B. Sc. Information Systems

Berlin School of Economics and Law

Department 1: Business and Economics

Bachelor's Thesis

Detecting Gender Bias in English-German Translations using Natural Language Processing

Khanh Linh Pham

Supervisors:	Prof. Dr. Diana Hristova, Prof. Dr. Markus Schaal
Semester:	Summer Semester 2025
Matrikel-Nr.:	77211916753
Email:	klpham04@gmail.com
Date:	**xx.xx.2025**

Abstract

XX

Sperrvermerk

XX

Berlin, den 01. Januar 2099

. . . .

(Unterschrift des Verfassers)

Contents

List of Figures

List of Tables

1 Introduction

Machine Translation (MT) helps millions of people communicate across languages, in daily life and in areas like healthcare, law, and business (Kappl, 2025). Services like Google Translate handle over 200 million users every day (Prates et al., 2019; Shrestha and Das, 2022). It is a fast-growing market. A report by SkyQuest (2025) valued it at 980 million USD in 2023, with projections reaching 2.78 billion USD. New and more advanced translation models keep appearing, and many of them are free to use. As a result, MT tools are now used to translate large volumes of content across domains.

With this widespread use, the output of MT systems increasingly shapes how people receive and interpret information. But automatic translations are not neutral. There is growing concern about the social effects of biased translations. One key issue is gender bias. MT systems are often trained on large datasets that reflect social norms and stereotypes. If the data contains gender bias, the system will likely reproduce it (Cho et al., 2019; Soundararajan and Delany, 2024; Smacchia et al., 2024).

A common case is the use of gendered terms in translations of gender-neutral input. For example, the English sentence “The nurse is hard-working” does not say anything about gender. But a translation system may render it in German as “Die Krankenschwester ist fleißig,” which uses the explicitly feminine term *Krankenschwester*. Similarly, “The surgeon is hard-working” may become “Der Chirurg ist fleißig,” using the masculine form *Chirurg*. These choices add gendered assumptions that were not present in the original. Such patterns are not just technical side effects. They can reinforce stereotypes, especially when they appear in job ads, reports, or other public texts.

1.1 Motivation

1.1.1 Social and Ethical Importance of Addressing Gender Bias

Academia has come to the consensus that MT systems do default to male pronouns when gender in the source sentence is ambiguous (Prates et al., 2019; Cho et al., 2019; Rescigno and Monti, 2023). In addition, translations often reflect traditional roles, like associating “nurse” with women and “surgeon” with men. This can affect people’s perceptions of jobs and reinforce gender roles.

When used in formal contexts like job descriptions or reference letters, biased translations can shape how a candidate is perceived. If a system always assigns male pronouns to leadership roles and female terms to caregiving roles, it may disadvantage those who do not match those stereotypes (Bolukbasi et al., 2016). This is not just a personal issue. It can reduce diversity and go against international standards. Organizations like the United Nations, UNESCO, and the European Union stress the importance of gender equality and inclusive language, making gender equality one of the 17 Sustainable Development Goals for 2030 (Sczesny et al., 2016; United Nations, 2023).

Language also shapes thought. Research shows that readers often interpret masculine forms as male-specific, even if they are supposed to be generic (Sczesny et al., 2016). Inclusive forms are more common in official documents, less so in everyday language. However, exposure matters. Frequent use of fair language makes it feel more normal. Detecting and addressing bias in MT can support this shift.

1.1.2 Why Detection Systems Are Needed

Current research on this topic tends to focus more on the quantitative measurement of gender bias (Rescigno and Monti, 2023; Barclay and Sami, 2024; Smacchia et al., 2024). Common methods include counting gendered forms in outputs and comparing them to demographic baselines or human expectations (Rescigno and Monti, 2023; Prates et al., 2019; Savoldi, Papi, et al., 2024). These are useful, but they do not help users identify specific biased translations in real-time. Evaluations are not enough for accountability.

Other domains, like facial recognition, have already seen progress in active bias detection. For example, Schwemmer et al. (2020) showed that systems tend to label women more accurately if they match stereotypical appearances (e.g., long hair). Some models even linked female images to words like “kitchen” or “cake” based on bias patterns in training data. For MT, a detection layer is still missing. Without such tools, biased translations are likely to spread unnoticed. A detection system could flag potential bias in real time, improving transparency and encouraging more careful use.

1.2 Problem Statement and Research Questions

DRAFT NEED TO REWRITE AFTER IMPLEMENTATION This thesis focuses on gender bias in English-to-German (EN-DE) MT. This language pair is widely used in research, with many open datasets and high-quality models available. It also involves a grammatical shift: English has limited gender marking, while German assigns gender to many nouns and pronouns. This structural difference makes gender bias more visible and easier to study in the translation outputs.

The core problem boils down to the significant bias towards the masculine form in EN-DE MTs, sometimes consituting 93-96% of translations for isolated words (Lardelli et al., 2024). These outputs often reflect social stereotypes rather than objective translations, yet current systems offer no mechanism to detect or signal when such bias occurs (Rescigno and Monti, 2023). To address this, this thesis deploys a blackbox approach to explore how fine-tuning a pre-trained multilingual BERT model can help detect gender bias in MT outputs. The model takes an input sentence and its corresponding German translation and predicts whether the translation introduces gender bias.

The translation system used is Opus-MT, an open-source neural MT model. It is widely used in research, supports EN-DE translation, and is trained on real-world corpora, making it suitable for studying translation bias (Tiedemann and Thottingal, 2020). Translations are then passed through BERT, trained on a dataset I have constructed by combining and adapting several existing datasets from other researchers. The classifier is lightweight and efficient, aiming for transparent behavior and easy integration into other tools (Devlin et al., 2019). The final tool highlights biased parts in a simple web demo. The goal is not a perfect classifier but a working prototype that shows how such detection could be integrated into translation workflows.

The main research question is therefore: **"How can a NLP-based binary classification model detect gender bias in English-German translations?"**.

1.3 Scope

WRITE AFTER IMPLEMENTATION PART This thesis focuses only on EN-DE MT. Other language pairs are out of scope.

1.4 Limitations

WRITE AFTER IMPLEMENTATION PART It becomes especially difficult to detect when sentences contain multiple subjects, indirect references, or ambiguous pronouns. For example, as Barclay and Sami (2024) explain, the sentence “He went to see her mother” clearly implies three people, while “He went to see his mother” could refer to either two or three. These types of structures introduce ambiguity that makes annotation and evaluation much harder. Creating a dataset that captures such linguistic complexity would require significant effort and careful control of variables. One broader limitation in building datasets for complex scenarios with multiple subjects is the difficulty of isolating the influence of each gendered entity (Lardelli et al., 2024). When working with natural language sources, it becomes hard to tell what caused the bias in the translation. Because of this, the focus of this thesis is on simpler sentence structures with a single subject. This makes it easier to identify and explain bias patterns. It also fits the intended use case: translating business texts like job advertisements or reports, which rarely involve multiple nested clauses or ambiguous pronouns.

1.5 Overview of Chapters

WRITE AFTER IMPLEMENTATION PART

2 Theoretical Background and Related Work

This section outlines key findings of related work on gender bias in MT, with a focus on the English-German (EN-DE) language pair to build the theoretical knowledge base. The research aims are to (1) define the core concept of gender bias in MT, (2) establish the relevance of the topic by reviewing related work, (3) identify the research gap, and (4) justify technical design choices.

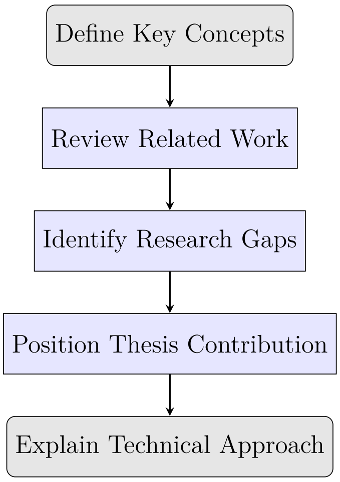


Figure 2.1: Overview of Chapter 2 Structure.

2.1 Definitions

This section explains the key terms and concepts needed to understand gender bias in EN-DE MT. It defines important ideas like Natural Language Processing (NLP), MT, and gender bias. These concepts provide the background necessary to follow the thesis.

2.1.1 Natural Language Processing vs. Machine Translation

NLP refers to the development of machine systems that can process and generate human language. The goal is to mimic and understand it as fluently as possible (Smacchia et al., 2024; Ullmann, 2022). Common applications are chatbots, translation tools, speech recognition, and image captioning.

MT is a direct application of NLP. It is used to automatically translate text from one language to another (Lin and Chien, 2009). MT systems have gone through several stages of development; earlier approaches like rule-based and statistical MT used manually defined grammar rules or pattern matching from large translation corpora (Chakravarthi et al., 2021).

Most modern systems like Google Translate and DeepL, use neural machine translation (NMT) (Y. Wu et al., 2016; DeepL, 2021). These systems are trained on large sets of translated texts. They learn to represent the meaning of whole sentences as mathematical structures and generate more fluent and accurate translations. Unlike earlier systems, they aim to consider the full context of a sentence, which helps reduce errors and improves the handling of ambiguous or idiomatic language. In this work, all MT systems mentioned or deployed are neural NMT systems.

2.1.2 Bias in Society and its Manifestations

Bias refers to a tendency to favour or disadvantage certain individuals or groups based on preconceived ideas. It often comes from stereotypes, which are fixed and oversimplified ideas about a group. While stereotypes describe what people think others are like, bias affects how they are treated.

There are many kinds of bias. It can be based on age, disability, gender, ethnicity, religion, or sexual orientation (Ullmann, 2022). These forms of bias often come from cultural and historical beliefs about how people in these groups should behave.

This work focuses on gender bias. It is the most visible form of bias in MT due to how language works. Gendered words, job roles, and grammar patterns can all affect translations and often repeat stereotypes.

Drawing on key studies that examine gender bias in EN-DE MT (Ullmann, 2022; Rescigno and Monti, 2023; Lardelli et al., 2024; Kappl, 2025), such bias typically manifests in the following forms:

Defaulting to Masculine Forms

In both singular and plural contexts, the *generic masculine* refers to the default use of the masculine grammatical gender. For example, the sentence "Die Studenten sind im Hörsaal" (translation: "The students are in the lecture hall") uses the masculine plural form to refer to a group of students regardless of their gender.

It is commonly used in spoken German and other gendered languages (Lardelli et al., 2024; Schmitz, 2022), although research has consistently shown that the generic masculine creates a male bias in mental representations, leading readers or listeners to think more of male than female examples (Sczesny et al., 2016).

Reinforcement of Stereotypes

Gender bias is deeply rooted in traditional views of men's and women's roles at work and at home (Godsil et al., 2016). Even though many of these roles are outdated, they still shape how people judge others' abilities and character. This can lead to correspondence bias, where people assume traits based on behavior or context. These ideas are reinforced by media like TV shows and advertisements, and they can also influence the way language is used and interpreted.

One common result of this is stereotypical job associations. People often link roles like doctors or pilots with he/him pronouns, and roles like nurses or flight attendants with she/her pronouns (Shrestha and Das, 2022). Prates et al. (2019) rates et al. also found clear patterns in how gender is associated with certain traits. Adjectives like "shy," "happy," "kind," and "ashamed" are often linked to women, while words like "arrogant," "cruel," and "guilty" are more often linked to men.

2.1.3 Gender Bias in Machine Translation

In MT, there is no clear definition of what counts as gender biased, nor is there a standard way to decide which features in text indicate it (Barclay and Sami, 2024).

Because of that, this work uses a simple rule-based definition to decide when a translation is considered gender biased:

- A gender-ambiguous subject in the source text is translated with a gendered term, often defaulting to the generic masculine (e.g., doctor → Arzt) or reflecting stereotypical gender roles (e.g., nurse → Krankenschwester).

- A gendered subject in the source text is assigned an incorrect gender in the translation, leading to semantic inconsistency (e.g., my mother is an engineer → meine Mutter ist ein Ingenieur).

This does not mean that all other cases are truly "unbiased". I will refer to anything that does not fall under these two cases as "neutral". This includes, but is not limited to:

- Sentences with no gendered terms, like "The weather is nice".
- Accurate translations of gendered input, like "The woman is a coder" → "Die Frau ist eine Programmiererin".
- The use of gender-fair alternatives (see subsection 2.2.5).

Biased Translation	**Neutral/Fair Translation**
Gender-ambiguous source is translated with a gendered term.	Gender ambiguity is preserved in the translation.
Gendered subject is assigned an incorrect gender.	Gender in the translation matches the gendered subject.
—	Use of gender-fair language alternatives (see subsection 2.2.5).

Table 2.1: Summary of gender bias scenarios in translation (original compilation).

We can observe that the translation errors often stem from two main sources: Model Overamplification and Semantic Bias. Both phenomena interact with skewed training data to produce the biased outputs outlined in Table 2.1.

Model Overamplification occurs during training when the system exaggerates patterns that already exist in its data. If a corpus frequently pairs cooking with women, the model may conclude that cooking is inherently a female activity. This overemphasis can then manifest as the introduction of a gendered term where none existed or as the misassignment of a subject's gender (Ullmann, 2022; Shah et al., 2020).

Semantic Bias arises from the associations the model learns between words and genders. When “he” co-occurs more often with “doctor,” the model will default to the masculine form in translation. Such defaults lead directly to outputs that either introduce an unwarranted gender or assign the wrong gender, as captured in the first two bias criteria.

Both overamplification and semantic bias worsen when training data are imbalanced. Studies report a high frequency of masculine forms in parallel corpora. One analysis found that the German term for “male doctor” appears 38 times more often than its female counterpart. Female and non-binary examples remain far less common in many datasets (Ullmann, 2022; Stanczak and Augenstein, 2021).

Finally, the sheer scale of modern MT training sets makes manual review impossible. When a model is trained on hundreds of billions of tokens, it may unknowingly absorb and replicate harmful or offensive content. This reinforces the patterns that lead to the biased translations, underlining the need for targeted mitigation strategies at both the data and model levels (Ullmann, 2022).

2.2 Related Works

2.2.1 Literature Search Process

For the literature review I combined incremental and conceptual literature review methods, where each source led to the identification of next. Based on this progression, I identified key concepts and used them to organize and interpret the literature, aligning with a conceptual approach. The structure followed the qualitative Information Systems framework by Schryen (2015) and was further informed by Shrestha and Das (2022) and Savoldi, Bastings, et al. (2025), who both conducted systematic reviews on gender bias in ML and MT respectively.

Search Sources and Tools

Sources were primarily searched on Google Scholar and Perplexity, which served as an additional search engine. Prompts and outputs from Perplexity have been saved and are included in the appendix. To organize and manage the collected sources, Zotero was used throughout the process.

Literature Review Framing

To answer the four research aims, I have defined the key concepts in Table 2.2. Key search terms consisted of *gender bias*, *machine translation*, *AI*, *machine learning*, *German*, *stereotypes*, and *detection*, which were combined with *AND/OR*. The focus was on literature published between 2019 and 2025 to maintain relevance and currency, while foundational and definitional works from earlier periods were selectively included. The initial search for the term *gender bias in machine translation* returned over 18,000 results. Through my iterative selection process, this was narrowed down to 34 core sources.

Key Concept	**Description**
Foundations of Gender Bias in Natural Language Processing	Traces early research that identified gender bias in language. Focuses on foundational studies that showed why the issue matters and how later work builds on these findings.
Sources and Manifestations of Bias	Explains how stereotypes shape language and persist over time. Describes how societal bias enters training data, model design, and system feedback. Shows how bias appears in machine translation and everyday language.
Linguistic Challenges in English-German Translations	Explores key grammatical differences between English and German that affect translation. Focuses on how the lack of gender in English and its presence in German can lead to biased outputs.
Mitigation Strategies and Current Limitations	Reviews how current research tries to reduce gender bias in NLP. Highlights what these methods can and cannot do. Helps identify where a classification-based approach could fill gaps and improve bias detection in translations.

Table 2.2: Key concepts relevant to this thesis

Citation Tracking

Backward citation searching involved reviewing references cited by selected papers, prioritizing frequently cited and foundational works relevant to gender bias in MT. Forward citation searching used Google Scholar's "cited by" function to identify newer research citing those key papers. Filtering with specific terms (e.g., *German* and *machine translation*) was applied during forward search to maintain focus. Beyond these systematic methods, I also included supplementary sources when needed while writing. These consist of contextual references, statistics, or secondary citations that support specific points but were not part of the core conceptual or methodological framework. Supplementary sources were defined as materials identified outside the systematic search, such as papers found through backward citations or targeted queries for statistics and news, which provided support for subordinate arguments without being central to the study's theoretical or analytical structure.

Selection Criteria and Screening Process

Titles and abstracts were manually screened to select relevant studies. Inclusion required sources to specifically address gender bias in MT, provide examples or discussions of gender-related errors, or explain the significance of gender bias in this context. Sources also had to be available in full text without access restrictions. Exclusion criteria filtered out studies focusing on general NLP bias without a direct link to MT, non-gender biases, and highly technical papers lacking contribution to the general understanding of gender bias or that did not provide additional knowledge beyond what was already found in previously published papers. Full texts were reviewed after initial screening to confirm relevance and extract insights. Redundant sources not providing new perspectives aligned with the thesis goals were excluded.

2.2.2 Foundational studies

The existence of gender bias in MT is well-documented. First mentions of this issue date back to over a decade ago, having been recognized by a paper by Schiebinger in 2014. Since then, there has been a general increase in research papers focusing on this topic, especially between 2019 and 2023 (Savoldi, Bastings, et al., 2025).

Prates et al. (2019) conducted a large-scale study using Google Translate to translate sentences like "[Gender-neutral pronoun] is an engineer" from twelve gender-neutral languages into English. The results showed a strong bias toward male pronouns, especially in STEM occupations. This could not be explained by real-world labor statistics, pointing instead to imbalances in the system's training data. The study received wide media attention, leading Google to change their translation policy: Google Translate began showing both feminine and masculine forms for ambiguous inputs (Google, 2018) (see Figure 3).

Building on this, Stanovsky et al. (2019) created WinoMT, a benchmark for evaluating gender bias in English-to-multilingual translations. It focused on occupations in contexts designed to challenge stereotypes. The study found that systems were more accurate for stereotypical gender roles but struggled in non-stereotypical cases, confirming the trends observed by Prates et al. Together, these studies helped spark the ongoing research interest in gender bias in MT.

2.2.3 Ongoing Impact of Gender Bias in Machine Translation

Gender bias in MT can lead to representational harm. This happens when certain genders are repeatedly shown in biased or limiting ways through language (Stanczak and Augenstein, 2021). These patterns can then again, enter training data and influence MT systems, and reflect back into society. This creates a regressive feedback loop.

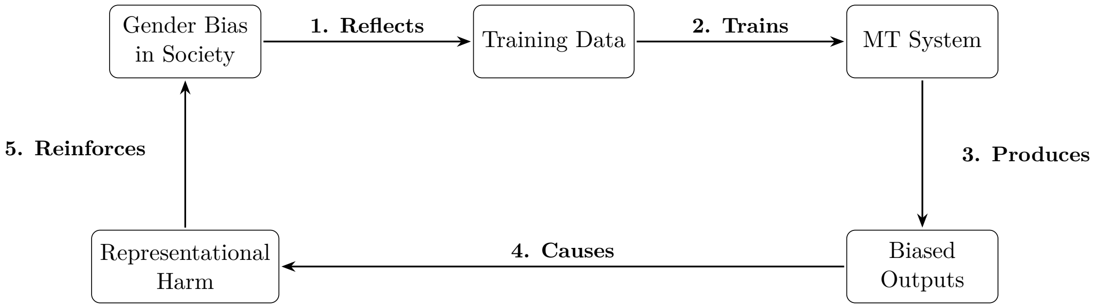


Figure 2.2: Regressive feedback loop of gender bias in MT.

The generic masculine in particular leads to inaccurate and unfair representations of gender in translated text. Rescigno and Monti (2023) observed a predominance of masculine forms in translation outputs (approximately 90% in Google Translate and 85–88% in DeepL for EN-IT and EN-DE), even when the original sentences contained relatively few masculine references. This shows that the bias is not minor but occurs quite heavily in those systems.

It also contributes to the invisibility of women in male-dominated professions (Kappl, 2025). Studies show that biased language in machine-generated text, such as children's stories or job ads, can influence how young people view themselves (Soundararajan and Delany, 2024; Kappl, 2025). It may shape their interests, hobbies, and career choices. This is especially visible in STEM fields (Prates et al., 2019), where stereotypes are more persistent. When job descriptions or mock interviews use gender-exclusive pronouns, women report feeling less belonging, lower motivation, and weaker identification with the role (Godsil et al., 2016). Many self-select out of applying, shrinking the female talent pool and reinforcing gender gaps in the workforce.

Research also shows that using GFL like "she and he" or "one" can improve how women respond to job ads. It reduces stereotype threat and helps them engage more positively with opportunities (Godsil et al., 2016).

Furthermore, a study by Savoldi, Papi, et al. (2024) measured how much effort people need to fix biased translations. They used metrics like the time it took to edit and how many edits were needed, based on human-targeted error rate. The results showed that fixing translations with feminine forms took almost twice as long and required four times more edits than those with masculine forms.

As a result, biased translations lead to higher economic costs and a quality gap that disproportionately affects women. Savoldi, Papi, et al. argued that current automatic bias metrics miss these human impacts. They called for better evaluation methods that reflect what users actually experience.

2.2.4 Linguistic Challenges in English-German Translation

Although both English and German originate from the Indo-European language family (Baldi, 2008), they have different characteristcs. English does not assign grammatical gender to nouns. The article "the" is used universally, independent of what it refers to. On the contrary, German assigns one of three grammatical gendered articles to nouns: "der" (m), "die" (f) and "das" (n). The form or ending of a noun may also change depending on its grammatical gender. While English has a few gendered word pairs, such as "actor" (m) and "actress" (f), gender distinctions in German apply broadly across the entire noun system. "Der Student" refers to a male student, whereas "die Studentin" refers to a female student.

Note that grammatical gender has no connection to societal or biological gender. It is a rule of the language rather than a reflection of identity. For example, the German word Mädchen (girl) is grammatically neuter and takes the article "das". This is not because the referent lacks gender, but because the suffix "-chen" automatically assigns neuter gender. Grammatical gender in German follows structural rules, even when they contradict real-world gender associations.

2.2.5 German Gender-Fair Language

Gender-fair language (GFL) refers to the use of language that treats all genders equally and aims to reduce stereotyping and discrimination (Sczesny et al., 2016). Three common approaches to plural mentionings in German are:

- **Gender-neutral rewording:** This uses neutral terms instead of gendered nouns, e.g., *die Studierenden lernen.* A challenge for this version is that neutral alternatives do not exist for every noun and cannot be consistently applied (Lardelli et al., 2024).

- **Gender-inclusive characters:** This combines masculine, feminine and non-binary forms by using a character like ***, *:*, or _, e.g., *die Student*innen lernen.* This method is consistent but may interrupt reading flow and lacks standardization (Lardelli et al., 2024).

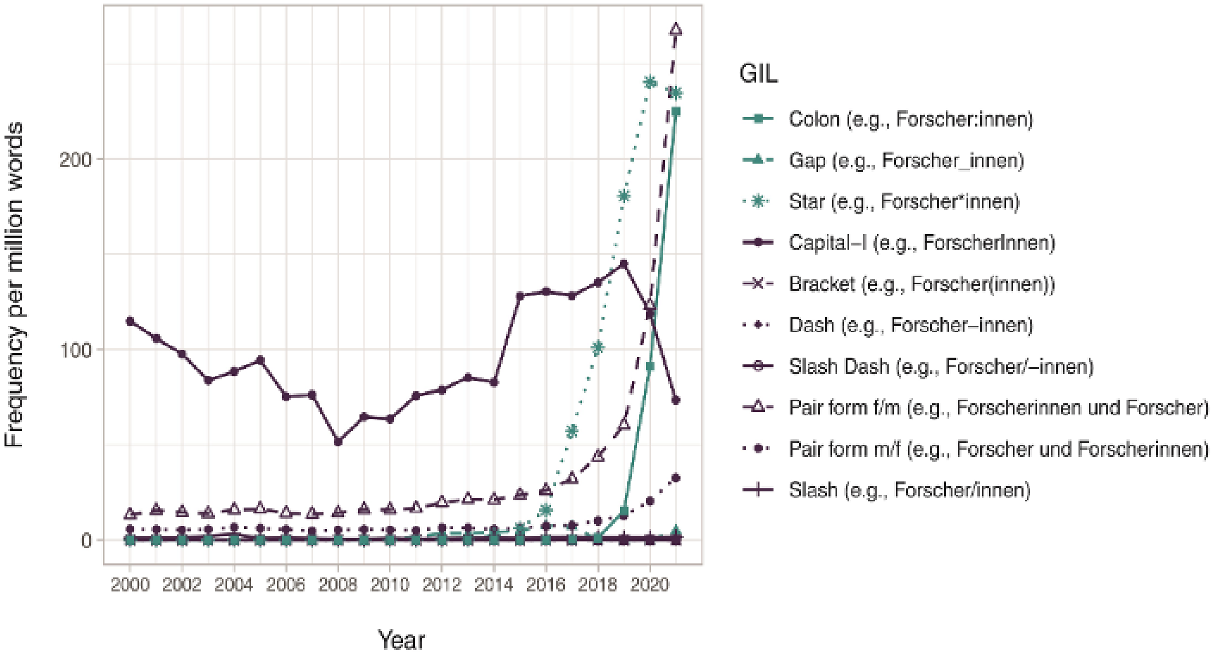


Figure 2.3: Frequency of different types of gender-inclusive language. Source: Waldendorf (2024) p. 367.

- **Pair form:** This names both gender forms, e.g., *die Studentinnen und Studenten lernen.* It is currently the most used GFL form in German (Waldendorf, 2024), briefly surpassing the star and colon characters as seen in Figure 2.3.

These examples apply when the gender of the subjects is ambiguous. But when gender is known, especially in singular mentions, the generic masculine should be avoided. However, in the same way as gender bias has no clear definition, there is no agreed standard for GFL (Lardelli et al., 2024; Savoldi, Bastings, et al., 2025). "Fairness" therefore heavily depends on personal views, culture, and context, which raises ethical questions about debiasing systems.

Challenge of Integrating Gender-Fair Language into NLP

Although the use of GFL has increased in recent years (Waldendorf, 2024), it is still generally low. This leads to a scarcity of relevant linguistic data. Few datasets include GFL variants, and existing resources often depend on manual translations or post-editing to add gender-inclusive forms (Lardelli et al., 2024). For this project, the limited availability of GFL data poses a significant challenge, especially when training the model to recognize gender-fair alternatives as neutral due to the lack of consistent examples.

2.2.6 Research Gaps

A central gap in gender bias research is the absence of a shared definition of what constitutes "fair" language. This lack of conceptual clarity makes it difficult to design systematic evaluation approaches, define accountability standards, or detect all relevant forms of harm (Barclay and Sami, 2024; Shrestha and Das, 2022; Stanczak and Augenstein, 2021).

A second major gap concerns the availability of high-quality EN-DE translation data that includes GFL. While a few datasets exist, they are not designed for bias detection tasks and often require manual post-editing to incorporate inclusive forms (Lardelli et al., 2024). This limits the development and evaluation of models that aim to identify biased output in a structured and reproducible way.

As Stanczak and Augenstein (2021) point out, findings on gender bias in English do not necessarily transfer to other languages such as German. This underlines the importance of typological diversity and language-specific solutions when addressing fairness in MT. Existing studies on EN-DE (Ullmann, 2022; Kappl, 2025; Lardelli et al., 2024) systems mostly confirm the presence of gender bias, propose mitigation strategies, or introduce evaluation metrics. However, few provide methods for systematically detecting bias in translated text.

This project addresses that gap by focusing on bias detection as a foundational step. To support this, I combine existing datasets from previous research to create a new dataset specifically for detecting gender bias in EN-DE translations. Given the lack of suitable data and tools, detection is a necessary starting point. Manual correction will likely remain necessary, as current research does not yet support reliable automatic debiasing. This project therefore focuses on flagging biased translations, not correcting them.

2.3 Approach and Justification of the Technical Setup

This section outlines the technical setup used in the project and explains the rationale behind design choices. It also provides background information on the underlying technologies to clarify how each component contributes to the overall goal of detecting gender bias in EN-DE translations.

2.3.1 Binary Classification in NLP

Binary classification means sorting items into two clear groups. It is the most common task in ML and is frequently found in every day life, such as automatically flitering e-mails as "spam" or "not spam" (Quemy, 2019) or deciding whether a transaction is "fraudulent" or "legitimate". For instance, a spam filter uses previously labeled e-mails to learn relevant patterns, such as specific keywords or sender information, and builds a model that applies these patterns to classify new messages accurately.

This thesis tries to label a translation as either "biased" or "neutral". While it is possible to extend the classification beyond two categories, such as distinguishing types of bias or including labels like "gender-fair", this would require much more data and training. Given the practical aim of this work, which is to help users quickly identify whether their text might contain gender bias, the model focuses on a simple binary decision.

2.3.2 Transformer Architecture

To provide some background on the BERT architecture, it is important to understand its foundation in the Transformer model. The Transformer is designed to process input sequences and *transform* them into output sequences. To do this effectively, it uses a self-attention mechanism (Phuong and Hutter, 2022).

Self-attention mechanism

The self-attention mechanism allows the model to weigh the significance of all input elements simultaneously (Xiao and Zhu, 2023), meaning it can look at all words in a sentence at once and decide which ones are most relevant to each word. Unlike traditional methods like Recurrent Neural Networks (RNNs), which process input step by step, self-attention captures global dependencies and contextual relationships more accurately, creating "context-aware" representations.

Encoder-Decoder Framework

The transformer architecture consists of two main components: the encoder and the decoder. The encoder's job is to read the input sentence and turn it into a series of vectors the model can understand. Each vector is a list of numbers representing the meaning and structure of each word (Xiao and Zhu, 2023). The encoder works as follows (see Figure 2.4):

1. It receives input embeddings, which represent the words, and positional encodings, which tell the model the order of the words.

2. The data then passes through several identical layers. Each layer has two main parts:

a. **Multi-head self-attention** runs several attention processes in parallel. Each attention head focuses on different details to help the model understand the sentence better.

b. A **Feed-forward network** processes each word vector separately, refining the information like a small filter.

c. **Add & Layer Norm** combines a shortcut connection (Add) and normalization (Layer Norm). The Add step passes the original input forward to keep useful information. Layer Norm adjusts the output values to a stable range.

3. Each layer builds on the output of the previous one, helping the model form more complex and abstract ideas about the input sentence.

4. Finally, the encoder outputs a sequence of *hidden states.* These are continuous vector representations for each input token. They encode contextual information from the entire sentence. For example, in the sentence "The cat sat on the mat," the vector for "cat" reflects its relationship to words like "sat" and "mat."

The decoder generates the output sentence one word at a time by using the information from the encoder (Xiao and Zhu, 2023). However, since BERT uses only an encoder-only architecture (see Figure 2.5), the decoder is not relevant for this work and is therefore excluded from the discussion.

2.3.3 BERT

BERT is a language model that stands for "Bidirectional Encoder Representations from Transformers" and was introduced by Google in 2018 (Devlin et al., 2019). After pre-training, BERT can be adapted to many NLP tasks by adding a simple output layer and fine-tuning, without needing major changes to its design. Since BERT uses only the encoder part of the Transformer architecture, it is designed to understand input rather than generate output. This makes it especially suitable for a binary classification task, where the goal is to analyze input texts and assign it to one of two categories.

There are multiple variants of the original BERT model. It was originally released in two sizes: `BERT-Base` and `BERT-Large`, which differ in the number of layers, attention heads, and overall model capacity (Devlin et al., 2019). Since then, many other versions have been developed. Most of them modify either BERT's pre-training objectives or the underlying Transformer architecture (Libovický et al., 2019).

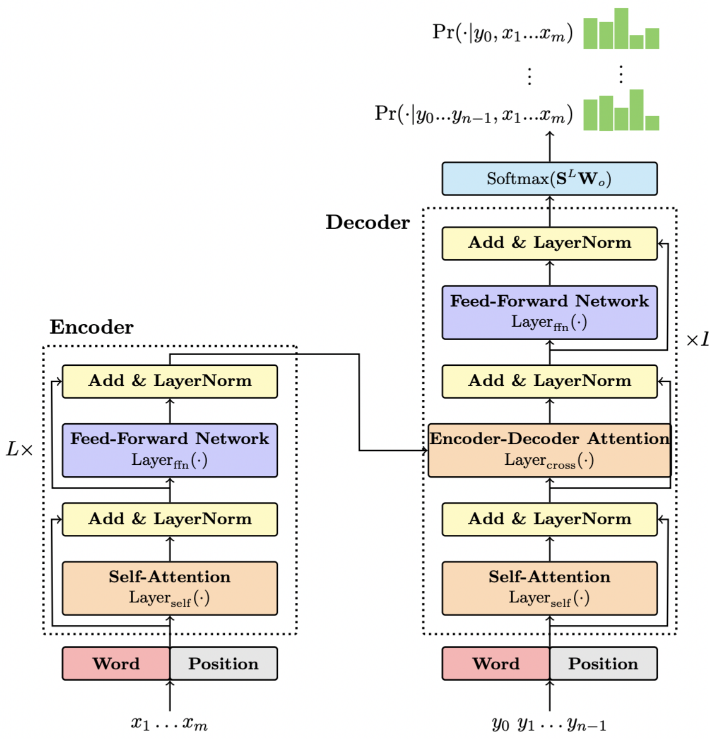

$\Pr(\cdot|y_0, x_1...x_m)$
$\Pr(\cdot|y_0...y_{n-1}, x_1...x_m)$
Softmax($\mathbf{S}^L\mathbf{W}_o$)
Decoder
Encoder
Add & LayerNorm
Feed-Forward Network
$\text{Layer}_{\text{ffn}}(\cdot)$
Add & LayerNorm
Self-Attention
$\text{Layer}_{\text{self}}(\cdot)$
Encoder-Decoder Attention
$\text{Layer}_{\text{cross}}(\cdot)$
$L\times$
$\times l$
Word
Position
$x_1 \dots x_m$
$y_0\ y_1 \dots y_{n-1}$

Figure 2.4: Transformer encoder-decoder architecture. The encoder (left) processes input tokens $x_1, \dots, x_m$ through: (1) a self-attention layer for contextual relationships, (2) a feed-forward network for feature transformation, and (3) residual connections with layer normalization. The decoder (right) generates outputs by attending to both the encoder's representations and its previous outputs (y_0 to y_{n-1}), producing the next-token probability distribution. Figure and description adapted from Xiao and Zhu (2023), p. 6.

2.3.4 Multilingual BERT (mBERT)

For this thesis, I use multilingual BERT (mBERT) (Devlin et al., 2019). mBERT uses the same configuration as BERT-Base, but it is pretrained on Wikipedia data from 104 languages, including both English and German. There is no explicit indication of the input language, nor is there a training objective that aligns languages bilingually. Instead, multilingual capabilities emerge naturally from its large multilingual text corpus (Pires et al., 2019).

Monolingual models like German BERT do not support English input. Larger multilingual models, such as XLM-RoBERTa, require more computational resources and training time, which was not feasible here. mBERT offers a good balance between language coverage, model size, and training efficiency, making it a practical choice detecting gender bias in EN-DE translations.

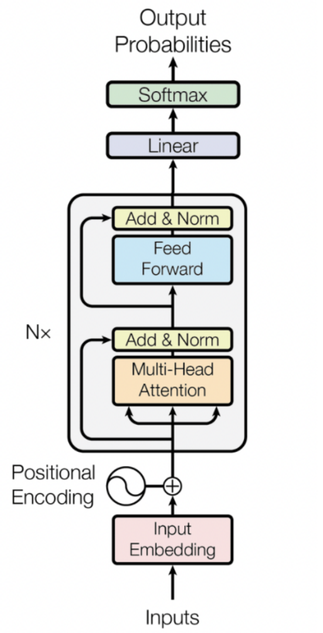


Figure 2.5: BERT's encoder-only architecture. Figure by Smith (2024).

Tokenization

`mBERT` processes input by splitting words or subword units into *tokens* (tokenization)[1]. It uses the WordPiece algorithm with a shared vocabulary of 110,000 tokens, and all texts are lowercased before tokenization (Devlin, 2018). To balance the training data, languages with large Wikipedia corpora are downsampled, while those with fewer resources are oversampled.

Pre-processing is the same for all supported languages: (1) converting text to lowercase and removing accents, (2) splitting punctuation, and (3) tokenizing based on whitespace. Removing accents helps reduce the vocabulary size, even though it can introduce ambiguity in languages where accents carry meaning. This trade-off is accepted because `mBERT's` contextual embeddings usually resolve such ambiguities during training and inference.

Special tokens are reserved tokens added to in. They indicate boundaries or roles, helping the model distinguish parts of the text and process it correctly.

- `[CLS]` (classification) marks the start of the sequence,
- `[SEP]` separates sentence pairs.

In this work, each input combines an English source sentence and its German translation as:

[1] This tokenization process applies to both BERT and mBERT.

```
[CLS] english sentence [SEP] german translation [SEP]

[CLS] the nurse is kind [SEP] die krankenschwester ist nett [SEP]
```

Mechanics of Fine-Tuning mBERT

Fine-tuning adjusts the base model for a specific task, in this case, detecting gender bias in translations.[2] To do so, a new labeled dataset is used to continue training the model, allowing it to adapt its weights to task-specific patterns.

A classification head, comprising a linear layer followed by a softmax function (see Figure 2.5), is added on top of `mBERT's` output. The linear layer applies a learned transformation to the final hidden state vector of the `[CLS]` token.

$$z = Wx + b$$

Here, x is the `[CLS]` embedding, W is the weight matrix, and b is the bias vector. Both W and b are parameters learned during training to help map `mBERT's` output to the task labels. This changes the output into two numbers (logits), one for each class: biased or neutral. Then, the softmax function turns these numbers into probabilities (Devlin et al., 2019; Xiao and Zhu, 2023). Short for "soft maximum," it maps raw scores to a probability distribution, emphasizing the highest values while still giving smaller ones some weight.

$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

Each logit z_i is exponentiated to ensure positivity. The result is then normalized by dividing by the sum of all exponentials, producing the probability distributions. K is the number of possible classes. The class with the highest probability is selected as the model's prediction.

Key Hyperparameters Explained

Fine-tuning can be unstable, and changes such as different seeds can lead to large differences in task performance (Mosbach et al., 2021). Tuning a set of key hyperparameters is therefore necessary. These are not learned by the model but must be set manually or through experimentation. Their values affect how fast the model learns, how stable training is, and how well the model generalizes to new data.

[2] This fine-tuning process applies to both BERT and mBERT.

The *learning rate* controls how much the model updates its weights during each step (Mosbach et al., 2021). If it is too high, the model may not converge and instead jump over good solutions. If it is too low, training can be very slow or get stuck in local minima.

Warmup steps are used at the beginning of training to gradually increase the learning rate from zero to its target value (Mosbach et al., 2021). This helps avoid instability in the early stages, where large updates can be harmful. After the warmup period, the learning rate is often decreased again using a scheduler, which controls how it changes over time.

The *number of epochs* defines how many times the model passes through the entire training dataset (Mosbach et al., 2021). More epochs mean more training iterations, which can help the model better fit the data. On small datasets, training for more epochs—sometimes up to 20 instead of the usual 3—helps reduce instability and improves generalization. This is because the model has more chances to learn meaningful patterns instead of stopping too early.

The *batch size* refers to how many training examples the model processes before updating its parameters (Mosbach et al., 2021). Commonly, a batch size of 16 is used during fine-tuning `mBERT`. Larger batches provide more stable gradient estimates but require more memory. Smaller batches can introduce noise in the updates but might help the model generalize better. While Mosbach et al. (2021) does not deeply analyze batch size effects on stability, it remains an important parameter to balance resource limits and training quality.

Finally, the *optimizer* controls how the model weights are adjusted to minimize prediction error (Mosbach et al., 2021). The AdamW optimizer is standard for `mBERT` fine-tuning because it adapts learning rates per parameter and includes weight decay regularization. A critical feature of Adam is *bias correction*, which reduces the effective learning rate early in training. This acts like an implicit warmup, preventing large unstable updates and vanishing gradients in the lower layers. Combining explicit warmup with Adam's bias correction allows training with higher learning rates more stably.

Layer Freezing During Fine-Tuning

Layer freezing refers to the practice of keeping certain layers of a pretrained model fixed during fine-tuning, meaning their weights are not updated. This approach reduces the number of trainable parameters. This not only speeds up training (Sorrenti et al., 2023) but also helps prevent overfitting on small datasets and preserves the broad language knowledge from pre-training.

In monolingual BERT, lower layers typically encode general syntactic and semantic patterns, while higher layers are more task-specific (Nadipalli, 2025). As a result, it is common to freeze the lower layers and only fine-tune the top layers and the classification head, especially in resource-constrained settings (Nadipalli, 2025).

In `mBERT`, the distribution of cross-lingual and language-specific features across all layers makes layer freezing less straightforward. S. Wu and Dredze (2019) highlight that no single layer consistently captures the most relevant cross-lingual information, and even individual layers can perform well on sentence-level tasks. They suggest that freezing the lower six layers may improve generalization, but emphasize that optimal strategies depend on the specific task and require empirical testing (S. Wu and Dredze, 2019).

Limitations of mBERT

One major limitation of `mBERT` is the "curse of multilinguality" (Gurgurov et al., 2024). Because it must represent 104 languages within a fixed parameter budget, the capacity available per language is limited. This causes reduced performance across languages compared to monolingual models. Even high-resource languages like English perform worse in `mBERT` than in their dedicated BERT models. Additionally, the shared vocabulary of 110,000 tokens is diluted, meaning it is less tailored to any single language. Languages with more data tend to get better performance, while others suffer.

Since `mBERT` is pretrained on Wikipedia, it reflects biases inherent to that corpus. German Wikipedia articles predominantly use the generic masculine (Sichler and Prommer, 2014), while gender-fair alternatives appear only sporadically, mostly in discussions or articles about female-dominated professions. These biases can influence the model's outputs and are especially important to consider in a gender bias detection context.

Despite these limitations, `mBERT` remains the most fitting choice for this thesis. Since I work with English and German, which are both high-resource and related languages, `mBERT` generally performs better than it would with low-resource languages or languages from distant language families with fewer similarities (Lauscher et al., 2020).

2.3.5 Chosen Evaluation Metrics

Evaluation metrics quantify how effectively the model identifies gender bias. They provide objective measures to assess and compare performance.

In this task, two types of errors are especially important to avoid: false positives, where neutral translations are incorrectly flagged as biased, and false negatives, where actual bias is missed. A model that only guesses or always plays it safe is not useful.

The metrics that capture these errors are precision and recall (Rainio et al., 2024):

- **Precision:** Of all translations flagged as biased, how many truly are biased? High precision means fewer false alarms.
- **Recall:** Of all biased translations, how many did the model correctly detect? High recall means fewer missed biases.

There is often a trade-off between precision and recall. A model with high precision but low recall misses many real biases, while one with high recall but low precision raises too many false warnings. To balance this trade-off, the F1 score is used. It combines precision and recall into a single number by calculating their harmonic mean:

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

While commonly used, accuracy (the overall percentage of correct predictions) is a less suitable metric for this task. Even with a balanced dataset, a model could achieve high accuracy by disproportionately favoring one class, thereby failing at the primary goal of reliably detecting bias.

2.3.6 Interactive Demo

The fine-tuned model is intended to be presented through an interactive demonstration. Since the focus lies on showcasing the model's functionality rather than creating a fully developed application, Streamlit was chosen. Streamlit allows for quick and easy development of lightweight user interfaces in Python, providing a simple setup and effective performance. For live translation, an open-source tool supporting EN-DE pairs was required. Opus-MT (Tiedemann and Thottingal, 2020) meets these criteria and integrates smoothly into the demonstration. While state-of-the-art translators like Google Translate or DeepL would have been preferred for their quality, they do not meet the requirements for this setup. Therefore, a separate tab for manual translation input was added, allowing users to paste translations directly and bypass this limitation.

3 Methodology

This chapter explains the overall approach and structure of the project. It covers how data is handled, how the model is built and trained, and how the demo application is designed.

3.1 Goal of the project

The goal is to build a gender bias detection model for real-world MT scenarios. This includes cases like translating everyday sentences or job descriptions. The focus is to flag bias at the sentence level, so users do not have to find the specific sentences causing bias themselves.

Thus, the model processes each sentence independently. If multiple sentences are inputted, bias is evaluated for each one separately. Context across sentences is not considered, as it does not reflect the intended use case. This approach is also reflected in the design of the training data, where each sentence pair is treated as a standalone instance.

3.2 Workflow

The project begins by selecting and combining datasets from previous work (see Figure 3.1). The model building phase then follows, as shown in the purple boxes. It starts with cleaning and preparing the data, followed by extracting features for training. A pre-trained multilingual BERT model is then fine-tuned for the classification task. Its performance is measured using standard evaluation metrics. In the final step, the trained model is integrated into the demo application.

3.3 Dataset Handling

Since there was no ready-to-use dataset for this task and no prior work that built a similar model, I first had to define: **(1)** the number of samples required, and **(2)** the desired content of my dataset.

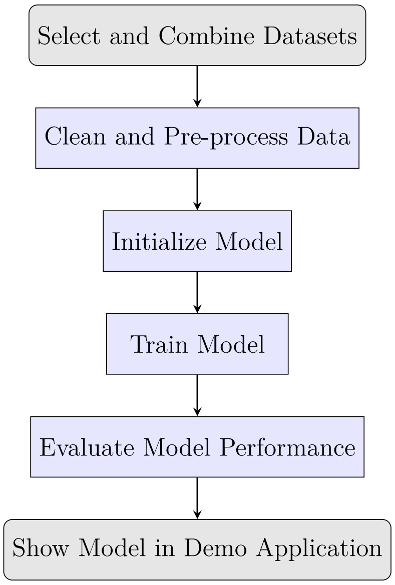

Select and Combine Datasets
Clean and Pre-process Data
Initialize Model
Train Model
Evaluate Model Performance
Show Model in Demo Application

Figure 3.1: Workflow of the project.

3.3.1 Number of Samples

For a binary classification task of detecting gender bias using mBERT, general guidelines suggest between 100 and 5000 labeled samples for fine-tuning (Pecher et al., 2024), while multi-class tasks need fewer samples (around 100). However, the complex nature of gender bias often requires a larger dataset for robust detection since the number of samples depends mainly on the task type.

3.3.2 Dataset Composition

Ideally, I wanted to make use of past EN-DE datasets to minimize manual labour. My options were mGeNTE en-de (Savoldi, Cupin, et al., 2025), Building Bridges Dictionary (Lardelli et al., 2024), and Translated Wikipedia Biographies (Stella et al., 2021).

However, While analyzing the Translated Wikipedia Biographies dataset, I found issues that prevented automatic reuse. For example, the perceivedGender column sometimes contained subject names instead of expected labels like Male, Female, or Neutral, which would require manual review. Moreover, the dataset only provided neutral labels (0) since the phrases were correctly gendered. Because my other two datasets are already balanced and include enough neutrally gendered examples, I decided to exclude this dataset.

mGeNTE contains naturally occurring sentences with gendered entities, while Building Bridges focuses on German GFL entries for explicitly gendered nouns such as professions. However, this setup lacked genuinely neutral examples; sentences that do not involve any gendered subject at all, such as *"The weather is nice"* or *"How are you"*. Including such sentences is important to help the model learn that not all translations are relevant for gender bias detection and that many ordinary sentences should be classified as neutral.

`Tatoeba.` Since no suitable dataset for this category was readily available, a supplementary set was created using random EN-DE sentence pairs from the `Tatoeba` corpus. A sample of 550 sentence pairs was selected. Manual filtering was applied to remove any pairs with incorrect or stereotypically gendered translations, as public contributions often default to male forms. The resulting subset consisted of 532 clearly neutral sentence pairs, all labeled with 0. [1]

Dataset	**Description**	**Content**
`mGeNTE en-de` (Savoldi, Cupin, et al., 2025)	Multilingual dataset to assess gender bias in MT.	~1,500 gender-ambiguous and gendered English sentences with gender-neutral and gendered German translations.
`Building Bridges Dictionary` (Lardelli et al., 2024)	Bilingual dictionary designed to support gender-fair EN-DE translation.	~230 German gender-neutral and gender-inclusive singular and plural sentences with English equivalents.

Table 3.1: Overview of suitable EN-DE datasets based on past works.

`mGeNTE en-de.` The mGeNTE dataset contained the following relevant information:

- `SET-G`: English sentences with a clearly gendered subject.
- `SET-N`: English sentences with neutral or ambiguous subject gender.
- `REF-G`: German translations that preserve or introduce gender.
- `REF-N`: German translations that are fully gender–neutral.

The bias definition used in this study classifies translations that omit the original gender as neutral since it does not rely on a male default or stereotype. While gender-neutral

[1] The transformed dataset can be found in `/datasets/deu_final.csv`.

translations may be imperfect, they are not considered "biased" under this framework. Initial experiments showed that including `REF-N` pairs during training overly penalized neutral outputs. Given the scarcity of neutral examples, I chose not to penalize neutral translations.

Each original entry was split into two paired examples and labeled as follows:

$$\begin{aligned}
\texttt{SET-G} + \texttt{REF-G} &\rightarrow 0 \quad \text{(neutral)}\\
\texttt{SET-G} + \texttt{REF-N} &\rightarrow 0 \quad \text{(neutral)}\\
\texttt{SET-N} + \texttt{REF-N} &\rightarrow 0 \quad \text{(neutral)}\\
\texttt{SET-N} + \texttt{REF-G} &\rightarrow 1 \quad \text{(biased)}
\end{aligned}$$

This procedure yields 3,000 total instances, of which 750 are labeled biased (1) and 2,250 are labeled neutral (0). [2]

`Building Bridges Dictionary.` This dataset did not contain full sentences but rather a gender-fair dictionary of nouns. While this made it a valuable resource for studying gender-fair language, I needed full sentences for my task. To address this, I used prompt engineering with Google Gemini 2.5 Flash to synthetically expand the dataset. The prompt used for generation is included in the appendix. The generated sentences can be found in the code files. I used the nouns from the original dataset to create multiple grammatically correct sentence variations, covering singular, plural, gender-neutral, and gender-inclusive forms. The dataset uses the star form (e.g., *Lehrer***innen*) as its gender-inclusive format. Since the colon form (e.g., *Lehrer:innen*) is also widely used, I added it manually via a script. The script duplicated all entries with stars and replaced the star with a colon to create additional variants.

This resulted in 3,381 total entries: 2,001 labeled as 0 (neutral) and 1,380 labeled as 1 (biased). [3]

3.3.3 Available Data Summary

Table 3.2 shows an overview of the labeled data from the three available sources.

[2] The transformed dataset can be found in `/datasets/mgente_final.csv`.

[3] The transformed dataset can be found in `/datasets/lardelli_final.csv`.

Dataset	Total	Neutral (0)	Biased (1)
`lardelli_final.csv`	3381	2001	1380
`mgente_final.csv`	3000	2250	750
`deu_final.csv`	532	532	0

Table 3.2: Summary of available labeled examples

The number of samples selected from each dataset was determined through iterative testing. Multiple dataset variants were created by upsampling or downsampling specific groups. The documentation of this process is discussed in subsection 3.7.3.

3.4 Data Pre-processing

I load and split the dataset into three parts: training (80%), validation (10%), and test sets (10%). This ratio is a common choice when working with limited data. It provides enough samples for the model to learn general patterns while reserving separate subsets for tuning and final evaluation. I used stratified sampling for this. That means the label distribution (biased vs. neutral) stays the same across all three sets, avoiding skewed splits. For example, if 30% of the full dataset is biased, each split will also have 30% biased samples. This helps the model learn from both classes equally and avoids misleading results in validation or testing.

I did not apply advanced text cleaning such as removing punctuation, lowercasing, or stemming because I use `bert-base-multilingual-cased`. This tokenizer is designed to handle raw, unmodified text and preserves casing. The model has been trained on large corpora containing natural language in its original form (Devlin et al., 2019). Altering the input, such as converting "Doctor" to "doctor", could remove distinctions the model has learned. Therefore, advanced preprocessing might reduce performance by disrupting patterns the tokenizer expects.

Further data preprocessing was not necessary since I concatenated already clean datasets and manually reviewed them.

3.5 Model Initialization

`mBERT` with a binary classification head is used to predict whether a translation is *biased* or *neutral*.

The tokenizer from the same model source encodes input pairs into token IDs and applies segment embeddings to distinguish between source and target sentences. All sequences are padded or truncated to a fixed length of 256 tokens. This length was chosen after tests with 128 tokens led to truncation warnings and content loss. With 256 tokens, most pairs remained complete while keeping memory use efficient.

Input features and labels (0 for neutral, 1 for biased) are converted to PyTorch tensors for training. The model runs on GPU if available to speed up processing. For classification, the output vector of the [CLS] token is used as a representation of the full input.

3.6 Training Pipeline

3.6.1 Fine-Tuning with the Trainer API.

I fine-tuned the model using the Hugging Face `Trainer` API. This library takes care of most of the training process. It handles evaluation, logging, and saving model checkpoints. It also supports automatic early stopping and loading the best model at the end.

Compared to writing a manual PyTorch training loop, `Trainer` is easier to use and requires less code. It also includes standard features like tracking metrics and scheduling learning rates. For my use case, this made it a better choice. I did not need custom loss functions or multi-GPU setups, so the added complexity of other frameworks (like PyTorch Lightning) was not necessary.

3.6.2 Layer Freezing.

I fine-tuned only the last two encoder layers (10 and 11), the pooler, and the classifier, about 25-30% of the model. Later layers specialize in the fine-tuning task by focusing on features relevant to it. Freezing earlier layers keeps the general language knowledge unchanged. This reduces overfitting by limiting changes to the most task-relevant parts and speeds up training because fewer parameters are updated (Nadipalli, 2025). This is helpful since the task is very specific and resources are limited.

3.6.3 Hyperparameters

I tested a few different settings but mostly kept values close to common recommendations for fine-tuning BERT models.

Epochs. I trained the model for up to 8 epochs, with early stopping enabled using a patience of two epochs. This means that training stopped if the F1 score did not improve for two consecutive epochs. The setup follows the recommendation by Pecher et al. (2024), who advise training until convergence with a maximum of 10 epochs and early stopping. In my case, the validation loss typically began to increase after 8 epochs, with no further gains in performance. Reducing the maximum to 8 epochs helped avoid overfitting and shortened training time.

Batch size. I used a batch size of 16. This is a common choice for fine-tuning on small datasets. It keeps GPU memory use low while still allowing stable gradient updates. Mosbach et al. (2021) also use this value when testing fine-tuning stability.

Learning rate. I set the learning rate to 2e-5. This is the default used in the original BERT paper (Devlin et al., 2019) and still works well in many cases. I also tested 1e-5 and 3e-5. Both gave worse results on the validation set. For example, training with 3e-5 led to sharp drops in F1 after the second epoch. The 2e-5 rate gave the most stable and consistent results in my runs.

Optimizer and scheduler. The `Trainer` API uses the AdamW optimizer by default. An optimizer is an algorithm that adjusts the model's parameters during training to minimize errors (Mosbach et al., 2021). AdamW is a popular choice for fine-tuning BERT because it includes weight decay, which helps prevent overfitting. It also uses bias correction to improve stability early in training, avoiding issues like vanishing gradients.

A scheduler controls how the learning rate changes during training (Mosbach et al., 2021). This affects how big each step is when updating parameters. The scheduler here uses a warmup-linear schedule: the learning rate starts low and increases gradually during the first 10% of training (warmup), then decreases linearly until the end. This helps the model learn smoothly and prevents unstable training.

3.7 Evaluation

3.7.1 Evaluation Strategy

The model was evaluated during and after training using the validation and test splits. As described in the data section, both sets were created using stratified sampling to preserve the label distribution. This ensured consistent evaluation across all splits and avoided misleading metrics due to imbalance.

During training, the model was evaluated on the validation set after every epoch. The F1 score was used to monitor progress and decide when to stop. The checkpoint with the highest validation F1 score was saved as the final model. This strategy helped avoid overfitting and improved generalization.

After training, I evaluated the model on two types of data:

- A small handcrafted test set that includes common bias patterns seen during training. This tested whether the model correctly learned to flag known types of bias.
- A set of real-world sentences taken from external sources like job ads. This tested how well the model handles unseen natural language and generalizes beyond training data.

3.7.2 Why F1 Score Was Used

In this task, both false positives (flagging neutral translations as biased) and false negatives (missing actual bias) are problematic. A model that just plays it safe or guesses randomly is not useful. The F1 score balances precision and recall, which are the two metrics most relevant here.

- **Precision:** Of the translations flagged as biased, how many were actually biased?
- **Recall:** Of the biased translations, how many did the model catch?

If the model has high precision but low recall, it avoids false accusations but misses many real issues. If it has high recall but low precision, it catches more bias but floods users with false warnings.

To balance these trade-offs, the F1 score is used. It is the harmonic mean of precision and recall:

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Accuracy, by contrast, is not helpful here. With a 50-50 dataset, a model could reach high accuracy by only learning to predict the majority class. That would ignore the actual task of bias detection.

The use of the F1 score ties back to the model design: sentence-level binary classification with a balanced dataset and no cross-sentence context. In this setup, F1 gives the clearest picture of how well the model performs in practice.

3.7.3 Dataset Sampling Configurations and Results

3.8 Demo Application Design

The demo application comprises three modules: the Streamlit interface, the bias detection model with its prediction functions, and the translation component. Figure 3.2 illustrates the workflow. Both input modes converge on a common prediction pipeline.

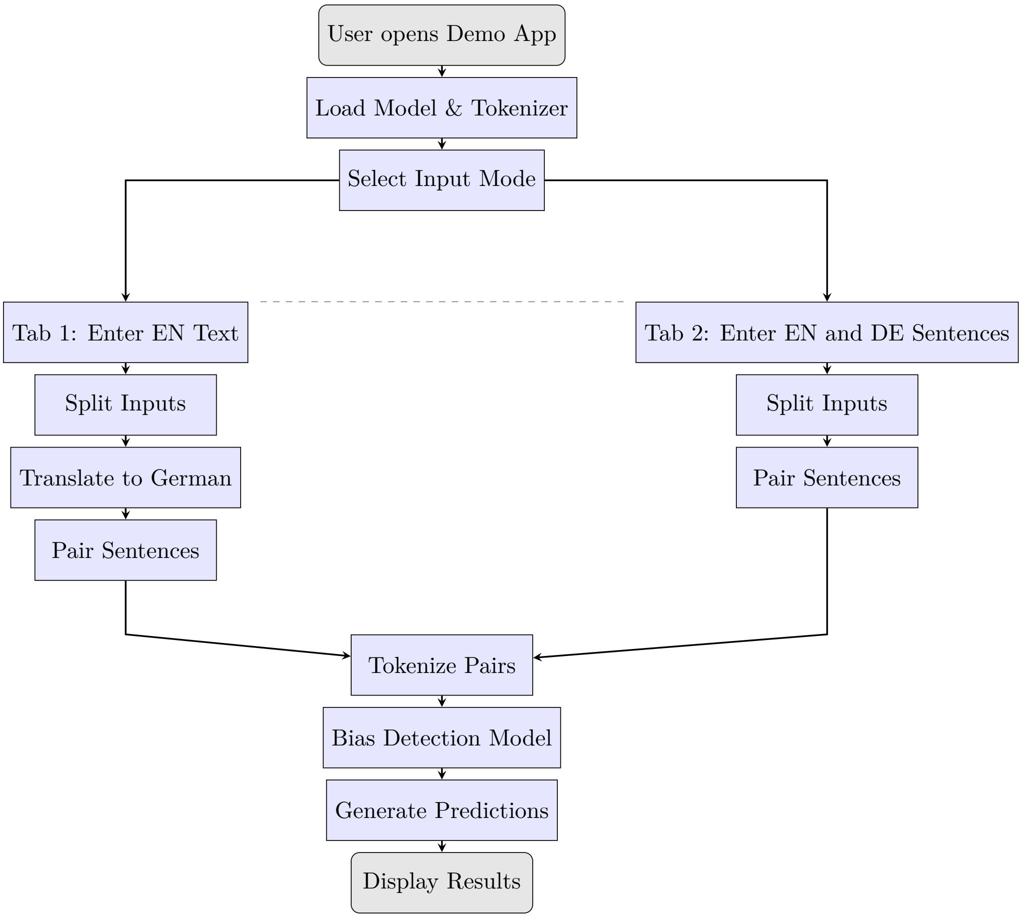


Figure 3.2: Workflow of the bias detection application. Automatic translation and manual sentence pairs follow the same prediction steps.

Initialization. On launch, the application loads the fine-tuned BERT bias detection model and its tokenizer from a local directory. These objects are cached in memory to avoid repeated loading. The model and tokenizer move to the available device, either CPU or GPU.

User Interface. The interface has two tabs:

1. **Automatic Translation.** The user inputs raw English text. The application splits the text into sentences and sends them in batches to the translation module. The module returns German translations. Each English sentence is paired with its translation before bias analysis.

2. **Manual Pairing.** The user supplies parallel English–German sentence pairs. After splitting and pairing, the application bypasses translation and proceeds directly to bias analysis.

Bias Detection Pipeline. Sentence pairs are processed in batches to reduce overhead and speed up analysis. Each sentence pair is first encoded with the tokenizer of the fine-tuned BERT model. The encoded input is then processed by the trained classifier, which produces raw bias scores. These scores are normalized into probabilities using softmax. The bias label corresponding to the highest probability is selected, and its confidence value is reported.

Results Presentation. The application displays a table of results. Each row contains:

- Original English sentence
- Corresponding German sentence
- Bias prediction (binary)
- Confidence score

4 Implementation

4.1 Project Structure

4.2 Environment Setup

4.3 Core components and their interaction

4.4 Demo Result

Bibliography

Baldi, P. (July 2008). “English as an Indo-European Language”. In: *A Companion to the History of the English Language*. Ed. by H. Momma and M. Matto. 1st ed. Wiley, pp. 127–141. ISBN: 978-1-4051-2992-3 978-1-4443-0285-1. DOI: 10.1002/9781444302851.ch12. (Visited on 06/06/2025).

Barclay, P. J. and A. Sami (Apr. 2024). *Investigating Markers and Drivers of Gender Bias in Machine Translations*. DOI: 10.48550/arXiv.2403.11896. arXiv: 2403.11896 [cs]. (Visited on 05/21/2025).

Bolukbasi, T. et al. (2016). “Man Is to Computer Programmer as Woman Is to Homemaker? Debiasing Word Embeddings”. In: *NIPS’16: Proceedings of the 30th International Conference on Neural Information Processing Systems*.

Chakravarthi, B. R. et al. (July 2021). “A Survey of Orthographic Information in Machine Translation”. In: *SN Computer Science* 2.4, p. 330. ISSN: 2662-995X, 2661-8907. DOI: 10.1007/s42979-021-00723-4. (Visited on 06/27/2025).

Cho, W. I. et al. (2019). *On Measuring Gender Bias in Translation of Gender-neutral Pronouns*. DOI: 10.48550/arXiv.1905.11684. arXiv: 1905.11684 [cs]. (Visited on 04/21/2025).

DeepL (Nov. 2021). *How Does DeepL Work?* https://www.deepl.com/en/blog/how-does-deepl-work. (Visited on 06/27/2025).

Devlin, J. (2018). *Multilingual BERT GitHub Readme*. https://github.com/google-research/bert/blob/a9ba (Visited on 07/25/2025).

Devlin, J. et al. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. DOI: 10.48550/arXiv.1810.04805. arXiv: 1810.04805 [cs]. (Visited on 04/09/2025).

Godsil, R. D. et al. (2016). “The Effects of Gender Roles, Implicit Bias, and Stereotype Threat on the Lives of Women and Girls”. In: *THE SCIENCE OF EQUALITY* 2.Perception Institute.

Google (Dec. 2018). *Reducing Gender Bias in Google Translate*. https://blog.google/products/translate/red gender-bias-google-translate/. (Visited on 06/05/2025).

Gurgurov, D., T. Bäumel, and T. Anikina (2024). "Multilingual Large Language Models and Curse of Multilinguality". In: DOI: 10.48550/arXiv.2406.10602. arXiv: 2406.10602 [cs]. (Visited on 07/27/2025).

Kappl, M. (2025). *Are All Spanish Doctors Male? Evaluating Gender Bias in German Machine Translation*. DOI: 10.48550/arXiv.2502.19104. arXiv: 2502.19104 [cs]. (Visited on 04/10/2025).

Lardelli, M., G. Attanasio, and A. Lauscher (2024). "Building Bridges: A Dataset for Evaluating Gender-Fair Machine Translation into German". In: *Findings of the Association for Computational Linguistics ACL 2024*. Bangkok, Thailand and virtual meeting: Association for Computational Linguistics, pp. 7542–7550. DOI: 10.18653/v1/2024.findings-acl.448. (Visited on 04/06/2025).

Lauscher, A. et al. (Nov. 2020). "From Zero to Hero: On the Limitations of Zero-Shot Language Transfer with Multilingual Transformers". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Ed. by B. Webber et al. Online: Association for Computational Linguistics, pp. 4483–4499. DOI: 10.18653/v1/2020.emnlp-main.363. (Visited on 07/27/2025).

Libovický, J., R. Rosa, and A. Fraser (Nov. 2019). *How Language-Neutral Is Multilingual BERT?* DOI: 10.48550/arXiv.1911.03310. arXiv: 1911.03310 [cs]. (Visited on 07/14/2025).

Lin, G. H.-c. and P. S. C. Chien (2009). "Machine Translation for Academic Purposes". In: *Proceedings of the International Conference on TESOL and Translation 2009*, pp.133–148.

Mosbach, M., M. Andriushchenko, and D. Klakow (Mar. 2021). *On the Stability of Fine-tuning BERT: Misconceptions, Explanations, and Strong Baselines*. DOI: 10.48550/arXiv.2006.04884. arXiv: 2006.04884 [cs]. (Visited on 07/14/2025).

Nadipalli, S. (Feb. 2025). *Layer-Wise Evolution of Representations in Fine-Tuned Transformers: Insights from Sparse AutoEncoders*. DOI: 10.48550/arXiv.2502.16722. arXiv: 2502.16722 [cs]. (Visited on 07/20/2025).

Pecher, B., I. Srba, and M. Bielikova (Apr. 2024). *Comparing Specialised Small and General Large Language Models on Text Classification: 100 Labelled Samples to Achieve Break-Even Performance*. DOI: 10.48550/arXiv.2402.12819. arXiv: 2402.12819 [cs]. (Visited on 04/27/2025).

Phuong, M. and M. Hutter (July 2022). *Formal Algorithms for Transformers*. DOI: 10.48550/arXiv.2207.09238. arXiv: 2207.09238 [cs]. (Visited on 07/04/2025).

Pires, T., E. Schlinger, and D. Garrette (June 2019). *How Multilingual Is Multilingual BERT?* DOI: 10.48550/arXiv.1906.01502. arXiv: 1906.01502 [cs]. (Visited on 07/14/2025).

Prates, M. O. R., P. H. C. Avelar, and L. Lamb (2019). *Assessing Gender Bias in Machine Translation – A Case Study with Google Translate*. DOI: 10.48550/arXiv.1809.02208. arXiv: 1809.02208 [cs]. (Visited on 04/03/2025).

Quemy, A. (Mar. 2019). *Binary Classification in Unstructured Space With Hypergraph Case-Based Reasoning*. DOI: 10.48550/arXiv.1806.06232. arXiv: 1806.06232 [cs]. (Visited on 07/04/2025).

Rainio, O., J. Teuho, and R. Klén (Mar. 2024). "Evaluation Metrics and Statistical Tests for Machine Learning". In: *Scientific Reports* 14.1. ISSN: 2045-2322. DOI: 10.1038/s41598-024-56706-x. (Visited on 07/20/2025).

Rescigno, A. A. and J. Monti (2023). "Gender Bias in Machine Translation: A Statistical Evaluation of Google Translate and DeepL for English, Italian and German". In: *Proceedings of the International Conference on Human-informed Translation and Interpreting Technology 2023*. UNIOR NLP Research Group, University of Naples "L'Orientale", Naples, Italy: INCOMA Ltd., Shoumen, Bulgaria, pp. 1–11. DOI: 10.26615/issn.2683-0078.2023_001. (Visited on 02/27/2025).

Savoldi, B., J. Bastings, et al. (May 2025). "A Decade of Gender Bias in Machine Translation". In: *Patterns*, p. 101257. ISSN: 26663899. DOI: 10.1016/j.patter.2025.101257. (Visited on 06/06/2025).

Savoldi, B., E. Cupin, et al. (2025). *mGeNTE: A Multilingual Resource for Gender-Neutral Language and Translation*. DOI: 10.48550/arXiv.2501.09409. arXiv: 2501.09409 [cs]. (Visited on 04/08/2025).

Savoldi, B., S. Papi, et al. (2024). "What the Harm? Quantifying the Tangible Impact of Gender Bias in Machine Translation with a Human-centered Study". In: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Miami, Florida, USA: Association for Computational Linguistics, pp. 18048–18076. DOI: 10.18653/v1/2024.emnlp-main.1002. (Visited on 04/06/2025).

Schiebinger, L. (Mar. 2014). "Scientific Research Must Take Gender into Account". In: *Nature* 507.7490, pp. 9–9. ISSN: 1476-4687. DOI: 10.1038/507009a. (Visited on 06/06/2025).

Schmitz, D. (Aug. 2022). *In German, All Professors Are Male*. DOI: 10.31234/osf.io/yjuhc. (Visited on 06/06/2025).

Schryen, G. (2015). "Writing Qualitative IS Literature Reviews—Guidelines for Synthesis, Interpretation, and Guidance of Research". In: *Communications of the Association for*

Information Systems 37. ISSN: 15293181. DOI: 10.17705/1CAIS.03712. (Visited on 05/09/2025).

Schwemmer, C. et al. (Jan. 2020). "Diagnosing Gender Bias in Image Recognition Systems". In: *Socius* 6, p. 2378023120967171. ISSN: 2378-0231. DOI: 10.1177/2378023120967171. (Visited on 05/28/2025).

Sczesny, S., M. Formanowicz, and F. Moser (2016). "Can Gender-Fair Language Reduce Gender Stereotyping and Discrimination?" In: *Frontiers in Psychology* 7. ISSN: 1664-1078. DOI: 10.3389/fpsyg.2016.00025. (Visited on 05/16/2025).

Shah, D., H. A. Schwartz, and D. Hovy (2020). "Predictive Biases in Natural Language Processing Models: A Conceptual Framework and Overview". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 5248–5264. DOI: 10.18653/v1/2020.acl-main.468. arXiv: 1912.11078 [cs]. (Visited on 06/13/2025).

Shrestha, S. and S. Das (2022). "Exploring Gender Biases in ML and AI Academic Research through Systematic Literature Review". In: *Frontiers in Artificial Intelligence* 5, p. 976838. ISSN: 2624-8212. DOI: 10.3389/frai.2022.976838. (Visited on 04/06/2025).

Sichler, A. and E. Prommer (2014). "Gender Differences within the German-language Wikipedia". In: *ESSACHESS - Journal for Communication Studies* 7.2, pp. 77–93. ISSN: 1775-352X.

SkyQuest (2025). *Machine Translation (MT) Market Size, Growth & Trends Report | 2032.* https://www.skyquestt.com/report/machine-translation-market. (Visited on 05/23/2025).

Smacchia, M., S. Za, and A. Arenas (2024). "Does AI Reflect Human Behaviour? Exploring the Presence of Gender Bias in AI Translation Tools". In: *Digital (Eco) Systems and Societal Challenges*. Ed. by A. M. Braccini, F. Ricciardi, and F. Virili. Vol. 72. Cham: Springer Nature Switzerland, pp. 355–373. ISBN: 978-3-031-75585-9 978-3-031-75586-6. DOI: 10.1007/978-3-031-75586-6_19. (Visited on 02/27/2025).

Smith, B. (May 2024). *A Complete Guide to BERT with Code.* (Visited on 07/11/2025).

Sorrenti, A. et al. (Oct. 2023). "Selective Freezing for Efficient Continual Learning". In: *2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW).* Paris, France: IEEE, pp. 3542–3551. DOI: 10.1109/iccvw60793.2023.00381. (Visited on 07/27/2025).

Soundararajan, S. and S. J. Delany (2024). "Investigating Gender Bias in Large Language Models Through Text Generation". In: *Association for Computational Linguistics* Proceedings of the 7th International Conference on Natural Language and Speech Processing (ICNLSP 2024), pp. 410–424.

Stanczak, K. and I. Augenstein (Dec. 2021). *A Survey on Gender Bias in Natural Language Processing*. DOI: 10.48550/arXiv.2112.14168. arXiv: 2112.14168 [cs]. (Visited on 05/13/2025).

Stanovsky, G., N. A. Smith, and L. Zettlemoyer (2019). *Evaluating Gender Bias in Machine Translation*. DOI: 10.48550/arXiv.1906.00591. arXiv: 1906.00591 [cs]. (Visited on 04/03/2025).

Stella, R. et al. (2021). *A Dataset for Studying Gender Bias in Translation*. https://research.google/blog/a-dataset-for-studying-gender-bias-in-translation/. (Visited on 04/10/2025).

Tiedemann, J. and S. Thottingal (2020). "OPUS-MT – Building Open Translation Services for the World". In: *European Association for Machine Translation* Proceedings of the 22nd Annual Conference of the European Association for Machine Translation, pp. 479–480.

Ullmann, S. (2022). "Gender Bias in Machine Translation Systems". In: *Artificial Intelligence and Its Discontents*. Ed. by A. Hanemaayer. Cham: Springer International Publishing, pp. 123–144. ISBN: 978-3-030-88614-1 978-3-030-88615-8. DOI: 10.1007/978-3-030-88615-8_7. (Visited on 05/16/2025).

United Nations (2023). *Achieve Gender Equality And Empower All Women and Girls*. https://sdgs.un.org/goals/goal5. (Visited on 05/28/2025).

Waldendorf, A. (Apr. 2024). "Words of Change: The Increase of Gender-Inclusive Language in German Media". In: *European Sociological Review* 40.2, pp. 357–374. ISSN: 0266-7215. DOI: 10.1093/esr/jcad044. (Visited on 06/08/2025).

Wu, S. and M. Dredze (Nov. 2019). "Beto, Bentz, Becas: The Surprising Cross-Lingual Effectiveness of BERT". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Ed. by K. Inui et al. Hong Kong, China: Association for Computational Linguistics, pp. 833–844. DOI: 10.18653/v1/D19-1077. (Visited on 07/27/2025).

Wu, Y. et al. (Oct. 2016). *Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation*. DOI: 10.48550/arXiv.1609.08144. arXiv: 1609.08144 [cs]. (Visited on 06/29/2025).

Xiao, T. and J. Zhu (Nov. 2023). *Introduction to Transformers: An NLP Perspective*. DOI: 10.48550/arXiv.2311.17633. arXiv: 2311.17633 [cs]. (Visited on 07/10/2025).

Appendix

1 Prompt and Output for Pre-training/Fine-tuning Explanation

Linear and Softmax Layers in BERT Fine-tuning: Formulas and Functions

In BERT fine-tuning, the **linear and softmax layers** form the classification head that processes the final hidden representations from the transformer to produce predictions. Here's a comprehensive explanation of their mathematical formulas and functions:

Linear Layer (Classification Head)

The linear layer is a fully connected layer that transforms the BERT output representation into class scores.

Formula:
$z = W \cdot h + b$

Where:

- z is the output logits (raw prediction scores)
- W is the weight matrix with dimensions $num\_classes \times hidden\_size$
- h is the hidden representation from BERT (typically the [CLS] token representation)
- b is the bias vector with dimensions $num\_classes$

What it does:

- **Transforms representations**: Converts the high-dimensional BERT output (usually 768 dimensions for BERT-base) into class-specific scores[1] [2]
- **Learns task-specific mappings**: The weight matrix W learns which features in the BERT representation are most important for each class
- **Provides linear transformation**: Each output neuron computes a weighted sum of all input features plus a bias term

Softmax Layer (Activation Function)

The softmax function converts the raw logits from the linear layer into probability distributions.

Formula:
$softmax(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$

Where:

- z_i is the logit for class i
- K is the total number of classes
- e is the exponential function

What it does:

- **Normalizes outputs**: Ensures all class probabilities sum to 1[3] [4]
- **Amplifies differences**: Uses exponential function to emphasize higher-scoring classes
- **Enables probabilistic interpretation**: Converts raw scores into meaningful probabilities for decision-making

Combined Process in BERT Fine-tuning

The complete classification pipeline follows this sequence:

1. **BERT Processing**: Input text → BERT layers → [CLS] token representation $h$
2. **Linear Transformation**: $z = W \cdot h + b$
3. **Softmax Activation**: $p = softmax(z)$
4. **Prediction**: $\hat{y} = \arg\max(p)$

Training Process

During fine-tuning, these layers are optimized using:

- **Cross-entropy loss**: $L = -\sum_{i=1}^{K} y_i \log(p_i)$
- **Backpropagation**: Gradients flow back through softmax → linear → BERT layers
- **Parameter updates**: Both the linear layer weights and BERT parameters are updated

Practical Implementation

In practice, the linear and softmax layers work together as follows[2] [5]:

```
# Conceptual implementation
linear_output = torch.matmul(bert_output, weight_matrix) + bias
probabilities = torch.softmax(linear_output, dim=-1)
```

Key Functions

Linear Layer Functions:

- Feature extraction and dimensionality reduction
- Task-specific weight learning
- Bias adjustment for class imbalances

Softmax Layer Functions:

- Probability normalization
- Confidence scoring
- Multi-class decision making

Variations and Enhancements

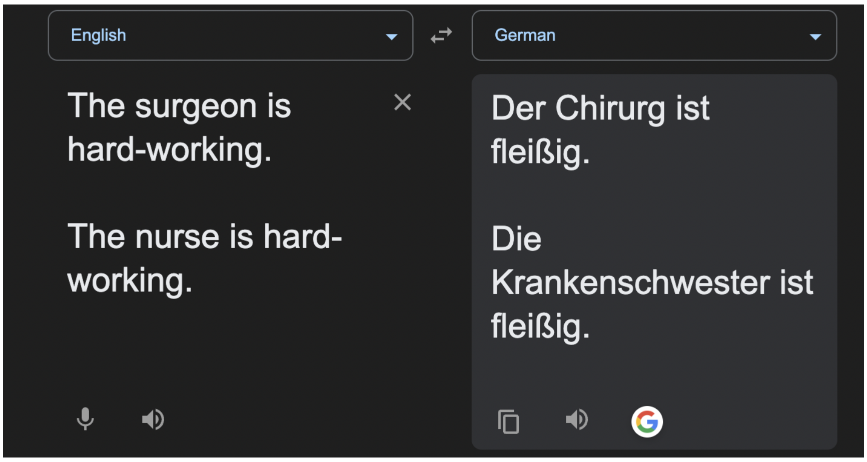
Recent research has explored alternatives to the standard linear+softmax approach:

- **Multiple linear layers**: Adding depth to the classification head[2]
- **Attention mechanisms**: Using attention to weight different parts of the representation[4]
- **Alternative activation functions**: Exploring functions beyond softmax for specific tasks[6]

The linear and softmax layers, while conceptually simple, are crucial for translating BERT's rich contextual representations into task-specific predictions. Their mathematical simplicity belies their importance in the overall fine-tuning process, serving as the bridge between pre-trained language understanding and downstream task performance.

⁂

1. https://www.semanticscholar.org/paper/b964afe5b755022f1f1e6915d23df9a7f65c911c
2. https://ieeexplore.ieee.org/document/10742347/
3. https://www.mdpi.com/1424-8220/23/3/1481
4. https://journals.sagepub.com/doi/10.3233/KES-230066
5. https://www.mdpi.com/2078-2489/14/8/467
6. https://arxiv.org/abs/2408.08803



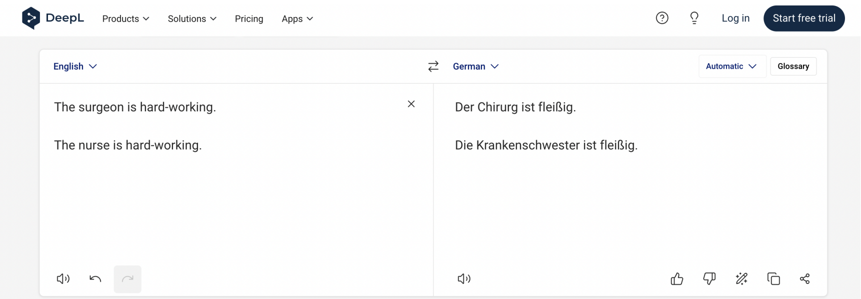
The surgeon is hard-working.

The nurse is hard-working.

Der Chirurg ist fleißig.

Die Krankenschwester ist fleißig.

Figure 1: Google Translate assigns stereotypical genders to occupational roles.



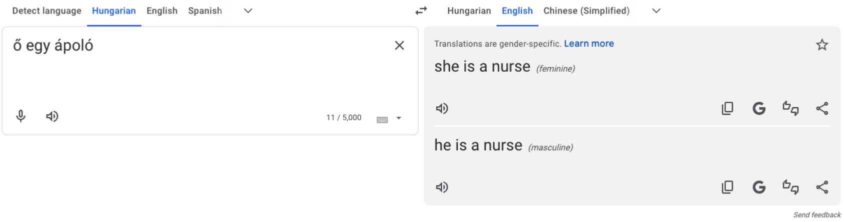
The surgeon is hard-working.

The nurse is hard-working.

Der Chirurg ist fleißig.

Die Krankenschwester ist fleißig.

Figure 2: DeepL shows a similar bias in the same sentence, highlighting consistent patterns across MT tools.



ő egy ápoló

Translations are gender-specific. Learn more

she is a nurse *(feminine)*

he is a nurse *(masculine)*

Figure 3: Gender-specific translation by Google Translate for ambiguous pronouns.

1. Hiermit versichere ich,
   - dass ich die von mir vorgelegte Arbeit selbständig abgefasst habe,
   - dass ich keine weiteren Hilfsmittel verwendet habe als diejenigen, die im Vorfeld explizit zugelassen und von mir angegeben wurden,
   - dass ich die Stellen der Arbeit, die dem Wortlaut oder dem Sinn nach anderen Werken (dazu zählen auch Internetquellen und KI-basierte Tools) entnommen sind, unter Angabe der Quelle kenntlich gemacht habe und
   - dass ich die vorliegende Arbeit noch nicht für andere Prüfungen eingereicht habe.
2. Mir ist bewusst,
   - dass ich diese Prüfung nicht bestanden habe, wenn ich die mir bekannte Frist für die Einreichung meiner schriftlichen Arbeit versäume,
   - dass ich im Falle eines Täuschungsversuchs diese Prüfung nicht bestanden habe,
   - dass ich im Falle eines schwerwiegenden Täuschungsversuchs ggf. die Gesamtprüfung endgültig nicht bestanden habe und in diesem Studiengang nicht mehr weiter studieren darf und
   - dass ich, sofern ich zur Erstellung dieser Arbeit KI-basierter Tools verwendet habe, die Verantwortung für eventuell durch die KI generierte fehlerhafte oder verzerrte (bias) Inhalte, fehlerhafte Referenzen, Verstöße gegen das Datenschutz- und Urheberrecht oder Plagiate trage.

Berlin, den July 27, 2025

. . . .

(Unterschrift des Verfassers)