48	0.64	0.73
Mistral	0	0.34	0.38	0.42	0.42	0.54	0.63	0.64
Mistral	1	0.33	0.38	0.40	0.43	0.55	0.54	0.68
Mistral	2	0.35	0.33	0.37	0.36	0.53	0.68	0.64

Table A.3: RQ2.2: Results with LoRA adapters on Schmidt.

A.3 RQ2.3

Dataset	Run	32	64	128	256	512	1024	2048	4096
GermEval	0	0.66	0.70	0.72	0.71	0.72	0.74	0.72	0.76
GermEval	1	0.66	0.70	0.70	0.71	0.72	0.75	0.75	
GermEval	2	0.68	0.67	0.72	0.73	0.74	0.72	0.74	
OMP	0	0.57	0.51	0.63	0.64	0.68	0.68		
OMP	1	0.55	0.58	0.66	0.65	0.68	0.69		
OMP	2	0.56	0.58	0.57	0.64	0.67	0.68		
Schmidt	0	0.60	0.68	0.78	0.80	0.80	0.79		
Schmidt	1	0.64	0.73	0.79	0.78	0.81	0.74		
Schmidt	2	0.63	0.72	0.74	0.80	0.77	0.79		

Table A.4: RQ2.3: Results of intfloat/multilingual-e5-large-instruct on different datasets.

References

Llama Team AI@Meta. 2024. The Llama 3 Herd of Models (July). (Cited on pages 10 sq., 23).

Wazir Ali, Naveed Ali, Yong Dai, Jay Kumar, Saifullah Tumrani, and Zenglin Xu. 2021. Creating and Evaluating Resources for Sentiment Analysis in the Low-resource Language: Sindhi. In *Proceedings of the Eleventh Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis,* 188–194. Online, April. (Cited on page 14).

Seth Aycock and Rachel Bawden. 2024. Topic-guided Example Selection for Domain Adaptation in LLM-based Machine Translation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop,* 175–195. St. Julian's, Malta, March. (Cited on pages 17, 40).

Alexandra Balahur, Ralf Steinberger, Mijail Kabadjov, Vanni Zavarella, Erik van der Goot, Matina Halkia, Bruno Pouliquen, and Jenya Belyaeva. 2010. Sentiment Analysis in the News. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10),* 2216–2220. Valletta, Malta, May. (Cited on page 1).

Francesco Barbieri, Luis Espinosa Anke, and Jose Camacho-Collados. 2022. XLM-T: Multilingual Language Models in Twitter for Sentiment Analysis and Beyond. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference,* 258–266. Marseille, France, June. (Cited on page 16).

Jeremy Barnes. 2023. Sentiment and Emotion Classification in Low-resource Settings. In *Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis,* 290–304. WASSA 2023. Toronto, Canada, July. (Cited on page 17).

Bethard, Steven, Carpuat, Marine, Apidianaki, Marianna, Mohammad, Saif M., Cer, Daniel, and Jurgens, David, eds. 2017. Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017). Vancouver, Canada, August. (Cited on page 16).

Yuri Bizzoni, Pascale Moreira, Mads Rosendahl Thomsen, and Kristoffer Nielbo. 2023. Sentimental Matters - Predicting Literary Quality by Sentiment Analysis and Stylometric Features. In *Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis,* 11–18. Toronto, Canada, July. (Cited on page 15).

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,* edited by H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, 33:1877–1901. online, December. (Cited on pages 8, 10, 12, 16).

Alessio Buscemi and Daniele Proverbio. 2024. ChatGPT vs Gemini vs LLaMA on Multilingual Sentiment Analysis, January. arXiv: 2402.01715. (Cited on page 25).

Mark Cieliebak, Jan Milan Deriu, Dominic Egger, and Fatih Uzdilli. 2017. A Twitter Corpus and Benchmark Resources for German Sentiment Analysis. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media,* edited by Lun-Wei Ku and Cheng-Te Li, 45–51. Valencia, Spain, April. (Cited on page 29).

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),* 4171–4186. Minneapolis, Minnesota, June. (Cited on pages 9 sq., 12).

Gemma Team et al. 2024. Gemma: Open Models Based on Gemini Research and Technology, arXiv:2403.08295, April. (Cited on pages 10 sq.).

Aniruddha Ghosh, Guofu Li, Tony Veale, Paolo Rosso, Ekaterina Shutova, John Barnden, and Antonio Reyes. 2015. SemEval-2015 Task 11: Sentiment Analysis of Figurative Language in Twitter. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015),* 470–478. Denver, Colorado, June. (Cited on page 2).

Dirk Goldhahn, Thomas Eckart, and Uwe Quasthoff. 2012. Building Large Monolingual Dictionaries at the Leipzig Corpora Collection: From 100 to 200 Languages. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12),* 759–765. Istanbul, Turkey, May. (Cited on page 20).

Maarten Grootendorst. 2022. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. arXiv: 2203.05794 [cs.CL]. (Cited on page 8).

Oliver Guhr, Anne-Kathrin Schumann, Frank Bahrmann, and Hans Joachim Böhme. 2020. Training a Broad-Coverage German Sentiment Classification Model for Dialog Systems. In *Proceedings of The 12th Language Resources and Evaluation Conference,* 1620–1625. Marseille, France, May. (Cited on pages 3, 10, 21, 24, 27).

Tim Highfield. 2017. Social media and everyday politics. John Wiley & Sons. (Cited on page 20).

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, no. 8 (November): 1735–1780. (Cited on page 8).

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-Efficient Transfer Learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning,* 97:2790–2799. Proceedings of Machine Learning Research. June. (Cited on page 13).

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations.* online, April. (Cited on pages 3, 13, 40).

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining,* 168–177. August. (Cited on page 14).

Xia Hu and Huan Liu. 2012. Text analytics in social media. *Mining text data* (January): 385–414. (Cited on page 2).

C. Hutto and Eric Gilbert. 2014. VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text. In *Proceedings of the eigth International AAAI Conference on Web and Social Media,* 216–225. Ann Arbor, MI, USA, May. (Cited on page 7).

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B, October. (Cited on page 11).

Young G. Jung, Kyung T. Kim, Byungjun Lee, and Hee Y. Youn. 2016. Enhanced Naive Bayes Classifier for real-time sentiment analysis with SparkR. In *2016 International Conference on Information and Communication Technology Convergence (ICTC),* 141–146. Jeju Island, Korea, October. (Cited on pages 3, 5, 15).

Daniel Jurafsky and James H. Martin. 2024. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition. Third. Prentice Hall, February. (Cited on page 7).

Anna Jurek, Maurice D. Mulvenna, and Yaxin Bi. 2015. Improved lexicon-based sentiment analysis for social media analytics. *Security Informatics* 4 (December): 1–13. (Cited on page 15).

Kiana Kheiri and Hamid Karimi. 2023. SentimentGPT: Exploiting GPT for Advanced Sentiment Analysis and its Departure from Current Machine Learning, July. arXiv: 2307.10234 [cs.CL]. (Cited on pages 3, 6, 16).

Christopher SG Khoo and Sathik Basha Johnkhan. 2018. Lexicon-based sentiment analysis: Comparative evaluation of six sentiment lexicons. *Journal of Information Science* 44, no. 4 (August): 491–511. (Cited on page 6).

Lisa M Kruse, Dawn R Norris, and Jonathan R Flinchum. 2018. Social media as a public sphere? Politics on social media. *The Sociological Quarterly* 59, no. 1 (January): 62–84. (Cited on page 20).

Moritz Laurer, Wouter Van Atteveldt, Andreu Casas, and Kasper Welbers. 2024. Less Annotating, More Classifying: Addressing the Data Scarcity Issue of Supervised Machine Learning with Deep Transfer Learning and BERT-NLI. *Political Analysis* 32, no. 1 (January): 84–100. (Cited on pages 10, 21).

Quanzhi Li and Sameena Shah. 2017. Learning Stock Market Sentiment Lexicon and Sentiment-Oriented Word Vector from StockTwits. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017),* 301–310. Vancouver, Canada, August. (Cited on page 3).

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, arXiv:1907.11692, July. arXiv: 1907.11692[cs]. (Cited on page 10).

Lxyuan. 2023. lxyuan/distilbert-base-multilingual-cased-sentiments-student. https://huggingface.co/lxyuan/distilbert-base-multilingual-cased-sentiments-student. Accessed: 2024-05-01, May. (Cited on pages 10, 24).

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the Role of Demonstrations: What Makes In-Context Learning Work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing,* 11048–11064. EMNLP 2022. Abu Dhabi, United Arab Emirates, December. (Cited on pages 16, 40).

Saif Mohammad and Peter Turney. 2010. Emotions Evoked by Common Words and Phrases: Using Mechanical Turk to Create an Emotion Lexicon. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text,* 26–34. Los Angeles, CA, USA, June. (Cited on page 14).

Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2023. MTEB: Massive Text Embedding Benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics,* edited by Andreas Vlachos and Isabelle Augenstein, 2014–2037. Dubrovnik, Croatia, May. (Cited on page 46).

Peter Norvig. 1987. Inference in text understanding. In *AAAI'87: Proceedings of the sixth National conference on Artificial intelligence - Volume 2,* 561–565. Seattle, WA, USA, July. (Cited on page 2).

Keiron O'Shea and Ryan Nash. 2015. An Introduction to Convolutional Neural Networks, arXiv:1511.08458, December. arXiv: 1511.08458[cs]. (Cited on page 8).

OpenAI. 2023. GPT-4 Technical Report. arXiv: 2303.08774 [cs.CL]. (Cited on page 16).

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (85): 2825–2830. (Cited on page 21).

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP),* edited by Alessandro Moschitti, Bo Pang, and Walter Daelemans, 1532–1543. EMNLP 2014. Doha, Qatar, October. (Cited on page 10).

Livia Polanyi and Annie Zaenen. 2006. Contextual Valence Shifters. In *Computing Attitude and Affect in Text: Theory and Applications,* 20:1–10. Springer. (Cited on pages 2, 6).

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training, (cited on page 16).

Christian Rauh. 2018. Validating a sentiment dictionary for German political language—a workbench note. *Journal of Information Technology & Politics* 15 (4): 319–343. (Cited on page 1).

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP),* 3982–3992. EMNLP-IJCNLP 2019. Hong Kong, China, November. (Cited on page 11).

Robert Remus, Uwe Quasthoff, and Gerhard Heyer. 2010. SentiWS - A Publicly Available German-language Resource for Sentiment Analysis. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10),* 1168–1171. Valletta, Malta, May. (Cited on pages 2, 7).

Irina Rish. 2001. An empirical study of the naive Bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence,* 41–46. August. (Cited on page 8).

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. In *The 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing @ NeurIPS 2019,* 1–5. Vancouver, Canada, December. (Cited on page 10).

Dietmar Schabus, Marcin Skowron, and Martin Trapp. 2017. One Million Posts: A Data Set of German Online Discussions. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval,* 1241–1244. New York, NY, USA, August. (Cited on pages 20 sq.).

Thomas Schmidt, Jakob Fehle, Maximilian Weissenbacher, Jonathan Richter, Philipp Gottschalk, and Christian Wolff. 2022. Sentiment Analysis on Twitter for the Major German Parties during the 2021 German Federal Election. In *Proceedings of the 18th Conference on Natural Language Processing (KONVENS 2022),* 74–87. Potsdam, Germany, September. (Cited on pages 5, 20 sq., 30).

Kim Schouten and Flavius Frasincar. 2016. Survey on Aspect-Level Sentiment Analysis. *IEEE Transactions on Knowledge and Data Engineering* 28, no. 3 (March): 813–830. (Cited on page 2).

Glorin Sebastian. 2023. Privacy and Data Protection in ChatGPT and Other AI Chatbots: Strategies for Securing User Information. *International Journal of Security and Privacy in Pervasive Computing (IJSPPC)* 15, no. 1 (July): 1–14. (Cited on page 3).

Fabrizio Sebastiani and Andrea Esuli. 2006. Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of the 5th international conference on language resources and evaluation,* 417–422. Genoa, Italy, June. (Cited on page 14).

Gayane Shalunts, Gerhard Backfried, and Katja Prinz. 2014. Sentiment Analysis of German Social Media Data for Natural Disasters. In *Proceedings of the 11th International ISCRAM Conference,* 752–756. State College, PA, USA, May. (Cited on page 5).

Kamran Shaukat, Ibrahim A. Hameed, Suhuai Luo, Imran Javed, Farhat Iqbal, Amber Faisal, Rabia Masood, Ayesha Usman, Usman Shaukat, Rosheen Hassan, Aliya Younas, Shamshair Ali, and Ghazif Adeem. 2020. Domain Specific Lexicon Generation through Sentiment Analysis. *International Journal of Emerging Technologies in Learning (iJET)* 15, no. 9 (May): 190–204. (Cited on page 6).

Xiaofei Sun, Xiaoya Li, Shengyu Zhang, Shuhe Wang, Fei Wu, Jiwei Li, Tianwei Zhang, and Guoyin Wang. 2023. Sentiment Analysis through LLM Negotiations, arXiv:2311.01876, November. (Cited on page 17).

Sayyida Tabinda Kokab, Sohail Asghar, and Shehneela Naz. 2022. Transformer-based deep learning models for the sentiment analysis of social media data. *Array* 14 (July): 1–12. (Cited on pages 8, 16).

Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. 2011. Lexicon-Based Methods for Sentiment Analysis. *Computational Linguistics* 37, no. 2 (June): 267–307. (Cited on page 14).

Kian L. Tan, Chin P. Lee, K. Anbananthen, and Kian M. Lim. 2022. RoBERTa-LSTM: A Hybrid Model for Sentiment Analysis With Transformer and Recurren Neural Network. *IEEE Access* 10:21517–21525. (Cited on pages 1, 15).

Mike Thelwall, Kevan Buckley, and Georgios Paltoglou. 2011. Sentiment in Twitter events. *Journal of the American Society for Information Science and Technology* 62, no. 2 (February): 406–418. (Cited on page 1).

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models, arXiv:2307.09288, August. arXiv: 2307.09288[cs]. (Cited on pages 11, 23).

Daniel Trottier, Christian Fuchs, et al. 2015. Social media, politics and the state. *Protests, Revolutions, Riots, Crime and Policing in the Age of Facebook, Twitter and YouTube* (January). (Cited on page 20).

Mikalai Tsytsarau and Themis Palpanas. 2012. Survey on mining subjective data on the web. *Data Mining and Knowledge Discovery* 24, no. 3 (May): 478–514. (Cited on page 1).

Lewis Tunstall, Nils Reimers, Unso Eun Seo Jo, Luke Bates, Daniel Korat, Moshe Wasserblat, and Oren Pereg. 2022. Efficient Few-Shot Learning Without Prompts, arXiv:2209.11055, September. arXiv: 2209.11055[cs]. (Cited on pages 12, 46).

Peter D. Turney. 2002. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics,* 417–424. Philadelphia, PA, USA, July. (Cited on page 14).

Karsten Michael. Tymann, Matthias Lutz, Patrick Palsbröker, and Carsten Gips. 2019. GerVADER-A German Adaptation of the VADER Sentiment Analysis Tool for Social Media Texts. In *Lernen, Wissen, Daten, Analysen 2019.* Berlin, Germany, September. (Cited on pages 2, 6 sq., 27).

Ellen van Kleef, Hans C. M. van Trijp, and Pieternel Luning. 2015. Consumer research in the early stages of new product development: a critical review of methods and techniques. *Food Quality and Preference* 16, no. 3 (April): 181–201. (Cited on page 1).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems,* 31:1–11. Long Beach, CA, USA, December. (Cited on page 9).

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual E5 Text Embeddings: A Technical Report, arXiv:2402.05672, February. arXiv: 2402.05672[cs]. (Cited on page 46).

Mayur Wankhade, Annavarapu Chandra Sekhara Rao, and Chaitanya Kulkarni. 2022. A survey on sentiment analysis methods, applications, and challenges. *Artificial Intelligence Review* 55, no. 7 (October): 5731–5780. (Cited on pages 1, 4 sq.).

Weaviate. n.d. Accessed: 2024-05-01. https://weaviate.io/. (Cited on page 54).

Ian H. Witten, Eibe Frank, and Mark A. Hall. 2011. Data Mining: Practical Machine Learning Tools and Techniques. 3rd ed. Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann. (Cited on pages 3, 8).

Michael Wojatzki, Eugen Ruppert, Sarah Holschneider, Torsten Zesch, and Chris Biemann. 2017. GermEval 2017: Shared Task on Aspect-based Sentiment in Social Media Customer Feedback. In *Proceedings of the GermEval 2017 – Shared Task on Aspect-based Sentiment in Social Media Customer Feedback,* 1–12. Berlin, Germany, September. (Cited on page 19).

Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. 2023. A Survey on Aspect-Based Sentiment Analysis: Tasks, Methods, and Challenges. *IEEE Transactions on Knowledge and Data Engineering* 35, no. 11 (November): 11019–11038. (Cited on pages 1, 25).

Andrea Zielinski, Calvin Spolwind, Henning Kroll, and Anna Grimm. 2023. A Dataset for Explainable Sentiment Analysis in the German Automotive Industry. In *Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis,* 138–148. Toronto, Canada, July. (Cited on page 3).