Research Master Thesis

A Comparative Study of Open-Source and Closed-Source Large Language Models for Native Language Identification

Yee Man Ng

a thesis submitted in partial fulfilment of the requirements for the degree of

ReMA Humanities
(Human Language Technology)

Vrije Universiteit Amsterdam

Computational Linguistics and Text Mining Lab
Department of Language and Communication
Faculty of Humanities

Supervised by: dr. Ilia Markov
2^{nd} reader: dr. Lucia Donatelli

Submitted: August 13, 2024

Abstract

With the rapid development of large language models (LLMs), it is crucial to explore the strengths and weaknesses of implementing LLMs for natural language tasks, which we aim to do for the task of Native Language Identification (NLI). This thesis investigates the difference in performance between open-source and closed-source LLMs on NLI, a text classification task in which a system automatically predicts authors' native language (L1) based on texts written in their second language (L2). The task is based on the assumption that second language speakers transfer certain properties from their L1 to the production of their L2, from which we can derive linguistic patterns that are indicative of one's L1.

Previous research has shown that closed-source LLMs like GPT-4 obtain state-of-the-art performance on the NLI task, outperforming previous efforts that relied on heavily feature-engineered supervised classification models. While closed-source LLMs achieve impressive results on the NLI task, they are accompanied by many risks to research in Natural Language Processing (NLP), such as the high costs associated with the use of closed-source LLMs and the limited access through APIs. With these risks, it is important to consider open-source LLMs for which we often have better insights into training procedures and openly publish their model weights, which allows for fine-tuning for downstream tasks.

Taking into account the advantages of open-source LLMs, this study aims to compare the performance of smaller open-source LLMs on the NLI task, when used out-of-the-box and after fine-tuning, to closed-source LLMs. We further explore the advantages that are unique to implementing generative models for NLI, by 1) leveraging open-source LLMs to provide natural language explanations for L1 classifications for explainability and 2) examining their ability to classify without a pre-defined set of L1s, i.e., open-set classification.

The results indicate that smaller open-source LLMs out-of-the-box perform considerably worse than closed-source LLMs, not only achieving significantly lower accuracy scores on the NLI benchmarks but also generating less coherent explanations of L1-indicative features. After fine-tuning, however, open-source LLMs can achieve state-of-the-art performance, with our fine-tuned Gemma model setting a new performance record of 96.6% accuracy on the ICLE-NLI benchmark, outperforming previous state-of-the-art approaches and GPT-4. Our study demonstrates the promising application of fine-tuning smaller open-source LLMs for text classifications like NLI.

Declaration of Authorship

I, Yee Man Ng, declare that this thesis, titled *A Comparative Study of Open-Source and Closed-Source Large Language Models for Native Language Identification* and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a Research Master degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Date: August 13, 2024

Signed: 

Acknowledgments

First and foremost, I would like to express my sincere gratitude to my supervisor dr. Ilia Markov. This thesis would not have been possible without your expertise on the topic (which is also evidenced by the dozens of Markov et al. in my works cited), your valuable feedback and guidance, and your support during this period in general. I greatly appreciate the chats we had about research, life, and academia, and I will carry that with me for the rest of my academic journey.

I would like to thank the CLTL staff for teaching me everything I know about Natural Language Processing and computational linguistics. Your passion for education and research is greatly inspiring to me. I also want to thank prof. Dr. Piek Vossen for his support with experiments for my thesis.

I would like to thank my classmates (better known as the HLT girlies and the thesis task force) for your wholehearted support during this time, and for being on campus with me every single day. I am so grateful for all of our frequent mental health outings, as these activities literally carried me and my thesis.

Specifically, thank you to Marina Munuera Esteller, Celonie Rozema, and Sidi Wang; I learned the absolute most during this Research Masters from working together with you. Thank you to Claire van Stolk, who I could always count on to send me silly memes during dire times. Thank you to my favorite historians, David Grantsaan and Josie Lauferts, for joining me on campus often and offering the much-needed support. Our side quest to break some silos in Leuven during thesis time, together with Celonie, was an unforgettable experience.

Finally, I thank my family and other friends for supporting me in my academic journey. I am especially grateful to Ilja van Oort for helping me stay sane during this process, being my main hype person and always offering your great insights into machine learning. Thank you to Tijmen van Gerwen for taking the time to proofread my work.

List of Figures

Contents

Chapter 1

Introduction

Native Language Identification (NLI) is the task of automatically identifying the native language (L1) of authors based on texts written in their second language (L2). The task is based on the language transfer hypothesis that states that L2 learners subconsciously transfer certain properties from their L1 into their L2 production, which allows us to distinguish between groups of speakers of particular L1s. This chapter introduces the task of NLI and its relation to the language transfer hypothesis. The chapter then outlines the applications of the NLI task, the recent implementation of large language models for this task, and the main contributions of this work.

1.1 Language transfer effect

One's native language and other previously acquired languages directly and indirectly influence the acquisition of a target language, which is also known as the process of *language transfer* or *cross-linguistic influence* (Odlin, 1989). In other words, an L2 learner's linguistic background predisposes them to display certain patterns in their L2 that are influenced by their native language. For example, native speakers of Mandarin tend to make more mistakes with determiners, such as 'the' or 'an', when writing in English than other learners, as there is no direct equivalent to English determiners in Mandarin (Malmasi, 2022). This, among other phenomena, would differentiate the use of English by native Mandarin speakers from, for example, native speakers of Spanish, which does have determiners like 'el' and 'la' and are used similarly to English. In Italian, there are only seven vowel sounds for the five different vowels whereas English has 15-20 possible vowel sounds (depending on the variety of English). Hence, it is very common for native speakers of Italian to confuse the use of vowels in English, which is manifested in particular spelling error patterns (Chen et al., 2017). These examples demonstrate how second language learners' L1 interferes with language production in their L2 and how this interference gives rise to different transfer patterns in relation to the L1.

1.2 Native language identification

Efforts have been made to computationally model these language transfer patterns and automatically predict learners' native language based on their writing in the second language, which is also known as the task of Native Language Identification (NLI).

From a machine learning perspective, NLI is commonly framed as a supervised multi-class classification task where an author's L1 is assigned from a predefined set of classes. On the basis of features extracted from L2 learner texts, a model is trained to predict the most likely native language of the speaker of each text. Figure 1.1 provides a schematic representation of the task, where an NLI model takes a set of English texts written by English as a second language speakers as input and then assigns the most likely native language of the author of the text, such as Spanish, German, or Chinese.

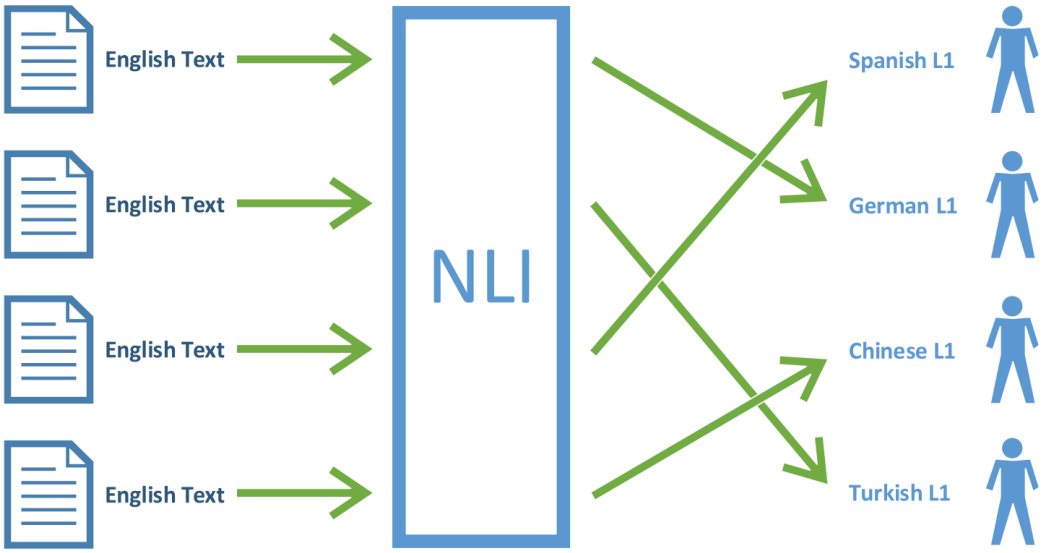


Figure 1.1: Simple representation of the Native Language Identification task as a multi-class classification problem, in which a model predicts the native language of authors based on texts written in their second language. Source: Malmasi (2022, p. 21).

The task of NLI is useful for various applications. In an educational context, NLI can help provide insights into language learners' patterns of language transfer. NLI systems can, for instance, be used to create writing tutor systems that can detect the likely native language of an author in order to provide more tailored and detailed feedback that connects certain errors to common properties of the learner's L1 (Tetreault et al., 2013). Additionally, NLI can aid in developing grammatical error detection and correction systems, such as including specific language profiles of users which can improve the performance of these systems (Rozovskaya and Roth, 2011; Malmasi, 2022).

Not only is NLI useful for educational purposes, but also for forensic analyses. NLI is regarded as a sub-task of author profiling, that attempts to find and describe the relation between stylistic features in texts and characteristics of the author, such as age, birthplace, or native language (Estival et al., 2007). NLI can be used in the context of tracing the likely native language of a suspect, or used to trace linguistic influence in multi-author texts (Malmasi et al., 2017; Malmasi, 2022). In this way, NLI models can be implemented as a linguistic profiling method and a tool for author profiling.

Furthermore, NLI can aid research in the field of linguistics. NLI-based analyses can enhance our understanding of language transfer patterns, allowing for further development of research in Second Language Acquisition (SLA). As Malmasi (2022) argues, while most work in SLA takes a more deductive corpus-based approach to test certain

language transfer theories, NLI takes a more inductive approach to SLA. In NLI, we derive certain patterns of L1 transfer from the available data. On the basis of patterns from the data, we can form certain hypotheses about likely causes of the differences between groups of speakers from a specific L1 background. Subsequently, NLI-based analyses can support research in cognitive linguistics to better inform researchers on how multilingual speakers process language.

1.3 Large language models for NLI

Previous research in NLI has obtained state-of-the-art results on the task using conventional machine learning approaches based on extensive feature engineering. Approaches using deep learning models, on the other hand, like long short-term memory networks (LSTM), convolutional neural networks (CNN), and bidirectional encoder representations from transformers (BERT), appear to yield poorer performance than conventional machine learning approaches for the NLI task (Markov et al., 2022; Steinbakken and Gambäck, 2020).

Recently, large language models (LLMs) have gained a lot of attention in the field of Natural Language Processing (NLP), achieving state-of-the-art (SOTA) results on a wide range of tasks. GPT-4, for example, at the time of its release achieved SOTA results on numerous LLM benchmarks for commonsense reasoning, language understanding, and reading comprehension, beating previous SOTA even with benchmark-specific training (OpenAI, 2023). When prompted in a zero-shot setting, the use of these models also eliminates the need of fine-tuning and training data for a specific NLP task. In this way, generative LLMs challenge the traditional approach that does require fitting a model to labeled data (Bucher and Martini, 2024).

Likewise, the use of LLMs has also been explored for the NLI task. Zhang and Salle (2023) performed experiments using the TOEFL11 dataset, the *de facto* benchmark dataset for NLI, with GPT-3.5 (Brown et al., 2020) and GPT-4 (OpenAI, 2023) in a zero-shot setting. Their results indicate that GPT-4 achieves a state-of-the-art accuracy score of 91.7% on the TOEFL11 test set, demonstrating remarkable out-of-the-box performance on this task.

In addition to reporting the remarkably high performance of LLMs on the NLI task, Zhang and Salle (2023) outline two advantages of implementing LLMs for NLI over traditional supervised approaches. First, LLMs used in a zero-shot setting can be implemented for NLI without specifying the set of known L1 classes, i.e., open-set classification. While traditional supervised models are limited in the sense that they can only predict a native language based on the predefined set of classes that have been seen in training, LLMs when used out-of-the-box do not have to be limited to the known set of classes. As there is no a priori knowledge of what an author's L1 would be in a real-world setting, leveraging the ability of generative LLMs to predict any possible L1 removes the restriction of a predefined set of L1s, a persistent shortcoming of previous NLI studies using supervised models for real-world applications, such as for forensic linguistic purposes.

Second, Zhang and Salle (2023) showcase the capability to leverage LLMs to provide explanations for their classifications. In language learning contexts, the focus of NLI lies not only on the ability to predict L2 learners' L1 accurately, but also what underlying features are indicative of learners' L1, expanding the measurement of classification accuracy with the analysis of language transfer features and patterns. For

real-world applications, such as for educational purposes, it is important to examine how second language learners' L1 influences their second language learning and production to improve the proficiency of their L2. For this reason, the authors leverage the LLMs' capability to provide reasoning for their prediction based on linguistic features in the text, such as spelling errors, word choice, and syntactic patterns. A manual examination of these explanations reveals seemingly plausible claims, suggesting that LLMs can be used as a tool for linguistic analysis of learner texts.

While Zhang and Salle (2023)'s results indicate that LLMs achieve state-of-the-art results on NLI, their approach only explores the implementation of GPT-3.5 and GPT-4. Given their closed-source nature, these models are accompanied by a multitude of limitations with respect to research and society (a more detailed explanation is provided in Section 2.5.1). Companies releasing closed-source models often disclose little to no information regarding the training data or procedure, hindering the evaluation of results achieved with these models and obscuring the biases in the training data and model. Moreover, the undisclosed nature of the training data has raised concerns among researchers about data contamination risks, as it is challenging to determine whether a model's high performance on tasks can indeed be attributed to the model's effective generalization or potential data leakage (Yu et al., 2023). In addition, closed-source models are typically only accessible via an API, causing lack of control over model updates and model versioning which are often not well-communicated to users (Yu et al., 2023). In turn, the reproducibility of experiments using closed-source LLMs cannot be guaranteed. The usage of closed-source LLMs is also highly costly, which negatively impacts the accessibility of LLMs and inhibits the growth of research in NLP. Thus, despite closed-source LLMs' high performance on many tasks, including NLI, the undisclosed nature of the training data, the API-only access, and the high costs of closed-source LLMs greatly inhibit the development of research in the field of NLP.

Open-source LLMs, on the other hand, often release more information regarding training data and procedures, allowing for a better understanding of possible biases in the model. In addition, the open release strategy somewhat reduces the impact of training LLMs on the environment, as the open release strategy of model weights means that other organizations do not have to make the same pretraining costs (Touvron et al., 2023). As model weights are released openly, open-source LLMs provide the additional advantage of the possibility of fine-tuning on a down-stream task, which is often highly costly or not supported for closed-source models. In this study, we aim to compare open-source LLMs out-of-the-box and after fine-tuning on the NLI task with closed-source LLMs to observe whether fine-tuned open-source LLMs can match the performance that closed-source models achieve on the task when used out-of-the-box.

1.4 Main contributions

With the negative impact of closed-source models like GPT-4 and the advantages of open-source LLMs, it is crucial to gain a better understanding of the gap in performance and explanation capabilities between open-source and closed-source LLMs. As previous research using LLMs for Native Language Identification shows very promising results using closed-source LLMs, the question is whether open-source LLMs can achieve similar results as closed-source LLMs for the NLI task.

Previous research comparing closed-source against open-source LLMs on classifi-

cation tasks have noted a large drop in performance between closed-source and open-source models when used out-of-the-box (Yu et al., 2023; Zhang et al., 2024b). With the recent development of techniques for efficiently fine-tuning open-source LLMs, such as Quantized Low-Rank Adaptation (QLoRA) (Dettmers et al., 2023), we are interested in fine-tuning open-source LLMs on the NLI task. Specifically, we examine how much fine-tuned open-source LLMs improve on the NLI task in comparison to open-source LLMs when used out-of-the-box, and whether fine-tuned open-source LLMs can match the performance of closed-source LLMs.

By implementing open-source LLMs for NLI, we aim to 1) investigate how smaller open-source LLMs perform on the task, and 2) further explore the advantages of using LLMs for NLI, i.e., gaining insights into linguistic features for NLI by leveraging the LLMs' ability to provide explanations for certain classifications, and performing open-set classification, as previously explored by Zhang and Salle (2023) using GPT-3.5 and GPT-4. Taking into account the advantages of using open-source LLMs over closed-source LLMs, we conduct a comparative study of open-source and closed-source LLMs for Native Language Identification, to better understand the difference in performance between these groups of models on the task. The code used for the experiments is openly available on GitHub[1].

The main research question of this thesis is as follows: **How do smaller open-source generative LLMs perform on Native Language Identification compared to closed-source LLMs?**

The research sub-questions guiding this work are as follows:

- Is the difference in performance between open-source and closed-source LLMs consistent across two benchmark NLI datasets, ICLE-NLI and TOEFL11, and across open-set and closed-set settings?
- How do closed-source LLMs out-of-the-box compare to open-source LLMs out-of-the-box and after fine-tuning in terms of performance on the NLI task?
- Can open-source LLMs provide targeted explanations for their L1 classifications?

1.5 Thesis structure

This thesis is structured as follows: Chapter 2 provides an overview of previous work on NLI with respect to the results of the NLI shared tasks, conventional machine learning methods and deep learning approaches for NLI. Chapter 3 outlines the experimental setup, the chosen NLI benchmark datasets, and open-source and closed-source large language models used in this study. Chapter 4 presents the results of our experiments in a comparative evaluation of open-source and closed-source models on NLI across different experimental settings. Chapter 5 contains a discussion of the results in light of the research questions, the limitations of our approach, and possibilities for future research. Chapter 6 summarizes our work and the overall impact of the findings.

[1]The relevant GitHub repository can be found here: https://github.com/yeem4n/thesis-NLI

Chapter 2

Background

2.1 Introduction

This literature review is dedicated to providing an overview of previous work on Native Language Identification (NLI), describing the NLI Shared Tasks and computational approaches that have been explored for NLI. This chapter first outlines traditional machine learning methods and features that have been proven to be useful for NLI, followed by an account of current approaches using large language models (LLMs) for the NLI task. We address the context of current developments in Natural Language Processing (NLP), i.e., the 'race of LLMs', in which LLMs are continuously being developed to match the performance of current state-of-the-art closed-source LLMs and emphasize the importance of investigating open-source LLMs in the context of NLI.

2.2 NLI Datasets and Shared Tasks

Before the organization of NLI shared tasks, most work on NLI had been performing experiments with the International Corpus of Learner English (ICLEv2), a dataset consisting of essays written by college-level English learners. However, as the dataset was not compiled specifically for the purpose of NLI, there were idiosyncrasies in the data, such as topic bias and the use of language-specific characters, that made the dataset less suitable for the NLI task (Brooke and Hirst, 2012; Tetreault et al., 2012). Variation across studies in the selection of L1s and the use of cross-validation also hindered the comparison of approaches to the NLI task Tetreault et al. (2013). To address these issues, Tetreault et al. (2012) sampled a subset of this corpus, referred to as ICLE-NLI, in which the topics and L1s are more balanced. ICLE-NLI covers 7 native languages: Bulgarian, Chinese, Czech, French, Japanese, Russian, and Spanish. Our experiments also make use of the ICLE-NLI dataset, which is explained in more detail in Section 3.1.

In 2013, Tetreault et al. (2013) organized the first shared task for English NLI, allowing researchers to compare approaches using a much larger corpus specifically designed for NLI, called TOEFL11 (Blanchard et al., 2013). The 2013 Native Language Identification Shared Task greatly boosted the popularity of NLI; with 29 participating teams working in a variety of fields, this shared task was one of the largest NLP competitions that year (Tetreault et al., 2013; Malmasi et al., 2017). The TOEFL11-13 dataset consists of essays written during a college entrance test and includes 11

different L1s: Arabic, Chinese, French, German, Hindi, Italian, Japanese, Korean, Spanish, Telugu, and Turkish. The dataset consisted of a training set with 900 essays per L1, a validation set of 100 essays per L1, and a test set of another 100 essays per L1, totaling in 12,100 essays. The best performance achieved in this shared task was 83.6% accuracy on the TOEFL11-2013 test set using a Support Vector Machine (SVM) classifier with lexical and POS *n*-grams. The TOEFL11-2013 dataset is also used in our experiments and is described in more detail in Section 3.1.

In 2017, Malmasi et al. (2017) organized another shared task for English NLI, which covered both speech-based and text-based NLI. The 2017 edition attracted 19 participating teams. For the essay-only NLI track, the training and validation data used in the 2013 Shared Task were combined to form the training data in the 2017 Shared Task, the test data from the previous task formed the validation dataset, and a newly released TOEFL11-2017 test set was used for evaluation. The TOEFL11-17 includes the same 11 different L1s as in the 2013 Shared Task and consisted of the training set with 11,000 essays (1,000 per L1), and a development set and test set with each 1,100 essays (100 per L1). The results indicated that ensembling approaches based on traditional classifiers, such as SVMs, with lexical and syntactic features were most effective and could not be outperformed by deep learning approaches (Malmasi et al., 2017).

While both the ICLE-NLI and TOEFL11 datasets consist of English student essays, there is also a multitude of NLI datasets in other domains [1]. Rabinovich et al. (2018) created the Reddit-L2 dataset, which consists of English posts and comments written by non-native speakers on Reddit, a social media platform. Brooke and Hirst (2013) compiled the Lang-8 learner corpus consisting of journal entries posted by English as a second language learners on the Lang-8 website. In addition, there have been numerous studies exploring the NLI task in L1s other than English, such as Arabic (Malmasi and Dras, 2014a), Chinese (Malmasi and Dras, 2014b), German (Malmasi and Dras, 2015b), Russian (Remnev, 2019), and Turkish (Uluslu, 2023). Nevertheless, the majority of previous research in NLI has focused on English learner corpora.

2.3 Traditional machine learning approaches to NLI

The NLI task has been commonly addressed using conventional machine learning techniques, which rely on explicit feature engineering. Different lexical and syntactic features have been explored for NLI. Popular lexical features include character, word, and lemma *n*-grams, while (morpho-)syntactic features are based on constituent parse trees, dependency parse features, and part-of-speech (POS) tags (Malmasi et al., 2017). While this work does not explore explicit feature engineering due to our focus on large language models, some of our baseline approaches are based on previous SOTA features, feature representations, and systems. The following sections provide an overview of traditional machine learning approaches. We first outline a variety of features that have been considered useful for the NLI task following different areas of linguistic analysis, followed by a summary of feature representations. We then provide an overview of classification systems that have been employed for the NLI task.

[1] See (Goswami et al., 2024) for a more detailed overview of available NLI datasets.

2.3.1 Features for the NLI task

Orthography The field of NLI has a long history of investigating spelling error features in L2 texts. This is based on the hypothesis that spelling errors are connected to differences in spelling conventions or pronunciation between the L1 and L2, and can thus be indicative of a L2 speaker's L1. Koppel et al. (2005) were one of the first to focus on spelling errors as features for NLI. The authors represent spelling errors according to various orthographic error types, such as repeated letters, double letters appearing only once, letter inversion, and frequency of types of spelling errors. A linear SVM trained on these spelling error types and other common NLI features like character n-grams, achieves an accuracy score of 80.2% on a subset of ICLEv1 covering 5 L1s (Bulgarian, Czech, French, Russian, Spanish). Similarly, Chen et al. (2017) examine the representation of spelling errors with character n-grams up to size 3. Their results indicate that misspelled parts of the word are strongly indicative of L2 speakers' L1. They found that using character n-grams extracted from spelling errors as features produced better results than directly using misspelled words. Given this previous research, it is clear that spelling features are highly useful for NLI and capture interesting language transfer patterns.

Previous research has also explored variation in punctuation usage amongst L2 speakers. Markov et al. (2018) explored the impact of punctuation on NLI, in a series of experiments comparing POS n-grams and word n-grams with and without punctuation marks. Their results indicate that an author's use of punctuation is a robust indicator of their L1, even despite their proficiency in the L2.

Lexical choices Word frequencies and word type frequencies are commonly-used features in NLI that capture lexical choices. The majority of the best-performing classification systems in the 2013 NLI Shared Task used a range of word n-grams (Tetreault et al., 2013). Jarvis et al. (2013), for example, trained the best-performing classifier in the 2013 NLI Shared Task using a combination of features including word 1-4-grams. Word n-grams provide insight into L2 learners' lexical choices and are generally useful for the NLI task. One major point of criticism against the use of lexical features is that these also inadvertently capture topical information (Brooke and Hirst, 2012). A classifier could learn to distinguish between L1 classes based on topical information, leading to topic bias.

Markov et al. (2019) analyze lexical choices with respect to variation in word spelling in L2 learner texts that might be indicative of a specific L1. They investigate the use of cognates, words that are derived from the same etymological ancestor, and L2-ed (i.e., anglicized) words, words deriving from the L2 learner's L1 that were adjusted to seem like valid L2 words. For example, in the case of English L2 learners, that would be by adding a typical English prefix or suffix to an existing word from their L1. Their results showed an increase in accuracy using these features, demonstrating that these features that capture spelling and lexical choice are highly useful for NLI.

(Morpho-)syntactic features Different types of syntactic features have also been explored for NLI and shown to boost the performance of NLI models. Wong and Dras (2011) investigated syntactic errors for NLI with the use of POS n-grams, Context-Free Grammar (CFG) features, and parse tree cross-sections. Their results indicated that including a binary representation of non-lexicalized parse rules boosted the performance of their NLI system, showing that syntactic features are useful for NLI. (Brooke and

Hirst, 2012) explored a range of different features, including syntactic features like the use of CFG features and POS n-grams. The authors found that adding CFG features generally boosts the performance of the system.

***N*-grams as features** When used in isolation, surface form features like character and word n-grams appear to be most informative for the NLI task (Malmasi et al., 2017). The results of multiple participants in the 2017 NLI Shared Task indicate that high-order character n-grams (up to $n = 10$) are very useful for NLI, likely due to these features not only capturing sub-word or morphological information but also dependencies between words (Malmasi et al., 2017). For example, Kulmizev et al. (2017) trained a linear SVM using character n-grams of lengths ranging from 1-9, which proved to be one of the best-performing models in the NLI shared task 2017, with an F1-score of 87.56%. The authors found that 7-9 range character n-grams, a relatively high number of characters, seem to capture the most information on a learner's native language. Similarly, Ionescu et al. (2014) found that a range of character n-grams captured a large number of linguistic features, such as stems of content words, prefixes and suffixes, and function words.

Feature representations The aforementioned features can be represented in different ways. These could be binary vectors or count vectors, but Term Frequency-Inverse Document Frequency (TF-IDF) is considered particularly useful in various NLP tasks, such as text classification and authorship identification. TF-IDF weighting considers the frequency of features in relation to the entire training corpus, which helps to identify features that might be highly discriminative (Goswami et al., 2024). For example, Kulmizev et al. (2017), who achieved their best results using an SVM with 1-9 character n-grams as mentioned previously, used binary feature representation normalized with TF-IDF weighting. Similarly to Kulmizev et al. (2017), we also use an SVM with a range of 1-9 character n-grams normalized using TF-IDF weighting as one of our baselines, combining the effectiveness of high-order character n-grams and TF-IDF weighting.

2.3.2 Classification models for NLI

Traditional machine learning algorithms have previously produced impressive results on NLI Malmasi et al. (2017). Specifically, SVMs have consistently produced the best results for NLI due to their high performance on text classification tasks in general, and their ability to handle large and sparse feature spaces (Jarvis et al., 2013; Kulmizev et al., 2017; Tetreault et al., 2013; Markov et al., 2017). The majority of the participants in the NLI Shared Tasks 2013 and 2017 used SVMs, including the top-ranked systems.

Besides traditional machine learning systems, ensemble approaches, where the strengths of multiple algorithms are combined (by, e.g., majority voting or metaclassifiers), also appear to be very effective for NLI. Multiple teams in the 2017 NLI Shared Task that achieved a high performance used a combination of different approaches. For example, Cimino and Dell'Orletta (2017) obtained the best result among all participants in the essay track using a classifier stacking approach. The authors used a 2-stack sentence-document classifier with a sentence-level classifier, of which its predictions are used by a second document-level classifier. Their results indicate that including those sentence-level features slightly improves the performance of the text classifier.

While deep learning models like Bidirectional Encoder Representations from Transformers (BERT) achieved state-of-the-art performance on numerous NLP tasks at the time of its release, deep learning models were often found to not outperform traditional machine learning models (e.g., feature-engineered SVMs) on the NLI task. For example, Steinbakken and Gambäck (2020) implemented the transformer model BERT on the TOEFL11 dataset and found that BERT is not able to compete with traditional state-of-the-art models. Markov et al. (2022), similarly, compared convolutional neural networks (CNN), long short-term memory networks (LSTM), and BERT to machine learning models like SVM, and found that deep learning models delivered lower results than a feature-engineered SVM.

There could be multiple reasons why deep learning models like LSTM, CNN, and BERT yield poorer performance than traditional machine learning approaches. According to Markov et al. (2022), this could be due to the nature of the data used in NLI, which contains very particular features pertaining to the L1 of authors that deep learning models struggle to capture. Deep learning models pre-trained on thousands of occurrences of words and character sequences from general corpora would have trouble capturing these very specific features that are important for NLI. Another potential reason for the poor performance of neural networks like CNN and LSTM, is the limited size of widely used NLI benchmarks.

2.4 Large Language Models

Currently, interest in utilizing Large Language Models (LLMs) for NLI has been growing, with the growing popularity of LLMs in the field of NLP. LLMs are pre-trained statistical language models trained on an enormous amount of data and have been shown to perform remarkably well on numerous language tasks. For more details on the architecture of LLMs, we refer to Section 3.3. In the following sections, we provide an overview of previous research using LLMs for NLI and other text classification tasks.

2.4.1 Large Language Models & NLI

Lotfi et al. (2020) first introduced the novel approach of using generative deep learning models for text classification and specifically on the task of NLI. They fine-tuned GPT-2 (Radford et al., 2019) models on training data grouped by each native language separately, with the intuition that each model learns distinctive characteristics of each native language. A label is assigned to an unseen text based on the model with the lowest language modeling (LM) loss. This method achieved an accuracy score of 89% on TOEFL11 and 94.2% on ICLE-NLI, outperforming the baseline and all previous state-of-the-art results on these datasets. This indicates that this approach using generative models, which does not require extensive feature engineering, can achieve promising results on the NLI task.

As mentioned in Chapter 1, Zhang and Salle (2023) conducted one of the first experiments using GPT models for NLI in a zero-shot setting, achieving a new performance record of 91.7% on the TOEFL11 test set. The authors experiment with GPT-3.5 (ChatGPT) (Brown et al., 2020) and GPT-4 (OpenAI, 2023) in a closed-set and open-set setting on the TOEFL11 benchmark. Their results indicated that GPT-4 performed significantly better than GPT-3.5 in both settings, with GPT-4 achieving 91.7% in a closed-set setting and 86.7% in an open-set setting, and GPT-3.5 74.0% and

73.4%, respectively. Both experienced a slight drop-off in performance from closed-set to open-set setting, with GPT-3.5 accuracy decreasing by 0.6% and GPT-4 by -4%. Based on a manual analysis by Zhang and Salle (2023), GPT-4 also seemed to be capable of providing reasonable justifications for their NLI predictions, demonstrating the utility of LLMs as tools for linguistic analysis. All in all, their experiments show the strengths of employing GPT models for NLI, in the potential to perform NLI experiments in an open-class setting, and for improved explainability of linguistic features, with the ability to prompt generative LLMs to explain their classification choices. As Zhang and Salle (2023) performed NLI using only closed-source LLMs, they express the possibility of exploring open-source LLMs on the NLI task for future research. The following sections outline the advantages and disadvantages of using closed-source models, and how open-source LLMs serve as a better alternative with respect to research and environment while providing an innovative approach to NLI.

2.5 Closed-source vs. open-source LLMs

Recently, the success of LLMs like ChatGPT has ignited a race of LLMs, in which research labs continue to train increasingly larger language models to surpass the performance of the previous state-of-the-art LLMs. Regrettably, some research labs choose to cease public disclosure of training data and methodology (Sun et al., 2023). Other research labs have responded by releasing open-source LLMs that attempt to match or surpass the results of closed-source LLMs, such as the LLaMA family by Meta (Touvron et al., 2023), BLOOM (BigScience, 2023), and many others. While there has been much discussion in the research community about how closed-source models perform consistently better than open-source ones, Balloccu et al. (2024) comment that this is sometimes merely driven by the hype surrounding these popular models.

2.5.1 Limitations of closed-source LLMs

While commercially available closed-source LLMs achieve high performance on many NLP tasks, they are accompanied by immense limitations for research and society. First, the lack of access to the model details, particularly the training data, has raised great concerns about data contamination among researchers (Balloccu et al., 2024). As the training data of closed-source LLMs is undisclosed, it is challenging to determine whether a model's high performance on a benchmark dataset can be attributed to effective generalization or potential data leakage (Yu et al., 2023).

Not only can data contamination originate from the model's training data in the pretraining stage, but also from user interactions with the model. Researchers commonly assume that using benchmarks available only to authorized parties guarantees that this data has not been leaked. However, as Balloccu et al. (2024) comment, this ignores the fact that models using reinforcement learning from human feedback (RLHF) can learn from user interactions. If user interactions with the models include benchmark data, the models would be contaminated even if the initial training data was free of such data. Balloccu et al. (2024) define this issue as 'indirect data leaking', where new data can leak to the model by feeding benchmarks that are not publicly available into closed-source LLMs, e.g., through the web interface of ChatGPT. They performed an analysis of 255 papers experimenting with GPT-3.5 and GPT-4, and they report that 4.7M samples coming from 263 different datasets have been exposed to the OpenAI

models in such a way that they could be used for training. Their results also indicate worrying trends concerning research practices regarding unfair comparisons of models (e.g., a lack of baselines, the practice of sampling test data for one model but not the other), and the low level of reproducibility. Their study shows the concerning direction in which the research area of LLMs is heading.

In addition to issues surrounding data leakage, closed-source models are typically accessed through application programming interfaces (API), which restricts how these LLMs can be used (Balloccu et al., 2024). APIs allow a user to use software without gaining access to the internal system details; in that way, the provider maintains control over the software. The limited access through APIs causes a lack of control over model versioning or model updates (Yu et al., 2023). The use of different model versions can significantly change the results of NLP experiments. (Chen et al., 2024) found that on various typical LLM tasks, like question-answering, solving mathematical problems, and code generation, the performance of GPT-3.5 and GPT-4 varies greatly over time. The variation in model performance in turn negatively impacts the reproducibility of experiments using closed-source models. While disclosing the specific model version when performing experiments using closed-source LLMs helps, in reality, updates in closed-source models are often poorly communicated or not communicated at all to users (Pozzobon et al., 2023). This makes it difficult to compare new results to previously reported results with an older version of the API. The dependence on access through API hinders fair comparisons of different techniques over time, possibly leading to biased conclusions (Pozzobon et al., 2023).

Moreover, the environmental and financial costs associated with training and running closed-source LLMs raise concerns within the field of NLP about the use of these models. The computing and financial resources required to run these models stifles creativity, as researchers might not have access to large-scale compute to execute their ideas (Strubell et al., 2019). This affects resource-poor research groups in particular, negatively impacting the accessibility of NLP research. The high financial costs associated with accessing closed-source models also negatively impact the reproducibility of experiments, as reproducing experiments using closed-source LLMs is costly, and by extension the progress within the research community (Pozzobon et al., 2023). The significant energy consumption needed to train and run these ever-larger LLMs also has a large environmental impact (Bender et al., 2021).

2.5.2 Advantages of open-source LLMs

Considering the multitude of downsides of closed-source models, the use of open-source LLMs promotes a more sustainable way of conducting research, with its relative transparency in model details and reduced environmental impact compared to closed-source models. According to Touvron et al. (2023), the open-release strategy means that other organizations will not need to make the same pretraining costs. By openly releasing the model weights, others in the NLP research community can make use of the model and fine-tune it for specific use cases. In this way, the open release of LLM weights potentially prevents the depletion of more global resources. Moreover, open-source LLMs provide the opportunity to fine-tune LLMs for specific downstream tasks, which is often either not supported or a highly costly process for closed-source commercial LLMs.

2.5.3 Comparative evaluations of closed- and open-source LLMs

Previous research has reported large drops in performance on text classification tasks between smaller open-source and closed-source LLMs when used out-of-the-box. Yu et al. (2023), for example, compare open-source, closed-source, and small language models on various text classification tasks like misinformation detection and political party prediction. In addition, they test the effects of different prompts and zero-shot vs. few-shot vs. fine-tuning setups. Their findings showed that large closed-source LLMs generally perform better on classification tasks than smaller open-source LLMs out-of-the-box, with a drop in accuracy of around 20% on different zero-shot classification tasks like implicit political ideology prediction between GPT-4 and the smaller LLaMA-2 (13B). This indicates that open-source LLMs out-of-the-box, especially when they are much smaller in size, do not yet match the performance of closed-source LLMs on text classification tasks.

While research suggests that, on text classification tasks, open-source LLMs when used out-of-the-box do not perform as well as closed-source LLMs, current research has presented some conflicting evidence when comparing closed-source generative LLMs to smaller LLMs after fine-tuning. Some studies have noted a gap in performance on classification tasks between closed-source LLMs and fine-tuned smaller LLMs, suggesting that closed-source prompt-based LLMs like ChatGPT have caught up with fine-tuned models. Zhang et al. (2024b) report a drop in accuracy of 16% on sentiment classification tasks and 24% on more complex sentiment analysis tasks such as irony detection between fine-tuned open-source smaller language model Flan-T5 (770M parameters) and ChatGPT out-of-the-box. Similarly, Qiu and Jin (2024) found that ChatGPT exhibits comparable performance to a fine-tuned BERT model on sentence-level classification tasks.

Other studies present evidence that suggests the opposite. Yu et al. (2023), for example, demonstrate that open-source LLMs like LLaMA-2 after fine-tuning can still outperform closed-source LLMs like GPT-3.5 on various text classification tasks, such as political party detection. Edwards and Camacho-Collados (2024) compare the performance of LLMs like GPT-3.5 in zero- and few-shot settings with that of fine-tuned smaller language models like Flan-T5 on classification tasks like topic analysis and sentiment analysis. Their results indicated that fine-tuned smaller language models outperform zero- and few-shot approaches of LLMs.

As there is no previous research, to our knowledge, on fine-tuning LLMs for the NLI task, a significant gap remains concerning fine-tuning LLMs for a text classification task like NLI. For this reason, we employ efficient fine-tuning techniques like Quantized Low-Rank Adaptation (QLoRA) in our experiments to observe to what extent fine-tuning open-source LLMs can boost performance. In addition, there is generally a lack of comparative studies of LLMs on classification tasks, as comparative evaluations often focus on other text comprehension tasks like natural language understanding, reasoning, or question-answering (Yu et al., 2023; Bucher and Martini, 2024). This further emphasizes the importance of analyzing the gap between open-source and closed-source LLMs on text classification tasks like Native Language Identification.

This background forms the basis of our approach to NLI. Traditional feature-engineered machine learning models have demonstrated impressive performance on the NLI task, as described previously. We draw upon previous approaches using traditional machine learning methods by implementing these as a baseline approach, e.g., a range of 1-9 character *n*-grams with an SVM model, similarly to Kulmizev et al. (2017). We

also implement BERT to better situate our findings and confirm previous findings on older deep learning models being less suitable for NLI than traditional machine learning approaches (Steinbakken and Gambäck, 2020; Markov et al., 2022). Current research demonstrates the impressive performance of closed-source LLMs like GPT-4 on the NLI task, leaving open-source LLMs unexplored for this task. Considering the negative impact of closed-source LLMs on the research community, it is worthwhile to investigate the performance of open-source LLMs and explore the possibility of fine-tuning LLMs for the NLI task. For this reason, we present a comparative study of closed-source and open-source LLMs for Native Language Identification.

Chapter 3

Methodology

This thesis aims to implement open-source generative large language models (LLMs) for Native Language Identification (NLI), the task of automatically predicting an author's native language (L1) based on texts written in their second language (L2), to examine the difference in performance between open-source and closed-source LLMs on this task. While closed-source LLMs have shown remarkable results for the task of NLI, the use of open-source LLMs like LLaMA (Touvron et al., 2023; Meta, 2024) and Gemma (Mesnard et al., 2024) has not yet been explored, despite the advantages of using open-source LLMs in NLP over closed-source models as discussed in Section 2.5.2. For this purpose, we conducted a series of experiments in which we compare the performance of various open-source LLMs with that of several baseline approaches and with the state-of-the-art performance of closed-source LLMs on two NLI benchmarks. Moreover, we evaluated the performance of open-source and closed-source LLMs in a closed-set setting with a predefined set of L1s in the prompt, and an open-set setting, without the set of L1s. The methodology used in this work closely follows that of Zhang and Salle (2023) to allow for a direct comparison with their results using GPT-3.5 and GPT-4 on TOEFL11. The following sections describe the selected datasets, models, and experimental setup for the conducted experiments.

3.1 Datasets

The experiments were carried out using two benchmark NLI datasets, namely TOEFL11 (Blanchard et al., 2013; Tetreault et al., 2012) and ICLE-NLI, a subset of ICLEv2 (Granger et al., 2009). These two datasets are commonly used in NLI, and both consist of English learner essays written by non-native English speakers. The sections below describe the main characteristics of each dataset.

3.1.1 TOEFL11

The TOEFL11 corpus (Blanchard et al., 2013) consists of essays written by English learners during a TOEFL (Test of English as a Foreign Language) iBT test that measures academic English proficiency. In the TOEFL11 test, the authors were asked to write an essay in response to a writing topic. TOEFL11 was created using the responses on this task. The following 11 native languages are represented in TOEFL11: Arabic (ARA), Chinese (CHI), French (FRE), German (GER), Hindi (HIN), Italian (ITA), Japanese (JPN), Korean (KOR), Spanish (SPA), Telugu (TEL), and Turkish (TUR).

Corpus	Languages	Topics	Avg. tokens /essay	No. essays per L1
TOEFL11	ARA, CHI, FRE, GER, HIN, ITA, JPN, KOR, SPA, TEL, TUR	8	348	Train: 1,000 Test: 100
ICLE-NLI	BUL, CHI, CZE, FRE, JPN, RUS, SPA	76	747	110

Table 3.1: The statistics of the datasets used.

The corpus covers low, medium, and high proficiency levels. In terms of distribution, it contains 1,100 texts per L1 that have been relatively evenly distributed per writing topic, totaling in 12,100 essays. On average, there are 343 tokens per essay in the test set. An overview of the dataset statistics is provided in Table 3.1. Due to its even distribution across native languages and topics, the corpus serves as a high-quality benchmark for NLI. TOEFL11 was also used for NLI Shared Task 2013 (Tetreault et al., 2013).

For our experiments, we concatenated the training set and development set used in the 2013 Shared Task (Tetreault et al., 2013) for training for all supervised approaches (i.e., Support Vector Machine (SVM), Bidirectional Encoder Representations from Transformers (BERT), and fine-tuned LLMs). The resulting training set contains 1,000 essays per L1, totaling 11,000 essays. We evaluated all systems on the test set from the 2013 Shared Task (Tetreault et al., 2013), to allow for comparison with previous research on NLI using LLMs, such as Lotfi et al. (2020) and Zhang and Salle (2023). The test set contains 100 essays per L1, totaling 1,100 essays.

3.1.2 ICLE-NLI

The International Corpus of Learner English (ICLEv2) (Granger et al., 2009) is another commonly used dataset for the NLI task, that consists of essays written by university undergraduates. Unlike TOEFL11 which covers different proficiency levels, the ICLE dataset only covers L2 learners with a high level of English proficiency. As mentioned in Chapter 2.2, because the corpus was not initially intended for computational modeling, there were some idiosyncrasies in the data concerning topic bias and encoding errors. Tetreault et al. (2012) attempted to resolve these by sampling a subset of which the topics and native languages were more balanced, and removing the instances with encoding errors. This resulted in a subset called ICLE-NLI. This subset consists of 110 essays, with an average of 747 words per essay, for each of the following 7 native languages: Bulgarian (BUL), Chinese (CHI), Czech (CZE), French (FRE), Japanese (JPN), Russian (RUS), and Spanish (SPA). For our experiments, we evaluated on the full ICLE-NLI subset using 5-fold cross-validation.

3.2 Baseline approaches

To better situate our results using LLMs, we implemented several baseline approaches. The following sections describe these baseline approaches.

SVM with BoW We implemented a simple linear Support Vector Machine (SVM) model with Bag-of-Words (BoW) features using Term Frequency (TF) representation. In other words, these features simply represent the counts of words in texts. We used the CountVectorizer and Linear Support Vector Machine (SVM) from the scikit-learn library[1]. Lotfi et al. (2020) also used this baseline approach in their experiments, which achieved 71.1% accuracy on TOEFL11 and 80.6% on ICLE-NLI.

1-9 TF-IDF SVM We also implemented a simple linear SVM with 1-9 character n-grams with Term Frequency-Inverse Document Frequency (TF-IDF) representation. As described in Section 2.3.1, an SVM with high-order character n-grams has demonstrated impressive performance in previous research, such as Kulmizev et al. (2017) who trained one of the best-performing systems in the 2017 NLI Shared Task using an SVM with 1-9 character n-grams and TF-IDF representation, which achieved 87.56% on the 2017 version of the TOEFL11 test set. For this reason, we selected this relatively simple approach as one of our baselines. We implemented LinearSVM and TfidfVectorizer using the scikit-learn library.

BERT In addition to our SVM baselines, we implemented fine-tuned BERT models (Devlin et al., 2019) for the NLI task. As outlined in Section 2.3.2, previous research has not achieved as much success on the NLI task with BERT as traditional machine learning approaches. Nonetheless, it would be interesting to compare our results using LLMs to a fine-tuned BERT to better situate our results. We fine-tuned BERT, a standard bert-base-uncased, for 12 epochs with a batch size of 12 using the Hugging Face transformers library[2] for the two NLI benchmarks, similarly to Lotfi et al. (2020). Their implementation of BERT achieved 80.8% on TOEFL11, and 76.8% on ICLE-NLI, which was lower than their approach using GPT-2.

3.3 Large language models

In the following paragraphs, we provide an overview of popular architectures and training methods used for LLMs. While there are different definitions for LLMs, we specifically focus on LLMs as pre-trained auto-regressive or generative language models that are relatively large in size[3]. We then describe the LLMs used in our experiments and available details regarding their training procedure and data.

3.3.1 Architecture

Generative LLMs are auto-regressive transformer-based language models, meaning they are generally trained to predict the next most probable token in the sequence (Wan et al., 2024). They have been trained on large amounts of data and contain billions of parameters, making them much larger than previous transformer models like BERT. Generative LLMs are considered general-purpose and versatile, i.e. can be applied to a variety of use cases, and they demonstrate better language understanding and generation abilities (Bucher and Martini, 2024; Minaee et al., 2024). While previous

[1] https://scikit-learn.org/stable/
[2] https://huggingface.co
[3] For more details on the training process of transformers in general, see (Wan et al., 2024).

models required further fine-tuning on a downstream task, generative LLMs can be prompted using textual input without additional fine-tuning on labeled data.

The training process of an LLM involves various major steps. Figure 3.1 provides an overview of common steps in the LLM training pipeline.

Data cleaning First, data cleaning techniques are applied, as they have been shown to have a big impact on model performance. This involves steps like removing false information from the data, standard text preprocessing, addressing biases in the data, and removing duplicate data, which all impact the quality of the training data (Minaee et al., 2024). Phi-3-mini (Microsoft, 2024) is an example of a language model which results indicate that heavily filtering the training data can significantly boost performance. The developers demonstrated that Phi-3-mini, a relatively small LLM with 3.8B parameters, can achieve performance on language understanding tasks rivaling that of GPT-3.5 and Mixtral 8x7B, despite the large difference in size. They argue that the innovation lies in the training data, which is composed of heavily filtered web data.

Tokenization Popular tokenizers for LLMs are usually based on sub-word tokenization as it can account better for words not seen in the training data. This includes tokenizers like BytePairEncoding (BPE), WordPieceEncoding (WPE), and SentencePieceEncoding (SPE) (Minaee et al., 2024). BPE is the tokenizer used for many LLMs, such as the GPT family and LLaMA family. BPE makes use of frequent patterns of subwords at the byte level. In this way, frequent words are kept in their original form, while uncommon ones are broken down into subwords. WPE is the tokenizer used for BERT and, similarly to BPE, builds the vocabulary based on frequent subword units. Both BPE and WPE assume that words are separated by white space, which is not true for some languages like Chinese and Japanese. Consequently, input sentences to BPE and WPE have to be pre-tokenized using language-dependent tokenizers. SPE tries to address this issue by taking white space as a normal symbol (Kudo and Richardson, 2018). SPE implements sub-word tokenization like BPE and extends it with direct training from raw sentences.

Positional encoding There are also several ways to encode positional information of tokens. Absolute Positional Embeddings (APE) were first used in the original transformer model (Vaswani et al., 2017). It encodes the absolute positional information in the input vectors but fails to account for relative distance between tokens. Shaw et al. (2018) therefore developed Relative Positional Encoding (RPE), which involves extending the self-attention mechanism to consider the links between elements in a sequence. Finally, many LLMs make use of Rotary Position Embeddings (RoPE), which use a rotation matrix to encode the absolute position of tokens as well as explicit relative position details through self-attention. LLaMA is an example of a model that implements RoPE.

Training and fine-tuning During pre-training, an LLM is trained on a large amount of typically unlabeled texts in a self-supervised manner. Here, auto-regressive language models train on the next token prediction task. After pre-training, the model goes through a process called supervised fine-tuning (SFT) to align their output with

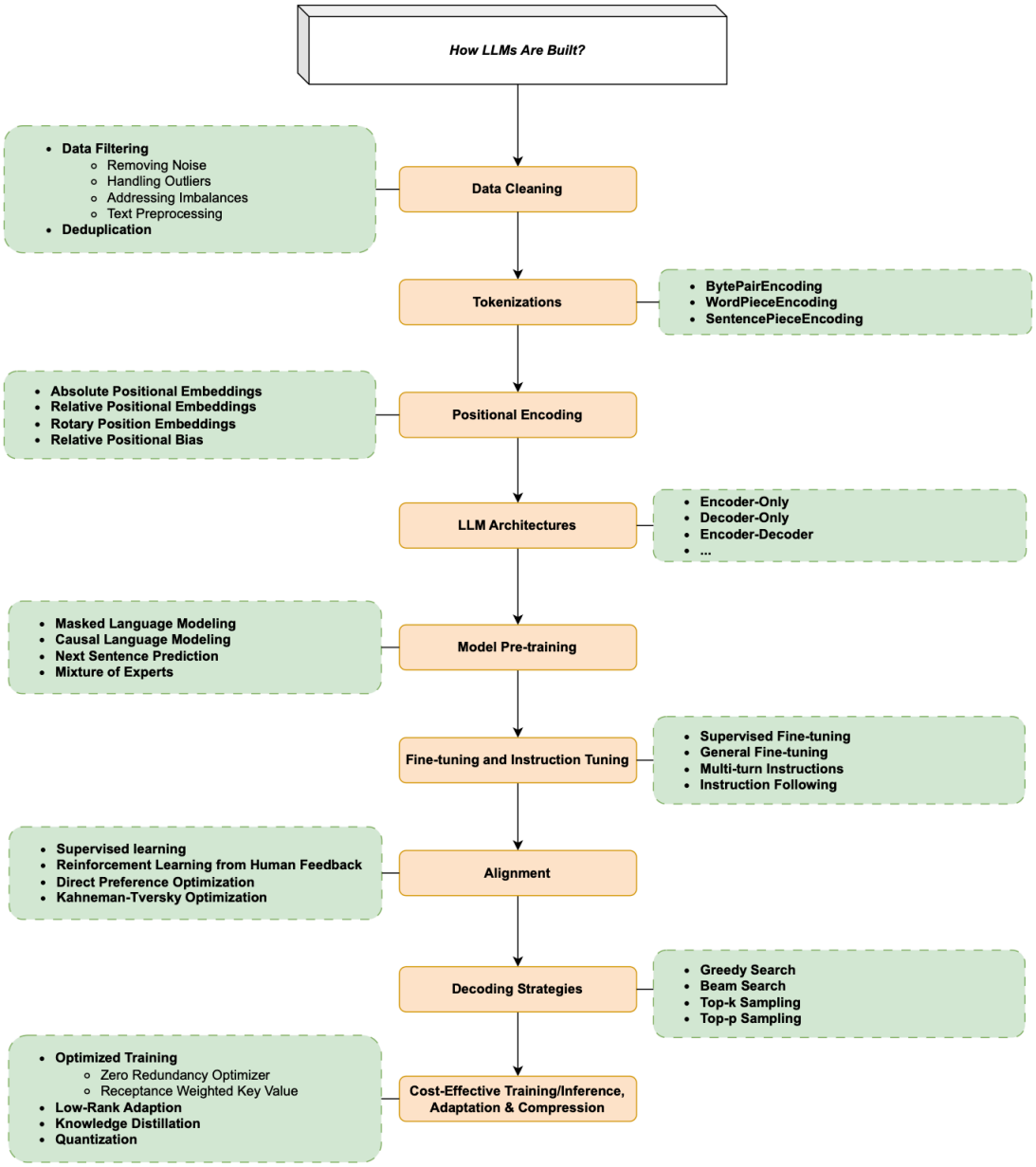


Figure 3.1: Overview of common steps in training and utilizing an LLM, such as tokenization, model pre-training, and instruction-tuning. Source: (Minaee et al., 2024)

humans' expectations. Instruction tuning is a form of fine-tuning, where LLMs are fine-tuned to generate output based on instructions that are provided in an input prompt. The instructions in these often not only include the specific task description and what the LLM should accomplish but also elements like examples of positive/negative output (Minaee et al., 2024). Instruction tuning serves to constrain the model output to align with desired model behavior, i.e., to follow humans' instructions and provide safer and more coherent answers (Zhang et al., 2024a).

While instruction-tuning greatly improves LLM behavior by steering it towards human preferences, LLMs often undergo more processes for better AI alignment, such as Reinforcement Learning from Human Feedback (RLHF) and Direct Preference Optimization (DPO). To learn alignment from human feedback, RLHF (Christiano et al., 2017) involves training a reward model that rates different model outputs and scores them according to human preferences. The LLM is further fine-tuned using the feedback from this reward model. Proximal Policy Optimization (PPO) (Schulman et al., 2017), a reinforcement learning algorithm, is often used in this process to optimize the reward signal (Xu et al., 2024). As RLHF involves separately training a reward model, followed by fine-tuning a large unsupervised model, the process can be quite complex and unstable. DPO (Rafailov et al., 2023) is another approach to alignment that addresses these issues of RLHF by identifying a mapping between optimal policies, i.e. model decision-making, and reward functions. In this way, DPO removes the reliance on separately training a reward model and simplifies the alignment process.

Cost-effective training and usage: quantization and LoRA Due to the size of LLMs, researchers have proposed different methods to run LLMs more cost-effectively. One popular method is quantization, which involves reducing the precision of the model weights. Dettmers et al. (2022) introduced 8-bit quantization, a method to perform 8-bit matrix multiplication by which we can convert large models with 16/32-bit representation to 8-bit, which significantly reduces the size of the parameters and speeds up the inference time. Similarly, Dettmers et al. (2023) introduced 4-bit quantization, which reduces the size of model weights even further while maintaining the same level of performance. In our experiments, we make use of 4-bit quantized open-source LLMs.

Models can also be efficiently fine-tuned for a specific task using training techniques like Low-Rank Adaptation (LoRA) and Quantized Low-Rank Adaptation (QLoRA). LoRA (Hu et al., 2021) is an efficient, lightweight fine-tuning approach that involves freezing the initial pre-trained weights, injecting low-rank matrices (adapters) into every layer over the model architecture, and fine-tuning. This reduces the number of trainable parameters, making training with LoRA significantly faster and memory-efficient. Similarly, QLoRA (Dettmers et al., 2023) involves fine-tuning low-rank matrices into pre-trained LLMs that have quantized to a 4-bit representation.

3.4 Open-source LLMs

For our experiments, we compare the performance of five state-of-the-art open-source generative decoder-only LLMs on the task of NLI. All of the models are hosted on Hugging Face[4].

The definition of what is truly 'open-source' is becoming increasingly more complex

[4] https://huggingface.co

Table 3.2: Overview of the open-source models used in our experiments and available details regarding the training dataset, number of parameters, training tokens, and type of architecture. The specific model sizes we used in our experiments are in bold.

Model	Release date	Cut-off date	Parameters	Tokens	Training data
LLaMA-2	18 Jul. 2023	Base: Sep. 2022 Instruct: Jul. 2023	**7B**, 13B, 70B	2.0T	A mix of publicly available online data
LLaMA-3	18 Apr. 2024	Mar. 2023	**8B**, 70B	15.0T	A mix of publicly available online data, of which 5% non-English data covering over 30 languages
Gemma	20 Feb. 2024	Unknown	2B, **7B**	6.0T	Primilarily English data containing web documents, mathematics, and code
Mistral	27 Sep. 2023	Unknown	7B	Unknown	Unknown
Phi-3-mini	23 Apr. 2024	Oct. 2023	3.8B	3.3T	Heavily filtered web data from various open sources and synthetic LLM-generated data

with the rise in proclaimed open-source generative LLMs. There is currently a rise in LLMs that claim to be open-source but are actually open in weights only. Under the current license-based definition of open-source AI, developers can refrain from disclosing other aspects of the training of the system, even though there are arguably various degrees of openness, ranging from the full release of training datasets to scientific and technical documentation to licensing and access methods (Liesenfeld and Dingemanse, 2024). LLaMA-2 and LLaMA-3, for example, two proclaimed open-source LLMs, do not release their training data openly, merely releasing minimal details regarding the training procedure and data.

For the purpose of our experiments, we considered open-source models that are open in weights to have a wider selection of models that have demonstrated impressive performance on natural language understanding and generation tasks. In the following sections, we provide an overview of each open-source LLM selected for our study. Table 3.2 contains a summary of available information regarding their training procedures and data.

LLaMA-2 LLaMA-2 is a family of open-source LLMs that was released in July 2023 by Touvron et al. (2023), ranging in size from 7B to 70B parameters and trained on 2 trillion tokens. The training data of LLaMA-2 consists of a mix of data from publicly available sources, of which data containing personal information has been explicitly filtered out. LLaMA-2 uses a SentencePiece tokenizer with BPE. Like other generative LLMs, LLaMA-2 is based on the standard transformer architecture (Vaswani et al., 2017), with pre-normalization using Root Mean Squared Normalization (RMSNorm), Swish Gated Linear Unit (SwiGLU) activation function, and RoPE embeddings. The post-training included supervised fine-tuning, iterative reward modeling, and RLHF to reduce the level of harmful language and steer the model into producing more desired output. At the time of its release, it outperformed other open-source LLMs like MPT and Falcon on most standard benchmarks for LLMs (Touvron et al., 2023). For our experiments on LLMs out-of-the-box, we used LLaMA-2 with 7B parameters, specifically

meta-llama/Llama-2-7b-chat-hf.

LLaMA-3 After the release of LLaMA-2, Meta (2024) released the next generation of Llama, called LLaMA-3 in April 2024. Similarly to LLaMA-2, LLaMA-3 is based on a standard decoder-only transformer (Vaswani et al., 2017) architecture. The key difference between LLaMA-2 and LLaMA-3 is that the latter uses a tokenizer with a vocabulary size of 128K tokens, which is much larger than that of LLaMA-2 (Meta, 2024). The number of tokens LLaMA-3 has been trained on is also seven times larger than that of LLaMA-2. LLaMA-3 also uses Grouped Query-Attention (GQA) instead of multi-head attention, used in the original transformer architecture. GQA greatly speeds up inference similarly to Multi-Query Attention, while achieving quality close to Multi-Head Attention (Ainslie et al., 2023).

The training data consists of publicly available online data, of which most are in English. Over 5% of the training dataset consists of non-English data covering over 30 languages to support multilingual tasks better. The data was then filtered using NSFW filters, heuristic filters, and text classifiers that predict data quality, trained on LLaMA-2-generated data. Post-training to align the model better with human preferences included a combination of supervised fine-tuning (SFT), rejection sampling, proximal policy optimization (PPO), and direct policy optimization (DPO) (Meta, 2024). For our experiments on LLMs out-of-the-box, we used LLaMA-3 with 8B parameters, specifically *meta-llama/Meta-Llama-3-8B-Instruct*.

Mistral Mistral 7B was released by Jiang et al. (2023) in September 2023. While the model is free to use, little to no details regarding the training data and the training procedure were released. In terms of attention mechanisms, Mistral 7B leverages grouped-query attention (GQA) and sliding window attention (SWA), which makes it able to process longer sequences at reduced computational cost and contributes to the enhanced performance and efficiency of the model (Jiang et al., 2023). At the time of its release, Mistral 7B outperforms the previous best LLaMA-2 13B model on several standard LLM benchmarks for common-sense reasoning, natural language understanding, and reading comprehension (Jiang et al., 2023). For our experiments on LLMs out-of-the-box, we implemented *mistralai/Mistral-7B-Instruct-v0.2*.

Gemma Gemma (Mesnard et al., 2024) is an open-source LLM released in February 2024, trained on 6 trillion tokens, and released in two sizes with 2B and 7B parameters. Gemma is a transformer-based model that makes use of Multi-Query Attention, and Generalized Gated Linear Unit (GeGLU) activation. It also includes RoPE embeddings and normalization with RMSNorm, similar to models from the LLaMA family.

The training data consists of primarily English web data, and data on topics such as code and mathematics, that have been filtered to remove undesired or harmful language, personal information, and low-quality data (Mesnard et al., 2024). Gemma was then fine-tuned for human alignment using two techniques: 1) SFT on a mix of English synthetic and human-generated prompts and responses, and 2) RLHF to align the model with English-only preference data. For our experiments on LLMs out-of-the-box, we implemented Gemma with 7B parameters, specifically *google/gemma-7b-it*.

Phi-3 Phi-3 is a novel open-source LLM, published by Microsoft (2024) in April 2024, that focuses on utilizing high-quality training data to improve the performance of small

language models. Phi-3-mini, the model used in our thesis experiments, has roughly 3.8B parameters and is trained on 3.3 trillion tokens, making it a relatively small LLM. The training data consists of heavily filtered, mostly English, web data, focusing on sources aimed at feeding the model language understanding, general knowledge, and logical reasoning (Microsoft, 2024). They then use two techniques post-training: 1) SFT to leverage highly curated data across various domains, and 2) direct preference optimization (DPO) to steer the model to be more aligned with a desired behavior. According to Microsoft (2024), the training method using high-quality data allows Phi-3 to achieve results that are similar to those of much larger closed-source LLMs like GPT-3.5 or Mixtral (8x7B) on standard LLM benchmarks for, e.g., common-sense reasoning and natural language understanding. In our experiments on LLMs out-of-the-box, we implemented *microsoft/Phi-3-mini-4k-instruct*.

3.5 Closed-source LLMs

This thesis aims to compare the results of open-source LLMs as described above to those of closed-source LLMs on NLI. Specifically, we examine the performance of GPT-3.5 and GPT-4 on NLI. We rely on the results by Zhang and Salle (2023) of GPT-3.5 and GPT-4 on TOEFL11 and implement their approach on ICLE-NLI. Below, we provide a brief overview of available information on GPT-3.5 and GPT-4.

GPT-3.5 GPT-3.5 is a series of language models with 175B parameters, released by OpenAI in November 2022, and an updated version of GPT-3 (Brown et al., 2020). At the time of its release, it received a lot of attention, especially with the release of ChatGPT (based on GPT-3.5). Not much is known about the training procedure or size, due to its closed-source nature. GPT-3.5 was trained using supervised fine-tuning and PPO to enhance instruction-following capabilities (OpenAI, 2022). In our experiments, we compared our results to those of Zhang and Salle (2023) who ran GPT-3.5, specifically *gpt-3.5-turbo*, on TOEFL11, and performed additional runs using GPT-3.5 on the ICLE-NLI dataset. We accessed the model through the OpenAI API.

GPT-4 GPT-4 is another closed-source model released by OpenAI in March 2023. GPT-4 is generally considered the state-of-the-art LLM on many NLP tasks, such as natural language understanding and common-sense reasoning, and even claimed to exhibit 'human-level performance' (OpenAI, 2023). As the use of GPT-4 has not yet been explored for the ICLE-NLI dataset, we ran GPT-4, specifically *gpt-4-0613* provided by the OpenAI API, on the entire ICLE-NLI dataset. We compared the results to those by Zhang and Salle (2023) of the same GPT-4 model on TOEFL11.

3.6 Experimental setup

We performed a series of open-set vs. closed-set, and out-of-the-box vs. fine-tuned experiments on five open-source LLMs to investigate the performance of open-source LLMs on Native Language Identification in various settings. All the models were evaluated on the TOEFL11 and ICLE-NLI datasets. We compared these to Zhang and Salle (2023)'s previous results of GPT-3.5 and GPT-4 (state-of-the-art closed-source LLMs) on the TOEFL11 test set, and our own results using GPT-4 on ICLE. In line

with prior work and because our datasets are balanced, we used accuracy score as our evaluation metric. Below, we discuss how the various experiments were conducted.

All experiments using LLMs were conducted in Google Colaboratory, using a 1xA100 GPU with 40GB RAM. The total computation time was roughly 120 hours. Total emissions are estimated to be 17.1 kgCO_2eq of which 100 percent were directly offset by the cloud provider[5].

3.6.1 Experiments using LLMs out-of-the-box

Open-source LLMs are often released in their base form and an instruction-tuned or chat form, which is better aligned with humans' expectations of model behavior, as described in Section 3.3.1. Instruction-tuned versions of LLMs are better at following instructions and do not necessarily require further fine-tuning on specific tasks (Minaee et al., 2024). When running inference on open-source LLMs in our experiments, we used the instruction-tuned versions, as these models are more adapted to follow the instructions in prompts, i.e., provide an L1 classification for a given text.

Moreover, instruction-tuned models are trained in a specific chat format or prompt template, so the model is able to complete chat sequences. The prompt template serves two purposes, which are indicating the roles in a conversation, as well as delineating the turns in a conversation (Mesnard et al., 2024). Figure 3.2 illustrates the structure of prompt templates for different models. Not using the same prompt template as the one used in training will likely make parts of the input out-of-distribution and, consequently, lead to less coherent output (Mesnard et al., 2024). Therefore, we used the *apply_chat_template()* function provided by Hugging Face to adapt each prompt to the prompt template accompanying each model tokenizer to align the prompt better with each model's expected structure of the input.

The prompt formatter typically includes roles like a system role, which allows the user to determine the behavior of the model, and the user role, which is mainly used for the instruction prompt. All our prompts are provided in Appendix A and include our inputs for these particular roles.

Not all open-source LLMs were instruction-tuned including a system role. For this reason, we adapted our prompts for some of the models of which its prompt formatter did not support a system role. For the LLaMA and GPT models, we entered the system and user prompts accordingly. For Gemma, Phi-3, and Mistral, we simply concatenated our system and user prompts and entered it as a user prompt.

All models when used out-of-the-box are loaded and 4-bit quantized using the BitsAndBytes library, supported by Hugging Face. We conducted three runs for each open-source LLM to account for stochasticity in model inference and training. The results of the closed-source models were all reported based on one run, taking into account the high financial costs.

When running the models out of the box, there are several hyperparameters that can be defined that could impact model behavior and output. Temperature, for example, is a key hyperparameter ranging between 0 and 1 that determines the level of randomness, or 'creativity' of the model output (Minaee et al., 2024). Setting the temperature to 0 leads to deterministic predictions, which can be useful for classification tasks (Zhang et al., 2024b). As preliminary experiments with the temperate set to 0 generally led to

[5]Estimations were conducted using the MachineLearning Impact calculator presented by Lacoste et al. (2019)

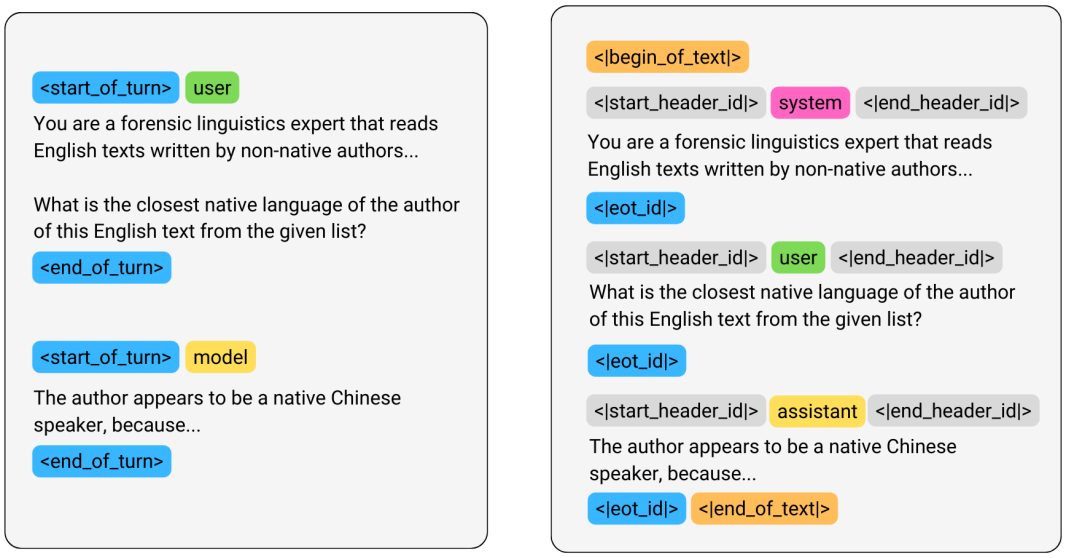


Figure 3.2: Examples of different prompt templates for Gemma (left) and LLaMA-3 (right). The special tokens used to delineate the different turns and roles in the sequence are highlighted to illustrate their purposes and differences per model. Orange: tokens that signify the beginning/end of the sequence. Blue: tokens signifying the beginning/end of the turns. Other colors: tokens corresponding to the different roles in the chat, e.g., user or system.

slightly worse performance compared to the model's default values, we did not set the hyperparameters to specific values when running LLMs out-of-the-box.

Experiment 1: Closed-set classification using LLMs out-of-the-box

In our first experiment, we ran inference in a zero-shot manner on the TOEFL11 test set and full ICLE-NLI dataset using open-source LLMs. We perform this as a closed-set task, in which the model's predictions are constrained to the predefined set of L1 classes of each dataset, mimicking traditional NLI classification approaches.

The prompts used for the closed-set experiments include a list of the possible 11 L1s for TOEFL11 and 7 L1s for ICLE-NLI (see Table 3.1). They are highly comparable to the ones used by Zhang and Salle (2023), which have yielded impressive performance for GPT-3.5 and GPT-4 on TOEFL11. After initial tests with running open-source LLMs using the exact same prompts as Zhang and Salle (2023) for different open-source LLMs, we found that these prompts caused a lot of variety in the LLM responses, particularly for LLaMA-2, which made it difficult to extract the predicted L1. In some cases, the generated response became very long as the LLMs would include a detailed analysis of the text, which negatively impacted the inference time. We attempted to resolve this by defining the maximum number of output tokens, but this sometimes led to truncating the predicted label from the output. For this reason, we instructed each model to only respond using JSON dictionaries, which generally helped to restrict the model output to one L1 classification. The exact prompts are provided in Appendix A.1. Using the data validation library Pydantic [6], we parsed the output into the predicted label.

[6] https://pydantic.dev

We experienced some difficulties parsing the model output to one prediction from the defined set of classes. In some cases, the model does not classify a predicted L1 out of the provided set of classes. For example, the models would sometimes predict English as the L1, despite the prompt specifically stating to not predict English, as it is an invalid response. Moreover, in some cases, a predicted L1 could not be extracted from the generated output, as the model would refuse to provide a prediction due to lack of information. We resolved these issues using iterative prompting, i.e., prompting the model again to provide an L1 prediction from the defined set of classes. We performed iterative prompting up to 5 times. If the model was unable to provide a prediction that corresponds with one of the possible classes after 5 attempts, the predicted label was set to 'other'. We ran the experiments using the five open-source LLMs as outlined in Section 3.4 and compared it to previous results by Zhang and Salle (2023) of the closed-source LLMs GPT-3.5 and GPT-4 on the TOEFL11 test set, baseline approaches, and our own results of GPT-4 on ICLE.

Experiment 2: Open-set classification using LLMs out-of-the-box

In our second experiment, we experimented with the use of LLMs for NLI in an open-set setting. The use of supervised models has been a persistent shortcoming of previous research in NLI, since supervised models are limited to predicting the L1s present in the training set (Zhang and Salle, 2023). In real-world applications, we have no a priori knowledge of the possible L1 of an author and should consider any L1 as a possibility. For this reason, our second experiment aims to assess the implementation of open-source LLMs in an open-set setting. Similarly to the first experiment, we ran inference on the TOEFL11 test set and full ICLE-NLI dataset in a zero-shot manner for the five open-source LLMs, but now without providing the list of possible classes in the prompt. This removes the output class restriction, allowing the LLM to predict any possible L1. Unlike in the closed-set experiment, we did not implement iterative prompting in the case that the model predicted English.

We implemented the prompt used by Zhang and Salle (2023) for the open-set experiments and adapted it by including the instruction to only respond using JSON dictionaries, similar to the closed-set experiment. This adaptation helped to restrict the variation in generated output by the different models, which was sometimes very long with irrelevant details about its observations. The exact prompts are provided in Appendix A.2. We again extracted the predicted label using the data validation library Pydantic based on the specified JSON output format. We then performed a post-processing step to parse predictions referring to the same language, e.g., 'FRA' and 'FRE' indicating French. We compared these results to previous results by Zhang and Salle (2023) of GPT-3.5 and GPT-4 in an open-set setting on TOEFL.

3.6.2 Experiment 3: Fine-tuning open-source LLMs

In addition to running inference on the open-source LLMs out-of-the-box, we fine-tuned each open-source LLM to improve the performance of the models on the task. For TOEFL11, we fine-tuned each model on the TOEFL11 training set, and evaluated it, similarly to the previous experiments, on the TOEFL11 test set. For ICLE, we fine-tuned each model under stratified 5-fold cross validation.

We used Unsloth[7], a library hosted on Hugging Face that makes LLM fine-tuning

[7] https://unsloth.ai

significantly faster and more efficient. We loaded pre-quantized versions of our five models provided by Unsloth, which are as follows:

- LLaMA-2: *unsloth/llama-2-7b-bnb-4bit*
- LLaMA-3: *unsloth/llama-3-8b-bnb-4bit*
- Gemma: *unsloth/gemma-7b-bnb-4bit*
- Mistral: *unsloth/mistral-7b-bnb-4bit*
- Phi-3: *unsloth/Phi-3-mini-4k-instruct*

In terms of hyperparameter settings, we used the AdamW optimizer, a learning rate of 1e-4, a batch size of 16, and 3 epochs for each model, similarly to Zhang et al. (2024b). We used an input prompt that is similar to the closed-set prompt. The exact prompt is provided in Appendix A.3). Similarly to our experiments on open-source LLMs out-of-the-box, we conducted three runs for each open-source LLM to account for stochasticity using a different random seed.

3.6.3 Experiment 4: Explainability

Gaining insights into model explainability is an important aspect of research in NLP in general, but it is particularly important for the task of NLI from the perspective of language learning and SLA research. Language learners benefit greatly from detailed explanations that give insights into particular errors they have made (Zhang et al., 2023). Explanations on how certain features, e.g., particular spelling mistakes or syntactic patterns, relate to a language learner's L1 could enhance awareness of these patterns and ultimately aid L2 learning. In the context of SLA research, the focus also lies more on insights into linguistic features that distinguish L1s rather than purely classification accuracy (Zhang and Salle, 2023). The ability of generative LLMs to provide explanations for their classification and provide insights on specific linguistic features makes them particularly useful for NLI.

Similarly to Zhang and Salle (2023) and Zhang et al. (2023), we attempted to gain insights into explainability by leveraging LLMs to generate explanations of classifications and provide cited examples in a zero-shot setting. A random sample of roughly 70 texts from the TOEFL11 test set and ICLE-NLI dataset was taken. We then prompted LLaMA-3, the open-source LLM that performed best on TOEFL11 and ICLE-NLI out-of-the-box, as well as GPT-4, in order to perform a comparative analysis between a closed-source and open-source LLM. The prompt is provided in Appendix A.4, which is the exact same as the one used by Zhang and Salle (2023). We then performed a qualitative analysis of the generated explanations to the best of our ability. We focused on the extent to which the explanations are targeted, i.e., they should highlight features in the text that are indicative of the L1 prediction rather than general features, and factually correct, i.e., presenting truthful information regarding the features, predicted L1, and language transfer patterns.

Ideally, the open-source LLM that performed best after fine-tuning would have been used for this experiment, as fine-tuning appeared to drastically boost the performance for all LLMs. However, after some initial experimenting, we found that the fine-tuned models failed to give any intelligible answer to the general reasoning prompt, only providing the classification label. It appears that the LLMs after fine-tuning lost the ability to answer general reasoning-related questions. Therefore, we selected LLaMA-3

out-of-the-box for this experiment, which demonstrated the best performance in closed-set and open-set experiments compared to other open-source models when used out-of-the-box.

3.7 Follow-up experiment

During testing, we observed that the performance of all open-source models when used out-of-the-box was extremely low compared to GPT-4's out-of-the-box performance on both NLI datasets. On the TOEFL11 test set, for example, we observed a drop in performance ranging between 34.9% to 78.1%. This raised the question: why does GPT-4 perform significantly better on the NLI task in comparison to open-source LLMs out-of-the-box?

We hypothesized that one possible reason could be data contamination. As we outlined in Section 2.5.1, data contamination is a considerable risk in using closed-source LLMs, and can be introduced in various stages of the training process. It would be plausible that OpenAI has at some point gained access to the TOEFL11 and ICLE-NLI benchmarks during pre-training or alignment with RLHF, despite both not being publicly available.

To verify whether GPT-4 has seen the data in training, we performed an additional experiment to test whether GPT-4 can achieve similarly high performance on data that it could not have seen in training. We selected a dataset with L2 learner texts that was released **after** the cut-off date of GPT-4 (specifically *gpt-4-0613*), September 2021, making it highly plausible that GPT-4 did not see this relatively novel dataset. Furthermore, this follow-up experiment on an additional dataset could verify our previous results that open-source LLMs achieve significantly poorer performance on the NLI task out-of-the-box and that fine-tuning LLMs greatly boosts the performance. The following sections describe the selected dataset and preprocessing steps, as well as the experimental setup of this follow-up experiment.

3.7.1 Dataset

For our follow-up experiment, we selected the Varieties of English for Specific Purposes dAtabase (VESPA) (Paquot et al., 2022). The VESPA corpus contains L2 learner texts written by university students from five European universities. The texts were annotated with the students' L1, of which the majority were written by speakers who have the same L1 as the official language of the country the university resides. The main languages represented are Dutch, French, Spanish, Norwegian, and Swedish. It comprises 941 texts, with an average of 1809 words per text. In comparison to TOEFL11 and ICLE-NLI, the texts are on average notably longer.

As this corpus was not made specifically for NLI, we took several preprocessing steps. First, because we only focus on speakers with one of the majority L1 classes, we took a subset of the corpus by selecting the texts written by authors who only had one of the five main languages as their L1. This entails removing the texts written by speakers with a multilingual background, and speakers with a different L1 background than the five main L1s. This subset consists of 697 texts and represents 5 L1 classes: Dutch (DUT), French (FRE), Spanish (SPA), Norwegian (NOR), and Swedish (SWE).

Second, the plain text files contain XML-style annotations of main sections, block quotes, and so-called "mentioned items" (Paquot et al., 2022, p. 8), which includes

Table 3.3: Distribution of the VESPA training set, with statistics regarding the number of texts, percentage of texts, and average number of words per essay for each L1 class.

Language	No. texts	Percentage	Avg. tokens /text
Dutch	57	8.81%	2618
French	130	20.1%	3286
Norwegian	378	58.4%	1403
Spanish	33	5.1%	1397
Swedish	49	7.57%	4858
Total	647		

citations, foreign words, and linguistic examples. We removed all XML tags and XML-tagged items, as well as in-text citations, using regular expressions, as these introduce more noise in the data when we are only interested in language produced by L2 learners.

We then randomly sampled a set of 10 texts per L1 to serve as a small test set, resulting in a test set of 50 texts in total, and a training set with the remaining 647 texts. The average length of the essays in the test set is 2672 words. We chose this size for the test set due to the limited number of samples for Spanish and to minimize the financial cost of running GPT-4 on this additional dataset. While the test set is balanced in terms of L1 classes, the training set is quite imbalanced, with over half of the training set consisting of texts written by Norwegian speakers. An overview of statistics for the training set can be found in Table 3.3.

3.7.2 Experimental setup

We followed the same experimental setup as some of our previous experiments, only substituting the dataset.

As a baseline, we implemented a LinearSVM with BoW using TF representation. We then conducted several classification experiments using LLMs. We selected three LLMs for this experiment: 1) Gemma, an open-source LLM that demonstrated SOTA performance on ICLE-NLI and near-SOTA performance on TOEFL11 after fine-tuning, 2) LLaMA-3, an open-source LLM that demonstrated best performance out-of-the-box compared to other open-source LLMs, and 3) GPT-4, the best-performing closed-source LLM. We first experimented with closed-set classification in a zero-shot setting using each LLM. We followed the same iterative prompting procedure as outlined in Section 3.6.1 to extract the predicted L1 class. As the set of L1s is different for this dataset, we slightly adapted the prompts. The exact prompts are provided in Appendix A.5.

Finally, we fine-tuned open-source LLMs, Gemma and LLaMA-3, on the training set to compare the performance of open-source LLMs out-of-the-box and after fine-tuning. We followed the same fine-tuning procedure as outlined in Section 3.6.2 using the Unsloth library. As the training set is unbalanced, we also fine-tuned open-source LLMs on a down-sampled version of the training set. We performed random down-sampling to 33 texts per L1 (the number of texts of the minority class in the VESPA dataset, Spanish), using the imbalanced-learn package (Lemaître et al., 2017).

For all experiments using LLMs, we truncated the input text from the right to a maximum of 8K tokens, to ensure that the input does not surpass the maximum context length of each model (8,192 tokens). We only evaluated the results based on one run,

as the standard deviation was generally relatively low in previous results.

Chapter 4

Results

In this thesis, we investigate the gap in performance between open-source and closed-source LLMs on Native Language Identification. This chapter contains the results of the experiments as described in Chapter 3. We first report and discuss the results of several baseline approaches in Section 4.1. These are as follows: a random guess baseline, SVM with Bag-of-Words using TF representation, SVM with character 1-9-grams using TF-IDF representation, and BERT. In Section 4.2 and 4.3, we report the performance of open- and closed-source LLMs in a zero-shot setting and out-of-the-box on the two NLI benchmarks, TOEFL11 and ICLE. We compare the difference in performance between a closed-set and open-set setting, i.e., with and without providing the set of possible L1s in the prompt. In Section 4.4, we compare the results of open-source LLMs that of closed-source LLMs, to observe whether smaller fine-tuned open-source LLMs can match the performance of closed-source LLMs out-of-the-box. Lastly, this chapter reports the use of open-source LLMs to generate explanations for particular classifications in comparison to those of closed-source LLMs. We attempt to closely analyze whether the linguistic features described in these explanations are sensible and correct, to the best of our ability.

All results are presented in Table 4.1. For the baseline approaches and closed-source LLMs, we report the accuracy score of one run. For the open-source LLM experiments, we report the average accuracy score and standard deviation across three runs to account for stochasticity when running and fine-tuning LLMs. Additional confusion matrices can be found in Appendix B.

4.1 Baseline approaches

We implemented several baseline approaches as outlined in Section 3.2 to better situate the results of the LLMs in the context of other machine learning and deep learning approaches, and directly compare the results of open-source LLMs to previous SOTA results, namely those achieved by Lotfi et al. (2020) who fine-tuned GPT-2 models representing each L1 for both TOEFL11 and ICLE-NLI and Zhang and Salle (2023), who evaluated GPT-3.5 and GPT-4 on the TOEFL11 benchmark.

The results indicate that the approaches using SVM generally outperform older transformer models like BERT. For both TOEFL11 and ICLE-NLI, a fine-tuned BERT model alone generally performs worse than SVM models in most cases. For TOEFL11, BERT achieves 75.3% accuracy, which is notably lower than SVM with 1-9-grams using TF-IDF representation that achieves 81.0% accuracy. On ICLE-NLI, SVM with 1-9-

Model	TOEFL11 (test set)		ICLE-NLI (5FCV/entire)	
	Closed-set	Open-set	Closed-set	Open-set
Baselines				
Random guess	9.1	–	14.3	–
BoW SVM	67.7	–	79.4	–
1-9 TF-IDF SVM	81.0	–	78.3	–
BERT	75.3	–	78.3	–
GPT-2 (Lotfi et al., 2020)	89.0	–	94.2	–
Closed-source LLMs				
GPT-3.5 (Zhang and Salle, 2023)	74.0	73.4	81.2	84.2
GPT-4 (Zhang and Salle, 2023)	**91.7**	**86.7**	95.5	**89.1**
Open-source LLMs				
LLaMA-2 (zero-shot)	29.2 ±0.9	22.1 ±0.7	29.2 ±1.0	15.5 ±0.3
LLaMA-2 (zero-shot, fine-tuned)	78.7 ±1.0	–	42.9 ±2.0	–
LLaMA-3 (zero-shot)	56.8 ±1.1	56.4 ±0.7	75.8 ±0.4	71.0 ±0.9
LLaMA-3 (zero-shot, fine-tuned)	85.3 ±0.1	–	78.5 ±2.5	–
Gemma (zero-shot)	13.6 ±0.0	7.0 ±0.0	28.2 ±0.1	13.1 ±0.0
Gemma (zero-shot, fine-tuned)	90.3 ±1.2	–	**96.6** ±0.2	–
Mistral (zero-shot)	35.6 ±1.6	24.2 ±0.1	53.1 ±1.1	41.5 ±0.1
Mistral (zero-shot, fine-tuned)	89.8 ±0.8	–	83.2 ±9.4	–
Phi-3 (zero-shot)	18.2 ±0.3	21.6 ±1.6	33.6 ±0.4	40.9 ±2.1
Phi-3 (zero-shot, fine-tuned)	65.6 ±0.4	–	51.4 ±1.7	–

Table 4.1: Results of open-source and closed-source LLMs on NLI, evaluated against conventional ML models. Evaluated on the TOEFL11 test set, as well as the full ICLE-NLI dataset for out-of-the-box models, and 5-fold CV for fine-tuned models. Reported in average accuracy (%) and standard deviation across 3 runs.

grams using TF-IDF representation and BERT both achieve 78.3% accuracy. A simple approach using SVM with BoW features, on the other hand, is able to achieve a higher accuracy score (79.4%) than a fine-tuned BERT. These observations are in line with previous results using BERT for NLI, such as those presented by Lotfi et al. (2020) and Steinbakken and Gambäck (2020), who found that BERT provides lower results than SOTA traditional machine learning approaches using SVM.

The 1-9-grams TF-IDF SVM approach provides good results for TOEFL11. On the TOEFL11 test set, SVM with 1-9 character n-grams using TF-IDF representation achieved an accuracy score of 81.0%. The high performance of a large range of character n-grams as features for NLI was also observed by Kulmizev et al. (2017). The authors achieved their highest results (F1 score of 87.56) with an SVM using only 1-9 character n-grams as features on the 2017 version of the TOEFL11 test set. A large range of character n-grams is capable of capturing morphological features, misspellings, and information about the writing style of an author, which are key features for the NLI task (Kulmizev et al., 2017).

While an SVM with 1-9 character n-grams and TF-IDF feature representation demonstrates a high performance on TOEFL11, an SVM with these features does not

achieve the same level of performance on the ICLE-NLI dataset. For the ICLE-NLI dataset, the SVM using TF-IDF 1-9-grams performs worse than the one with BoW count features, with the first achieving an accuracy score of 78.3% compared to the latter's accuracy of 79.4%. This relatively low accuracy score might be an indication of overfitting and supports the findings by Markov et al. (2022), who observed that when using character n-grams as features for NLI, the optimal length of character n-grams varies from corpus to corpus.

4.2 LLMs in closed-set setting

We first replicated Zhang and Salle (2023)'s approach using GPT-3.5 and GPT-4 on ICLE-NLI to solidify the high performance of closed-source LLMs on the NLI task. Our results using GPT-3.5 and GPT-4 on ICLE-NLI confirm the findings by Zhang and Salle (2023). GPT-4 out-of-the-box achieves a remarkably high accuracy score of 95.5% on ICLE-NLI in a closed-set setting.

The results as shown in Table 4.1 indicate that open-source LLMs out-of-the-box perform considerably worse than closed-source LLMs out-of-the-box. While GPT-4 achieves an accuracy of 91.7%, the five open-source models achieve an accuracy ranging between 13.64% and 56.76%. All open-source LLMs do perform better than the random guess baseline of 9.09%, but perform worse than all other baseline approaches, including the simple SVM model with BoW representation.

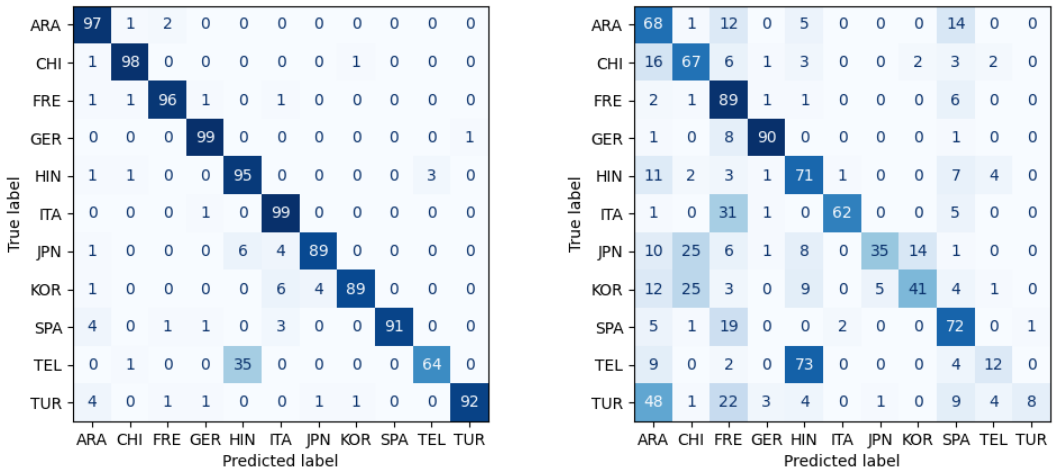
The following sections discuss the results of open-source and closed-source LLMs when used out-of-the-box in a closed-set setting in more detail. We examine the performance of these LLMs through the accuracy scores as well as confusion matrices.

4.2.1 TOEFL11 & ICLE-NLI

Out of the open-source models, LLaMA-3 is best out of the five in a closed-set setting when used out-of-the-box on the NLI task. In this setting, LLaMA-3 achieves an accuracy score of 56.76% on the TOEFL11 test set, which is still relatively low compared to our baseline approaches. Based on the confusion matrix as presented in Figure 4.1, French and German appear to be the easiest L1 to identify for LLaMA-3, while it often fails to correctly identify Turkish and Telugu as L1s. Similarly, GPT-4 also presents relatively many misclassifications when classifying texts with Hindi as the L1 when the actual L1 is Telugu. The high degree of confusion between Hindi and Telugu as L1s has been observed in previous research and the results of the 2013 and 2017 editions of the NLI shared task (Tetreault et al., 2013; Malmasi et al., 2017; Markov et al., 2022). LLaMA-3 also has a tendency to wrongly classify texts as French, particularly for texts written by speakers with L1s from the same language family, Italian and Spanish.

Similarly to the results on TOEFL11 in the closed-set setting, LLaMA-3 achieves the highest accuracy score compared to the five open-source models when used out-of-the-box on ICLE-NLI. While all models achieve accuracy scores ranging between 28.2% and 53.1%, LLaMA-3 achieves an accuracy score of 75.8%. When observing the confusion matrices of the best-performing closed-source and open-source LLM in Figure 4.3, we can observe that LLaMA-3 frequently confuses the L1s Bulgarian and Czech with Russian. LLaMA-3 also tends to wrongly classify Spanish as French but is able to accurately identify Chinese, French, and Russian as L1s.

Compared to results achieved by closed-source models, however, LLaMA-3's score



(a) GPT-4 (Zhang and Salle, 2023) (b) LLaMA-3

Figure 4.1: Confusion matrix of GPT-4 and LLaMA-3 when used out-of-the-box on the TOEFL11 test set and in a closed-set setting.

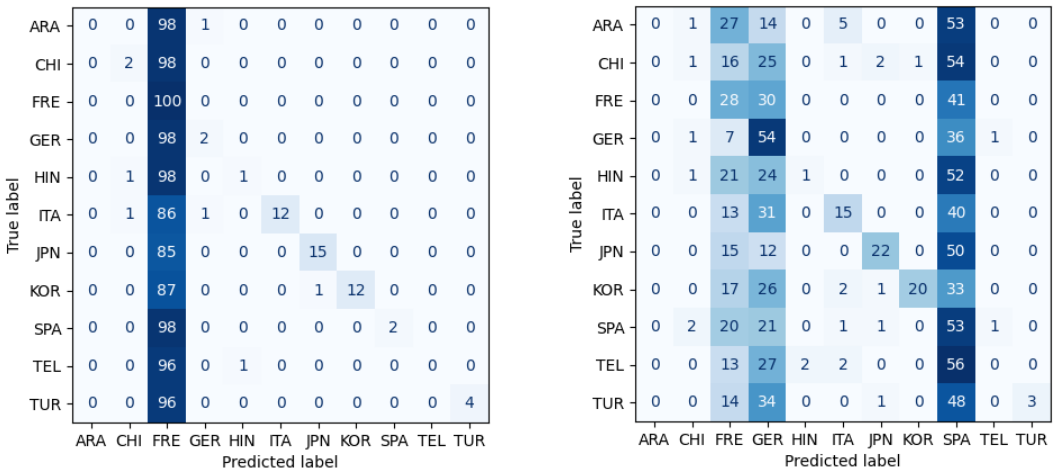
is significantly lower. GPT-4 performs remarkably better than all other models when used out-of-the-box, as it achieves a near-perfect accuracy score of 95.5%, beating even previous SOTA results achieved by Lotfi et al. (2020) using GPT-2 models. GPT-3.5 performs considerably worse than GPT-4, achieving an accuracy score of 81.2%. This is however still higher than all five tested open-source LLMs.

Gemma demonstrates the worst performance for the NLI task when used out-of-the-box, achieving an accuracy score of 13.6% which is close to the random guess baseline. When examining the confusion matrix (Figure 4.2), we observe that Gemma when used out-of-the-box predicts French as the L1 for almost every text. Interestingly, of all samples for which the model predicts the L1 to be Italian, Japanese, and Korean, it is correct between 93.8% and 100% of the time. These results indicate that Gemma out-of-the-box does not perform well on the NLI task, but might be more conservative in making any predictions outside of French.

Similarly, Phi-3 when used out-of-the-box demonstrates particularly low performance on the NLI task, obtaining only 18.2% accuracy on the TOEFL11 test set. Observing the confusion matrix (Figure 4.2), Phi-3 has a tendency to predict almost every author's L1 as Spanish, German, or French.

In terms of the worst-performing models on the ICLE-NLI dataset, both Gemma and LLaMA-2 demonstrate performance that is close to the random baseline, achieving an accuracy score of 28.2% and 29.2% on ICLE-NLI, respectively. When closely examining the confusion matrix (Figure 4.4), we observe that Gemma classifies nearly every L1 as French, only correctly predicting roughly 5-20 samples for all other L1s. LLaMA-2 similarly predicts the majority as having one L1, Chinese.

Taking the above into account, it appears that similar patterns of performance can be observed across the two datasets in a closed-set setting. For both TOEFL11 and ICLE, GPT-4 when used out-of-the-box performs extraordinarily well compared to baseline approaches, previous SOTA approaches, and all five open-source LLMs. The following section will report the results of LLMs when used in an open-set setting to compare the difference in performance in the two settings across different LLMs.



(a) Gemma (b) Phi-3

Figure 4.2: Confusion matrix of Gemma and Phi-3 when used out-of-the-box on TOEFL11 test set in a closed-set setting.

4.3 LLMs in open-set setting

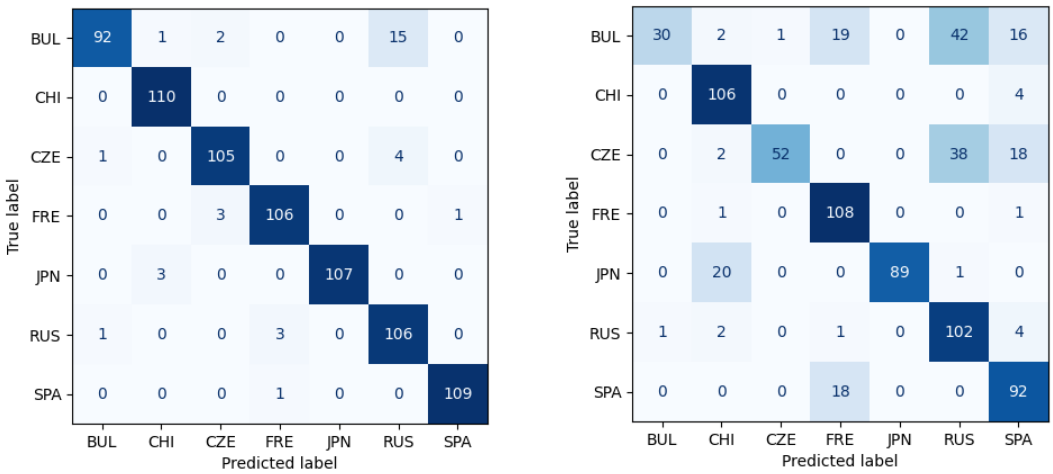
As LLMs can be used in a zero-shot setting, the L1 predictions do not have to be limited to a predefined set of L1s in a given dataset. In this experiment, we evaluate the performance of open-source LLMs in an open-set setting, analyzing the likely decrease in performance when removing this restriction. We first focus on comparing the performance of the different LLMs in an open-set setting compared to a closed-set setting in terms of accuracy score. To better understand the differences in performance, we then examine the out-of-set L1 predictions of various LLMs, directly comparing our results to Zhang and Salle (2023)'s previous results for GPT-3.5 and GPT-4 on TOEFL11.

For most open-source models, a drop in performance from closed-set to open-set setting can be observed. Surprisingly, some models perform better in an open-set setting compared to a closed-set setting. The following sections describe the performance of the LLMs on each of the benchmark datasets.

4.3.1 TOEFL11 & ICLE-NLI

When examining the results on TOEFL11 in terms of accuracy score (Table 4.1), a drop in accuracy between 5-11 percentage points (p.p.) from a closed-set to open-set setting can be observed for LLaMA-2, Gemma, and Mistral. A drop in accuracy can also be observed for most open-source LLMs when evaluated on the ICLE-NLI dataset. When removing the predefined set of L1s from the prompt, the accuracy decreases roughly with 12-15 p.p. for Gemma, Mistral, and LLaMA-2 compared to a closed-set setting. This drop in performance is expected: when removing the restriction of a predefined set of L1s in the prompt, the pool of possible L1s becomes significantly larger.

The level of performance decrease is similar to one that the closed-source LLMs GPT-3.5 and GPT-4 generally experience, as reported by Zhang and Salle (2023). GPT-4 demonstrates a drop in accuracy of 5 p.p. on TOEFL11 and 6 p.p. on ICLE-NLI. This is a greater relative drop than the one demonstrated by GPT-3.5 on TOEFL11, which is 0.6 p.p.



(a) GPT-4 (b) LLaMA-3

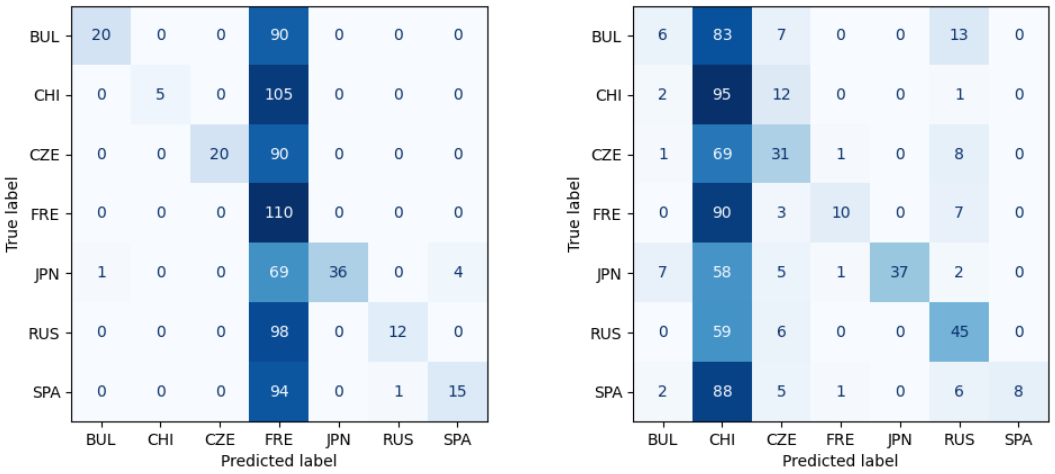
Figure 4.3: Confusion matrix of GPT-4 and LLaMA-3 out-of-the-box on the ICLE-NLI dataset in a closed-set setting.

In general, GPT-4 showcases the best performance in an open-set setting out of all other LLMs, achieving the highest accuracy score of 89.1% on the ICLE-NLI dataset and 86.7% on TOEFL11. GPT-4 again drastically outperforms all five open-source LLMs in an open-set setting. This indicates that GPT-4 is able to perform NLI without specifying the specific set of L1 classes.

Similarly to the results on the closed-set setting, LLaMA-3 demonstrates the best performance out of the five open-source LLMs in an open-set setting on the TOEFL11 test set. The performance of LLaMA-3 drops 0.4% compared to a closed-set setting. Out of the five open-source LLMs, LLaMA-3's drop in performance is the most marginal. On ICLE-NLI, LLaMA-3 also shows more promising results, as LLaMA-3 achieves an accuracy score of 71.0% in an open-set setting. This is significantly higher than the second-best performance of the open-source LLMs on ICLE-NLI, which is 41.5% accuracy achieved by Mistral. Again, the drop in performance of LLaMA-3 in an open-set setting is most marginal (4.8% on ICLE-NLI) relative to the other four models. These results indicate that similar patterns appear across the two benchmark datasets.

Surprisingly, not all models experience a decrease in performance going from a closed-set setting to an open-set setting. While one would expect that model performance drops in a set-up in which the pool of possible L1 classes is much larger, this is not the case for all models. Phi-3 demonstrates an improvement in performance compared to the closed-set setting, as the accuracy increases with 3.4% for the TOEFL11 dataset and 7.3% for the ICLE-NLI dataset. GPT-3.5's accuracy score also increases by 3% on the ICLE-NLI dataset.

To better understand the differences in L1 predictions between a closed-set and open-set setting for the different models, we must examine the out-of-set predictions, i.e., predictions that are outside of the training labels, for several LLMs. We specifically examine the out-of-set predictions for LLaMA-3, the open-source LLM that showcased the best performance in a closed-set and open-set setting compared to the other four LLMs, and compare these to the out-of-set predictions of GPT-3.5 and GPT-4 as



(a) Gemma (b) LLaMA-2

Figure 4.4: Confusion matrix of Gemma and LLaMA-2 when used out-of-the-box on the full ICLE-NLI dataset in a closed-set setting.

Table 4.2: Counts of out-of-set L1 classes (rows) predicted by LLaMA-3 (out-of-the-box) on the TOEFL11 test set, compared to the true L1s (columns).

LLaMA-3 Predicted L1	ARA	CHI	FRE	GER	HIN	ITA	JPN	KOR	SPA	TEL	TUR	Total
English	32	11	5	5	65	3	8	16	13	41	20	219
Portuguese	0	0	1	0	0	6	1	0	24	0	2	34
Russian	1	0	0	0	0	0	1	1	0	0	10	13
Indonesian	0	1	0	1	0	0	3	3	0	0	4	12
Persian (Farsi)	1	0	0	0	0	0	0	0	0	0	7	8
Indian	1	0	0	0	2	0	0	0	0	4	0	7
Urdu	0	0	0	0	1	0	0	0	0	1	1	3
Malay	0	2	0	0	0	0	0	0	0	0	0	2
Vietnamese	0	1	0	0	0	0	1	0	0	0	0	2
Thai	0	1	0	0	0	0	0	0	0	0	0	1

reported by Zhang and Salle (2023).

As shown in Table 4.2, LLaMA-3 incorrectly predicted English as the L1 for 219 samples, particularly for L2 texts with Hindi or Telugu as the L1 for the TOEFL11 dataset. Table 4.3 shows that LLaMA-3 also predicted the majority of out-of-set predictions as English for the ICLE-NLI dataset. The relatively large amount of English as the L1 predictions demonstrates LLaMA-3's failure to recognize non-native texts. The other out-of-set predictions are mostly related to the actual L1 through geographical location or due to being in the same language family. For example, LLaMA-3 misclassified 24 texts with the L1 Spanish as Portuguese. As Spanish and Portuguese are both Romance languages, there are many similarities between the two that likely caused this type of misclassification in an open-set setting.

When comparing LLaMA-3's out-of-set predictions to those of GPT-3.5 (Table 4.4), we observe a similar pattern in which GPT-3.5 wrongly classified relatively many texts, 126 texts in total, as English being the L1, which indicates a failure to identify non-native texts. The actual L1 for the majority of the texts classified as English being the L1 is Hindi and Telugu, similarly to LLaMA-3's out-of-set predictions. Moreover,

Table 4.3: Counts of out-of-set L1 classes (rows) predicted by LLaMA-3 on the ICLE-NLI dataset, compared to the true L1s (columns).

LLaMA-3 Predicted L1	BUL	CHI	CZE	FRE	JPN	RUS	SPA	Total
English	19	3	1	1	4	7	0	35
Polish	2	0	19	0	0	1	0	22
German	5	0	4	8	0	0	0	17
Italian	2	0	0	1	0	11	0	14
Portuguese	0	0	0	0	0	0	9	9
Dutch	1	0	1	3	0	0	0	5
Serbo-Croatian	2	0	2	0	0	0	0	4
Korean	0	0	0	0	4	0	0	4
Slovak	0	0	2	0	0	0	0	2
Arabic	1	0	0	0	0	0	0	1
Croatian	0	0	1	0	0	0	0	1
Greek	1	0	0	0	0	0	0	1
Romanian	1	0	0	0	0	0	0	1
Serbian	0	0	1	0	0	0	0	1
Turkish	1	0	0	0	0	0	0	1

Table 4.4: Counts of out-of-set L1 classes (rows) predicted by GPT-3.5 on the TOEFL11 test set, compared to the true L1s (columns). (Zhang and Salle, 2023)

GPT-3.5 Predicted L1	ARA	CHI	FRE	GER	HIN	ITA	JPN	KOR	SPA	TEL	TUR	Total
English	6	2	1	2	53	1	2	3	4	44	8	126
Tamil	0	0	0	0	0	0	0	0	0	12	1	13
Portuguese	0	0	0	0	1	0	0	0	3	0	1	5
Bengali	0	0	0	0	0	0	0	0	0	3	0	3
Persian	0	0	0	0	0	0	0	0	0	0	2	2
Dutch	0	0	1	0	0	0	0	0	0	0	0	1
Indeterminable	0	0	0	0	1	0	0	0	0	0	0	1
Malay	0	1	0	0	0	0	0	0	0	0	0	1
Vietnamese	0	0	0	0	0	0	0	1	0	0	0	1

Table 4.5: Counts of out-of-set L1 classes (rows) predicted by GPT-4 on the TOEFL11 test set, compared to the true L1s (columns). (Zhang and Salle, 2023)

GPT-4 Predicted L1	CHI	FRE	HIN	ITA	KOR	SPA	TEL	TUR	Total
Russian	0	1	0	0	1	0	0	5	7
Persian (Farsi)	0	0	0	0	0	0	0	4	4
Dutch	0	0	0	0	1	1	1	0	3
Indian language	0	0	0	0	0	0	2	0	2
Amharic	0	0	0	0	1	0	0	0	1
Bengali	0	0	1	0	0	0	0	0	1
Malay (Malaysian)	1	0	0	0	0	0	0	0	1
Portuguese	0	0	0	0	0	1	0	0	1
Romanian	0	0	0	1	0	0	0	0	1
Tamil	0	0	1	0	0	0	0	0	1

Table 4.6: Counts of out-of-set L1 classes (rows) predicted by GPT-4 on the ICLE-NLI dataset, compared to the true L1s (columns).

GPT-4 Predicted L1	BUL	CHI	CZE	FRE	JPN	Total
German	0	0	1	7	0	8
Slavic	2	0	5	0	0	7
Dutch	0	0	0	3	0	3
Korean	0	0	0	0	3	3
Cantonese	0	1	0	0	0	1
English	0	0	0	0	1	1
Italian	0	0	0	1	0	1
Romanian	1	0	0	0	0	1
Slovak	0	0	2	0	0	2

the other out-of-set L1 predictions are all related to the actual L1s either linguistically or geographically. These patterns in the out-of-set predictions indicate that LLaMA-3 makes similar misclassifications to GPT-3.5 when performing the NLI task in an open-set setting.

GPT-4, on the other hand, does not predict English as the L1 for any of the samples on the TOEFL11 dataset, and only once on the ICLE-NLI dataset. This suggests that GPT-4 is more sensitive to the prompt and As Zhang and Salle (2023) observe regarding the out-of-set predictions of GPT-4 on TOEFL11, while some of the out-of-set predicted L1s are related to the actual L1s either linguistically or geographically, others are not. On the ICLE-NLI dataset, most of the out-of-set L1 predictions do seem to be related to the ground truth labels, e.g., Chinese being identified as Cantonese and Japanese misclassified as Korean.

4.4 Fine-tuning LLMs for NLI

Compared to closed-source LLMs, open-source LLMs when used out-of-the-box on the NLI task appear to achieve significantly poorer performance in both closed-set and

open-set settings. For this reason, we examined to what extent the results of open-source LLMs could improve after fine-tuning it on the NLI task. We fine-tuned the same five open-source LLMs on the TOEFL11 training set and under 5-fold cross-validation on the ICLE-NLI dataset. The following section reports the results of fine-tuned open-source LLMs and compares these to the performance of closed-source LLMs used out-of-the-box (Table 4.1).

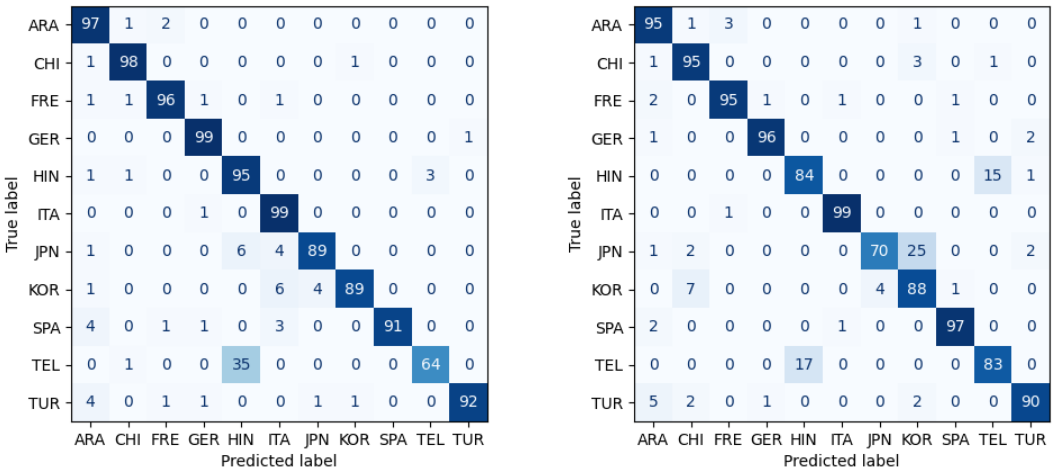
The results indicate that the performance of open-source LLMs improves drastically after fine-tuning for the NLI task. For both datasets, Gemma after fine-tuning achieves the best performance out of the five open-source LLMs. Fine-tuned Gemma achieves an accuracy score of 90.3% on the TOEFL11 dataset, nearly matching the results of GPT-4 as reported by Zhang and Salle (2023), and a near-perfect accuracy score of 96.6% on the ICLE-NLI dataset, outperforming GPT-4 by 1.1%. Mistral after fine-tuning achieves the second-best performance of the five open-source LLMs, with an accuracy score of 89.8% on the TOEFL11 test set, and 83.2% on the ICLE-NLI dataset. On TOEFL11, fine-tuned Mistral outperforms the previous SOTA achieved by Lotfi et al. (2020) using GPT-2. Phi-3, on the other hand, demonstrates relatively poor performance when used out-of-the-box, and also achieved the lowest accuracy score on both datasets after fine-tuning relative to the other fine-tuned models. Fine-tuned Phi-3 achieved an accuracy score of 65.6% on the TOEFL11 test set, and 51.4% on the ICLE-NLI dataset. All in all, these results show that open-source LLMs, particularly Gemma, can get (near-)SOTA results on the NLI task after fine-tuning.

When comparing the confusion matrices of the best-performing fine-tuned open-source LLM and the best-performing closed-source LLM, Gemma and GPT-4 (Fig. 4.5 and Fig. 4.6), we observe slightly different error patterns. On the TOEFL11 test set, as previously described, GPT-4 tends to misclassify texts with Hindi as the L1 when the actual L1 is Telugu. While fine-tuned Gemma also showcases some degree of confusion between Hindi and Telugu, it also has a tendency to misclassify samples with the L1 Japanese as Korean. On the ICLE-NLI dataset, GPT-4 has made some errors in classifying Bulgarian as Russian. Gemma only misclassifies roughly 6-8 samples with the L1s Czech and Russian as Bulgarian. The L1s that are confused by both models are either related through geographical location or language family.

Interestingly, the open-source LLMs that perform best out-of-the-box do not necessarily perform best after fine-tuning. In fact, the model that demonstrated the worst performance when used out-of-the-box, Gemma, shows the best performance out of the five open-source LLMs after fine-tuning. LLaMA-3, on the other hand, showed the best performance when used out-of-the-box, but the relative increase in performance is the smallest for LLaMA-3 out of the five models.

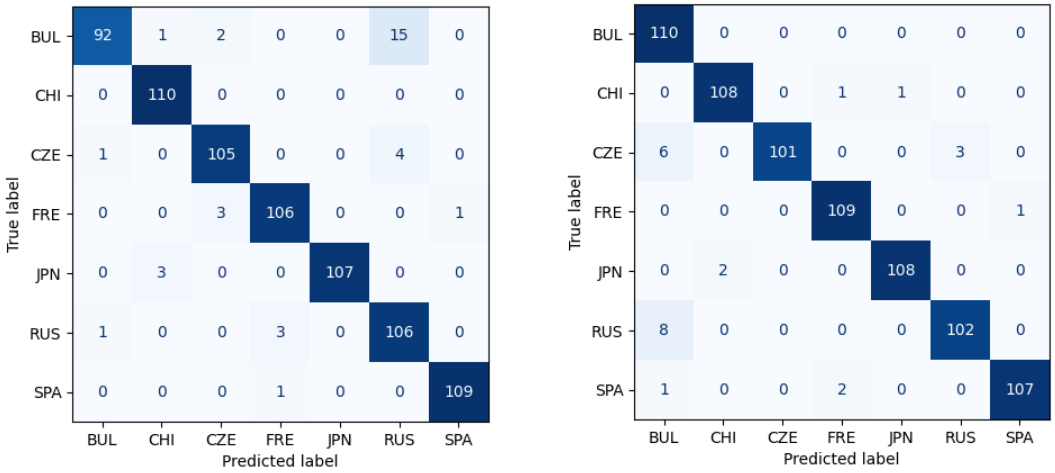
4.5 LLMs for explainability

In addition to measuring the performance of LLMs on the NLI task in terms of classification accuracy, we explored the usage of open-source LLMs for the explainability of linguistic features that distinguish L1s. As shown by Zhang and Salle (2023), sufficiently-large LLMs like GPT-4 can provide useful explanations that can aid linguistic analysis of learner language. Our experiment aims to examine whether an open-source LLM is able to provide viable explanations similarly to GPT-4. We selected LLaMA-3, as this open-source LLM demonstrated the best performance on NLI out-of-the-box based on our previous experiments. Following Zhang and Salle (2023)'s



(a) GPT-4 (Zhang and Salle, 2023) (b) Gemma (FT)

Figure 4.5: Confusion matrix of GPT-4 and fine-tuned Gemma in zero-shot setting on the TOEFL11 test set.



(a) GPT-4 (b) Gemma (FT)

Figure 4.6: Confusion matrix of GPT-4 and fine-tuned Gemma in zero-shot setting on the ICLE-NLI dataset.

approach, we leveraged LLaMA-3 in a zero-shot fashion on a small random sample from ICLE-NLI and TOEFL11. In the following sections, we perform a qualitative analysis, highlighting some common patterns in the explanations generated by LLaMA-3 in comparison to those generated by GPT-4.

4.5.1 Comparison between LLaMA-3 and GPT-4

We found that LLaMA-3's explanations were more general and ambiguous compared to GPT-4's explanations. LLaMA-3 often repeats the same phrases for different explanations. Below is one example of a part of an explanation generated by LLaMA-3:

> Grammar and sentence structure: The text features a mix of simple and complex sentences, but with a tendency towards shorter sentences and a lack of varied sentence structures. This is a common trait of non-native English writers who may not be as comfortable with sentence-level syntax and cohesion.
> Vocabulary: The text uses a limited range of vocabulary, with many words repeated throughout the essay. This is not unusual for non-native writers who may not have a strong command of English vocabulary.

While the explanation addresses some general features in the text that are indicative of L2 learner writing, the explanation is quite coarse-grained and provides little to no examples from the given text. The analysis includes observations like 'a mix of simple and complex sentences', a phrase that is repeated in multiple other generated explanations, and 'a limited range of vocabulary', which are features that are not specific to the predicted L1. In this way, the LLM fails to provide an explanation that gives insights into L1-indicative features. These results are in line with Zhang et al. (2023), who also prompted LLMs in a zero-shot setting for explanations, and found that LLMs like GPT-3.5 and LLaMA-2 used in a zero-shot setting produce explanations that lack informative and meaningful content.

Below is another example of a Llama-3-generated explanation, in which the model describes features that might be indicative of non-native writing, but not necessarily indicative of the specific L1 prediction.

> Spelling errors: "it it" instead of "it is", "easier" instead of "easier", "dificult" instead of "difficult", "rebeemeber" instead of "remember", "rememeber" instead of "remember", and "usderstand" instead of "understand". These errors are characteristic of non-native speakers who are still learning to navigate the nuances of English spelling.

While the explanations contains features indicative of non-native writing and mentions examples from the given text, LLaMA-3 fails to address why these errors are particular to their L1 prediction.

GPT-4, on the other hand, generally generates seemingly viable explanations that relate the observations of certain linguistic patterns to a likely source, e.g., direct translations from the L1 or spelling conventions in the L1. Below is an example of an explanation generated using GPT-4:

> Overuse of the word "have" is one common mistake made by Chinese speakers learning English as they often directly translate "有" into English as "have". In the sentences "There also have some countries have capital punishment" and "There are many for and against argument in this particular topic", the redundant use of "have" is evident.

In this explanation, GPT-4 highlights a specific linguistic feature, namely the overuse of the word 'have', and relates this to a possible direct translation from Chinese. It also quotes several examples from the text where the verb 'have' is used in an unconventional manner. This demonstrates that GPT-4 can provide targeted explanations with interesting L1-indicative features, including direct examples from the text. This is in line with Zhang and Salle (2023)'s findings, who observed that GPT-4

is able to provide sound reasoning for the NLI task. Note that the explanation contains a hallucination, as it includes a quotation from the text that does not contain the linguistic feature in question. Section 4.5.2 addresses more hallucinations found in LLM-generated explanations.

The following example further highlights the ability of GPT-4 to provide seemingly viable linguistic analyses.

> The main clue that led to this conclusion is the author's use of the preposition "to" in the phrase "addressing to him". In French, the verb for addressing someone ("s'adresser à") is reflexive and requires a preposition before the object, which appears to be the source of this mistake; English would typically use either "addressing him" or "speaking to him". The author's phrasing "condamn the acts", which mimics the French verb "condamner", is another clue, as been the correct English word would be "condemn". Lastly, the phrase "a kind of anthithesis" shows the author's tendency to use more literal translation from French (anti-thèse). These clues collectively point toward French being the author's native language.

GPT-4 again provides an explanation that is targeted to the specific L1 prediction and provides several examples of errors in the text. GPT-4 also outlines possible sources of errors, such as literal translations from French.

In this way, GPT-4 generally makes more targeted and contextualized linguistic analyses based on the text and the predicted L1 in comparison to LLaMA-3. These results further confirm the gap between closed-source and open-source LLMs when used out-of-the-box on the NLI task, and further confirm Zhang and Salle (2023)'s finding that large-enough LLMs can be used as tools for linguistic analysis.

4.5.2 Accuracy and hallucinations

Despite the generally positive results of GPT-4, a manual examination also revealed hallucinations, "generated content that is nonsensical or unfaithful to the provided source content" (Ji et al., 2023), made by both LLaMA-3 and GPT-4. The occurrence of hallucinations puts into question the accuracy of the generated explanations. These hallucinations include factual errors and misquotations.

The explanations generated by LLaMA-3 in particular contain many hallucinations. The following example highlights a hallucination in which the model presents factual errors.

> The text includes characteristic Japanese sentence endings, such as the use of the copula "desu" and the polite language particles "wa" and "ga". For instance, the sentence "I am against their opinion, but granted that Japanese today is somewhat corrupted" features the polite language particle "wa".

The explanation identifies linguistic features that are not present in the text, such as the text including 'characteristic Japanese sentence endings' when a manual examination of the given text shows that this is not the case. The explanation also features several quotes from the text as examples of certain errors, which do not actually contain the given error.

Similarly, the following example illustrates how LLaMA-3's explanations can contain factual errors.

> Word order: The author tends to use a subject-verb-object word order, which is typical in Russian but less common in English.

The factual statement that subject-verb-object (SVO) word order is less common in English is incorrect. As English is one of the many languages that follows SVO word order, the use of SVO word order in this L2 text cannot be an insightful L1-distinguishing feature.

Some explanations generated by GPT-4 also present several hallucinations. Below is an example of such an explanation:

> The text has a strong fluent nature indicating good proficiency in writing in English. There are no spelling or punctuation errors. However, certain word choices and sentence structure seem to be influenced by Spanish language. For example, using 'the' before proper nouns like 'the European Union' would be typical in Spanish but not in English. The extensive use of commas and long sentence construction often reflect a Spanish language style, as Spanish sentences are often longer and more complex compared to English ones. The use of 'both...together' such as 'both concepts, sovereignty and nation, forcefully should go together', seems to be a direct translation from Spanish structure. Also the construction 'I do n't believe Europe 1993 carries' conveys a Spanish-like syntax. It indicates the usage of the verb 'carry' in a similar way to the Spanish verb 'llevar' which can mean 'carry' as well as 'imply'. Therefore, even though the author has an advanced level of English writing, these elements suggest that the author is a native Spanish speaker.

A manual analysis of this sample carried out with a native speaker of Spanish[1], revealed several factual errors regarding Spanish linguistic features. First, the Spanish verb 'llevar' carries various meanings, but it does not carry the meaning 'imply'. In addition, the use of 'both...together' does not seem to be a direct translation from Spanish, nor is it a construction that is used particularly more in Spanish than in other languages. Moreover, the explanation also includes factual errors regarding linguistic features of English. The explanation addresses the use of the article 'the' before a proper noun as an influence of Spanish as their L1, implying that this feature is incorrect in English. However, the example of 'the European Union' is grammatically correct in English, and thus cannot be considered as an L1-distinguishing feature. This example showcases GPT-4's capacity to present factually incorrect information in a rhetorically sound manner.

4.6 Follow-up experiment

Lastly, as the performance of GPT-4 on the other benchmarks raised concerns about data contamination, we performed a follow-up experiment to verify whether GPT-4 has seen the NLI benchmarks in training. We selected a dataset that was compiled after

[1] Thank you to Marina Munuera Esteller for contributing to this analysis.

GPT-4's cut-off date, namely a subset of the VESPA dataset, and performed several experiments using open-source LLMs and closed-source LLMs. Table 4.7 contains the results of our experiment on the VESPA dataset.

4.6.1 VESPA

The results indicate that GPT-4 can achieve relatively high performance on VESPA. On the VESPA test set, GPT-4 achieves an accuracy score of 82%, the highest accuracy score out of all approaches using LLMs. Compared to its performance on the TOEFL11 and ICLE-NLI benchmark, the accuracy drops roughly 10%. Chapter 5 provides possible interpretations of these results.

While GPT-4 achieves high performance, the performance of the baseline approach using SVM with BoW is also notably high. With an accuracy score of 82.0%, the BoW SVM approach achieves an accuracy score that is identical to GPT-4's score. The high performance of BoW SVM suggests that there might be idiosyncratic properties in the training dataset that are captured by this baseline. The result is 2 accuracy points higher than the BoW SVM results for ICLE-NLI, despite the fewer number of classes. The high performance of this baseline highlights that the VESPA dataset is not an NLI benchmark and has not yet been tested extensively for NLI.

Furthermore, open-source LLMs still perform drastically worse than GPT-4 on this dataset, emulating the performance patterns of these LLMs on TOEFL11 and ICLE-NLI. When used out-of-the-box, LLaMA-3 achieves an accuracy score of 50.0% while Gemma achieves 20.0% accuracy.

While fine-tuning boosts the performance significantly on both TOEFL11 and ICLE-NLI, fine-tuning on the VESPA training set does not necessarily improve the results. For LLaMA-3, fine-tuning degrades the performance, as the accuracy score drops from 50.0% to 22.0%. For Gemma, fine-tuning slightly improves the performance to 52.0%, which does not match GPT-4's results (as it does for the other datasets). As the training set is imbalanced, we performed additional experiments with an under-sampled training set for the three datasets. All models fine-tuned on the under-sampled training sets demonstrate poorer performance than when fine-tuned on the entire training set. Surprisingly, Gemma demonstrates a marginal decrease of 3.6% and 5.6% on under-sampled TOEFL11 and ICLE-NLI, compared to LLaMA-3 and Mistral, which experience drops in accuracy ranging between 28.5%-73.4%. This indicates that only some of the models require a sufficient number of instances per L1 for some datasets. All in all, the results indicate that fine-tuning can boost the performance of open-source LLMs on the NLI task, but only with a sufficient number of training samples per L1, depending on the model and dataset.

4.7 Summary

First, our results indicate that the performance of GPT-4 is consistently high across ICLE-NLI and TOEFL11, and with that, in line with Zhang and Salle (2023)'s previous results. GPT-4 out-of-the-box demonstrates remarkably high performance on both NLI benchmarks. Compared to GPT-4, Open-source LLMs when used out-of-the-box exhibited drastically poorer performance on the NLI datasets, with accuracy scores that were all lower than a simple baseline approach using an SVM model with Bag-of-Word features. After fine-tuning, however, the performance of most open-source

Table 4.7: Results of open-source and closed-source LLMs on the VESPA test set (50 samples) in a closed-set setting, presented alongside results of these models on the TOEFL11 and ICLE-NLI dataset. Additional results of fine-tuned open-source LLMs on the three datasets under-sampled to 33 texts (the number of samples of the minority class in the VESPA training set) are presented. All results are reported in accuracy score (%) based on one run, except for previously reported results of LLaMA-3, Mistral, and Gemma (both zero-shot & fine-tuned), for which we report the average accuracy score across three runs.

Model	VESPA (test set)	TOEFL11 (test set)	ICLE-NLI (5CV/entire)
BoW SVM	82.0	67.7	79.4
GPT-4	82.0	91.7	95.5
LLaMA-3 (zero-shot)	50.0	56.8±1.1	75.8±0.4
LLaMA-3 (fine-tuned)	30.0	85.3±0.1	78.5±2.5
LLaMA-3 (fine-tuned, under-sampled)	20.0	11.9	15.3
Gemma (zero-shot)	20.0	13.6±0.0	28.2±0.1
Gemma (fine-tuned)	60.0	90.3±1.2	96.6±0.2
Gemma (fine-tuned, under-sampled)	42.0	86.7	91.0
Mistral (zero-shot)	30.0	35.6±1.6	53.1±1.1
Mistral (fine-tuned)	52.0	89.8±0.8	83.2±9.4
Mistral (fine-tuned, under-sampled)	24.0	61.3	14.2

models improved significantly. While Gemma when used out-of-the-box achieved the lowest accuracy score on both datasets in comparison to the other four open-source models, after fine-tuning, Gemma achieved (near-)SOTA performance, even outperforming GPT-4 on ICLE-NLI. In contrast, LLaMA-3 out-of-the-box achieved the best results on TOEFL11 and ICLE-NLI compared to other open-source LLMs out-of-the-box, but LLaMA-3's performance displayed minimal improvement after fine-tuning. This indicates that the open-source models that perform best out of the box do not necessarily achieve the best performance after fine-tuning. We observed all of these patterns of performance across both datasets, which further strengthens the generalizability of our findings. The results of the follow-up experiment on another dataset indicate that fine-tuning requires a sufficient number of samples per L1 depending on the model and dataset.

In addition, we surprisingly found that some models out-of-the-box perform better in an open-set setting, but overall, performance decreases slightly for most LLMs in an open-set setting. Finally, open-source LLMs out-of-the-box demonstrate less promising results regarding prompting for explainability in comparison to closed-source LLMs.

Chapter 5

Discussion

This thesis aimed to investigate how smaller open-source LLMs compare to closed-source LLMs on the NLI task and to further investigate the use of LLMs for prompting for explainability and open-set classification. We directly compared the results of five open-source LLMs to the results achieved by Zhang and Salle (2023) using GPT-3.5 and GPT-4 on the TOEFL11 test set and expanded the results by running GPT-3.5 and GPT-4 on the ICLE-NLI benchmark.

The results reveal a large gap in performance between closed-source and open-source LLMs when used out-of-the-box on the NLI task, with GPT-4 outperforming previous SOTA approaches and open-source LLMs on two NLI benchmarks. After fine-tuning, the performance of open-source LLMs on the NLI task improves drastically. Our results indicate that fine-tuned open-source LLMs can match, and in some cases even outperform, GPT-4 on the NLI benchmarks, with fine-tuned Gemma setting a new performance record of 96.6% on ICLE-NLI. These patterns were found for both datasets, further strengthening the generalizability of our findings. This chapter discusses these findings, the limitations, and possible directions for future research in NLI.

5.1 Performance gap between closed- and open-source LLMs

As mentioned previously, our results indicated that GPT-4 outperformed open-source LLMs when used out-of-the-box on the NLI task by a large margin. We hypothesize a few possible reasons for this performance gap, which we discuss in the sections below.

5.1.1 Potential data contamination

We hypothesize that a possible reason for this performance gap could be data leakage. While the TOEFL11 and ICLE-NLI datasets are both inaccessible to the public, as OpenAI models appear to have been exposed to hundreds of other benchmarks (Balloccu et al., 2024), data leakage is not improbable. TOEFL11 and ICLE-NLI are both *de facto* NLI benchmarks released before the cut-off date of GPT-4 training data, which raises the concern that these might have been seen in training.

For this reason, we performed a follow-up experiment implementing GPT-4, LLaMA-3, and Gemma on VESPA, an English L2 learner corpus that was released after the cut-off date of GPT-4 (September 2021), which makes it highly plausible that this data was not seen in training. The results from this follow-up experiment indicate that GPT-

4 still outperforms open-source LLMs when used out-of-the-box on an NLI dataset that most likely could not have been seen in training. However, the accuracy drops roughly 10 percentage points (p.p.) in comparison to GPT-4's performance on TOEFL11 and ICLE-NLI. The substantial drop on this novel dataset could be interpreted as significant enough to suggest possible data contamination of TOEFL11 and ICLE-NLI in GPT-4. On the other hand, GPT-4's high accuracy score relative to other LLMs could also indicate that GPT-4 maintains a high level of performance on the NLI task, suggesting that GPT-4 has not seen any of the NLI benchmarks. Additional research is required to test the possibility of data leakage of these NLI benchmarks, i.e., by examining whether a model has memorized a given text using perplexity measurements (Carlini et al., 2021).

Our follow-up experiment also raised possible doubts surrounding data contamination of ICLE-NLI in LLaMA-3. While LLaMA-3 when used out-of-the-box achieves 50% and 56.6% accuracy on VESPA and TOEFL11 respectively, the model achieves a much higher score on ICLE-NLI, with 75.8% accuracy. Moreover, while all other open-source LLMs after fine-tuning gain a large boost in performance for both datasets, LLaMA-3's accuracy after fine-tuning on ICLE-NLI increases by 2.7 p.p. only. LLaMA-3's relatively high performance out-of-the-box and marginal performance boost after fine-tuning are inconsistent with the results of other open-source LLMs. This could indicate possible data leakage of ICLE-NLI in LLaMA-3. As suggested previously, additional research is required to test the possibility of leakage of NLI benchmarks in LLMs.

5.1.2 Model size

A possible reason for the gap in performance between closed-source and open-source LLMs out-of-the-box could also be the differences in model size. GPT-3.5 has roughly 175B parameters (Brown et al., 2020), and GPT-4's size is unknown, but likely much larger than that of GPT-3.5. On the other hand, the open-source LLMs used in our experiments are quantized and significantly smaller than the closed-source LLMs, with sizes ranging between 3.8B and 8B parameters. Phi-3, the smallest LLM used in this study with 3.8B parameters (Microsoft, 2024), demonstrates relatively low performance on NLI when used out-of-the-box and the lowest performance of all open-source LLMs after fine-tuning. This indicates that the size of the model could impact the performance of LLMs on the NLI task.

5.1.3 Training data

Another possible reason for the performance gap could be the difference in training data: perhaps the training data of closed-source models include more L2 learner data than open-source models that allows the first to achieve better performance on the NLI task. Smaller open-source LLMs trained on less data often rely on heavy filtering to obtain the same level of performance as much larger LLMs (Microsoft, 2024). Phi-3, for example, relies on heavily filtering web data to obtain data of "high quality" or "textbook quality" (Microsoft, 2024; Gunasekar et al., 2023). This process of filtering would likely negatively impact the performance on a task that is based on L2 learner data which typically contains errors. The difference in the type of data the models are trained on could have increased the performance gap between the two types of models. With the lack of insights into the training data of both open-source and closed-source LLMs, this hypothesis cannot be verified.

5.2 LLMs for open-set classification

Most open-source and closed-source LLMs demonstrated poorer performance in an open-set setting compared to a closed-set one, as expected. With the relatively poor performance of open-source LLMs on both NLI benchmarks in an open-set setting, it appears that open-source LLMs cannot yet be implemented in an open-set manner for real-world applications of NLI, unlike closed-source LLMs.

Surprisingly, Phi-3 and GPT-3.5 demonstrated an increase in performance in an open-set setting. This might be because the prompt used in the closed-set experiments is relatively longer than the one used in open-set experiments, and includes many restrictions, such as "DO NOT USE ANY OTHER CLASS" and "Do not classify any input as "ENG" (English)". While Zhang and Salle (2023) achieved SOTA results with GPT-4 using a nearly identical prompt, this might not be the case for other LLMs, as models appear to display different levels of sensitivity to the prompt, depending on the prompts used (Lu et al., 2024). Future research can experiment with different prompts or prompt engineering methods, such as few-shot learning or Chain-of-Thought prompting, to improve the current results of open-source LLMs on the NLI task.

5.3 Fine-tuning LLMs for NLI

We found that smaller open-source LLMs after fine-tuning can match the performance of, and in some cases even outperform, closed-source LLMs. Gemma after fine-tuning achieved an accuracy score of 90.3% on TOEFL11 and 96.6% on ICLE-NLI, while GPT-4 achieved 91.7% on TOEFL11 and 95.5% on ICLE-NLI. This shows promising results for fine-tuning smaller open-source LLMs for other classification tasks in the future, in particular with the recent development of more efficient fine-tuning techniques, like LoRA (Hu et al., 2021) and QLoRA (Dettmers et al., 2023). Using these novel techniques to adapt open-source LLMs to specific tasks can drastically improve the performance of open-source LLMs.

While fine-tuning open-source LLMs improves the performance of these models in terms of classification accuracy, we found that LLMs after fine-tuning could not be used for generating natural language explanations. The 'general-purpose' property is often framed as one of the main strength of LLMs (Minaee et al., 2024), which allows us to prompt LLMs for reasoning, such as natural language explanations for NLI. When prompting the fine-tuned LLMs for NLI explanations, however, the fine-tuned LLMs had a tendency to only generate an L1 prediction. This behavior might be a sign of overfitting on this task and suggests that LLMs that are fine-tuned for a specific task cease to perform well on other language tasks. For this reason, we could not implement fine-tuned LLMs for our experiments on explainability.

5.4 Using LLMs for explainability

Our results confirm Zhang and Salle (2023)'s findings in that GPT-4 can provide insightful analyses regarding L1-indicative features, albeit with occasional hallucinations. Open-source LLMs like LLaMA-3 seem to not have reached the level of closed-source LLMs yet with regard to generating L2 feature explanations. A qualitative analysis of explanations generated by LLaMA-3 revealed that LLaMA-3 often generates explanations that are rather vague and coarse-grained. The open-source LLM often identi-

fies specific features as indicators of non-native English writing, rather than providing possible sources of particular errors in relation the predicted L1. It is expected that LLaMA-3 can generate less targeted and coherent explanations compared to GPT-4, as it has only 7B parameters and is trained on less data.

Both models present hallucinations in their generated explanations, which raises some doubts about the accuracy of some of the LLM-generated explanations. Our analysis showed that LLaMA-3- and GPT-4-generated explanations can contain misquotations or factual errors, often formulated in a convincing manner. The occurrence of these errors put into question the argument put forward by Zhang and Salle (2023) that LLMs can potentially be used as tools for linguistic analysis by educators and linguists. If a user implements LLMs for linguistic analysis while being unaware of the risk of hallucinations, they risk relying on rhetorically convincing but false or biased explanations (Kunz and Kuhlmann, 2024). This could lead to false conclusions about what differentiates language produced by native speakers and L2 learners. With hallucinations presented by both models, it is important to analyze LLM-generated explanations critically and be aware of the risks when applying it as a method for feature explainability.

5.5 Limitations

Several limitations to our study should be addressed. In what follows, we discuss the limitations concerning our focus on English L2 speakers, our experimental setup, and our definition of open-source and closed-source. These limitations then point toward avenues for future research.

Our study focuses purely on native language identification in English, which is the most well-studied L2 in the NLI task (Goswami et al., 2024). We do not take into account speakers with a multi-L1 background, or NLI in other languages. It would be interesting to study NLI for other L2s, as it is unclear whether the high performance of LLMs on NLI can hold for L2s other than English. In addition, we limit ourselves to investigating language produced by L2 speakers with one L1. It would be interesting to investigate NLI identification using texts written by speakers with multi-L1 backgrounds.

We also focus on datasets comprised of texts written by (under)graduate university students. As Goswami et al. (2024) note, most datasets used for NLI were collected in educational settings. Future experiments can explore the use of LLMs on NLI data from other domains, such as social media, blogs, or online reviews, where L2 learner texts are also prevalent.

Additionally, our study investigated a small sample of generated explanations to gain insights into explainability to the best of our ability. Verifying the validity of the claims presented in the explanations remains difficult, as this requires expert knowledge of linguistic features of the L1s in the dataset. Future research could include a more detailed analysis of LLM-generated NLI explanations with expert knowledge of the different L1s in the dataset.

As previously described, model size seems to impact the performance of LLMs on the NLI task. We were unable to implement larger open-source LLMs (>70B parameters) to fully test the impact of model size on performance due to computational limitations. All experiments were conducted on Google Colaboratory, where we made use of the most powerful GPU (A100) available. However, testing bigger open-source models, such

as LLaMA-3 with 70B parameters (Meta, 2024), requires much larger GPU memory. Future research with access to more computational resources could explore the use of larger open-source LLMs on the NLI task.

In fine-tuning, we used the same hyperparameters for each model and the different datasets. Hyperparameter optimization was not explored for our experiments to minimize the computational costs. Future research could study the effect of hyperparameter optimization on the performance when fine-tuning LLMs for the NLI task.

Moreover, while fine-tuning improves the performance of open-source LLMs drastically, the prerequisite of fine-tuning for optimal performance is a disadvantage to implementing open-source LLMs compared to closed-source LLMs. For high performance, the model has to be fine-tuned on large amounts of labeled domain-specific data. Previous research has found that NLI models suffer from performance degradation in a cross-corpus or cross-domain setting, and thus cannot be applied directly to different corpora (Markov et al., 2022; Malmasi and Dras, 2015a). Future research could explore the use of fine-tuned open-source LLMs for NLI in a cross-corpus setting to evaluate whether these models demonstrate performance loss in this setting.

More broadly, in our study, we define open-source and closed-source relatively loosely, treating the terms open and closed as a binary feature to perform a comparative analysis between open-source and closed-source LLMs for NLI. However, there are various dimensions of openness, as a model release involves different components ranging from the release of training datasets to model access (Solaiman, 2023; Liesenfeld and Dingemanse, 2024). There is currently an alarming trend within the area of LLM development in which companies take credit for releasing open-source LLMs, without actually disclosing crucial information regarding the training procedure, also referred to as 'open-washing' (Liesenfeld and Dingemanse, 2024). LLMs are often released by blog post rather than through a peer-reviewed scientific article, lacking fine-grained analyses of the performance. In turn, this makes it hard to determine whether an open-source model's performance can be attributed to the model's learning or possible data contamination. The lack of insights into the training data of proclaimed open-source LLMs also hindered our evaluation of LlaMA-3 on ICLE-NLI. In the future, researchers working with large language models should be aware of the complexities of defining the 'open-source'-ness of large language models.

Chapter 6

Conclusion

In this thesis, we presented, to the best of our knowledge, the first results using open-source LLMs on the Native Language Identification task, comparing these to the performance of closed-source LLMs like GPT-4. We implemented a variety of open-source LLMs, namely LLaMA-2, LLaMA-3, Mistral, Gemma, and Phi-3, running them out-of-the-box as well as fine-tuning for the NLI task. In addition, we tested the ability of open-source LLMs to predict authors' L1s in an open-set setting, without specifying the list of possible L1s, and explored the use of open-source LLMs to provide explanations regarding their L1 classification.

The results showed that open-source LLMs when used out-of-the-box demonstrate considerably poorer performance than closed-source LLMs like GPT-4 on two NLI benchmarks. They also generally showcased a decrease in performance in an open-set setting and demonstrated lesser performance with respect to generating explanations. This indicates that small open-source LLMs when used out-of-the-box cannot achieve the same level of performance as much larger closed-source LLMs on the NLI task.

When fine-tuned for the NLI task, however, open-source LLMs generally demonstrate a significant boost in performance. Our results indicated that fine-tuned open-source LLMs can achieve (near-)state-of-the-art performance on two NLI benchmarks, with fine-tuned Gemma setting a new performance record of 96.6% on ICLE-NLI. Taking into account the negative impact of closed-source LLMs like GPT-4 on research, small fine-tuned open-source LLMs present an alternative that shows considerable promise for text classification tasks like NLI.

Future research can explore the multilingual capabilities of LLMs for the NLI task by examining the use of LLMs for languages other than English, or explore NLI datasets from different domains, such as data collected from social media. Current results could also be improved by implementing prompt engineering methods, such as few-shot learning or exploring different prompt formats.

Appendix A

Prompts

A.1 Closed-set prompts

For the closed-set experiments for the TOEFL11 dataset, we used the prompt below.

> You are a forensic linguistics expert that reads English texts written by non-native authors to classify the native language of the author as one of:
>
> "ARA": Arabic
> "CHI": Chinese
> "FRE": French
> "GER": German
> "HIN": Hindi
> "ITA": Italian
> "JPN": Japanese
> "KOR": Korean
> "SPA": Spanish
> "TEL": Telugu
> "TUR": Turkish
> Use clues such as spelling errors, word choice, syntactic patterns, and grammatical errors to decide on the native language of the author.
>
> DO NOT USE ANY OTHER CLASS.
> IMPORTANT: Do not classify any input as "ENG" (English). English is an invalid choice.
> Valid output formats:
> Class: "ARA"
> Class: "CHI"
> Class: "FRE"
> Class: "GER"
>
> You ONLY respond in JSON files. The expected output from you is: json {"native_lang": The chosen class, ARA, CHI, FRE, GER, HIN, ITA, JPN, KOR, SPA, TEL, or TUR}

If possible, this was entered as a System prompt. If the system role not supported

by the prompt formatter, this was entered as part of the User prompt.

We then input the given text and used the prompt below as a User prompt:

> <TOEFL11 ESSAY TEXT>
> Classify the text above as one of ARA, CHI, FRE, GER, HIN, ITA, JPN, KOR, SPA, TEL, or TUR. Do not output any other class - do NOT choose "ENG" (English). What is the closest native language of the author of this English text from the given list?

For the ICLE-NLI dataset, as the set of labels are different from TOEFL11, we used the following prompt as system prompt if possible:

> You are a forensic linguistics expert that reads English texts written by non-native authors to classify the native language of the author as one of:
>
> "BUL": Bulgarian
> "CHI": Chinese
> "CZE": Czech
> "FRE": French
> "JPN": Japanese
> "RUS": Russian
> "SPA": Spanish
> Use clues such as spelling errors, word choice, syntactic patterns, and grammatical errors to decide on the native language of the author.
>
> DO NOT USE ANY OTHER CLASS.
> IMPORTANT: Do not classify any input as "ENG" (English). English is an invalid choice.
>
> Valid output formats: Class: "BUL"
> Class: "CHI"
> Class: "CZE"
> Class: "SPA"
>
> You ONLY respond in JSON files. The expected output from you has to be:"json {"native_lang": The chosen class, BUL, CHI, CZE, FRE, JPN, RUS, or SPA}"

We then used the following prompt as input prompt:

> <ICLE-NLI ESSAY TEXT>
> Classify the text above as one of BUL, CHI, CZE, FRE, JPN, RUS, or SPA. Do not output any other class - do NOT choose "ENG" (English). What is the closest native language of the author of this English text from the given list?

In the closed-set experiments, if the L1 was incorrectly predicted as English, we prompted the model below again using the prompt below:

> You previously mistakenly predicted this text as "ENG" (English). The class is NOT English. Please classify the native language of the author of the text again.

If we were unable to parse the prediction or the predicted L1 was not in the set of possible classes, we prompted the model again. For the TOEFL11 experiments we used the prompt below:

> Your classification is not in the list of possible languages.
> Please try again and choose only one of the following classes: ARA, CHI, FRE, GER, HIN, ITA, JPN, KOR, SPA, TEL, or TUR

For out-of-set predictions on the ICLE-NLI dataset, we prompted the model again using the prompt below:

> Your classification is not in the list of possible languages.
> Please try again and choose only one of the following classes: BUL, CHI, CZE, FRE, JPN, RUS, or SPA

A.2 Open-set prompts

For the open-set experiments, we used the prompt below as input prompt for all models:

> You are a forensic linguistics expert that reads texts written by non-native authors in order to identify their native language.
> Analyze each text and identify the native language of the author.
> Use clues such as spelling errors, word choice, syntactic patterns, and grammatical errors to decide.
>
> You ONLY respond in JSON files. The expected output from you has to be: "json {"native_lang": ""}"

When the predicted L1 could not be extracted from the generated output, we used the prompt below to apply iterative prompting to get a valid prediction:

> Your previous classification was not in the correct format. Please only respond in the following JSON format:
> "json {"native_lang": ""}"

A.3 Fine-tuning prompts

We used prompts for our fine-tuning experiments that are very similar to the one used in closed-set classification.

For the TOEFL11 dataset, we implemented the following prompt:

Instruction:
You are a forensic linguistics expert that reads English texts written by non-native authors to classify the native language of the author as one of:

“ARA”: Arabic
“CHI”: Chinese
“FRE”: French
“GER”: German
“HIN”: Hindi
“ITA”: Italian
“JPN”: Japanese
“KOR”: Korean
“SPA”: Spanish
“TEL”: Telugu
“TUR”: Turkish
Use clues such as spelling errors, word choice, syntactic patterns, and grammatical errors to decide on the native language of the author.

DO NOT USE ANY OTHER CLASS.
IMPORTANT: Do not classify any input as “ENG” (English). English is an invalid choice.

Valid output formats:
Class: “ARA”
Class: “CHI”
Class: “FRE”
Class: “GER”

Classify the text above as one of ARA, CHI, FRE, GER, HIN, ITA, JPN, KOR, SPA, TEL, or TUR. Do not output any other class - do NOT choose ”ENG” (English). What is the closest native language of the author of this English text from the given list?

Input:
<TOEFL11 ESSAY TEXT>

Response:
<L1 LABEL>

For the ICLE-NLI dataset, we implemented the prompt below:

Instruction:
You are a forensic linguistics expert that reads English texts written by non-native authors to classify the native language of the author as one of:

"BUL": Bulgarian
"CHI": Chinese
"CZE": Czech
"FRE": French
"JPN": Japanese
"RUS": Russian
"SPA": Spanish
Use clues such as spelling errors, word choice, syntactic patterns, and grammatical errors to decide on the native language of the author.

DO NOT USE ANY OTHER CLASS.
IMPORTANT: Do not classify any input as "ENG" (English). English is an invalid choice.

Valid output formats: Class: "BUL"
Class: "CHI"
Class: "CZE"
Class: "SPA"

Classify the text above as one of BUL, CHI, CZE, FRE, JPN, RUS, or SPA. Do not output any other class - do NOT choose "ENG" (English). What is the closest native language of the author of this English text from the given list?

Input:
<ICLE-NLI ESSAY TEXT>

Response:
<L1 LABEL>

A.4 Explainability prompts

For the explainability experiments, we used the system prompt as previously described in Appendix A.1 in accordance with the dataset, and the following prompt as input prompt:

You must provide a guess. Output two named sections: (1) "Native Language" with the name of the language, and (2) "Reasoning" with a detailed explanation of your judgement with examples from the text.

A.5 Follow-up experiment prompts

For the follow-up experiments, we used prompts that are very similar to previous experiments, adapting only the L1s in the dataset.

We used the system prompt below:

> You are a forensic linguistics expert that reads English texts written by non-native authors to classify the native language of the author as one of:
>
> "DUT": Dutch
> "FRE": French
> "NOR": Norwegian
> "SPA": Spanish
> "SWE": Swedish
> Use clues such as spelling errors, word choice, syntactic patterns, and grammatical errors to decide on the native language of the author.
>
> DO NOT USE ANY OTHER CLASS.
> IMPORTANT: Do not classify any input as "ENG" (English). English is an invalid choice.
>
> Valid output formats: Class: "DUT"
> Class: "SWE"
> Class: "NOR"
> Class: "SPA"
>
> You ONLY respond in JSON files. The expected output from you has to be:"json {"native_lang": The chosen class, DUT, FRE, NOR, SPA, or SWE}"

We entered the following prompt as User prompt:

> <VESPA ESSAY TEXT>
> Classify the text above as one of DUT, FRE, NOR, SPA, or SWE. Do not output any other class - do NOT choose "ENG" (English). What is the closest native language of the author of this English text from the given list?

If the L1 was English, we prompted again using the same prompt as implemented in previous experiments (Appendix A.1). If the L1 was not present in the set of labels, we applied iterative prompting using the following prompt:

> Your classification is not in the list of possible languages.
> Please try again and choose only one of the following classes: DUT, FRE, NOR, SPA, or SWE

We used the following prompt for the fine-tuning experiments:

Instruction:
You are a forensic linguistics expert that reads English texts written by non-native authors to classify the native language of the author as one of:

“DUT”: Dutch
“FRE”: French
“NOR”: Norwegian
“SPA”: Spanish
“SWE”: Swedish
Use clues such as spelling errors, word choice, syntactic patterns, and grammatical errors to decide on the native language of the author.

DO NOT USE ANY OTHER CLASS.
IMPORTANT: Do not classify any input as “ENG” (English). English is an invalid choice.

Valid output formats:
Class: “DUT”
Class: “SWE”
Class: “NOR”
Class: “SPA”

Classify the text above as one of DUT, FRE, NOR, SPA, or SWE. Do not output any other class - do NOT choose ”ENG” (English). What is the closest native language of the author of this English text from the given list?

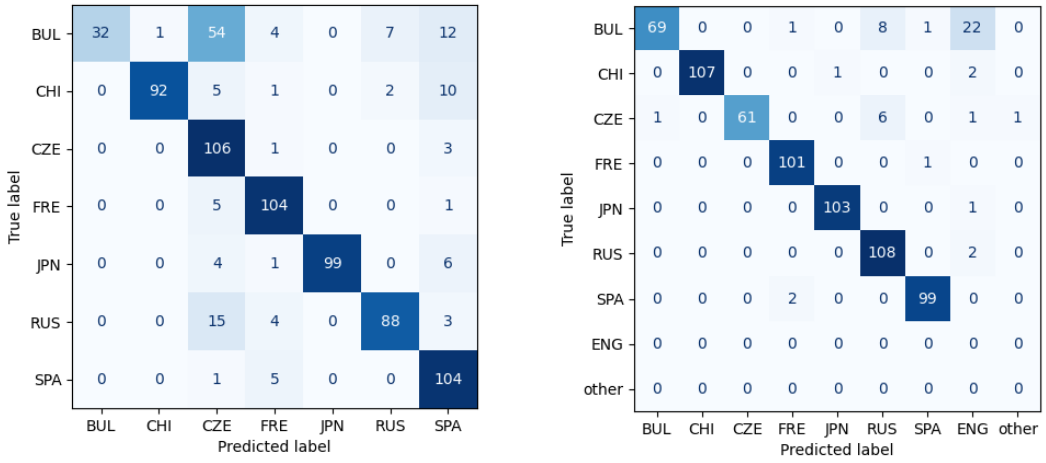
Input:
<VESPA ESSAY TEXT>

Response:
<L1 LABEL>

Appendix B

Confusion matrices

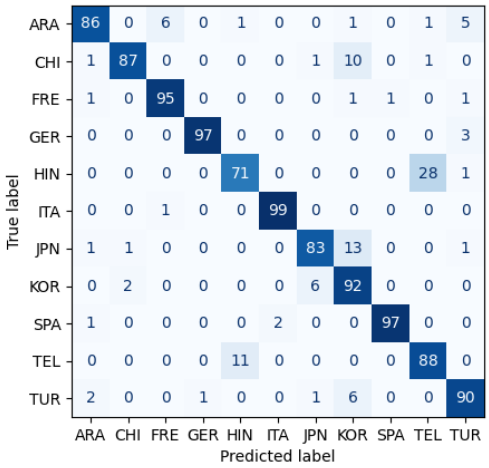
This appendix contains additional relevant confusion matrices.



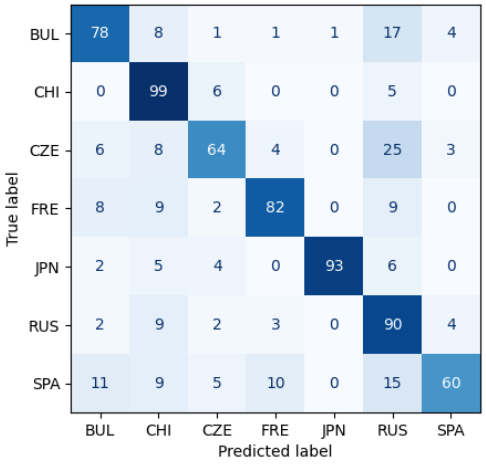
(a) GPT-3.5 (closed)

(b) GPT-3.5 (open)

Figure B.1: Confusion matrix of GPT-3.5 evaluated on the entire ICLE-NLI dataset in a closed-set and open-set setting.

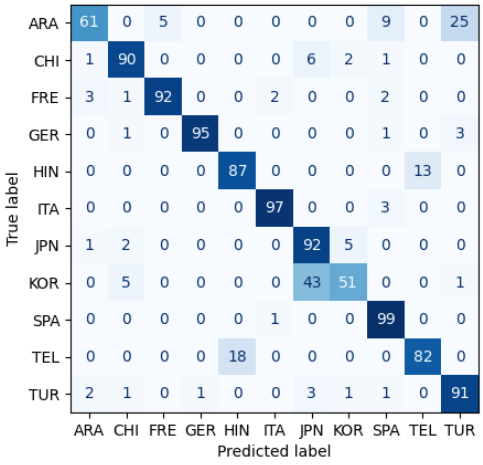


(a) Mistral (FT TOEFL11)

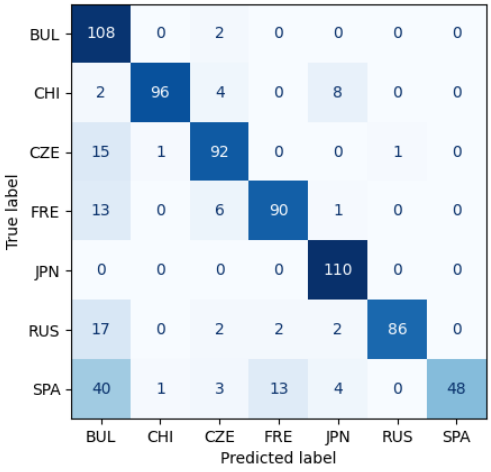


(b) Mistral (FT ICLE-NLI)

Figure B.2: Confusion matrix of Mistral fine-tuned on the TOEFL11 training set, evaluated on the TOEFL11 test set, and Mistral fine-tuned on the ICLE-NLI dataset using 5-fold CV.

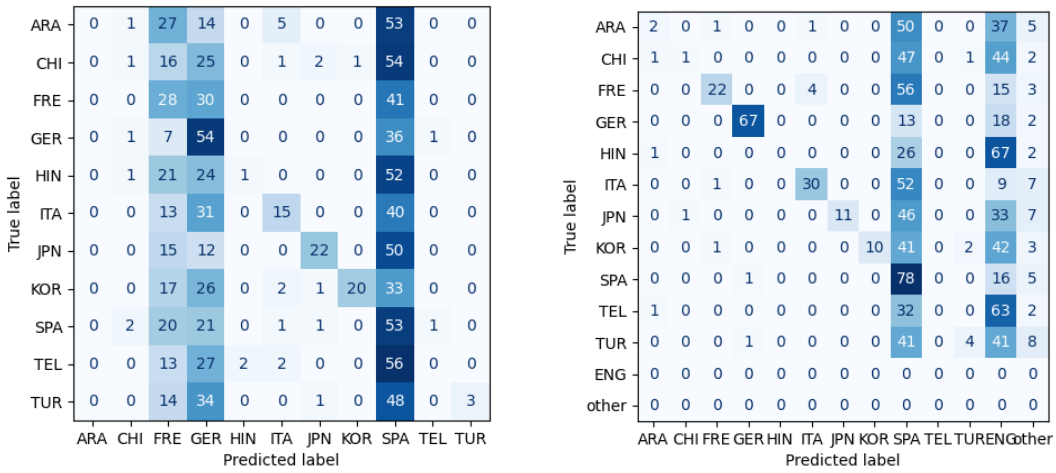


(a) LLaMA-3 (FT TOEFL11)



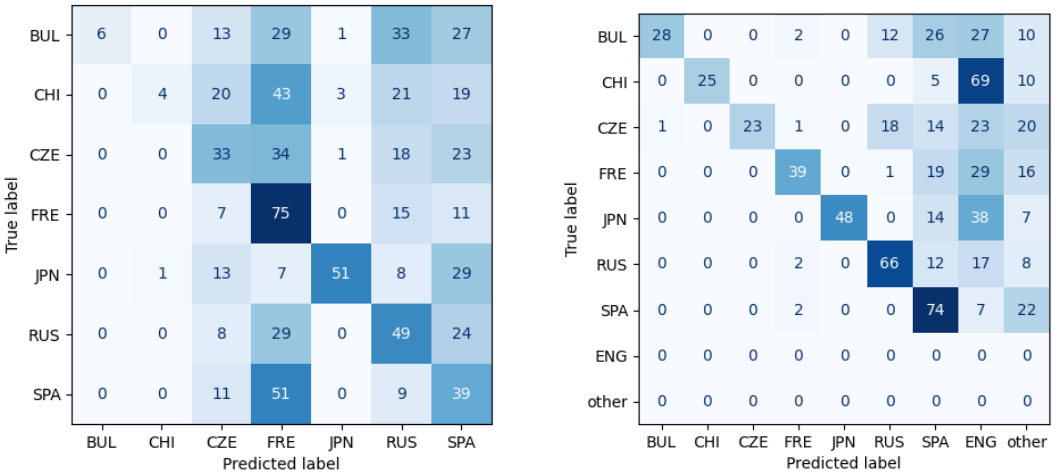
(b) LLaMA-3 (FT ICLE-NLI)

Figure B.3: Confusion matrix of LLaMA-3 fine-tuned on the TOEFL11 training set, evaluated on the TOEFL11 test set, and fine-tuned on the ICLE-NLI dataset using 5-fold CV.



(a) Phi-3 (closed) (b) Phi-3 (open)

Figure B.4: Confusion matrix of Phi-3 used out-of-the-box on the TOEFL11 test set in a closed-set and open-set setting.



(a) Phi-3 (closed) (b) Phi-3 (open)

Figure B.5: Confusion matrix of Phi-3 used out-of-the-box on the ICLE-NLI dataset in a closed-set and open-set setting.

Bibliography

J. Ainslie, J. Lee-Thorp, M. de Jong, Y. Zemlyanskiy, F. Lebrón, and S. Sanghai. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. *ArXiv*, 2023. doi: 2305.13245. URL https://arxiv.org/abs/2305.13245.

S. Balloccu, P. Schmidtová, M. Lango, and O. Dusek. Leak, cheat, repeat: Data contamination and evaluation malpractices in closed-source LLMs. In Y. Graham and M. Purver, editors, *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 67–93, St. Julian's, Malta, Mar. 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.eacl-long.5.

E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *FAccT 2021 - Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 610–623. Association for Computing Machinery, Inc, 3 2021. ISBN 9781450383097. doi: 10.1145/3442188.3445922. URL https://dl.acm.org/doi/10.1145/3442188.3445922.

BigScience. Bloom: A 176b-parameter open-access multilingual language model major contributors prompt engineering architecture and objective engineering evaluation and interpretability broader impacts. *ArXiv*, 2023. doi: 2211.05100. URL https://arxiv.org/abs/2211.05100.

D. Blanchard, J. Tetreault, D. Higgins, A. Cahill, and M. Chodorow. Toefl11: A corpus of non-native english. *ETS Research Report Series*, 2013:1–15, 2013. doi: 10.1002/j.2333-8504.2013.tb02331.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/j.2333-8504.2013.tb02331.x.

J. Brooke and G. Hirst. Robust, lexicalized native language identification. In M. Kay and C. Boitet, editors, *Proceedings of COLING 2012*, pages 391–408, Mumbai, India, Dec. 2012. The COLING 2012 Organizing Committee. URL https://aclanthology.org/C12-1025.

J. Brooke and G. Hirst. Native language detection with 'cheap' learner corpora. In S. Granger, G. Gilquin, and F. Meunier, editors, *Twenty Years of Learner Corpus Research: Looking back, Moving ahead*, pages 37–47. Corpora and Language in Use - Proceedings 1, Louvain-la-Neuve: Presses universitaires de Louvain, 2013. URL http://ftp.cs.toronto.edu/pub/gh/Brooke+Hirst-LCRbook-2013.pdf.

T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger,

T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, 2020. ISBN 9781713829546. URL https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

M. J. J. Bucher and M. Martini. Fine-tuned 'small' llms (still) significantly outperform zero-shot generative ai models in text classification. *ArXiv*, 2024. doi: 2406.08660. URL https://arxiv.org/abs/2406.08660.

N. Carlini, F. Tramèr, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. Brown, D. Song, Ú. Erlingsson, A. Oprea, and C. Raffel. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650. USENIX Association, Aug. 2021. ISBN 978-1-939133-24-3. URL https://www.usenix.org/conference/usenixsecurity21/presentation/carlini-extracting.

L. Chen, C. Strapparava, and V. Nastase. Improving native language identification by using spelling errors. In R. Barzilay and M.-Y. Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 542–546, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-2086. URL https://aclanthology.org/P17-2086.

L. Chen, M. Zaharia, and J. Zou. How is chatgpt' s behavior changing over time? *Harvard Data Science Review*, 6, 3 2024. doi: 10.1162/99608f92.5317da47. URL https://hdsr.mitpress.mit.edu/pub/y95zitmz/release/2.

P. F. Christiano, J. Leike, T. B. Brown, M. Martic, S. Legg, and D. Amodei. Deep reinforcement learning from human preferences. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Neural Information Processing Systems (NIPS) '17, page 4302–4310, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/d5e2c0adad503c91f91df240d0cd4e49-Paper.pdf.

A. Cimino and F. Dell'Orletta. Stacked sentence-document classifier approach for improving native language identification. In J. Tetreault, J. Burstein, C. Leacock, and H. Yannakoudakis, editors, *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 430–437, Copenhagen, Denmark, Sept. 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-5049. URL https://aclanthology.org/W17-5049.

T. Dettmers, M. Lewis, Y. Belkada, and L. Zettlemoyer. Llm.int8(): 8-bit matrix multiplication for transformers at scale, 2022. URL https://arxiv.org/abs/2208.07339.

T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *ArXiv*, 2023. doi: 2305.14314. URL https://arxiv.org/abs/2305.14314.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, and T. Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

A. Edwards and J. Camacho-Collados. Language models for text classification: Is in-context learning enough? In N. Calzolari, M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti, and N. Xue, editors, *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 10058–10072, Torino, Italia, May 2024. ELRA and ICCL. URL https://aclanthology.org/2024.lrec-main.879.

D. Estival, T. Gaustad, S. Pham, W. Radford, and B. Hutchinson. Author profiling for english emails. *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics*, pages 263–272, December 2007. URL http://www.dominique-estival.net/PACLING07Final.pdf.

D. Goswami, S. Thilagan, K. North, S. Malmasi, and M. Zampieri. Native language identification in texts: A survey. In K. Duh, H. Gomez, and S. Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3149–3160, Mexico City, Mexico, June 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.naacl-long.173.

S. Granger, E. Dagneaux, F. Meunier, and M. Paquot. *International Corpus of Learner English. Version 2. Handbook and CD-ROM.* Presses Universitaires de Louvain, 01 2009. ISBN 978-2-87463-143-6.

S. Gunasekar, Y. Zhang, J. Aneja, C. C. T. Mendes, A. D. Giorno, S. Gopi, M. Javaheripi, P. Kauffmann, G. de Rosa, O. Saarikivi, A. Salim, S. Shah, H. S. Behl, X. Wang, S. Bubeck, R. Eldan, A. T. Kalai, Y. T. Lee, and Y. Li. Textbooks are all you need. *arXiv*, 2023. doi: 2306.11644. URL https://arxiv.org/abs/2306.11644.

E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. Lora: Low-rank adaptation of large language models. *ArXiv*, 2021. doi: 2106.09685. URL https://arxiv.org/abs/2106.09685.

R. T. Ionescu, M. Popescu, and A. Cahill. Can characters reveal your native language? a language-independent approach to native language identification. In A. Moschitti, B. Pang, and W. Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1363–1373, Doha, Qatar, Oct. 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1142. URL https://aclanthology.org/D14-1142.

S. Jarvis, Y. Bestgen, and S. Pepper. Maximizing classification accuracy in native language identification. In J. Tetreault, J. Burstein, and C. Leacock, editors, *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 111–118, Atlanta, Georgia, June 2013. Association for Computational Linguistics. URL https://aclanthology.org/W13-1714.

Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and P. Fung. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12), mar 2023. ISSN 0360-0300. doi: 10.1145/3571730. URL https://doi.org/10.1145/3571730.

A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed. Mistral 7b. *ArXiv*, 10 2023. doi: 2310.06825. URL http://arxiv.org/abs/2310.06825.

M. Koppel, J. Schler, and K. Zigdon. Determining an author's native language by mining a text for errors. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, KDD '05, page 624–628, New York, NY, USA, 2005. Association for Computing Machinery. ISBN 159593135X. doi: 10.1145/1081870.1081947. URL https://doi.org/10.1145/1081870.1081947.

T. Kudo and J. Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In E. Blanco and W. Lu, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium, Nov. 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-2012. URL https://aclanthology.org/D18-2012.

A. Kulmizev, B. Blankers, J. Bjerva, M. Nissim, G. van Noord, B. Plank, and M. Wieling. The power of character n-grams in native language identification. In J. Tetreault, J. Burstein, C. Leacock, and H. Yannakoudakis, editors, *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 382–389, Copenhagen, Denmark, Sept. 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-5043. URL https://aclanthology.org/W17-5043.

J. Kunz and M. Kuhlmann. Properties and challenges of LLM-generated explanations. In S. L. Blodgett, A. C. Curry, S. Dey, M. Madaio, A. Nenkova, D. Yang, and Z. Xiao, editors, *Proceedings of the Third Workshop on Bridging Human–Computer Interaction and Natural Language Processing*, pages 13–27, Mexico City, Mexico, June 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.hcinlp-1.2.

A. Lacoste, A. Luccioni, V. Schmidt, and T. Dandres. Quantifying the carbon emissions of machine learning. *ArXiv*, 2019. doi: 1910.09700. URL https://arxiv.org/abs/1910.09700.

G. Lemaître, F. Nogueira, and C. K. Aridas. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5, 2017. URL http://jmlr.org/papers/v18/16-365.html.

A. Liesenfeld and M. Dingemanse. Rethinking open source generative ai: openwashing and the eu ai act. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '24, page 1774–1787, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400704505. doi: 10.1145/3630106.3659005. URL https://doi.org/10.1145/3630106.3659005.

E. Lotfi, I. Markov, and W. Daelemans. A deep generative approach to native language identification. In D. Scott, N. Bel, and C. Zong, editors, *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1778–1783, Barcelona, Spain (Online), Dec. 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.159. URL https://aclanthology.org/2020.coling-main.159.

S. Lu, H. Schuff, and I. Gurevych. How are prompts different in terms of sensitivity? In K. Duh, H. Gomez, and S. Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5833–5856, Mexico City, Mexico, June 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.naacl-long.325.

S. Malmasi. *Native language identification: explorations and applications*. PhD thesis, Macquarie University, Sydney, Australia, 3 2022. URL https://figshare.mq.edu.au/articles/thesis/Native_language_identification_explorations_and_applications/19437986.

S. Malmasi and M. Dras. Arabic native language identification. In N. Habash and S. Vogel, editors, *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 180–186, Doha, Qatar, Oct. 2014a. Association for Computational Linguistics. doi: 10.3115/v1/W14-3625. URL https://aclanthology.org/W14-3625.

S. Malmasi and M. Dras. Chinese native language identification. In S. Wintner, S. Riezler, and S. Goldwater, editors, *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 95–99, Gothenburg, Sweden, Apr. 2014b. Association for Computational Linguistics. doi: 10.3115/v1/E14-4019. URL https://aclanthology.org/E14-4019.

S. Malmasi and M. Dras. Large-scale native language identification with cross-corpus evaluation. In R. Mihalcea, J. Chai, and A. Sarkar, editors, *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1403–1409, Denver, Colorado, May–June 2015a. Association for Computational Linguistics. doi: 10.3115/v1/N15-1160. URL https://aclanthology.org/N15-1160.

S. Malmasi and M. Dras. Multilingual native language identification. *Natural Language Engineering*, 23:163–215, 2015b. doi: 10.1017/S1351324915000406. URL https://doi.org/10.1017/S1351324915000406.

S. Malmasi, K. Evanini, A. Cahill, J. Tetreault, R. Pugh, C. Hamill, D. Napolitano, and Y. Qian. A report on the 2017 native language identification shared task. In J. Tetreault, J. Burstein, C. Leacock, and H. Yannakoudakis, editors, *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 62–75, Copenhagen, Denmark, Sept. 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-5007. URL https://aclanthology.org/W17-5007.

I. Markov, L. Chen, C. Strapparava, and G. Sidorov. CIC-FBK approach to native language identification. In J. Tetreault, J. Burstein, C. Leacock, and H. Yannakoudakis, editors, *Proceedings of the 12th Workshop on Innovative Use of NLP*

for Building Educational Applications, pages 374–381, Copenhagen, Denmark, Sept. 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-5042. URL https://aclanthology.org/W17-5042.

I. Markov, V. Nastase, and C. Strapparava. Punctuation as native language interference. In E. M. Bender, L. Derczynski, and P. Isabelle, editors, *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3456–3466, Santa Fe, New Mexico, USA, Aug. 2018. Association for Computational Linguistics. URL https://aclanthology.org/C18-1293.

I. Markov, V. Nastase, and C. Strapparava. Anglicized words and misspelled cognates in native language identification. In H. Yannakoudakis, E. Kochmar, C. Leacock, N. Madnani, I. Pilán, and T. Zesch, editors, *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 275–284, Florence, Italy, Aug. 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4429. URL https://aclanthology.org/W19-4429.

I. Markov, V. Nastase, and C. Strapparava. Exploiting native language interference for native language identification. *Natural Language Engineering*, 28:167–197, 2022. doi: 10.1017/S1351324920000595. URL https://doi.org/10.1017/S1351324920000595.

T. Mesnard, C. Hardin, R. Dadashi, S. Bhupatiraju, S. Pathak, L. Sifre, M. Rivière, M. S. Kale, J. Love, P. Tafti, L. Hussenot, P. G. Sessa, A. Chowdhery, A. Roberts, A. Barua, A. Botev, A. Castro-Ros, A. Slone, A. Héliou, A. Tacchetti, A. Bulanova, A. Paterson, B. Tsai, B. Shahriari, C. L. Lan, C. A. Choquette-Choo, C. Crepy, D. Cer, D. Ippolito, D. Reid, E. Buchatskaya, E. Ni, E. Noland, G. Yan, G. Tucker, G.-C. Muraru, G. Rozhdestvenskiy, H. Michalewski, I. Tenney, I. Grishchenko, J. Austin, J. Keeling, J. Labanowski, J.-B. Lespiau, J. Stanway, J. Brennan, J. Chen, J. Ferret, J. Chiu, J. Mao-Jones, K. Lee, K. Yu, K. Millican, L. L. Sjoesund, L. Lee, L. Dixon, M. Reid, M. Mikuła, M. Wirth, M. Sharman, N. Chinaev, N. Thain, O. Bachem, O. Chang, O. Wahltinez, P. Bailey, P. Michel, P. Yotov, R. Chaabouni, R. Comanescu, R. Jana, R. Anil, R. McIlroy, R. Liu, R. Mullins, S. L. Smith, S. Borgeaud, S. Girgin, S. Douglas, S. Pandya, S. Shakeri, S. De, T. Klimenko, T. Hennigan, V. Feinberg, W. Stokowiec, Y. hui Chen, Z. Ahmed, Z. Gong, T. Warkentin, L. Peran, M. Giang, C. Farabet, O. Vinyals, J. Dean, K. Kavukcuoglu, D. Hassabis, Z. Ghahramani, D. Eck, J. Barral, F. Pereira, E. Collins, A. Joulin, N. Fiedel, E. Senter, A. Andreev, and K. Kenealy. Gemma: Open models based on gemini research and technology. *ArXiv*, 3 2024. doi: 2403.08295. URL http://arxiv.org/abs/2403.08295.

Meta. Llama 3. *Meta Blog*, Apr 2024. URL https://ai.meta.com/blog/meta-llama-3/. Accessed: 20 May 2024.

Microsoft. Phi-3 technical report: A highly capable language model locally on your phone. *ArXiv*, 2024. doi: 2404.14219. URL https://arxiv.org/abs/2404.14219.

S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, and J. Gao. Large language models: A survey. *ArXiv*, 2024. doi: 2402.06196. URL https://arxiv.org/abs/2402.06196.

T. Odlin. *Language Transfer*. Cambridge Applied Linguistics. Cambridge University Press, 1989.

OpenAI. Introducing chatgpt, November 2022. URL https://openai.com/index/chatgpt/. Accessed: 20 May 2024.

OpenAI. Gpt-4 technical report. *ArXiv*, abs/2303.08774, 2023. URL https://arxiv.org/abs/2303.08774.

M. Paquot, T. Larsson, H. Hasselgård, S. O. Ebeling, D. D. Meyere, L. Valentin, N. J. Laso, I. Verdaguer, and S. van Vuuren. The varieties of english for specific purposes database (vespa): Towards a multi-l1 and multi-register learner corpus of disciplinary writing. *Research in Corpus Linguistics*, 10:1–15, 2022. ISSN 2243-4712. doi: 10.32714/ricl.10.02.02. URL https://ricl.aelinco.es/index.php/ricl/article/view/223.

L. Pozzobon, B. Ermis, P. Lewis, and S. Hooker. On the challenges of using black-box APIs for toxicity evaluation in research. In H. Bouamor, J. Pino, and K. Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7595–7609, Singapore, Dec. 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.472. URL https://aclanthology.org/2023.emnlp-main.472.

Y. Qiu and Y. Jin. Chatgpt and finetuned bert: A comparative study for developing intelligent design support systems. *Intelligent Systems with Applications*, 21:1–16, 3 2024. ISSN 26673053. doi: 10.1016/j.iswa.2023.200308. URL https://www.sciencedirect.com/science/article/pii/S2667305323001333?ssrnid=4516782&dgcid=SSRN_redirect_SD.

E. Rabinovich, Y. Tsvetkov, and S. Wintner. Native language cognate effects on second language lexical choice. *Transactions of the Association for Computational Linguistics*, 6:329–342, 2018. doi: 10.1162/tacl_a_00024. URL https://aclanthology.org/Q18-1024.

A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 2019. URL https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

R. Rafailov, A. Sharma, E. Mitchell, S. Ermon, C. D. Manning, and C. Finn. Direct preference optimization: Your language model is secretly a reward model. *ArXiv*, 2023. doi: 2305.18290. URL https://arxiv.org/abs/2305.18290.

N. Remnev. Native language identification for russian. In *2019 International Conference on Data Mining Workshops (ICDMW)*, pages 1–7, Nov 2019. doi: 10.1109/ICDMW48858.2019.9024756. URL https://ieeexplore.ieee.org/document/9024756.

A. Rozovskaya and D. Roth. Algorithm selection and model adaptation for ESL correction tasks. In D. Lin, Y. Matsumoto, and R. Mihalcea, editors, *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 924–933, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL https://aclanthology.org/P11-1093.

J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *ArXiv*, 2017. doi: 1707.06347. URL https://arxiv.org/abs/1707.06347.

P. Shaw, J. Uszkoreit, and A. Vaswani. Self-attention with relative position representations. In M. Walker, H. Ji, and A. Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2074. URL https://aclanthology.org/N18-2074.

I. Solaiman. The gradient of generative ai release: Methods and considerations. *ArXiv*, arXiv.2302.04844, 2023. URL https://arxiv.org/abs/2302.04844.

S. Steinbakken and B. Gambäck. Native-language identification with attention. In P. Bhattacharyya, D. M. Sharma, and R. Sangal, editors, *Proceedings of the 17th International Conference on Natural Language Processing (ICON)*, pages 261–271, Indian Institute of Technology Patna, Patna, India, Dec. 2020. NLP Association of India (NLPAI). URL https://aclanthology.org/2020.icon-main.35.

E. Strubell, A. Ganesh, and A. McCallum. Energy and policy considerations for deep learning in NLP. In A. Korhonen, D. Traum, and L. Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1355. URL https://aclanthology.org/P19-1355.

S. Sun, Y. Zhang, J. Yan, Y. Gao, D. Ong, B. Chen, and J. Su. Battle of the large language models: Dolly vs LLaMA vs vicuna vs guanaco vs bard vs ChatGPT - a text-to-SQL parsing comparison. In H. Bouamor, J. Pino, and K. Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11225–11238, Singapore, Dec. 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.750. URL https://aclanthology.org/2023.findings-emnlp.750.

J. Tetreault, D. Blanchard, A. Cahill, and M. Chodorow. Native tongues, lost and found: Resources and empirical evaluations in native language identification. In M. Kay and C. Boitet, editors, *Proceedings of COLING 2012*, pages 2585–2602, Mumbai, India, Dec. 2012. The COLING 2012 Organizing Committee. URL https://aclanthology.org/C12-1158.

J. Tetreault, D. Blanchard, and A. Cahill. A report on the first native language identification shared task. In J. Tetreault, J. Burstein, and C. Leacock, editors, *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 48–57, Atlanta, Georgia, June 2013. Association for Computational Linguistics. URL https://aclanthology.org/W13-1706.

H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee,

D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, and T. Scialom. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, 2023. doi: 2307.09288. URL https://arxiv.org/abs/2307.09288.

A. Y. Uluslu. Turkish native language identification. In M. Abbas and A. A. Freihat, editors, *Proceedings of the 6th International Conference on Natural Language and Speech Processing (ICNLSP 2023)*, pages 303–307, Online, Dec. 2023. Association for Computational Linguistics. URL https://aclanthology.org/2023.icnlsp-1.32.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Z. Wan, X. Wang, C. Liu, S. Alam, Y. Zheng, J. Liu, Z. Qu, S. Yan, Y. Zhu, bosonai Quanlu Zhang, M. Chowdhury, and M. Zhang. Efficient large language models: A survey. *ArXiv*, 2024. doi: 2312.03863v3. URL https://arxiv.org/abs/2312.03863.

S.-M. J. Wong and M. Dras. Exploiting parse structures for native language identification. In R. Barzilay and M. Johnson, editors, *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1600–1610, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics. URL https://aclanthology.org/D11-1148.

S. Xu, W. Fu, J. Gao, W. Ye, W. Liu, Z. Mei, G. Wang, C. Yu, and Y. Wu. Is dpo superior to ppo for llm alignment? a comprehensive study. *ArXiv*, 2024. doi: 2404.10719. URL https://arxiv.org/abs/2404.10719.

H. Yu, Z. Yang, K. Pelrine, J. F. Godbout, and R. Rabbany. Open, closed, or small language models for text classification? *ArXiv*, 2023. doi: 2308.10092. URL https://arxiv.org/abs/2308.10092.

S. Zhang, L. Dong, X. Li, S. Zhang, X. Sun, S. Wang, J. Li, R. Hu, T. Zhang, F. Wu, and G. Wang. Instruction tuning for large language models: A survey. *ArXiv*, 2024a. doi: 2308.10792. URL https://arxiv.org/abs/2308.10792.

W. Zhang and A. Salle. Native language identification with large language models. *ArXiv*, abs/2312.07819, 2023. URL https://arxiv.org/abs/2312.07819.

W. Zhang, Y. Deng, B. Liu, S. Pan, and L. Bing. Sentiment analysis in the era of large language models: A reality check. In K. Duh, H. Gomez, and S. Bethard, editors, *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3881–3906, Mexico City, Mexico, June 2024b. Association for Computational Linguistics. URL https://aclanthology.org/2024.findings-naacl.246.

Z. Zhang, M. Mita, and M. Komachi. ClozEx: A task toward generation of English cloze explanation. In H. Bouamor, J. Pino, and K. Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5228–5242, Singapore, Dec. 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.347. URL https://aclanthology.org/2023.findings-emnlp.347.