Karlsruhe Institute of Technology

AI4LT

# Towards a Better Understanding of In-Context Learning in Large Language Models for Machine Translation

Bachelor's Thesis of

Florian Raith

Artificial Intelligence for Language Technologies (AI4LT) Lab
Institute for Anthropomatics and Robotics (IAR)
KIT Department of Informatics

| | |
|---|---|
| Reviewer: | Prof. Dr. Jan Niehues |
| Second reviewer: | Prof. Dr.-Ing. Rainer Stiefelhagen |
| Advisor: | M.Sc. Maike Züfle |
| Second advisor: | M.Sc. Danni Liu |

15. December 2024 – 15. April 2025

Karlsruher Institut für Technologie
Fakultät für Informatik
Postfach 6980
76128 Karlsruhe

I declare that I have developed and written the enclosed thesis completely by myself, and have not used sources or means without declaration in the text.

**PLACE, DATE**

. . . . . .

(Florian Raith)

# Abstract

Large Language Models (LLMs) have shown strong performance on machine translation (MT) tasks through in-context learning (ICL), where models generate translations conditioned on a few examples in the prompt. While ICL has been widely studied in classification tasks, its behavior in generative settings like translation remains underexplored — especially under imperfect prompting conditions. This thesis investigates how the quality of in-context examples impacts translation performance, with a focus on grammatical noise, incorrect alignments, and random mismatches. We conduct a structured evaluation comparing two distinct model types: Llama 3.1, a general-purpose instruction-tuned model, and Tower, a translation-optimized LLM fine-tuned on multilingual MT data. Using controlled perturbations of prompt examples, we assess model robustness across language pairs and error types. Our findings reveal that translation-optimized models are substantially more robust to noisy in-context examples than general-purpose models. For language pairs included in their instruction fine-tuning, optimized models demonstrate the ability to mitigate or ignore incorrect or grammatically flawed examples, whereas general-purpose models show a strong reliance on example quality and often degrade under misleading inputs. Moderate grammatical errors tend to have limited impact. Errors in target sentences consistently cause more severe degradation than those in source sentences. These results highlight the critical role of example quality and model specialization in in-context machine translation. They suggest that improving translation through ICL may require careful prompt design and targeted fine-tuning, rather than relying solely on general-purpose scaling.

# Zusammenfassung

Large Language Models (LLMs) haben starke Leistungen bei maschinellen Übersetzungsaufgaben (MT) durch In-Context Learning (ICL) gezeigt, bei dem Modelle Übersetzungen auf Grundlage weniger Beispiele im Prompt generieren. Während ICL bereits umfassend in Klassifikationsaufgaben untersucht wurde, ist das Verhalten in generativen Szenarien wie der Übersetzung bislang wenig erforscht – insbesondere unter unvollkommenen Prompt-Bedingungen. Diese Arbeit untersucht, wie sich die Qualität der In-Context-Beispiele auf die Übersetzungsleistung auswirkt, mit Fokus auf grammatikalisches Rauschen, fehlerhafte Zuordnungen und zufällige Fehlanpassungen. Wir führen eine strukturierte Evaluation durch, in der zwei unterschiedliche Modelltypen verglichen werden: Llama 3.1, ein allgemein einsetzbares, instruktionstuniertes Modell, und Tower, ein auf maschinelle Übersetzung spezialisiertes LLM, das auf mehrsprachigen MT-Daten feinabgestimmt wurde. Durch gezielte Störungen der Beispiele im Prompt bewerten wir die Robustheit der Modelle über Sprachpaare und Fehlertypen hinweg. Unsere Ergebnisse zeigen, dass auf Übersetzung optimierte Modelle deutlich robuster gegenüber fehlerhaften In-Context-Beispielen sind als allgemein einsetzbare Modelle. Für Sprachpaare, die in das instruktionstunierte Training einbezogen wurden, zeigen optimierte Modelle die Fähigkeit, fehlerhafte oder grammatikalisch mangelhafte Beispiele abzumildern oder zu ignorieren. Allgemeine Modelle hingegen sind stark von der Qualität der Beispiele abhängig und zeigen bei irreführenden Eingaben häufig eine verschlechterte Leistung. Moderat ausgeprägte Grammatikfehler wirken sich meist nur geringfügig aus. Fehler in den Zielsätzen führen durchweg zu einer stärkeren Verschlechterung als Fehler in den Quellsätzen. Diese Ergebnisse unterstreichen die entscheidende Rolle der Beispielqualität und Modellspezialisierung in der maschinellen Übersetzung mit In-Context Learning. Sie legen nahe, dass Fortschritte in der ICL-basierten Übersetzung sorgfältiges Prompt-Design und gezieltes Fine-Tuning erfordern – anstatt sich ausschließlich auf die Skalierung allgemeiner Modelle zu verlassen.

# Contents

# List of Figures

# List of Tables

# 1. Introduction

## 1.1. Problem Context and Motivation

Machine translation (MT) addresses a fundamental human need: communication across language barriers. While people seek connection, the diversity of languages complicates understanding, and manual translation remains time-consuming and resource-intensive (Baker, 1992). Recent advances in large language models (LLMs) have introduced a powerful alternative – systems capable of learning translation patterns directly from vast textual corpora without manual supervision (Vaswani et al., 2023). LLMs such as GPT-3 (T. B. Brown et al., 2020) and Llama (Touvron et al., 2023; Grattafiori et al., 2024) have demonstrated remarkable capabilities across a wide spectrum of natural language processing (NLP) tasks, including MT, question answering, and summarization (Devlin et al., 2019; Lewis et al., 2019; Yinhan Liu et al., 2020). A key breakthrough behind their flexibility is in-context learning (ICL) (T. B. Brown et al., 2020; Dong et al., 2022), which enables models to generalize to new tasks simply by conditioning on a small number of input-output examples provided in the prompt, without requiring any weight updates or gradient-based fine-tuning.

Real-world applications highlight the value of ICL. For instance, GitHub Copilot (GitHub and OpenAI, 2025) leverages previous code snippets as in-context examples to generate follow-up code in a consistent style. Similarly, in machine translation, multilingual support bots can benefit from ICL by conditioning on prior translation examples – particularly for domain-specific or technical terminology – to improve consistency and adequacy across languages.

While the phenomenon of ICL has been extensively analyzed for classification tasks (Min et al., 2022a; Yoo et al., 2022; Jerry Wei et al., 2023; Pan et al., 2023), its application to generative tasks – such as machine translation – has received comparatively less attention. Translation poses unique challenges for ICL, given its open-ended nature, reliance on bilingual alignment, and sensitivity to linguistic subtleties.

Recent work has demonstrated that the quality and structure of in-context examples play a central role in determining translation success (Agrawal et al., 2022; Zhang, Haddow, and Birch, 2023; Vilar et al., 2023). In some cases, even a single low-quality example can significantly deteriorate performance (Agrawal et al., 2022). However, much of the current literature on translation via ICL remains descriptive, lacking controlled experiments to quantify how different types of noise or imperfections impact performance. This gap is particularly relevant given the practical realities of many real-world applications: low-resource languages often lack high-quality parallel corpora, and in-context examples may contain typos, grammatical inconsistencies, or domain-specific terminology. Moreover, instruction-tuned generalist models like Llama 3.1 (Grattafiori et al., 2024) are frequently

preferred in industry and academia due to their flexibility, but their sensitivity to noisy or misleading prompts for MT tasks remains poorly understood. In contrast, domain-specialized models such as Tower (Alves et al., 2024) have been explicitly fine-tuned on machine translation tasks and might behave differently in the presence of noisy in-context examples. Yet, a direct comparison between generalist and translation-focused LLMs under noisy prompting conditions has not been systematically studied.

## 1.2. Research Questions

This thesis aims to fill this gap by conducting a structured, empirical investigation into how the quality of in-context examples affects translation performance across two contrasting model types. Specifically, we address the following research questions:

- **RQ1:** How does in-context learning performance differ between general-purpose and translation-optimized language models?
- **RQ2:** Does using incorrect or random translations as in-context examples hurt the performance of machine translation tasks?
- **RQ3:** How do grammatical errors — such as word reordering or spelling mistakes — in in-context examples affect the translation quality?

## 1.3. Thesis Structure

To address our research questions, the thesis is structured as follows. Chapter 2 introduces the technical background on large language models and their application to language tasks via in-context learning. Chapter 3 reviews relevant literature on in-context learning, particularly in the context of classification and translation, and examines recent findings on the underlying mechanisms of ICL. Chapter 4 outlines the experimental design, including the construction of perturbed prompts featuring grammatical errors, mismatches, and omissions, and details the evaluation methodology used with Llama 3.1 and Tower. Chapter 5 presents the empirical findings, analyzing how different types of in-context degradation affect translation output across both model types. Finally, Chapter 6 summarizes the main insights, discusses model-specific sensitivities, and reflects on the implications for designing robust prompting strategies in practical translation applications.

## 1.4. Summary of Contributions

Our findings reveal that translation-optimized models like Tower (Alves et al., 2024) are substantially more robust to noisy in-context examples than general-purpose models like Llama 3.1 (Grattafiori et al., 2024). Optimized models demonstrate the ability to mitigate or ignore incorrect or grammatically flawed examples, particularly for language pairs included in their instruction fine-tuning, whereas general-purpose models tend to be more sensitive to example quality and degrade under misleading inputs. While moderate

grammatical errors are generally well-tolerated, errors in target sentences consistently cause more significant performance degradation than those in source sentences. These results highlight the critical role of example quality and model specialization in in-context machine translation.

# 2. Background

Early machine translation systems relied on rule-based methods using handcrafted linguistic rules and bilingual dictionaries, but these proved rigid and struggled with linguistic nuance (Wang et al., 2022). Statistical machine translation (SMT) (P. F. Brown et al., 1990) improved adaptability by learning probabilistic mappings from parallel corpora, yet still faced limitations in contextual understanding (Och, 2003). Recurrent Neural Networks (RNNs), particularly LSTMs (Hochreiter and Schmidhuber, 1997) and GRUs (Cho, Merrienboer, Gulcehre, et al., 2014), advanced translation by modeling variable-length sequences and capturing long-range dependencies. Recent advances in large language models (LLMs) have introduced a powerful alternative — systems that can learn translation patterns directly from vast textual data, without manual supervision (Vaswani et al., 2023), often leveraging in-context learning to generalize from limited examples at inference time (T. B. Brown et al., 2020; Zhang, Haddow, and Birch, 2023).

This chapter provides the technical background necessary to understand such models and their application to language tasks. Section 2.1 introduces the language modeling task, the foundation of modern natural language processing (NLP). Section 2.2 outlines the Transformer architecture that underlies LLMs, with a focus on decoder-only models (Section 2.2.1). Section 2.2.2 discusses how these models scale into LLMs. Section 2.3 introduces in-context learning, a core capability enabling LLMs to perform translation and other tasks without fine-tuning.

## 2.1. Language Modeling Task

Language modeling is a foundational task in NLP, aiming to predict the next word in a sequence given its preceding context. For instance, in the sentence "This document is about Natural Language ____", the model should predict "Processing" based on prior context. A language model thus assigns probabilities to sequences of words, learning which word is most likely to follow a given context. This seemingly simple task captures many linguistic phenomena — syntactic, semantic, and contextual — as successful prediction requires nuanced language understanding.

Language modeling has widespread applications. Tasks such as predictive text input, autocorrect, spell-checking, machine translation, code completion (e.g., GitHub Copilot (GitHub and OpenAI, 2025)), and conversational agents all rely on language models to select contextually appropriate words or phrases. In machine translation, for example, a language model ensures that the generated sentence is both fluent and semantically plausible in the target language.

Historically, language models enhanced larger NLP systems such as statistical machine translation (Koehn et al., 2007; Cho, Merrienboer, Gülçehre, et al., 2014) and speech

recognition (Graves, Mohamed, and Hinton, 2013) by favoring more probable hypotheses. More recently, they have become central to generative systems — e.g., OpenAI's GPT series (T. B. Brown et al., 2020; OpenAI et al., 2024) — which produce responses by generating a sequence of tokens conditioned on a user's prompt. Language modeling is particularly attractive due to its *self-supervised* nature: models train on raw text by predicting hidden or subsequent words, without requiring annotated data. The abundance of digital text enables training at large scales, paving the way for powerful models and advanced architectures discussed below.

## 2.2. Transformers

Modern language models are built on the *Transformer* architecture (Vaswani et al., 2023), which processes sequences in parallel using *self-attention*, unlike earlier RNN-based (Sherstinsky, 2018) models that processed words sequentially. Self-attention enables the model to consider all positions in the input simultaneously, identifying which tokens are most relevant when encoding a given word. This design allows Transformers to capture long-range dependencies and complex contextual relationships more effectively than RNNs. They use stacked layers of self-attention and feed-forward networks, along with positional encodings to model word order.

Transformers are trained via backpropagation and gradient descent, adjusting parameters to minimize loss on language modeling tasks. Their elimination of recurrence makes training highly parallelizable, leveraging GPUs/TPUs for efficient large-scale learning. Transformers now underpin state-of-the-art models in NLP (Vaswani et al., 2023). Their key innovation — *multi-head attention* — enables the model to attend to information from multiple representational subspaces, enhancing contextual understanding. Intuitively, the model determines which input elements are most relevant to each prediction and weights them accordingly.

The Transformer architecture includes both an *encoder* and *decoder*, originally designed for sequence-to-sequence tasks like machine translation. Variants include encoder-only models (e.g., BERT, Devlin et al., 2019) for representation tasks, and encoder–decoder and decoder-only models for generation. This thesis focuses on the decoder-only variant.

### 2.2.1. Decoder-Only

Decoder-only Transformers omit the encoder and operate purely as generative models. Given a prompt, they generate text *autoregressively*, predicting one token at a time. They use *masked self-attention* to ensure each prediction depends only on prior context.

These models are trained via *causal language modeling*, where the objective is next-token prediction. Once trained, they generate coherent continuations for a wide range of prompts — from story completion to question answering.

Their generative flexibility supports diverse tasks like summarization, translation, or instruction following by phrasing the task as a prompt. For example, inputting "Translate to French: [sentence]" prompts the model to generate the translation, all within the autoregressive framework.

Modern systems like ChatGPT (OpenAI, 2025) are based on large decoder-only Transformers, fine-tuned for conversational behavior. These models exemplify the strengths of this architecture for open-ended generation.

### 2.2.2. Large Language Models

Transformer-based models have scaled dramatically, giving rise to Large Language Models (LLMs), typically defined as models with billions of parameters. LLMs are trained on massive text corpora using self-supervised objectives, acquiring broad linguistic and factual knowledge through text prediction.

For example, GPT-3 has 175 billion parameters (T. B. Brown et al., 2020) — over 100 times the size of the original Transformer — while Llama 3 reaches 405 billion (Grattafiori et al., 2024). Larger models exhibit improved performance across tasks, often displaying emergent abilities not present in smaller models (Jason Wei et al., 2022).

A key emergent property is *in-context learning* (T. B. Brown et al., 2020), where the model performs tasks based on examples in the input, without parameter updates. These capabilities scale with model size, highlighting the benefits of large-scale training.

## 2.3. In-Context Learning

In-context learning is the ability of LLMs to perform tasks using instructions or examples provided directly in the prompt, without updating model parameters (T. B. Brown et al., 2020; Dong et al., 2022). This was notably demonstrated by GPT-3, which performs tasks such as translation or question answering using only a few prompt examples (*few-shot prompting*) (T. B. Brown et al., 2020).

Prior to LLMs, new tasks typically required fine-tuning on labeled datasets. GPT-3 showed that sufficiently large models could generalize from in-prompt examples alone. For instance, when prompted with: "English: I am happy. French: Je suis heureux.
English: Thank you. French:", the model correctly continues with "Merci." It infers the task structure purely from the prompt.

In-context learning mimics human learning by example and is computationally efficient — eliminating the need for costly fine-tuning on large models. This has led to the rise of *prompt engineering*, where task success depends on the quality and structure of the input prompt (White et al., 2023; Yi Liu et al., 2023). Research confirms that this ability emerges with scale: larger models outperform smaller ones at prompt-based learning (Jason Wei et al., 2022).

# 3. Related Work

Recent advances in large language models (LLMs) have demonstrated impressive capabilities across a wide range of natural language processing tasks, including language understanding, question answering, translation, and summarization (Devlin et al., 2019; Lewis et al., 2019; Yinhan Liu et al., 2020). In-context learning (ICL) has emerged as an inherent property of larger models, enabling them to adapt to new tasks using only a handful of examples provided in the prompt — thereby circumventing the computational expense of fine-tuning (T. B. Brown et al., 2020; Dong et al., 2022; Jason Wei et al., 2022). In this chapter, we survey prior work on in-context learning, with a focus on three key areas: (Section 3.1) the use of contextual demonstrations in classification tasks and the factors that contribute to robust ICL performance; (Section 3.2) strategies for selecting and structuring examples in translation tasks, where prompt quality can dramatically affect outcomes; and (Section 3.3) recent theoretical and empirical insights into the mechanisms underlying ICL, including task recognition, attention head specialization, and latent task representations.

## 3.1. Learning from Contextual Demonstrations

Research on in-context learning (ICL) in classification tasks has established that the precision of ground-truth demonstrations provided to large language models (LLMs) is not strictly necessary to achieve effective performance (Min et al., 2022a). Rather, the demonstrations serve multiple specific purposes: they introduce and define the set of possible labels, reflect realistic distributions of input texts, and demonstrate the sequential input-output structure expected by the model (Min et al., 2022a). This aligns with our findings in translation tasks, where even when we intentionally prefixed in-context examples with an incorrect language label (e.g., labeling German text as French), the models effectively disregarded the inaccurate label and successfully leveraged the provided examples to achieve performance improvements (see Section 5.3).

Yoo et al. (2022) further specify that the robustness of ICL to imprecise or noisy demonstrations depends significantly on two main factors: prompt verbosity and model size. Prompt verbosity refers to the extent and richness of contextual detail provided within the prompt. Increased verbosity generally improves the clarity of the expected task structure and label definitions, thereby enhancing the model's tolerance to noise in demonstrations (Yoo et al., 2022). Conversely, overly terse prompts may fail to sufficiently convey task requirements, negatively affecting performance (Yoo et al., 2022). Model size, on the other hand, influences the ability of an LLM to generalize from noisy examples. Larger models tend to have greater representational capacity, thus better accommodating variability and noise within demonstrations, leading to improved ICL outcomes (Yoo et al., 2022).

Instruction tuning further enhances performance by leveraging semantic priors – pre-existing knowledge encoded within the model about relationships between inputs and their plausible outputs (Jerry Wei et al., 2023). Semantic priors effectively serve as implicit guidelines that help models more reliably map new inputs to appropriate labels. Models with robust semantic priors are better equipped to generalize accurately from fewer and noisier demonstrations, as these priors compensate for ambiguity or imprecision in examples provided at inference time (Jerry Wei et al., 2023). This aligns with our findings that models instruction-tuned specifically for translation tasks exhibit greater robustness to mismatched translation examples (see Section 5.4.2) compared to general-purpose models (see Section 5.4.1).

Further insights from Pan et al. (2023) highlight the distinction between task recognition and task learning in the context of ICL. Task recognition pertains to a model's ability to correctly identify the nature of a task from its description alone, independent of the specific input-output mapping provided. This capability generally plateaus beyond a certain scale of the model and number of demonstrations, suggesting limited incremental benefit from further increases in these parameters (Pan et al., 2023). In contrast, task learning – the capacity to adapt and accurately apply novel input-label mappings – continues to improve significantly with additional in-context examples, emphasizing the importance of sufficient and well-chosen demonstrations to maximize performance (Pan et al., 2023).

## 3.2. In-Context Examples Selection for Translation Tasks

Multiple studies have demonstrated that the effectiveness of ICL in machine translation heavily relies on both the number (Zhang, Haddow, and Birch, 2023) and the quality (Agrawal et al., 2022; Vilar et al., 2023) of the prompt examples provided. Specifically, selecting and composing these examples carefully can significantly enhance translation outcomes, while poorly chosen examples can substantially degrade performance (Agrawal et al., 2022; Vilar et al., 2023).

The quality of prompt examples is critical; those with high semantic relevance and n-gram overlap with the input consistently improve translation performance, outperforming strong baselines like kNN-MT, especially in out-of-domain settings (Agrawal et al., 2022). Conversely, even a single noisy or unrelated example can have a catastrophic impact, drastically reducing translation accuracy (Agrawal et al., 2022).

Moreover, positional bias in the prompt sequence has been identified as a significant determinant of performance, with earlier examples in a sequence generally exerting a stronger influence on the translated output (Zaranis, Guerreiro, and Martins, 2024). This bias underscores the importance of ordering examples thoughtfully. Additionally, Zaranis, Guerreiro, and Martins (2024) observed that the source part of few-shot examples appears to contribute more significantly to the translation than its corresponding target part, irrespective of the translation direction. This aligns with our findings that omitting the source text and providing only the target text in few-shot translation examples significantly degrades translation performance, underscoring the importance of the source segment (see Section 5.2). However, we observed a somewhat contradictory outcome: grammatical errors in the target side had a greater negative impact on translation quality than errors in

the source (see Section 5.5.3). This discrepancy highlights the complexity of source-target interactions in in-context learning and indicates a need for further investigation.

Additional investigations (Zhang, Haddow, and Birch, 2023; Vilar et al., 2023) reveal nuanced factors influencing example effectiveness. Zhang, Haddow, and Birch (2023) identified features such as semantic similarity and example quality as having significant, though weak, correlations with translation performance, suggesting that relying solely on semantic similarity metrics is insufficient for optimal prompt selection. Instead, they propose constructing pseudo-parallel examples through zero-shot prompting of monolingual data, which can improve translation outcomes, emphasizing the value of example quality and relevance (Zhang, Haddow, and Birch, 2023).

Vilar et al. (2023) reinforced these findings by systematically assessing various example selection strategies and concluded that example quality is the most critical factor, surpassing both domain alignment and semantic proximity to the input text. Their analyses using modern MT metrics and human evaluation indicate that carefully selected, high-quality examples substantially improve translation performance, although state-of-the-art supervised systems still maintain an overall advantage (Vilar et al., 2023).

Our own findings further reinforce the critical role of example quality, as we observed substantial performance degradation when deliberately introducing errors into in-context examples, such as mismatched source-target pairs, translations labeled with incorrect languages, and grammatical inaccuracies.

## 3.3. Understanding the Mechanisms of In-Context Learning

The underlying mechanisms enabling large language models (LLMs) to perform in-context learning (ICL) remain an active area of research. Recent theoretical (Xie et al., 2021) and empirical (Sia, Mueller, and Duh, 2024; Tao, Chen, and N. F. Liu, 2024; Yin and Steinhardt, 2025) studies have contributed to a clearer understanding of how LLMs effectively leverage provided examples.

Theoretical explanations (Xie et al., 2021) have proposed that ICL can emerge naturally through implicit Bayesian inference when pretraining data exhibits long-range coherence. Xie et al. (2021) showed that in-context learning occurs as LLMs infer a latent, document-level concept from examples provided in the prompt, even when there is a distribution mismatch between prompts and pretraining data. They demonstrated this mechanism formally using a synthetic dataset (GINC (Xie et al., 2021)), highlighting that the quality of in-context learning improves with model size and the number of examples, despite maintaining identical pretraining losses (Xie et al., 2021).

Empirical studies by Sia, Mueller, and Duh (2024) have localized a critical "task recognition" point within the model layers, beyond which the specific task (e.g., translation or code generation) is effectively encoded into internal representations. At this point, further attention to the input context becomes redundant. Experiments demonstrated that masking input context from later layers still maintained task performance, suggesting substantial computational savings and implications for efficient fine-tuning (Sia, Mueller, and Duh, 2024). Moreover, this research underscored that parameter-efficient fine-tuning

methods, such as LoRA (Hu et al., 2021), are particularly effective at layers preceding the task recognition point (Sia, Mueller, and Duh, 2024).

In parallel, Tao, Chen, and N. F. Liu (2024) identified two sequential processes within LLMs during ICL: an inference function that generates a latent task representation and a subsequent verbalization function that maps this representation to specific output labels. Their controlled interventions revealed that the inference function remains invariant to changes in label spaces, reinforcing the idea of a shared underlying inference mechanism across different ICL settings. Further analysis localized these two functions within distinct sets of layers, suggesting that the model's internal structure differentiates inference and verbalization explicitly (Tao, Chen, and N. F. Liu, 2024).

Adding to these findings, Yin and Steinhardt (2025) distinguished two specific attention head mechanisms crucial to ICL: induction heads and function vector (FV) heads. Induction heads specialize in identifying and replicating relevant token patterns, while FV heads encode task-specific latent representations. Through detailed ablation studies, they demonstrated that FV heads are predominantly responsible for improved few-shot ICL performance, especially in larger models. Many FV heads initially function as induction heads early in the training process before transitioning to their more sophisticated roles, indicating a developmental interplay between these two mechanisms (Yin and Steinhardt, 2025).

# 4. Experimental Setup

To examine the impact of in-context example quality on Large Language Model (LLM) performance in machine translation (MT), we design controlled experiments that reflect common issues in real-world data. These experiments are grounded in the systematic construction and manipulation of prompts containing in-context examples derived from benchmark datasets (Section 4.1), outlining both the baseline prompt construction process and specific modifications applied to assess the impact of example quality. This chapter then explains the technical setup employed for inference (Section 4.2), including the selection of models, datasets and languages, and computational infrastructure. Finally, we introduce the metrics used to quantitatively evaluate translation outputs, as well as describe the language identification method implemented to ensure the validity of generated translations (Section 4.3). The results of these experimental evaluations are presented and analyzed in Chapter 5.

## 4.1. Experiments

To comprehensively investigate the influence of in-context example quality on the translation capabilities of large language models (LLMs), we designed a series of controlled experiments. These experiments will systematically evaluate how different aspects of example quality — such as language resource level, grammatical correctness, and the presence of incorrect or missing translations — affect translation performance. Each of these experimental manipulations simulates common issues encountered in real-world translation datasets — such as missing parallel corpora, incorrect language annotations, and human-induced noise. Understanding the specific ways in which example quality impacts performance is crucial for optimizing LLM-based translation systems, particularly given the variability and noise inherent in real-world translation data.

We begin by establishing baseline performance using standardized prompts containing correctly aligned in-context examples. These baseline prompts provide a reliable reference point against which the impact of manipulated example quality can be measured. Subsequent experiments systematically explore conditions reflecting realistic challenges: the absence of source sentences (Section 4.1.2), mislabeled or entirely incorrect target languages (Section 4.1.3), mismatches between source sentences and translations (Section 4.1.4), and grammatical inaccuracies such as reordered words and spelling mistakes (Section 4.1.5).

### 4.1.1. Baseline

To establish baseline performance for evaluating the impact of in-context example quality on machine translation using Large Language Models (LLMs), we construct prompts following standardized chat templates. Each prompt includes a system instruction that explicitly defines the translation task, specifying both the source and target languages. Although the in-context examples are presented in various languages, both the system instruction and the language labels are written in English, as English templates have been shown to perform best for machine translation tasks (Zhang, Haddow, and Birch, 2023). In-context examples are uniformly randomly sampled from parallel sentence datasets (see Section 4.2.2) relevant to the specified language pair. These examples are then presented as a chat exchange: the user message contains a source-language sentence[1] prefixed by its language, and the assistant message provides the corresponding translation. Four prompt variants are used, incorporating 0, 2, 4, or 8 in-context examples. We limit the number of in-context examples to 8, as prior work indicates that increasing beyond a certain threshold yields only marginal performance gains (Zhang, Haddow, and Birch, 2023). Additionally, using 8 examples fully saturates our available GPU memory (24 GB). The final user message presents the sentence to be translated by the LLM. The exact prompt syntax depends on the specific model and is dynamically generated using the Huggingface API[2]. Figure 4.1 shows an example prompt with two in-context examples using the Llama 3.1 chat template syntax.

### 4.1.2. Target-Only Translations

In some cases, instance–translation pairs may not be available — for example, when translations are collected independently from their source texts or when only monolingual corpora exist for low-resource languages. Target-only sentences also often reflect the desired text style or domain of interest, which may guide the model's generation and improve translation quality. This experiment investigates whether language models still benefit when only target-language sentences are provided as in-context examples. The original prompt is modified by removing the source instances from the in-context examples, as illustrated in Figure 4.2.

### 4.1.3. Wrong Target Language

Previous work has examined the importance of correct label mappings for in-context examples across various classification tasks (Min et al., 2022b; Yoo et al., 2022; Jerry Wei et al., 2023). However, the impact of incorrect label mappings in machine translation remains unexplored. This study addresses this gap through two experiments. In the first, we explicitly set the prefixed language label to French for all translations, as shown in

[1] Although the actual input may consist of multiple sentences, for simplicity we refer to them collectively as a single sentence.

[2] `https://huggingface.co/docs/transformers/main/en/chat_templating`

[3] `https://www.llama.com/docs/model-cards-and-prompt-formats/llama3_1/`

```
<|begin_of_text|>

<|start_header_id|>system<|end_header_id|>
You are a professional translator.
Translate the German sentence into English.<|eot_id|>

<|start_header_id|>user<|end_header_id|>
German: Alex fing seinen Ball.
English:<|eot_id|>

<|start_header_id|>assistant<|end_header_id|>
Alex caught his ball.<|eot_id|>

<|start_header_id|>user<|end_header_id|>
German: Mia schrieb eine Notiz.
English:<|eot_id|>

<|start_header_id|>assistant<|end_header_id|>
Mia wrote a note.<|eot_id|>

<|start_header_id|>user<|end_header_id|>
German: Sam schwang seinen Schläger.
English:<|eot_id|>
```

Figure 4.1.: Example prompt for German-to-English translation using the Llama 3.1 chat template syntax[3]with two in-context examples. The system instruction has been simplified compared to the original, and the in-context examples use manually selected, simplified sentences rather than actual instances from datasets.

```
German: Alex fing seinen Ball.
English: Alex caught his ball.
German: Mia schrieb eine Notiz.
English: Mia wrote a note.

German: Sam schwang seinen Schläger.
English: ?
```

→

```
English: Alex caught his ball.
English: Mia wrote a note.

German: Sam schwang seinen Schläger.
English: ?
```

Figure 4.2.: Prompt example with the *Target Only Translations* modification. The syntax is simplified; the exact prompt format is shown in Figure 4.1.

Figure 4.3. In the second, we use Spanish translations instead of the correct target language, as illustrated in Figure 4.4.

```
German: Alex fing seinen Ball.          German: Alex fing seinen Ball.
English: Alex caught his ball.          French: Alex caught his ball.
German: Mia schrieb eine Notiz.   →     German: Mia schrieb eine Notiz.
English: Mia wrote a note.              French: Mia wrote a note.

German: Sam schwang seinen Schläger.    German: Sam schwang seinen Schläger
English: ?                              English: ?
```

Figure 4.3.: Prompt example with the *Wrong Target Language Label* modification.

```
German: Alex fing seinen Ball.          German: Alex fing seinen Ball.
English: Alex caught his ball.          Spanish: Alex encontró su balón.
German: Mia schrieb eine Notiz.   →     German: Mia schrieb eine Notiz.
English: Mia wrote a note.              Spanish: Mia escribió una nota.

German: Sam schwang seinen Schläger.    German: Sam schwang seinen Schläger
English: ?                              English: ?
```

Figure 4.4.: Prompt example with the *Wrong Target Language* modification.

### 4.1.4. Mismatched Translations

To further investigate which components of in-context examples most influence the model's ability to perform in-context learning, we propose an experiment where all in-context examples contain incorrect translations, as shown in Figure 4.5. Each source sentence is paired with a randomly selected target-language sentence from the dataset, resulting in fully mismatched instance–translation pairs. This experiment tests whether the model relies on semantic alignment between source and target sentences or can still leverage target-language priors (e.g., exposure to fluent English) or structural patterns (e.g., consistent punctuation or sentence structure).

### 4.1.5. Grammatical Errors

Existing translations are often produced by humans and may contain grammatical errors or spelling mistakes. Ideally, the model should remain robust to such noise in the in-context examples. However, prior work has not systematically investigated the impact of these types of disturbances. Since such imperfections are common in real-world data (Dahlmeier,

**German**: Alex fing seinen Ball.
**English**: Alex caught his ball.
**German**: Mia schrieb eine Notiz.
**English**: Mia wrote a note.

**German**: Sam schwang seinen Schläger.
**English**: ?

→

**German**: Alex fing seinen Ball.
**English**: *Noah found a coin*.
**German**: Mia schrieb eine Notiz.
**English**: *Sophia took a photo*.

**German**: Sam schwang seinen Schläger
**English**: ?

Figure 4.5.: Prompt example with the *Mismatched Translations* modification.

Ng, and Wu, 2013; Belinkov and Bisk, 2018; Hagiwara and Mita, 2019), understanding their effect on in-context learning is essential for practical deployment. To address this gap, we propose two experiments: (1) randomly reordering words within a sentence (Section 4.1.5.1), and (2) introducing spelling errors into sentences (Section 4.1.5.2).

#### 4.1.5.1. Reordered Words

We simulate grammatical errors by reordering words within a sentence. This is implemented by iterating over each word and, with a probability of 20% or 40%, removing the word and reinserting it at a random position within the sentence. Figure 4.6 illustrates this procedure applied to target-language translations. We apply this method under three conditions: reordering only the source sentence, only the target sentence, or both. Combined with the two noise levels (20% and 40%), this yields a total of six experimental settings.

**German**: Alex fing seinen Ball.
**English**: Alex caught his ball.
**German**: Mia schrieb eine Notiz.
**English**: Mia wrote a note.

**German**: Sam schwang seinen Schläger.
**English**: ?

→

**German**: Alex fing seinen Ball.
**English**: Alex <> his ball *caught*.
**German**: Mia schrieb eine Notiz.
**English**: Mia *a* wrote <> note.

**German**: Sam schwang seinen Schläger
**English**: ?

Figure 4.6.: Prompt example with the *Reordered Words* modification applied to target sentences. Angle brackets <> indicate the original word positions for visualization purposes only and are not included in the actual prompt.

#### 4.1.5.2. Spelling Mistakes

We simulate spelling errors by introducing character-level noise, following a procedure similar to that used for word reordering in Section 4.1.5.1. Specifically, we iterate over each adjacent pair of characters and, with a probability of 20% or 40%, swap the two. Figure 4.7

demonstrates this applied to target-language translations. As before, we consider three conditions: modifying only the source sentence, only the target sentence, or both, resulting in six experimental settings.

**German**: Alex fing seinen Ball.
**English**: Alex caught his ball.
**German**: Mia schrieb eine Notiz.
**English**: Mia wrote a note.

**German**: Sam schwang seinen Schläger.
**English**: ?

→

**German**: Alex fing seinen Ball.
**English**: Alex cau*hg*t his ball.
**German**: Mia schrieb eine Notiz.
**English**: Mia wrote a n*to*e.

**German**: Sam schwang seinen Schläger
**English**: ?

Figure 4.7.: Prompt example with the *Spelling Mistakes* modification applied to target sentences. Spelling errors are indicated by italicized character pairs.

## 4.2. Technical Setup

The prompts generated in Section 4.1 are used as input to the Large Language Models (LLMs) to produce output translations. Inference is executed via the HuggingFace Transformers API, which downloads the models given their identifiers. The prompts are tokenized before inference. LLMs are run using greedy decoding, and generation is performed in batches. To save time and avoid repeated failures due to variable memory demands across batches, a dynamic batching algorithm monitors VRAM usage and adjusts the batch size accordingly: it increases the batch size if sufficient VRAM is available and decreases it when out-of-memory errors occur. All experiments are conducted on a university server equipped with an NVIDIA Titan RTX GPU with 24 GB of VRAM.

### 4.2.1. Large Language Models

We aim to assess the translation capabilities of both general-purpose instruction-tuned models and models specifically fine-tuned for machine translation, to understand when it is beneficial to switch to a more specialized model. General-purpose models are more commonly used in real-world applications (e.g., GPT, Gemini, Llama), making their evaluation critical for practical relevance. For this role, we select Llama 3.1 8B Instruct (Grattafiori et al., 2024), an open-source, state-of-the-art model frequently used in scientific benchmarks. To compare, we include TowerInstruct 7B v0.2 (Alves et al., 2024), which is specifically fine-tuned for machine translation instructions and has demonstrated strong zero-shot performance across multiple languages. The "Instruct" designation indicates that these models are fine-tuned for instruction-following tasks. For brevity, we refer to them as Llama 3.1 and Tower. Both models follow a decoder-only transformer architecture (Vaswani et al., 2023) and are pretrained using next-token prediction on unlabeled multilingual text. They

| Model | Parameters | HuggingFace-ID |
|---|---|---|
| Llama 3.1 | 8B | meta-llama/Meta-Llama-3.1-8B-Instruct |
| Tower v0.2 | 7B | Unbabel/TowerInstruct-7B-v0.2 |
| Llama 2 | 7B | meta-llama/Llama-2-7b-chat-hf |

Table 4.1.: Large Language Models used for inference, along with their parameter sizes and corresponding HuggingFace identifiers.

are subsequently fine-tuned on a selected set of languages, including English and German. Czech, Ukrainian, and Nepali are not part of the fine-tuning set.

#### 4.2.1.1. Llama 3.1

Llama 3.1 adopts a slightly modified architecture compared to Llama 2 (Touvron et al., 2023). However its main advancements lie in improved training data quality and fine-tuning procedures. The model is pretrained on 15T multilingual tokens. Although Llama supports multiple languages, in contrast to Tower, Llama is not specifically fine-tuned for machine translation tasks. Instead, it serves a more general-purpose role, with capabilities in code generation, mathematical reasoning, and tool use such as interacting with search engines or code interpreters (Grattafiori et al., 2024).

#### 4.2.1.2. Tower

Tower builds on the pretrained Llama 2 (Touvron et al., 2023) model and applies additional fine-tuning strategies with a primary focus on machine translation tasks. It is initially pretrained on 1.8T tokens (Touvron et al., 2023), followed by further pretraining on 20B cross-lingual tokens. Unlike Llama 3.1, Tower incorporates post-training data that includes few-shot translation prompts. While translation is the main focus, 43% of the post-training data consists of general-purpose tasks such as code generation and conversational interactions (Alves et al., 2024).

#### 4.2.1.3. Llama 2

We include Llama 2 (Touvron et al., 2023) to assess the effectiveness of Tower's additional pretraining and fine-tuning strategies. To ensure comparability and reduce computational cost, we evaluate Llama 2 only on unperturbed in-context examples, aligning with the Tower baseline configuration. Llama 2 is primarily pretrained (on 1.8T tokens) and instruction fine-tuned on English data. Unlike Tower, no specific fine-tuning targeting multilinguality or translation tasks is applied (Touvron et al., 2023). This setup allows us to isolate the impact of Tower's translation-oriented fine-tuning relative to its base model.

### 4.2.2. Datasets

We require high-quality parallel texts across multiple languages with varying resource levels to systematically evaluate translation performance under controlled conditions. For

| Dataset | Languages | Instances | Avg. Instance Length |
|---|---|---|---|
| Flores+ *devtest* | English | 1012 | 131.40 |
| Flores+ *devtest* | German | 1012 | 152.99 |
| Flores+ *devtest* | Czech | 1012 | 126.75 |
| Flores+ *devtest* | Ukrainian | 1012 | 133.91 |
| Flores+ *devtest* | Nepali | 1012 | 126.40 |
| Flores+ *dev* | English | 997 | 126.57 |
| Flores+ *dev* | German | 997 | 148.00 |
| Flores+ *dev* | Czech | 997 | 123.18 |
| Flores+ *dev* | Ukrainian | 997 | 130.28 |
| Flores+ *dev* | Nepali | 997 | 122.15 |
| WMT 2023 | English-German | 557 | EN: 354.78, DE: 413.42 |
| WMT 2023 | English-Czech | 2074 | EN: 96.45, CS: 95.22 |
| WMT 2023 | English-Ukrainian | 2074 | EN: 96.45, UK: 99.14 |
| WMT 2023 | Czech-Ukrainian | 2017 | CS: 81.94, UK: 87.82 |
| WMT 2024 | English-German | 998 | EN: 185.64, DE: 216.03 |
| WMT 2024 | English-Czech | 998 | EN: 185.64, CS: 181.66 |
| WMT 2024 | English-Ukrainian | 998 | EN: 185.64, UK: 191.45 |
| WMT 2024 | Czech-Ukrainian | 2317 | CS: 79.74, UK: 85.18 |

Table 4.2.: Summary of dataset statistics, including the number of instances per language and average instance length in characters. Flores contains parallel instances shared across all listed languages. WMT provides distinct instances for each language pair. Flores+ *devtest* and WMT 2023 are used to sample in-context examples; Flores+ *dev* and WMT 2024 serve as translation targets.

this purpose, we use the Flores+ (NLLB Team et al., 2024) and WMT 2023/2024 (Kocmi, Avramidis, Bawden, Bojar, Dvorkovich, Federmann, Fishel, Freitag, Gowda, Grundkiewicz, Haddow, Koehn, et al., 2023; Kocmi, Avramidis, Bawden, Bojar, Dvorkovich, Federmann, Fishel, Freitag, Gowda, Grundkiewicz, Haddow, Karpinska, et al., 2024) datasets. These publicly available, widely adopted resources are specifically designed for machine translation tasks and provide parallel sentence pairs in a broad range of languages. These publicly available and widely used resources are designed for machine translation tasks, providing parallel sentences in multiple languages. A detailed overview of the datasets is provided in Table 4.2. To avoid any overlap between in-context (IC) examples and instances to be translated, we use separate sub-datasets. Flores+[4] provides two subsets: "dev" and "devtest." We use "dev" for IC examples and "devtest" for translation targets. WMT provides a single test set per domain; therefore, we use different years to ensure separation. Specifically, WMT 2023[5] for IC examples, and WMT 2024[6] for instances to translate. The models mentioned in Section 4.2.1 do not include these datasets in their training data.

### 4.2.3. Languages

We aim to evaluate translation performance across both high- and low-resource languages, and to examine how language model behavior varies depending on whether a language was seen during instruction fine-tuning. For this reason, we select language pairs that span a range of resource levels and training exposure. Czech and Ukrainian are included due to their presence in both WMT and Flores+, providing a large and diverse set of examples. Nepali, available only in Flores+, is chosen to represent an additional low-resource language. Given that Ukrainian shares significant linguistic features with Russian and Nepali with Hindi, these pairs provide an opportunity to test whether the models can accurately translate into the intended target language or if they inadvertently conflate them with their closely related counterparts. German, a high-resource language, is included for reference but limited to combinations with English to reduce inference cost. Figure 4.3 lists the languages used for prompt generation. In total, 14 directed language pairs are employed:

```
DE → EN, EN → DE, CS → EN, EN → CS
UK → EN, EN → UK, NE → EN, EN → NE
CS → UK, UK → CS, NE → UK, UK → NE
NE → CS, CS → NE
```

The selection of Czech and Ukrainian is further constrained by WMT coverage: only language pairs present in both WMT 2023 and WMT 2024 are considered.

[4] At the time of development, Flores+ was available at `https://github.com/openlanguagedata/flores`. At the time of writing, it has been migrated to `https://huggingface.co/datasets/openlanguagedata/flores_plus`

[5] `https://github.com/wmt-conference/wmt23-news-systems`

[6] `https://github.com/wmt-conference/wmt24-news-systems`

[7] `https://en.wikipedia.org/wiki/List_of_ISO_639_language_codes`

| Language | Code | Resource Level | Included in Original Fine-Tuning |
|---|---|---|---|
| English | EN | high | yes |
| German | DE | high | yes |
| Czech | CS | medium | no |
| Ukrainian | UK | low | no |
| Nepali | NE | low | no |

Table 4.3.: Languages used for prompt generation, including ISO 639 codes[7], resource levels, and whether the models (Llama 3.1 and Tower) were fine-tuned on them.

## 4.3. Evaluation

To extract meaningful insights from raw LLM-generated translations, we require quantitative measures that allow for systematic comparison. In Section 4.3.1, we introduce automatic metrics that assign scores reflecting semantic adequacy, fluency, and overall translation quality — crucial for assessing how well the models convey intended meaning. In Section 4.3.2, we also apply automatic language identification to verify whether models adhere to target language instructions, as incorrect language usage undermines validity. The results of these evaluations are examined further in Chapter 5.

### 4.3.1. Metrics

To estimate how well each LLM-generated translation conveys the intended meaning, we compute three complementary evaluation scores. The first is COMET-22 (Rei et al., 2022), a neural metric trained on human judgments that uses both the source and a reference translation to evaluate semantic adequacy and fluency with respect to the intended output. We also include COMET-Kiwi (Rei et al., 2022), a reference-free variant that relies solely on the source input. While it offers insight into the general quality of the output sentence, it may fail to detect translation errors as it lacks access to the intended target language. Finally, we use SacreBLEU[8] (Post, 2018), a fast n-gram-based metric comparing the output to a reference. Despite its limitations in handling synonymous or semantically equivalent phrasing, SacreBLEU provides a useful complementary signal and can be more robust when COMET struggles with language recognition.

### 4.3.2. Output Language Identification

During experimentation, we observed that LLMs occasionally fail to translate into the intended target language, instead producing output in an unintended language. This phenomenon has also been reported in prior work (Bawden and Yvon, 2023). To better analyze this behavior, we perform language identification on each LLM output to verify correct language usage. For this, we use the fastText model `lid.179.ftz`[9] from Meta (Joulin, Grave, Bojanowski, Douze, et al., 2016; Joulin, Grave, Bojanowski, and Mikolov,

[8] BLEU|nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.0.0

[9] `https://fasttext.cc/docs/en/language-identification.html`

2016), which supports 176 languages, including all those considered in this thesis (see Section 4.2.3).

# 5. Results and Analysis

This chapter presents the experimental results gathered through the process described in chapter 4 and accompanying analyses, aiming to answer the research questions regarding the influence of in-context example quality on machine translation performance. The findings are derived from a series of controlled experiments that systematically vary the in-context examples—ranging from the number of examples provided to specific manipulations such as target-only translations, language mismatches, and induced grammatical errors. The models used are Llama 3.1 and Tower. We evaluate outputs using SacreBLEU, COMET-22, and COMET-Kiwi, and also perform language identification to verify adherence to target language specifications. For simplicity, we only report COMET-22 scores unless other metrics exhibit divergent trends.

First, in Baseline (Section 5.1), we establish reference performance using unmodified prompts and examine how language selection and the number of few-shot examples influence model outputs. Subsequent sections explore specific experimental conditions designed to test the robustness and adaptability of the models. Specifically, we examine the efficacy of target-only in-context examples in scenarios with limited or monolingual data (Section 5.2), investigate the consequences of incorrect language information through mislabeled or entirely incorrect translations (Section 5.3), analyze the importance of semantic alignment via mismatched source-target pairs (Section 5.4), and assess the resilience of models to input noise caused by controlled grammatical disruptions (Section 5.5).

An overall discussion of the findings is provided in Chapter 6.

## 5.1. Baseline

In this section, we establish baseline performance to serve as a reference point for subsequent analyses. We examine how the choice of language pairs and the number of few-shot examples provided influence translation quality, focusing primarily on assessing the extent to which the models—Llama 3.1 and Tower—exhibit in-context learning capabilities. By evaluating these baseline conditions, we provide a foundation for interpreting the effects of the experimental manipulations detailed in later sections.

### 5.1.1. Llama Exhibits In-Context Learning

Across all evaluated language pairs, Llama 3.1 demonstrates improved translation quality as the number of in-context examples increases, as shown in Figures 5.1 and 5.3. This aligns with the general expectation that more examples enhance translation performance, as previously demonstrated by T. B. Brown et al. (2020). This suggests that fine-tuning

strategies can further improve performance until few-shot prompting becomes redundant, with zero-shot performance approaching optimality.

### 5.1.2. Tower Shows Limited In-Context Learning on Fine-Tuned Languages

Tower outperforms Llama 3.1 only on the English–German and German–English language pairs. In both directions, Tower's zero-shot performance exceeds Llama's 8-shot results (see Figure 5.1). These are the only pairs for which Tower was fine-tuned on both source and target languages, indicating that Tower performs competitively only when both languages were seen during fine-tuning. This is further supported by the results in Figure 5.3, which show that Tower underperforms compared to Llama 3.1 on language pairs not included in its fine-tuning set.

Notably, Tower's performance does not improve with additional in-context examples for German–English, suggesting it fails to leverage in-context learning. In fact, for both German–English and English–German, performance degrades beyond 4-shot prompts, even falling below zero-shot levels. Tower is fine-tuned with up to 5-shot instruction prompts (Alves et al., 2024), which may explain its limited robustness to longer prompts. Architectural constraints also play a role: Llama 3.1 supports a 128K-token context window (Grattafiori et al., 2024), while Tower is limited to 4K tokens (Alves et al., 2024). Empirical analysis shows that 8-shot prompts can occupy half of Tower's context window, likely reducing attention to earlier tokens containing translation instructions.

Since Tower is based on Llama 2 (Touvron et al., 2023) – only further pretrained and instruction fine-tuned for translation tasks (Alves et al., 2024) – we report Llama 2 scores in Figure 5.2 for direct comparison. Tower clearly outperforms Llama 2 by a wider margin than it does Llama 3.1 (cf. Figure 5.1). In all settings, Llama 2 demonstrates in-context learning, with few-shot scores exceeding zero-shot performance. This gain is more pronounced for German→English, but even for English→German, the improvement from zero-shot to 4-shot is higher for Llama 2 (1.8 percentage points) than for Tower (0.4 percentage points), which represents the maximum gain observed in that direction. These results highlight the effectiveness of Tower's translation-optimized fine-tuning strategies, even in the absence of strong in-context learning behavior.

### 5.1.3. Tower's Language Output Improves with In-Context Examples

While Llama 3.1 handles a wide range of target languages reliably, Tower struggles with low-resource languages, especially when English is not the source language. In several cases, Tower fails to generate output in the correct target language. We observe three distinct error modes:

- Translating into English instead of the target language, particularly in zero-shot settings. When English is also the source language, Tower either copies the input or paraphrases it in approximately half of the tested cases.
- Translating into the source language instead of the target.

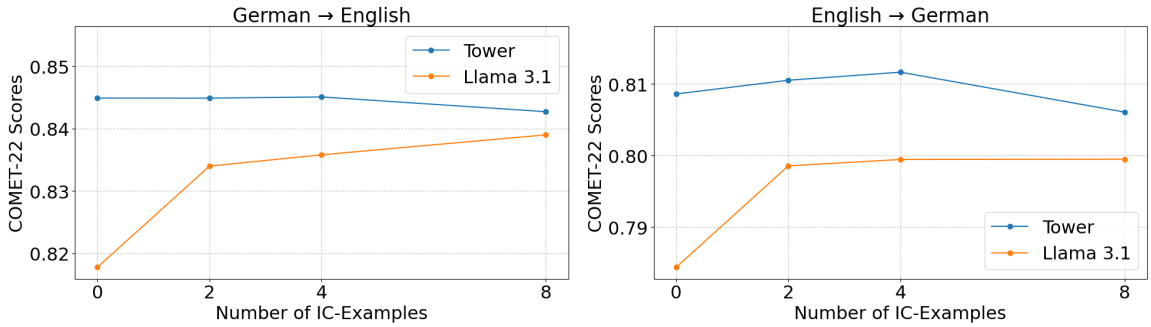



Figure 5.1.: Tower outperforms Llama 3.1 in average baseline COMET-22 scores when all translation directions involve languages seen during fine-tuning. Left: German-to-English translations. Right: English-to-German translations. Scores are shown for Tower (blue) and Llama 3.1 (orange).

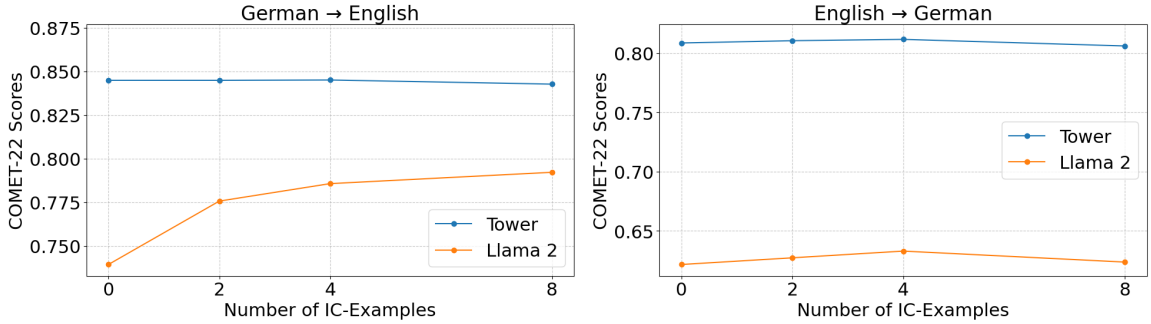



Figure 5.2.: Tower achieves higher average baseline COMET-22 scores than Llama 2. Llama 2 demonstrates in-context learning for German→English, whereas Tower does not. Left: German-to-English translations. Right: English-to-German translations. Scores are shown for Tower (blue) and Llama 2 (orange).

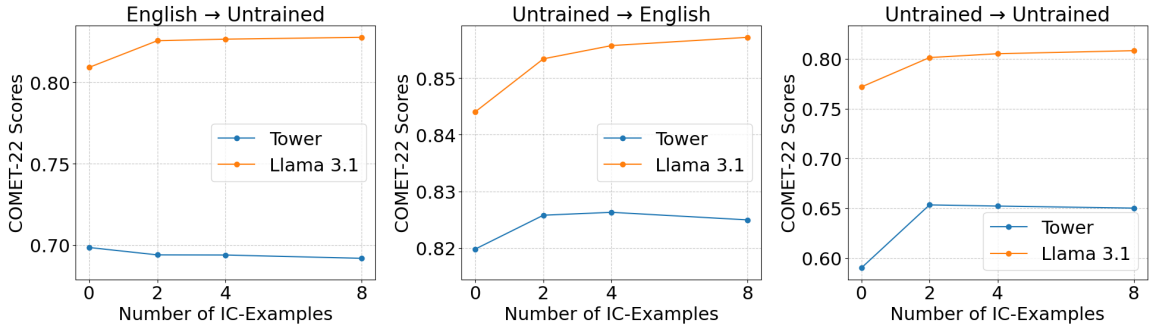



Figure 5.3.: Llama 3.1 outperforms Tower in average baseline COMET-22 scores when at least one language in the translation direction was not seen during fine-tuning. Left: English to untrained language translations. Middle: Untrained language to English translations. Right: Untrained language to untrained language translations. Untrained languages are Czech, Ukrainian, and Nepali – languages on which neither model was fine-tuned. Scores are shown for Tower (blue) and Llama 3.1 (orange).

- Translating into a linguistically related but incorrect language, such as Russian instead of Ukrainian or Hindi instead of Nepali.

These behaviors are illustrated in Figure 5.4. Despite these issues, increasing the number of in-context examples improves Tower's ability to generate outputs in the correct target language. Similar improvements with few-shot prompting have been observed in prior work (Bawden and Yvon, 2023), where incorrect language generation is greatly reduced in the few-shot setting. In Figure 5.5 We also report SacreBLEU scores for translations into non-fine-tuned languages. As the output language quality improves, BLEU scores increase accordingly – an expected outcome, given BLEU's reliance on n-gram overlap. However, contradicting results emerge when comparing SacreBLEU with COMET-22 scores for English → Untrained (Compare Figures 5.3 and 5.5). While SacreBLEU scores rise, COMET-22 scores decrease. An increase in BLEU may suggest better surface-level alignment with references, but this can come at the cost of semantic adequacy and fluency. COMET, which better correlates with human judgments, may penalize overly literal translations that miss contextual nuances. With additional in-context examples, the model may overfit to specific structures or phrasings, increasing BLEU due to more n-gram matches, yet decreasing COMET as output flexibility and meaning preservation decline. Ultimately, this phenomenon requires further investigation.

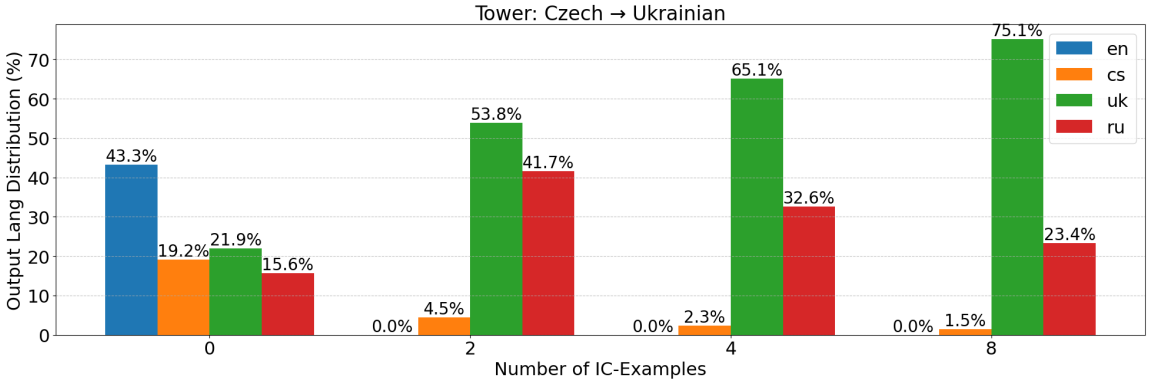



Figure 5.4.: Tower translation outputs for the Czech-to-Ukrainian direction. In the zero-shot setting, some outputs are incorrectly translated into English. Across all few-shot settings, two persistent error types appear: translations into the source language (Czech) and into a related language (Russian instead of Ukrainian).

## 5.2. Target-Only Translations

In this section, as described in Section 4.1.2, we investigate how translation performance is affected when in-context examples contain only target-side translations, simulating conditions of limited or monolingual data. We analyze whether exposure solely to target language sentences, without corresponding source-target alignments, can still enhance translation quality. This evaluation provides insights into the extent to which the models rely on explicit source-target mappings versus general target language patterns.

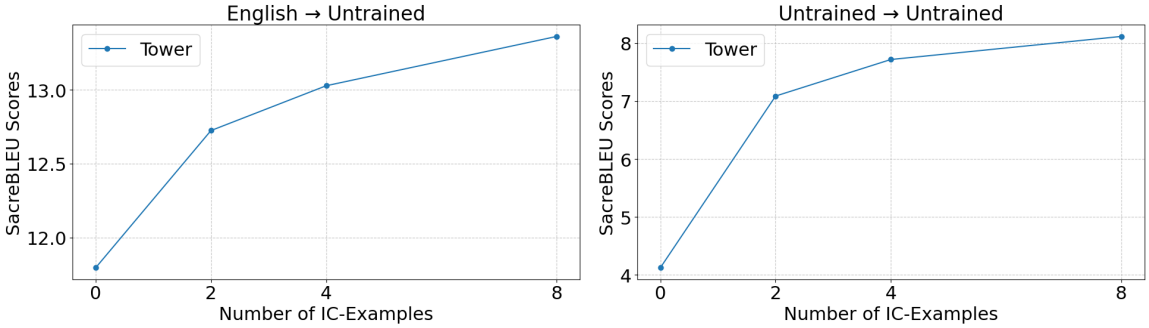


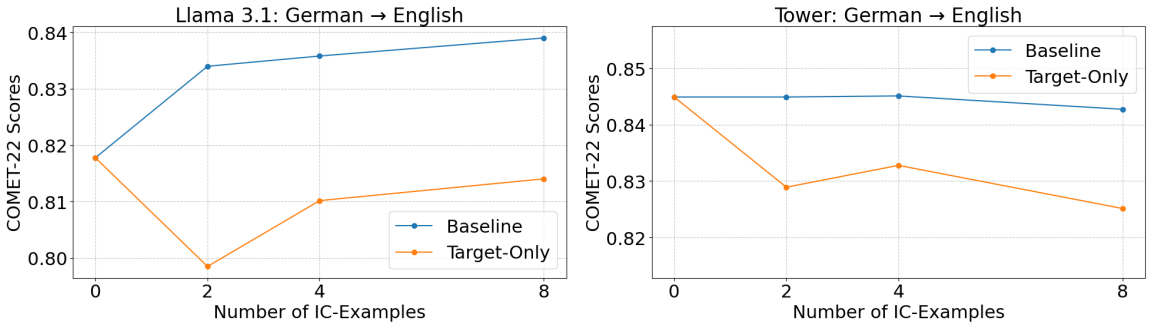

Figure 5.5.: According to SacreBLEU scores, Tower improves translation quality for languages outside the fine-tuned set. Left: English to untrained language translations. Right: Untrained language to untrained language translations. Untrained languages are Czech, Ukrainian, and Nepali—languages on which neither model was fine-tuned. Y-axis scales are consistent across models (offsets may differ) for comparability.

### 5.2.1. Baseline Outperforms Target-Only Prompts

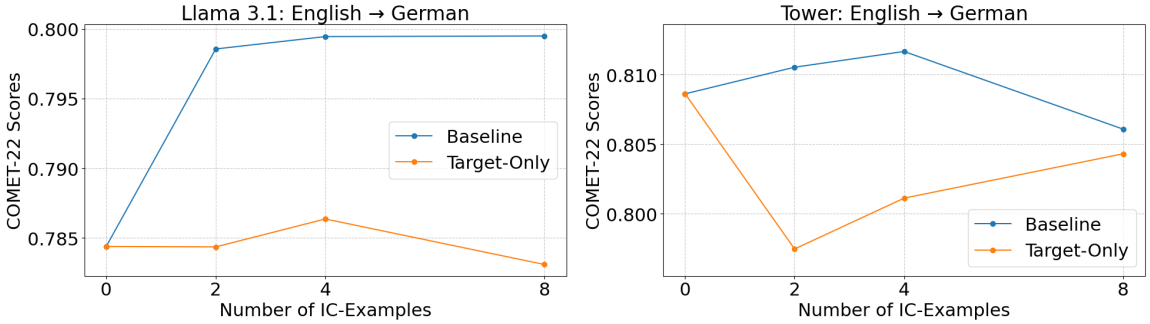
As Figures 5.6 and 5.7 show, both Llama 3.1 and Tower consistently achieve better translation quality in baseline settings compared to target-only prompts across all language pairs. This demonstrates that both models significantly leverage source-to-target mappings provided by full in-context examples, rather than simply relying on exposure to target language patterns. This finding aligns with prior work by Zhang, Haddow, and Birch (2023), which shows that using monolingual examples for prompting degrades translation quality. Specifically, for Llama, zero-shot baseline performance surpasses few-shot target-only results in English→German and German→English translations, suggesting minimal utility from target-only prompts in settings involving fine-tuned languages, as illustrated in Figure 5.6. Tower exhibits similar behavior, generally showing little to no improvement from target-only prompts compared to baseline performance.

### 5.2.2. Translation Improvements Limited to Non-Fine-Tuned Languages

As shown in Figure 5.7 for both models, improvements from target-only prompts are primarily observed when translating into languages not included during fine-tuning. Llama achieves slightly better performance in these scenarios with target-only prompts compared to zero-shot translations, indicating that exposure to target language structures alone can marginally boost translation quality. Tower also benefits in settings involving translations between languages not seen during fine-tuning, with modest improvements evident in few-shot target-only scenarios. This suggests that when source-target mappings are unavailable, any exposure to target translations can partially assist in generating more accurate outputs.

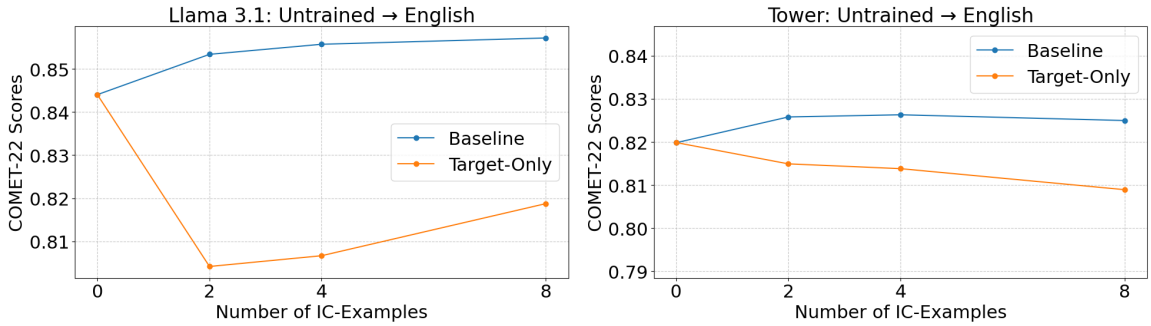



(a) German → English

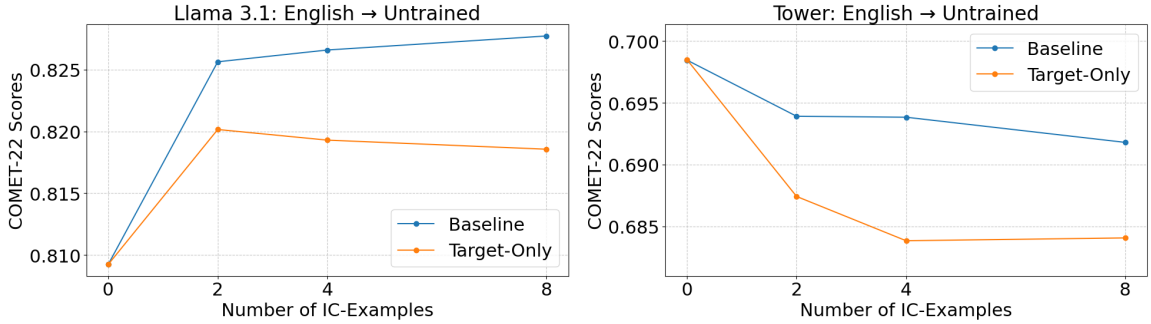



(b) English → German

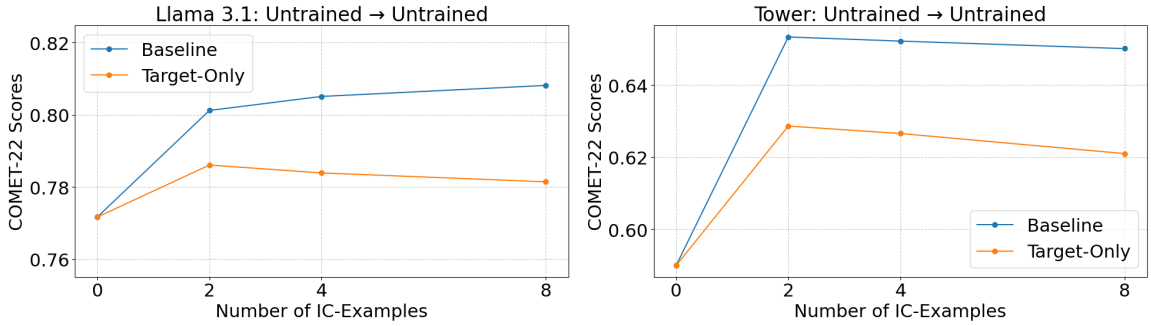
Figure 5.6.: Average COMET-22 scores for German–English translations in the Target-Only setting. In-context examples include only the target translation. Left: Llama 3.1; Right: Tower. Y-axis scales are consistent across models (offsets may differ) for comparability.




(a) Untrained → English




(b) English → Untrained

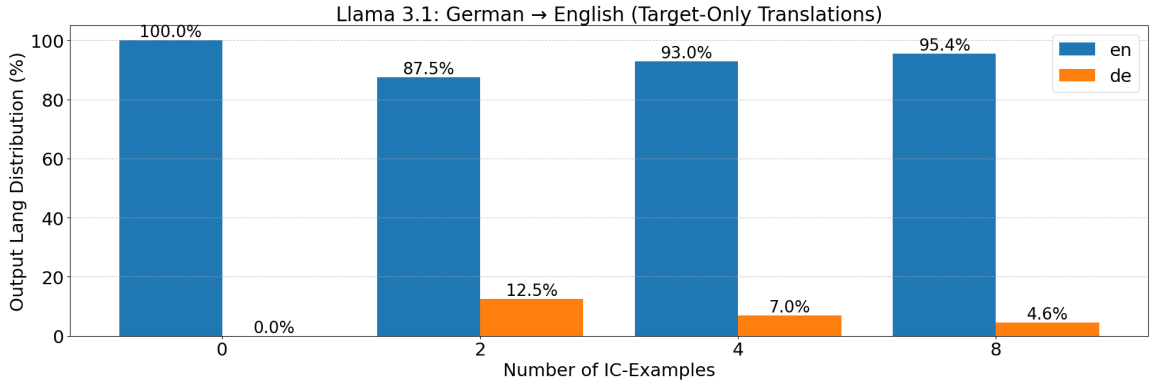



(c) Untrained → Untrained

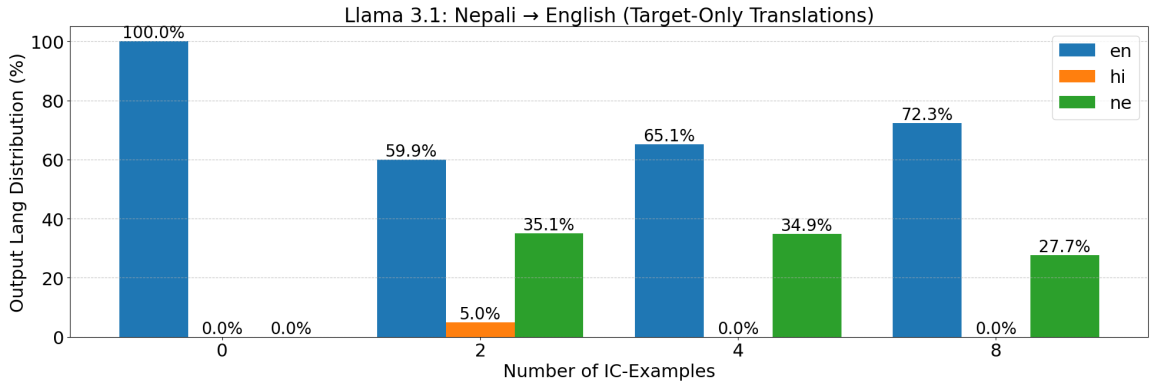
Figure 5.7.: Average COMET-22 scores for translations involving non-fine-tuned languages in the Target-Only setting. In-context examples include only the target translation. Left: Llama 3.1; Right: Tower. *Untrained* refers to languages neither model was fine-tuned on (Czech, Ukrainian, Nepali). Y-axis scales are consistent across models (offsets may differ) for comparability.

### 5.2.3. Target-Only Prompts Reduce Language Output Accuracy

Introducing target-only in-context examples negatively impacts language accuracy, particularly evident in Llama's performance when translating into English (see Figure 5.6a and 5.7a). For instance, transitioning from zero-shot to two-shot target-only prompts triggers a significant drop in performance due to the model incorrectly generating outputs in the source language or an entirely different language. Specifically, in German→English translations (Figure 5.8a), German outputs begin to appear at the two-shot level. Similarly, for Nepali→English translations (Figure 5.8b), both Nepali and Hindi outputs emerge starting from two-shot prompts. However, accuracy partially recovers with four-shot and eight-shot prompts, indicating that increased exposure eventually aids language stabilization. Tower exhibits increased difficulty in maintaining correct target language outputs when using target-only examples, as shown in Figure 5.9 for English→Ukrainian translations. The distribution reveals that target-only prompting leads to less accurate language outputs compared to the baseline, highlighting Tower's reliance on explicit source-target mappings.




(a) German → English




(b) Nepali → English

Figure 5.8.: Language distributions in Target-Only translations using Llama 3.1. Additional outputs in languages other than English emerge due to the absence of source sentences and are not observed in the baseline.

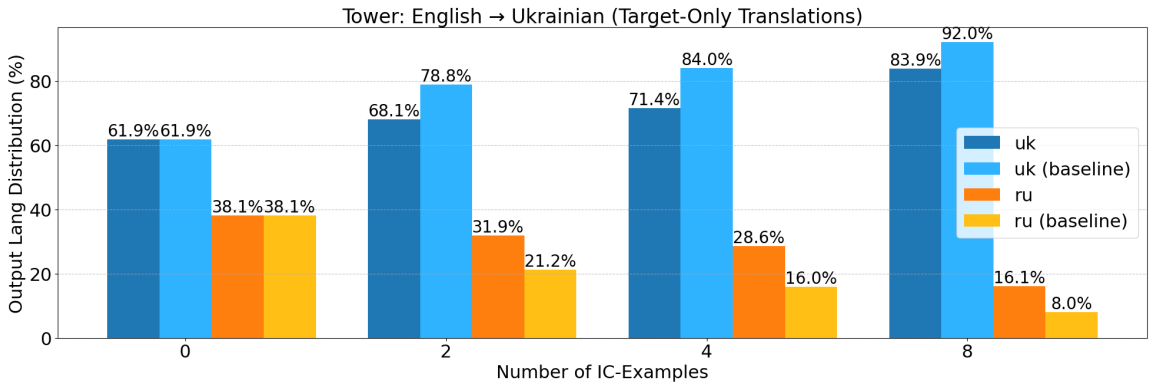



Figure 5.9.: Language distributions for English→Ukrainian Target-Only translations using Tower. Darker shades indicate Target-Only outputs; lighter shades represent corresponding baseline outputs. The results show that Target-Only translations yield less accurate language outputs compared to the baseline.

### 5.2.4. Importance of Source-Target Mappings

The observed results strongly suggest that explicit mappings between source sentences and their translations significantly influence translation performance. Both models clearly struggle more with correct language identification and generate lower-quality translations in target-only settings compared to their baseline counterparts. Hence, the presence of aligned source-target examples is essential for robust translation quality, particularly when dealing with languages outside the models' fine-tuned repertoire. While this study only examined the absence of source sentences in in-context examples, future work could investigate the impact of omitting target translations. Notably, prior evidence indicates that source sentences contribute more substantially to performance (Zaranis, Guerreiro, and Martins, 2024).

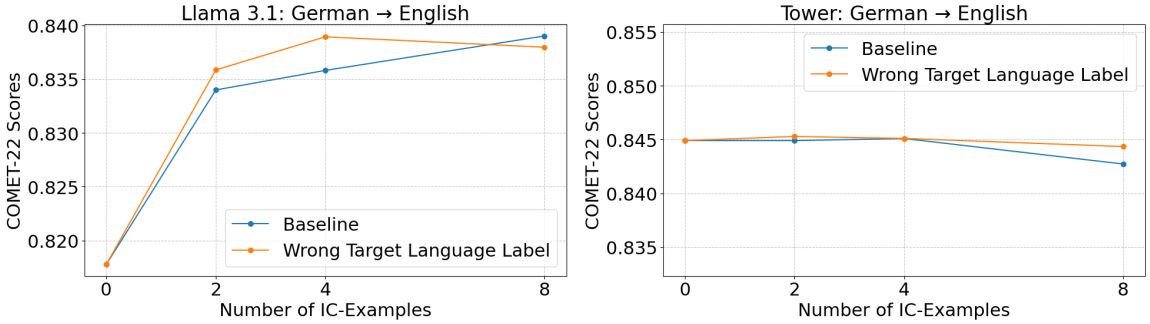
## 5.3. Wrong Target Language

In this section, as described in Section 4.1.3, we explore how translation performance is influenced when in-context examples contain incorrect target language information. Specifically, we evaluate scenarios where either language labels or entire translations provided in the examples do not match the intended target language. These experiments reveal contrasting behaviors between the Llama and Tower models, shedding light on the robustness of their in-context learning strategies and their reliance on explicit language identification cues versus semantic and contextual information.

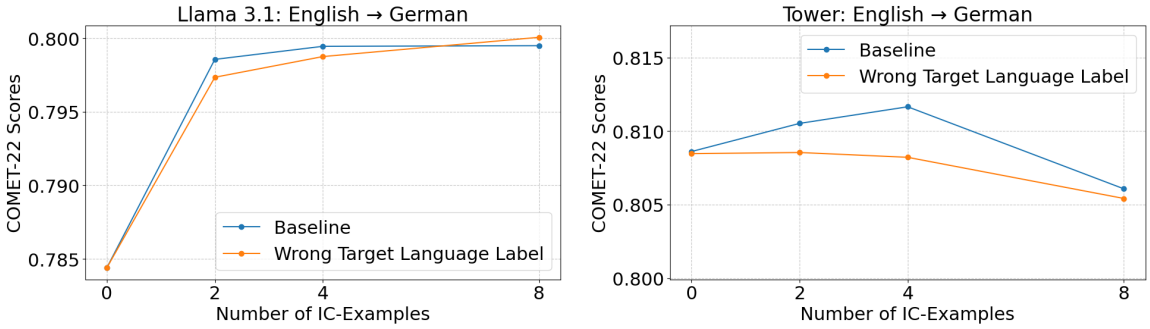
### 5.3.1. Models Ignore Wrong Language Labels

We first investigate the influence of incorrect language labels by setting the target language labels in the in-context examples to French, regardless of their actual target language. For instance, an English-to-German example would be labeled as "*English: <English sentence> French: <German translation>*". Both models demonstrate only negligible performance

variations compared to the baseline, which are likely due to random sampling of the few-shot examples. This behavior is consistent across fine-tuned and non-fine-tuned languages (see Figures 5.10 and 5.11). This strongly suggests that neither Llama nor Tower rely significantly on the explicit language labels within the prompts. Instead, both models prioritize meaningful cues from the translation pairs themselves, effectively ignoring superficial language labeling inaccuracies. These findings align with existing literature on label mapping robustness in classification tasks, where large language models have demonstrated adaptability to incorrect label mappings (Min et al., 2022b; Yoo et al., 2022; Jerry Wei et al., 2023). Further experimentation might explore scenarios where each example's language label is individually randomized rather than uniformly incorrect.




(a) German → English

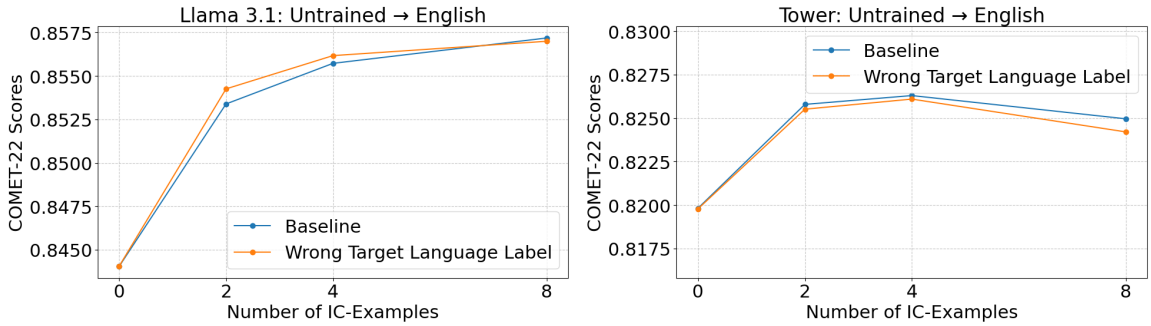



(b) English → German

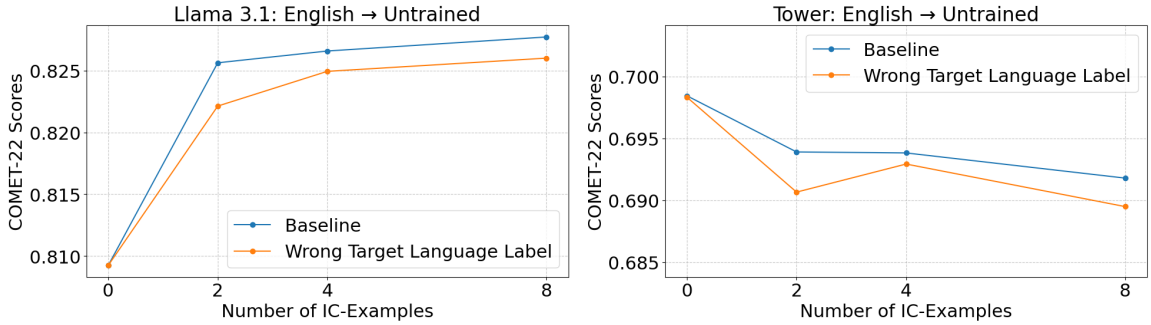
Figure 5.10.: Average COMET-22 scores for translations involving fine-tuned languages in the Wrong-Target-Language-Label setting. In-context translations are incorrectly prefixed with *French* instead of the actual target language. Left: Llama 3.1; Right: Tower. Y-axis scales are consistent across models (offsets may differ) for comparability. Performance differences relative to the baseline are negligible, suggesting that the models are robust to incorrect target language labels.

### 5.3.2. No In-Context Learning When Translating Into English for Tower

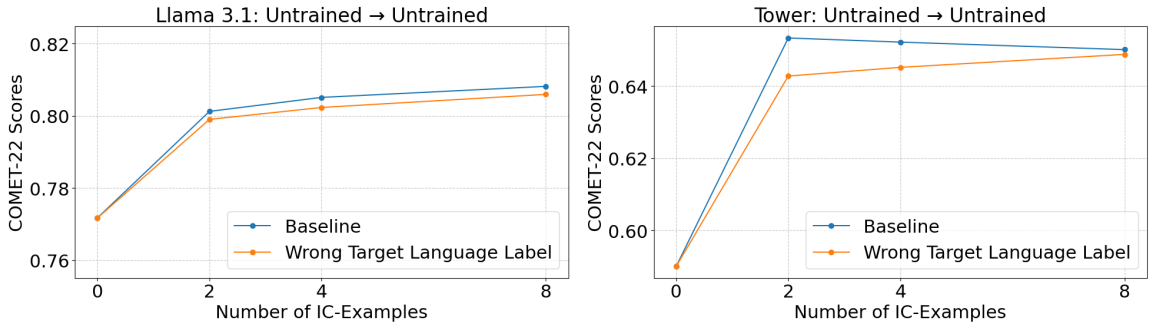
In the second experiment, we consistently used Spanish translations in all in-context examples, irrespective of the task's actual (non-Spanish) target language. When translating




(a) Untrained → English



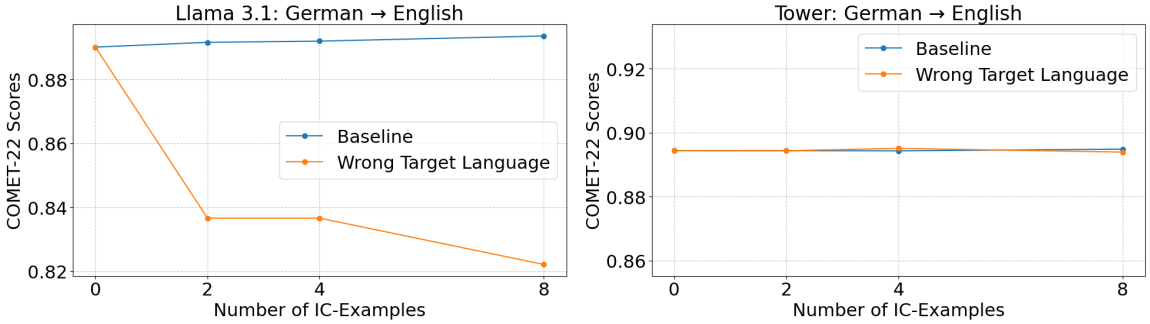

(b) English → Untrained



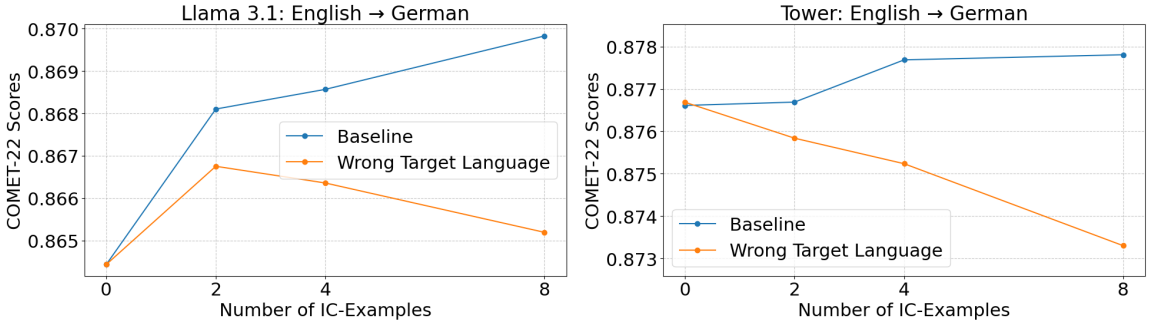

(c) Untrained → Untrained

Figure 5.11.: Average COMET-22 scores for translations involving non-fine-tuned languages in the Wrong-Target-Language-Label setting. In-context translations are incorrectly prefixed with *French* instead of the actual target language. Left: Llama 3.1; Right: Tower. *Untrained* refers to languages neither model was fine-tuned on (Czech, Ukrainian, Nepali). Y-axis scales are consistent across models (offsets may differ) for comparability. Performance differences relative to the baseline are negligible, suggesting that the models are robust to incorrect target language labels.

from any language into English, Tower's performance remains nearly unchanged compared to its baseline (see Figures 5.12a and 5.13a). Tower appears not to leverage the provided in-context examples at all in this setting. Given Tower's optimization specifically for translations into English and its fine-tuning on few-shot scenarios, this suggests that Tower may have learned to selectively disregard in-context examples when they provide no additional benefit, relying solely on its learned prior knowledge.




(a) German → English



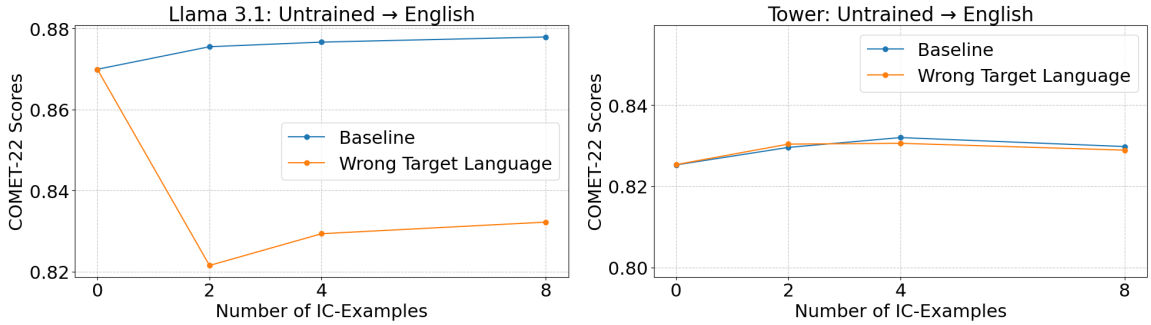

(b) English → German

Figure 5.12.: Average COMET-22 scores for translations involving fine-tuned languages in the Wrong-Target-Language setting. In-context translations are incorrectly prefixed with *French* instead of the actual target language. Left: Llama 3.1; Right: Tower. Y-axis scales are consistent across models (offsets may differ) for comparability. Llama and Tower exhibit contrasting behaviors, reflecting differences in fine-tuning strategies.

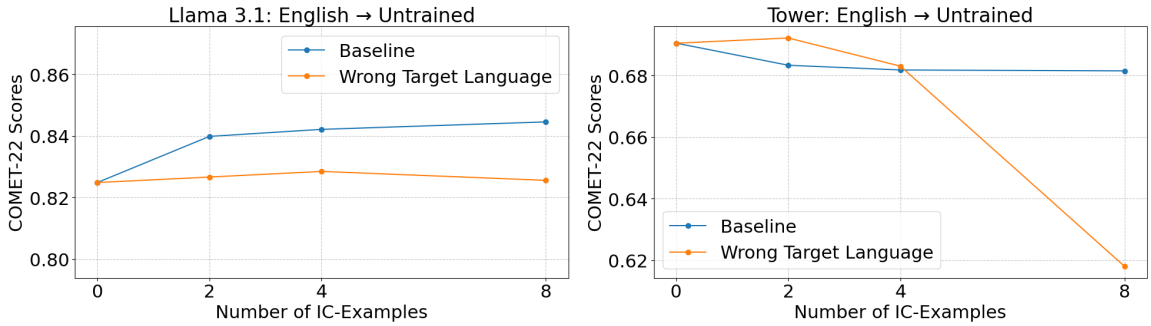
*Note: At first glance, English → German appears to show a significant performance drop. However, closer inspection of the COMET-22 y-axis reveals only minor deviations from baseline, indicating that both models handle this condition relatively well.*

### 5.3.3. Tower Relies on In-Context Examples for Non-Fine-Tuned Languages

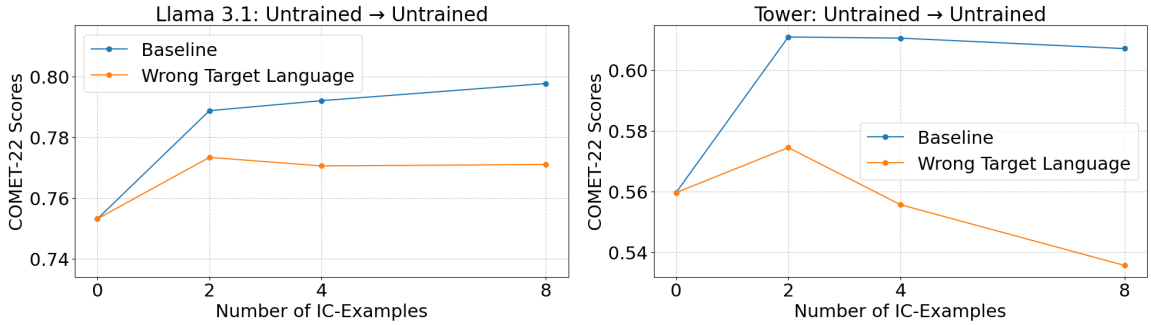
For translation tasks involving target languages Tower was not fine-tuned on (Czech, Ukrainian, Nepali), the presence of incorrect Spanish translations in the in-context ex-




(a) Untrained ⟶ English



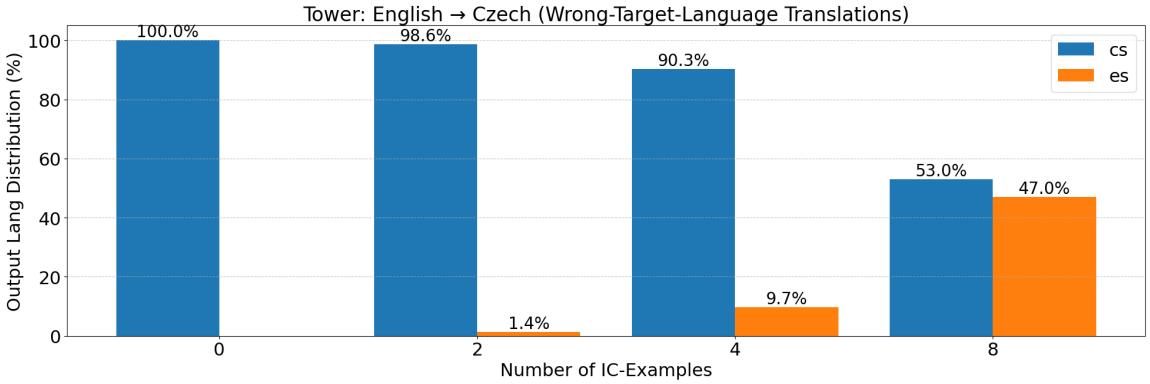

(b) English ⟶ Untrained



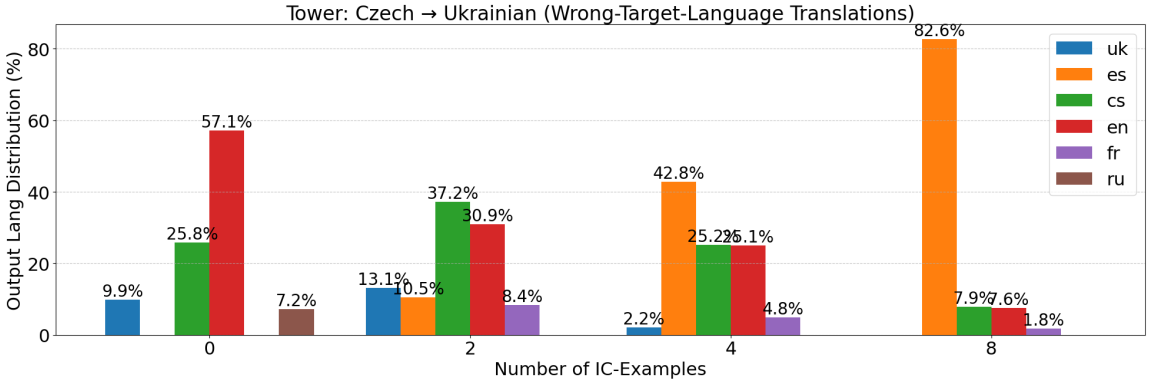

(c) Untrained ⟶ Untrained

Figure 5.13.: Average COMET-22 scores for translations involving non-fine-tuned languages in the Wrong-Target-Language setting. In-context translations are incorrectly prefixed with *French* instead of the actual target language. Left: Llama 3.1; Right: Tower. *Untrained* refers to languages neither model was fine-tuned on (Czech, Ukrainian, Nepali). Y-axis scales are consistent across models (offsets may differ) for comparability. Llama and Tower exhibit contrasting behaviors, reflecting differences in fine-tuning strategies.

amples significantly influences Tower's outputs (see Figures 5.13b and 5.13c). The model progressively shifts toward translating into Spanish as the number of few-shot examples increases — an effect clearly illustrated in Figure 5.14 — despite instructions explicitly specifying other target languages. This reveals Tower's heavy reliance on in-context examples for languages not encountered during fine-tuning, presumably due to a lack of prior learned linguistic knowledge. Thus, Tower's behavior sharply contrasts with scenarios translating into English, highlighting its differential reliance on in-context examples conditioned on prior linguistic familiarity.




(a) English → Czech



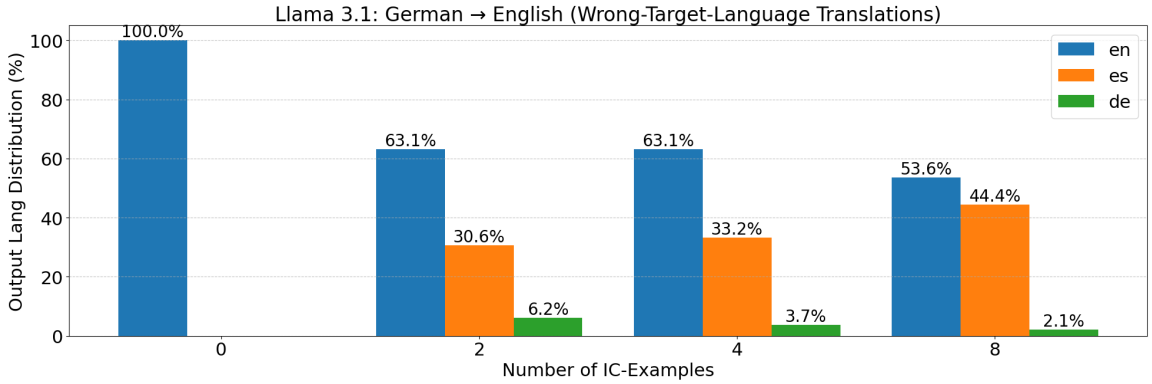

(b) Czech → Ukrainian

Figure 5.14.: Language distributions for Wrong-Target-Language translations with Tower. In-context examples contain *Spanish* translations instead of the actual target language. Tower increasingly defaults to Spanish as the number of few-shot examples grows, despite explicit target language instructions.

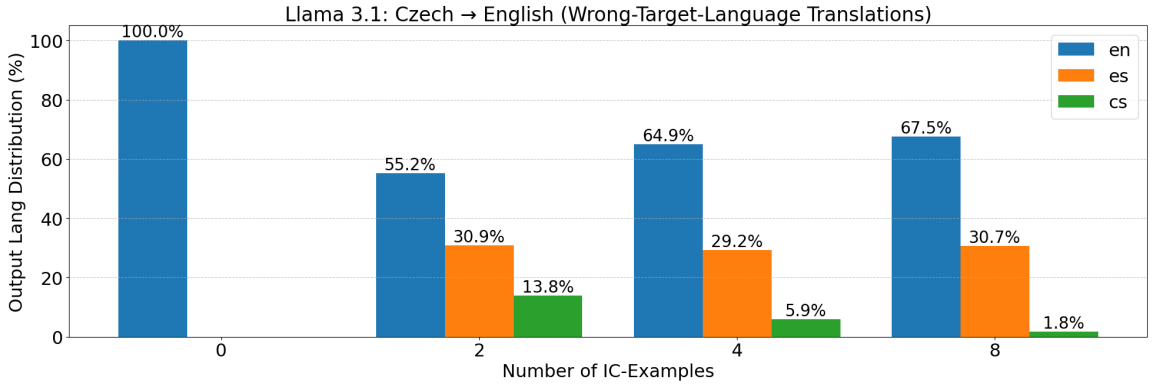
### 5.3.4. In-Context Examples Override Llama's Knowledge

Llama demonstrates a different vulnerability to incorrect in-context examples. Specifically, when tasked with translations into English (a language on which Llama possesses substantial prior knowledge), Llama incorrectly starts translating into Spanish as prompted by the misleading in-context examples. This trend is illustrated in Figure 5.15. Such behavior contradicts both the task instructions and pre-existing knowledge, suggesting

that Llama's translation process may prioritize pattern completion over semantic or task-specific comprehension. Consequently, Llama's prior knowledge can be unintentionally overwritten or misled through contradictory cues provided by misleading in-context examples, particularly for language pairs involving well-trained target languages.




(a) English → Czech




(b) Czech → Ukrainian

Figure 5.15.: Language distributions for Wrong-Target-Language translations with Llama 3.1. In-context examples contain *Spanish* translations instead of the actual target language. Llama 3.1 increasingly defaults to Spanish as the number of few-shot examples grows, despite explicit target language instructions.

### 5.3.5. Llama Does In-Context Learning When Translating Into Non-Fine-Tuned Languages

Contrary to Tower, Llama manages to leverage incorrect Spanish translations moderately well when translating into languages it was not fine-tuned on (see Figures 5.13b and 5.13c). From English to these languages, performance remains comparable to the zero-shot baseline, indicating limited reliance on the provided examples (see Figure 5.13b). More notably, translations from languages the model was not fine-tuned on (*Untrained* → *Untrained*) consistently benefit, with all few-shot prompts yielding higher performance than the zero-shot baseline (see Figure 5.13c). This suggests that Llama effectively exploits

semantic priors contained within in-context examples, despite incorrect language labeling. Unlike translations into English (as presented in Section 5.3.4), in these settings Llama's prior knowledge is not overridden, likely due to the absence of fine-tuning in the targeted languages. Instead, the model successfully extracts useful semantic information from translations provided in the wrong target language to enhance performance. This ability is likely enabled by the fact that the incorrect target language — *Spanish* — is one Llama was fine-tuned on. It may therefore be of interest for further research to explore similar setups with incorrect target languages the model was not fine-tuned on, to examine whether it can still leverage semantic cues without prior exposure.

These experiments highlight markedly contrasting behaviors between Llama and Tower regarding their reliance on and sensitivity to provided in-context examples.

## 5.4. Mismatched Translations

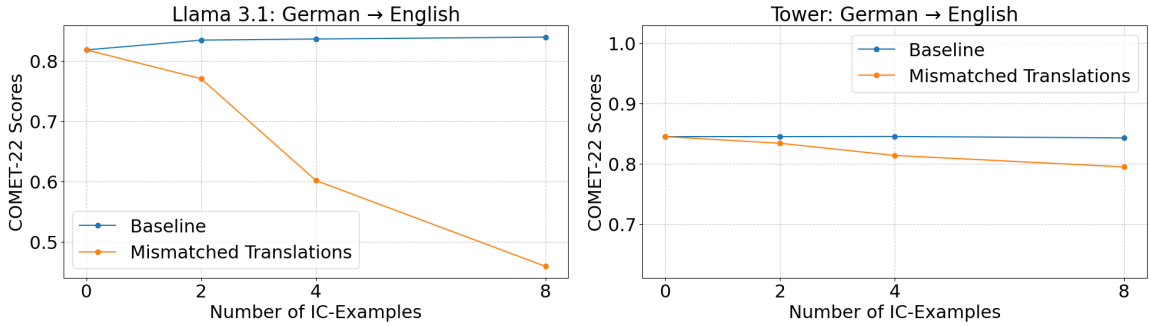
Previous findings suggest that the presence of a source-target mapping (see Section 5.2), even if the target is in another language (see Section 5.3), can improve translation quality. The latter suggests that contextual cues provided by these mappings may play an important role. To further investigate this, in this section, as described in Section 4.1.4, we analyze an experiment in which we purposefully mismatch the instances and translations in the in-context examples. Specifically, each source sentence is paired with a randomly selected target-language sentence from the dataset, resulting in fully mismatched instance–translation pairs.

### 5.4.1. Llama's Misaligned In-Context Learning Negatively Affects Performance

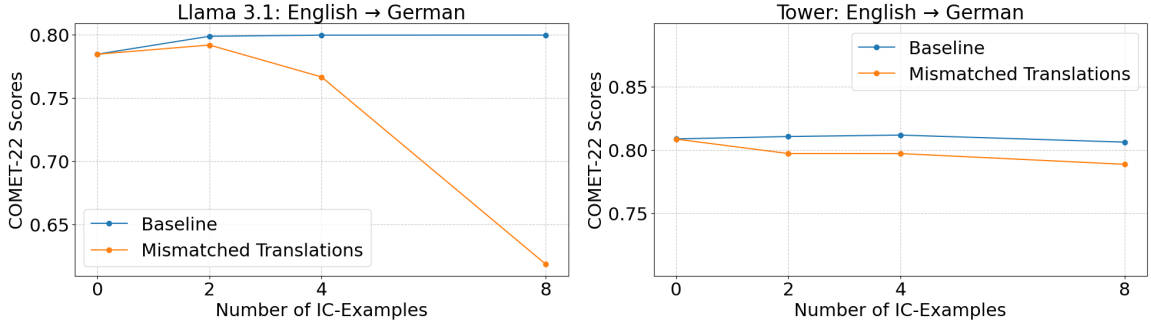
Across all tested language pairs, Llama's translation performance deteriorates rapidly when provided with mismatched source-target pairs (see Figures 5.16 and 5.17). This degradation intensifies as the number of provided mismatched in-context examples increases. Notably, this trend occurs even for language pairs that Llama was explicitly fine-tuned on, indicating that mismatched translations can significantly override or distort Llama's pre-existing linguistic knowledge. This observation aligns with previous findings (Section 5.3.4), confirming that Llama prioritizes pattern completion based on provided contextual cues, potentially at the expense of task comprehension.

### 5.4.2. Tower is More Robust to Mismatched Translations

Although Tower's translation quality also declines when given mismatched source-target examples, the deterioration is notably slower compared to Llama (see Figures 5.16 and 5.17). While Tower's performance still falls below the zero-shot baseline, the impact of misalignment remains comparatively limited, particularly in language pairs included during fine-tuning. This robustness likely arises from Tower's specialized fine-tuning on translation tasks with structured instruction prompts, potentially enabling it to detect and disregard incoherent contextual mappings more effectively.

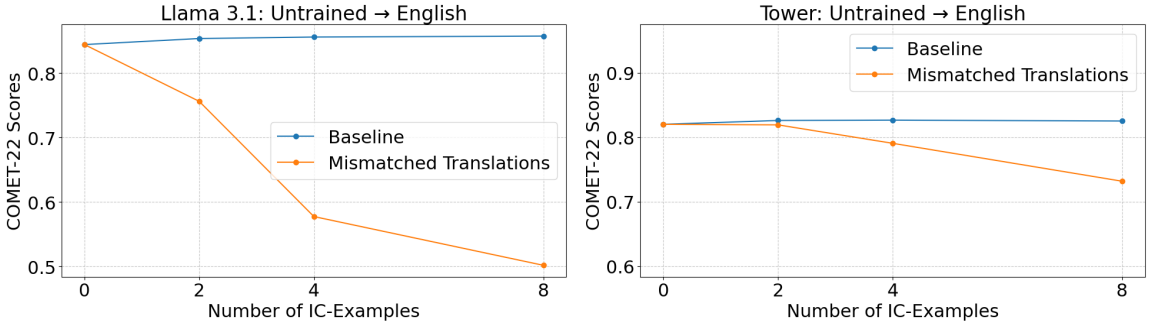



(a) German → English

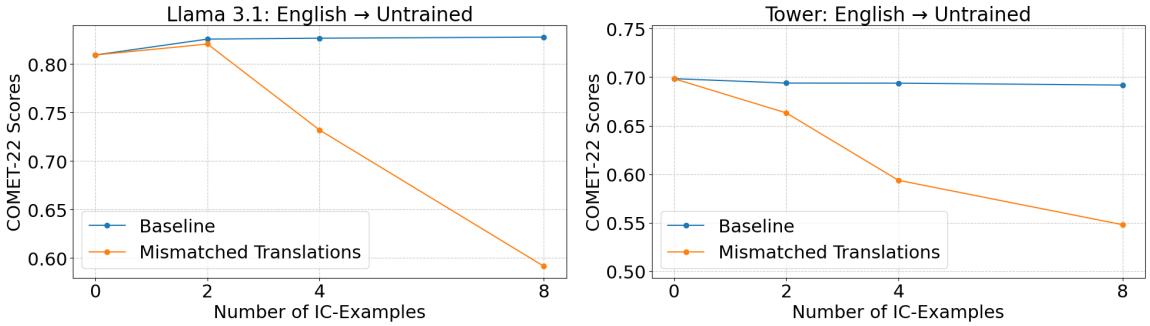



(b) English → German

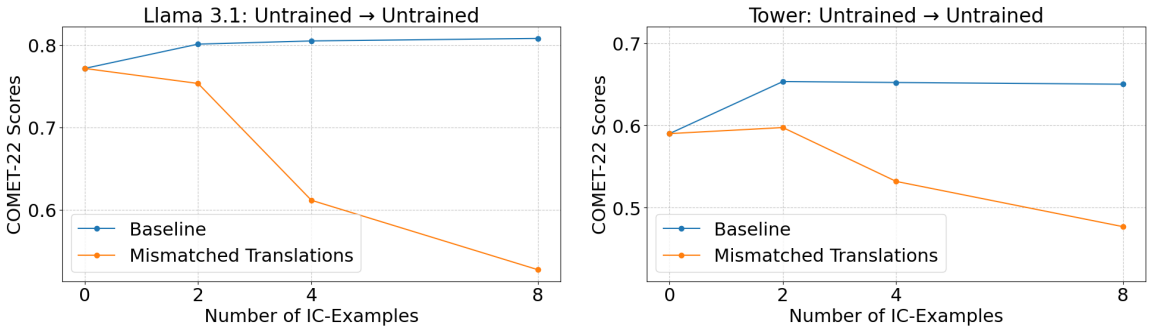
Figure 5.16.: Average COMET-22 scores for translations involving fine-tuned languages in the Mismatched-Translations setting. Each source sentence is paired with a randomly selected target translation. Left: Llama 3.1; Right: Tower. Y-axis scales are consistent across models (offsets may differ) for comparability. Both models exhibit degraded performance, indicating that semantic alignment in in-context examples is crucial for effective translation.




(a) Untrained ⟶ English




(b) English ⟶ Untrained




(c) Untrained ⟶ Untrained

Figure 5.17.: Average COMET-22 scores for translations involving non-fine-tuned languages in the Mismatched-Translations setting. Each source sentence is paired with a randomly selected target translation. Left: Llama 3.1; Right: Tower. *Untrained* refers to languages neither model was fine-tuned on (Czech, Ukrainian, Nepali). Y-axis scales are consistent across models (offsets may differ) for comparability. Both models exhibit degraded performance, indicating that semantic alignment in in-context examples is crucial for effective translation.

### 5.4.3. Llama and Tower Do Not Leverage Target Language Priors

Both models show no benefit from exposure to correctly formed but misaligned source and target sentences. They also fail to exploit structural patterns such as fluency, consistent punctuation, or syntactic regularity (see Figure 5.17). These observations hold even for languages not included in fine-tuning (see Figure 5.16). Translation quality consistently falls below zero-shot levels, indicating that the essential value of in-context examples resides primarily in their semantic and contextual alignment between source and target sentences. This underscores the critical role of coherent source-target mappings for effective in-context learning in machine translation.

## 5.5. Grammatical Errors

Existing translations used in practice often originate from human translators and may contain grammatical or spelling errors. Ideally, machine translation models should demonstrate robustness to such noise and maintain effective in-context learning capabilities. This section, as described in Section 4.1.5, evaluates the robustness of Llama 3.1 and Tower models against two types of introduced grammatical errors: randomized word order and typos. We investigate the models' sensitivity to these errors at varying noise levels (20% and 40%) as described in Sections 4.1.5.1 and 4.1.5.2. More extensive results are provided in Appendix A.1.

### 5.5.1. Both Models are Reasonably Robust to Grammar Errors

Both Llama 3.1 and Tower exhibit a reasonable degree of robustness to grammatical errors introduced through randomized word order. Across all language pairs tested, the impact on translation performance was negligible. Interestingly, both models demonstrated particular robustness when translating into English, suggesting that English's prevalence in training data and fine-tuning contributes to resilience against such perturbations. Increasing the noise level from 20% to 40% slightly amplified performance differences, but overall effects remained minimal, emphasizing the models' ability to extract semantic meaning despite significant syntactic disruptions.

### 5.5.2. Llama is More Sensitive to Grammar Errors than Tower

When comparing the robustness of the two models, Llama 3.1 showed greater sensitivity to grammatical disruptions than Tower, particularly regarding spelling errors (typos) (see Figure 5.18). While Tower maintained near-baseline performance even at higher typo levels, Llama exhibited larger deviations from its baseline performance. This behavior aligns with previous observations (see Sections 5.3.4 and 5.4.1), suggesting that Llama prioritizes in-context examples — even when faulty — potentially overriding its prior knowledge. Conversely, Tower appears to be better equipped to disregard erroneous in-context information, thus maintaining stable performance similar to previous findings (see Sections 5.3.2 and 5.4.2).

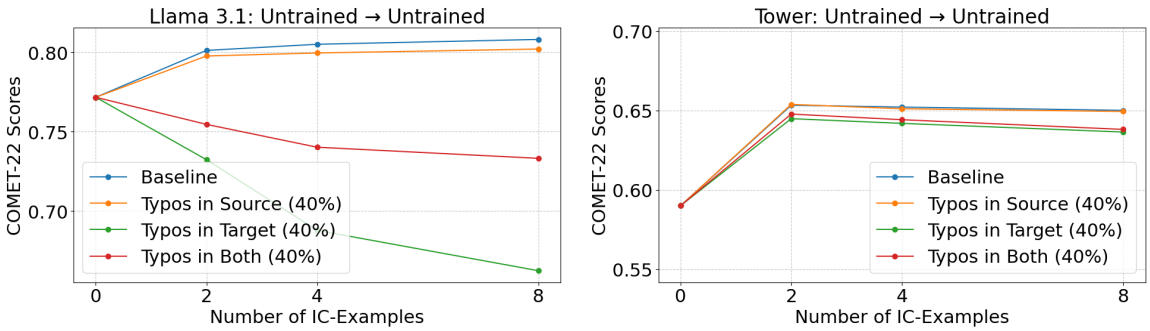



Figure 5.18.: Average COMET-22 scores for Untrained ⟶ Untrained translations with typos in the in-context examples. Typos are simulated by swapping 40% of character pairs in each sentence. Left: Llama 3.1; Right: Tower. *Untrained* refers to languages neither model was fine-tuned on (Czech, Ukrainian, Nepali). Y-axis scales are consistent across models (offsets may differ) for comparability. Tower shows greater robustness to typos than Llama 3.1.

### 5.5.3. Target Errors Hurt Performance More Than Source Errors

Consistently across both experiments (randomized word order and typos), errors introduced in target sentences of in-context examples had a more significant negative impact than errors in source sentences (see Figure 5.18). This pattern held true for both models and across varying noise levels, highlighting the importance both models place on the quality of target translations provided in in-context examples. It underscores that the semantic coherence and correctness of target-language examples are especially critical for effective translation performance.

For Llama, another pattern emerges: errors in both source and target sentences degrade performance less than errors in the target alone, as shown in Figure 5.18. This is not the case for Tower, as demonstrated in Figure 5.19. The uniformity of the error patterns may prevent the introduction of asymmetries between input and output, leading the model to prioritize semantic understanding over surface-level grammatical correctness. Ultimately, this behavior warrants further research.

### 5.5.4. Typos Significantly Impact Tower's Performance on Fine-Tuned Languages

An important exception to Tower's overall robustness emerged specifically when translating between fine-tuned languages (English and German). Here, the simultaneous presence of typos in both source and target sentences caused a notable performance degradation (see Figure 5.19). Tower handles isolated source or target errors effectively, indicating a cumulative negative effect only when errors were present on both sides simultaneously. Furthermore, this vulnerability was exclusive to translations involving fine-tuned language pairs; when at least one language was not included during fine-tuning, Tower's performance differences remained negligible. This finding highlights a nuanced limitation

in Tower's ability to maintain in-context learning robustness under compounded errors in familiar linguistic contexts.

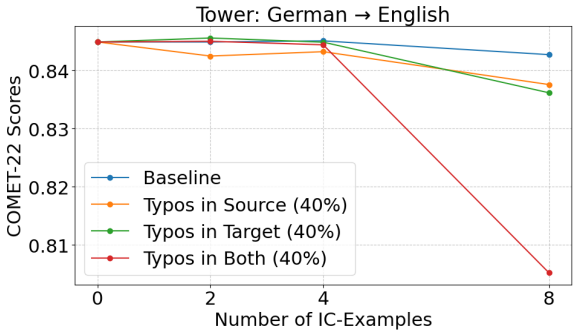



Figure 5.19.: Average COMET-22 scores for German → English translations with typos in the in-context examples. Typos are simulated by swapping 40% of character pairs in each sentence. Tower's performance degrades substantially when typos are present in both the source and target, compared to when they appear in only one.

# 6. Conclusion

## 6.1. How Does In-Context Learning Performance Differ Between General-Purpose and Translation-Optimized Language Models?

To explore this question, we conducted comparative experiments using two state-of-the-art models with contrasting design philosophies: **Llama 3.1**, a general-purpose instruction-tuned model, and **Tower**, a model specifically fine-tuned for machine translation tasks.

- **Llama 3.1** demonstrates significant susceptibility to the quality of in-context examples, with its prior knowledge frequently overridden by incorrect or mismatched examples. This suggests that Llama heavily relies on contextual cues and semantic mappings provided within the examples, sometimes at the expense of accurate task comprehension.

- **Tower**, conversely, exhibits a higher capacity to ignore or disregard misleading in-context examples, particularly in language pairs it has been fine-tuned on. This robustness appears to stem from its specialized fine-tuning on structured, translation-specific instruction prompts.

## 6.2. Does Using Incorrect or Random Translations as In-Context Examples Hurt the Performance of Machine Translation Tasks?

Yes, incorrect or randomly mismatched translations significantly degrade the performance of machine translation tasks, especially for Llama 3.1, which heavily prioritizes pattern matching from in-context examples. Llama often overrides its prior knowledge, even when the examples are clearly faulty, resulting in compromised translation quality. Tower, while more robust, still suffers performance drops when translations are completely misaligned. However, for language pairs it was explicitly fine-tuned on, Tower relies more on its prior knowledge and effectively ignores misleading examples. This indicates that semantic and contextual coherence in in-context examples remains crucial, but model-specific fine-tuning can mitigate some negative effects.

## 6.3. How Do Grammatical Errors in In-Context Examples Affect the Translation Quality?

Both models demonstrated reasonable robustness against grammatical errors, such as randomized word order and spelling mistakes. However, Llama is notably more sensitive, especially to spelling errors, which can substantially impact its translation quality. Tower maintains stronger resilience to grammatical disruptions, particularly for fine-tuned languages, unless errors are simultaneously present in both source and target examples, indicating a nuanced limitation. Importantly, for both models, errors introduced in target sentences consistently hurt performance more significantly than errors in source sentences, underscoring the greater importance of accurate target translations in in-context examples.

## 6.4. Model Selection Recommendations for In-Context Machine Translation

Our findings underline that the effectiveness of in-context learning strongly influences translation performance, albeit in different ways depending on the model's fine-tuning strategy. With this focus, we provide the following recommendations:

- **Fine-Tuned Language Pairs:** For translations involving languages that have been extensively featured during a model's fine-tuning – where structured, translation-specific prompts were used – the role of in-context examples is relatively diminished. In these scenarios, **translation-optimized models** are recommended since their specialized fine-tuning reinforces robust translation performance even when the in-context examples are limited or when their quality might not be ideal. Their design allow them to effectively disregard misleading in-context cues.

- **Unseen and Low-Resource Languages:** For language pairs outside the explicit scope of fine-tuning, the model must rely more heavily on in-context learning. In such cases, **general-purpose models** are preferred, provided that high-quality and semantically coherent examples are available. Their translation performance improves significantly when supported by carefully curated examples that help reconstruct accurate semantic mappings and reduce susceptibility to poor-quality contextual information.

- **Quality of Contextual Examples:** Across both settings, the intrinsic performance of a model is closely tied to the quality of the in-context examples. When the examples are aligned with the task's semantics and free from errors – especially in the target language –even general-purpose models can achieve substantial performance gains. Conversely, in cases with noisy or mismatched examples, the robust fine-tuning of translation-optimized models can provide a critical safeguard by reducing the negative influence of erroneous contextual cues.

Ultimately, the choice between these models should reflect the balance between the availability of high-quality in-context examples and the degree of fine-tuning available for the target language pair. Fine-tuning strategies that are translation-specific diminish reliance on in-context cues, whereas models optimized for general-purpose use can be effectively boosted by leveraging meticulously curated contextual prompts.

## 6.5. Suggestions for Further Research

Future studies should explore:

- How different fine-tuning strategies specifically tailored to MT tasks can further enhance the robustness and adaptability of models like Llama 3.1.
- The impact of varying levels of semantic coherence in in-context examples, beyond the binary mismatched or correct scenario.
- Further investigation into the robustness of these models against grammatical and semantic noise in real-world translation scenarios.

# Bibliography


Agrawal, Sweta et al. (2022). "In-context Examples Selection for Machine Translation". In: *Annual Meeting of the Association for Computational Linguistics*. URL: https://api.semanticscholar.org/CorpusID:254246450.

Alves, Duarte M. et al. (2024). *Tower: An Open Multilingual Large Language Model for Translation-Related Tasks*. arXiv: 2402.17733 [cs.CL]. URL: https://arxiv.org/abs/2402.17733.

Baker, Mona (1992). *In Other Words: A Coursebook on Translation*. URL: https://api.semanticscholar.org/CorpusID:57710440.

Bawden, Rachel and François Yvon (2023). "Investigating the Translation Performance of a Large Multilingual Language Model: the Case of BLOOM". In: *European Association for Machine Translation Conferences/Workshops*. URL: https://api.semanticscholar.org/CorpusID:257353790.

Belinkov, Yonatan and Yonatan Bisk (2018). *Synthetic and Natural Noise Both Break Neural Machine Translation*. arXiv: 1711.02173 [cs.CL]. URL: https://arxiv.org/abs/1711.02173.

Brown, Peter F. et al. (June 1990). "A statistical approach to machine translation". In: *Comput. Linguist.* 16.2, pp. 79–85. ISSN: 0891-2017.

Brown, Tom B. et al. (2020). *Language Models are Few-Shot Learners*. arXiv: 2005.14165 [cs.CL]. URL: https://arxiv.org/abs/2005.14165.

Cho, Kyunghyun, Bart van Merrienboer, Caglar Gulcehre, et al. (2014). *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*. arXiv: 1406.1078 [cs.CL]. URL: https://arxiv.org/abs/1406.1078.

Cho, Kyunghyun, Bart van Merrienboer, Çaglar Gülçehre, et al. (2014). "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation". In: *Conference on Empirical Methods in Natural Language Processing*. URL: https://api.semanticscholar.org/CorpusID:5590763.

Dahlmeier, Daniel, Hwee Tou Ng, and Siew Mei Wu (June 2013). "Building a Large Annotated Corpus of Learner English: The NUS Corpus of Learner English". In: *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*. Ed. by Joel Tetreault, Jill Burstein, and Claudia Leacock. Atlanta, Georgia: Association for Computational Linguistics, pp. 22–31. URL: https://aclanthology.org/W13-1703/.

Devlin, Jacob et al. (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *North American Chapter of the Association for Computational Linguistics*. URL: https://api.semanticscholar.org/CorpusID:52967399.

Dong, Qingxiu et al. (2022). "A Survey on In-context Learning". In: *Conference on Empirical Methods in Natural Language Processing*. URL: https://api.semanticscholar.org/CorpusID:255372865.

GitHub and OpenAI (2025). *GitHub Copilot*. https://github.com/features/copilot. AI-powered code completion tool.

Grattafiori, Aaron et al. (2024). *The Llama 3 Herd of Models*. arXiv: 2407.21783 [cs.AI]. URL: https://arxiv.org/abs/2407.21783.

Graves, Alex, Abdel-rahman Mohamed, and Geoffrey E. Hinton (2013). "Speech recognition with deep recurrent neural networks". In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6645–6649. URL: https://api.semanticscholar.org/CorpusID:206741496.

Hagiwara, Masato and Masato Mita (2019). *GitHub Typo Corpus: A Large-Scale Multilingual Dataset of Misspellings and Grammatical Errors*. arXiv: 1911.12893 [cs.CL]. URL: https://arxiv.org/abs/1911.12893.

Hochreiter, Sepp and Jürgen Schmidhuber (Nov. 1997). "Long Short-Term Memory". In: *Neural Computation* 9.8, pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. eprint: https://direct.mit.edu/neco/article-pdf/9/8/1735/813796/neco.1997.9.8.1735.pdf. URL: https://doi.org/10.1162/neco.1997.9.8.1735.

Hu, Edward J. et al. (2021). "LoRA: Low-Rank Adaptation of Large Language Models". In: *CoRR* abs/2106.09685. arXiv: 2106.09685. URL: https://arxiv.org/abs/2106.09685.

Joulin, Armand, Edouard Grave, Piotr Bojanowski, Matthijs Douze, et al. (2016). "FastText.zip: Compressing text classification models". In: *arXiv preprint arXiv:1612.03651*.

Joulin, Armand, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov (2016). "Bag of Tricks for Efficient Text Classification". In: *arXiv preprint arXiv:1607.01759*.

Kocmi, Tom, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Marzena Karpinska, et al. (Nov. 2024). "Findings of the WMT24 General Machine Translation Shared Task: The LLM Era Is Here but MT Is Not Solved Yet". In: *Proceedings of the Ninth Conference on Machine Translation*. Ed. by Barry Haddow et al. Miami, Florida, USA: Association for Computational Linguistics, pp. 1–46. DOI: 10.18653/v1/2024.wmt-1.1. URL: https://aclanthology.org/2024.wmt-1.1/.

Kocmi, Tom, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Philipp Koehn, et al. (Dec. 2023). "Findings of the 2023 Conference on Machine Translation (WMT23): LLMs Are Here but Not Quite There Yet". In: *Proceedings of the Eighth Conference on Machine Translation*. Ed. by Philipp Koehn et al. Singapore: Association for Computational Linguistics, pp. 1–42. DOI: 10.18653/v1/2023.wmt-1.1. URL: https://aclanthology.org/2023.wmt-1.1/.

Koehn, Philipp et al. (2007). "Moses: Open Source Toolkit for Statistical Machine Translation". In: *Annual Meeting of the Association for Computational Linguistics*. URL: https://api.semanticscholar.org/CorpusID:794019.

Lewis, Mike et al. (2019). "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension". In: *Annual Meeting of the Association for Computational Linguistics*. URL: https://api.semanticscholar.org/CorpusID:204960716.

Liu, Yi et al. (2023). "Jailbreaking ChatGPT via Prompt Engineering: An Empirical Study". In: *ArXiv* abs/2305.13860. URL: https://api.semanticscholar.org/CorpusID:258841501.

Liu, Yinhan et al. (2020). “Multilingual Denoising Pre-training for Neural Machine Translation”. In: *Transactions of the Association for Computational Linguistics* 8, pp. 726–742. URL: https://api.semanticscholar.org/CorpusID:210861178.

Min, Sewon et al. (2022a). *Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?* arXiv: 2202.12837 [cs.CL]. URL: https://arxiv.org/abs/2202.12837.

– (2022b). *Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?* arXiv: 2202.12837 [cs.CL]. URL: https://arxiv.org/abs/2202.12837.

NLLB Team et al. (2024). “Scaling neural machine translation to 200 languages”. In: *Nature* 630.8018, pp. 841–846. ISSN: 1476-4687. DOI: 10.1038/s41586-024-07335-x. URL: https://doi.org/10.1038/s41586-024-07335-x.

Och, Franz Josef (July 2003). “Minimum Error Rate Training in Statistical Machine Translation”. In: *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*. Sapporo, Japan: Association for Computational Linguistics, pp. 160–167. DOI: 10.3115/1075096.1075117. URL: https://aclanthology.org/P03-1021/.

OpenAI (2025). *ChatGPT (GPT-4o)*. https://chat.openai.com/. Large language model.

OpenAI et al. (2024). *GPT-4 Technical Report*. arXiv: 2303.08774 [cs.CL]. URL: https://arxiv.org/abs/2303.08774.

Pan, Jane et al. (2023). “What In-Context Learning "Learns" In-Context: Disentangling Task Recognition and Task Learning”. In: *Annual Meeting of the Association for Computational Linguistics*. URL: https://api.semanticscholar.org/CorpusID:258740972.

Post, Matt (2018). *A Call for Clarity in Reporting BLEU Scores*. arXiv: 1804.08771 [cs.CL]. URL: https://arxiv.org/abs/1804.08771.

Rei, Ricardo et al. (Dec. 2022). “COMET-22: Unbabel-IST 2022 Submission for the Metrics Shared Task”. In: *Proceedings of the Seventh Conference on Machine Translation (WMT)*. Ed. by Philipp Koehn et al. Abu Dhabi, United Arab Emirates (Hybrid): Association for Computational Linguistics, pp. 578–585. URL: https://aclanthology.org/2022.wmt-1.52/.

Sherstinsky, Alex (2018). “Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network”. In: *CoRR* abs/1808.03314. arXiv: 1808.03314. URL: http://arxiv.org/abs/1808.03314.

Sia, Suzanna, David Mueller, and Kevin Duh (2024). *Where does In-context Translation Happen in Large Language Models*. arXiv: 2403.04510 [cs.CL]. URL: https://arxiv.org/abs/2403.04510.

Tao, Junyi, Xiaoyin Chen, and Nelson F. Liu (2024). *Inference and Verbalization Functions During In-Context Learning*. arXiv: 2410.09349 [cs.LG]. URL: https://arxiv.org/abs/2410.09349.

Touvron, Hugo et al. (2023). *Llama 2: Open Foundation and Fine-Tuned Chat Models*. arXiv: 2307.09288 [cs.CL]. URL: https://arxiv.org/abs/2307.09288.

Vaswani, Ashish et al. (2023). *Attention Is All You Need*. arXiv: 1706.03762 [cs.CL]. URL: https://arxiv.org/abs/1706.03762.

Vilar, David et al. (2023). *Prompting PaLM for Translation: Assessing Strategies and Performance*. arXiv: 2211.09102 [cs.CL]. URL: https://arxiv.org/abs/2211.09102.

Wang, Haifeng et al. (2022). “Progress in Machine Translation”. In: *Engineering* 18, pp. 143–153. ISSN: 2095-8099. DOI: https://doi.org/10.1016/j.eng.2021.03.023. URL: https://www.sciencedirect.com/science/article/pii/S2095809921002745.

Wei, Jason et al. (2022). *Emergent Abilities of Large Language Models*. arXiv: 2206.07682 [cs.CL]. URL: https://arxiv.org/abs/2206.07682.

Wei, Jerry et al. (2023). *Larger language models do in-context learning differently*. arXiv: 2303.03846 [cs.CL]. URL: https://arxiv.org/abs/2303.03846.

White, Jules et al. (2023). "A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT". In: *ArXiv* abs/2302.11382. URL: https://api.semanticscholar.org/CorpusID:257079092.

Xie, Sang Michael et al. (2021). "An Explanation of In-context Learning as Implicit Bayesian Inference". In: *ArXiv* abs/2111.02080. URL: https://api.semanticscholar.org/CorpusID:241035330.

Yin, Kayo and Jacob Steinhardt (2025). *Which Attention Heads Matter for In-Context Learning?* arXiv: 2502.14010 [cs.LG]. URL: https://arxiv.org/abs/2502.14010.

Yoo, Kang Min et al. (Dec. 2022). "Ground-Truth Labels Matter: A Deeper Look into Input-Label Demonstrations". In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Ed. by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, pp. 2422–2437. DOI: 10.18653/v1/2022.emnlp-main.155. URL: https://aclanthology.org/2022.emnlp-main.155/.

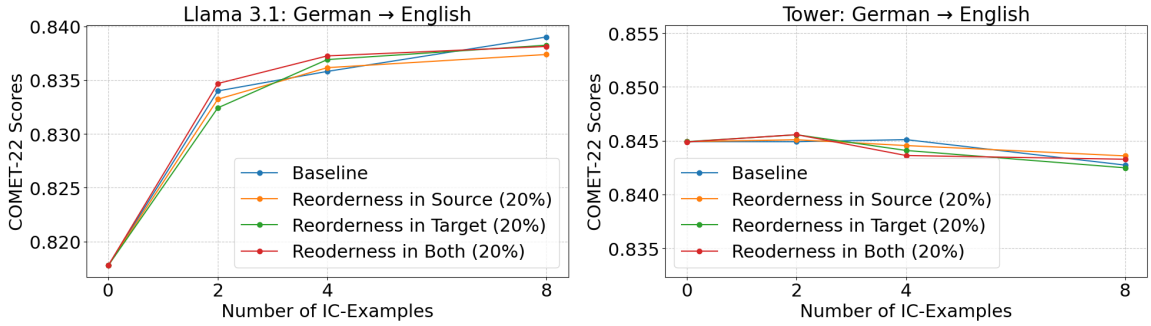
Zaranis, Emmanouil, Nuno M. Guerreiro, and André F. T. Martins (2024). *Analyzing Context Contributions in LLM-based Machine Translation*. arXiv: 2410.16246 [cs.CL]. URL: https://arxiv.org/abs/2410.16246.

Zhang, Biao, Barry Haddow, and Alexandra Birch (2023). *Prompting Large Language Model for Machine Translation: A Case Study*. arXiv: 2301.07069 [cs.CL]. URL: https://arxiv.org/abs/2301.07069.

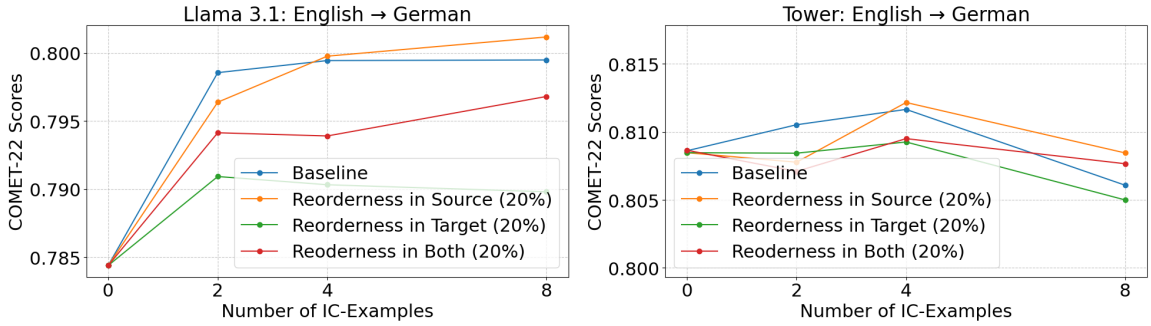
# A. Appendix

## A.1. Grammar Error Reports

### A.1.1. Reordered Words with 20% noise level

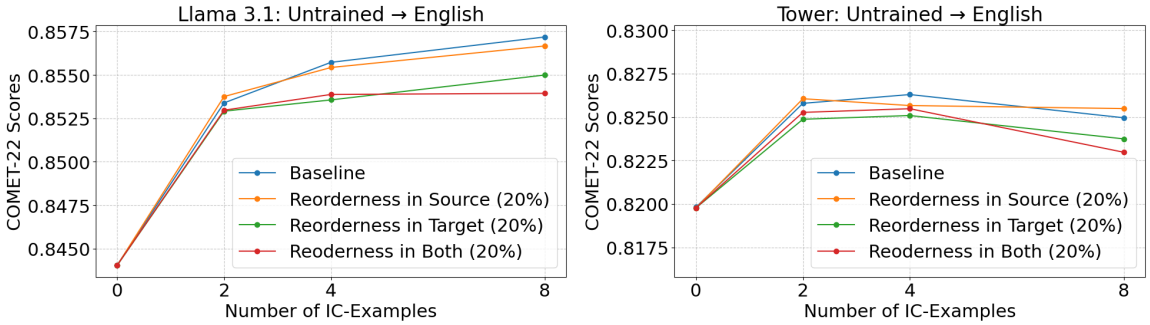



(a) German → English

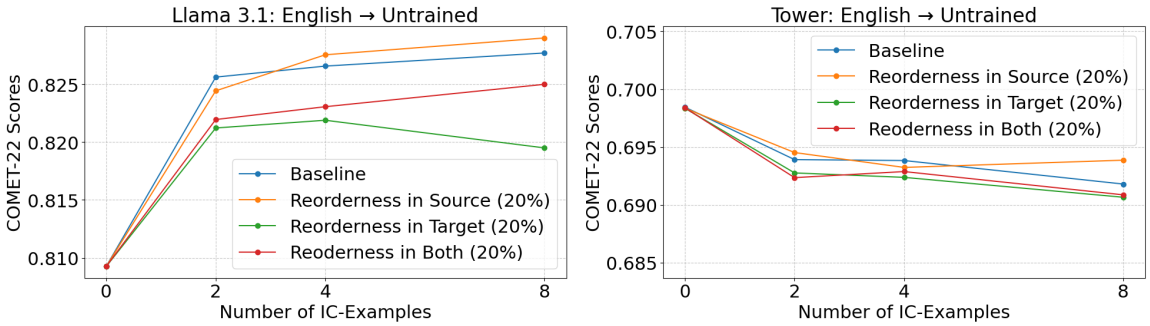



(b) English → German

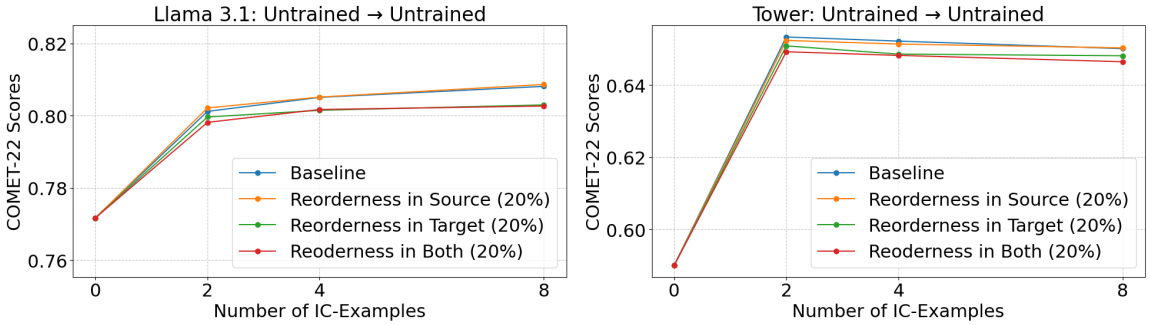
Figure A.1.: Average COMET-22 scores for translations involving fine-tuned languages in the Reordered-Words setting with 20% noise level. 20% of the words in each sentence are randomly repositioned within the same sentence. Results are shown for three conditions: reordering applied to the source sentences, the target translations, or both. Left: Llama 3.1; Right: Tower. Y-axis scales are consistent across models (offsets may differ) for comparability.




(a) Untrained → English



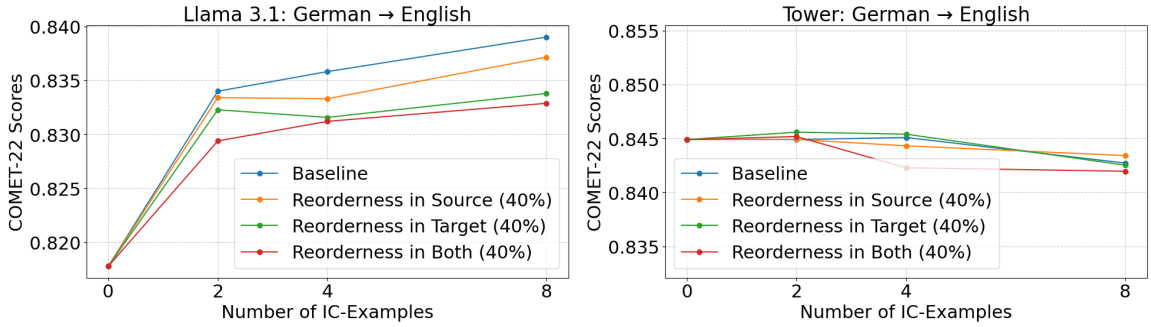

(b) English → Untrained



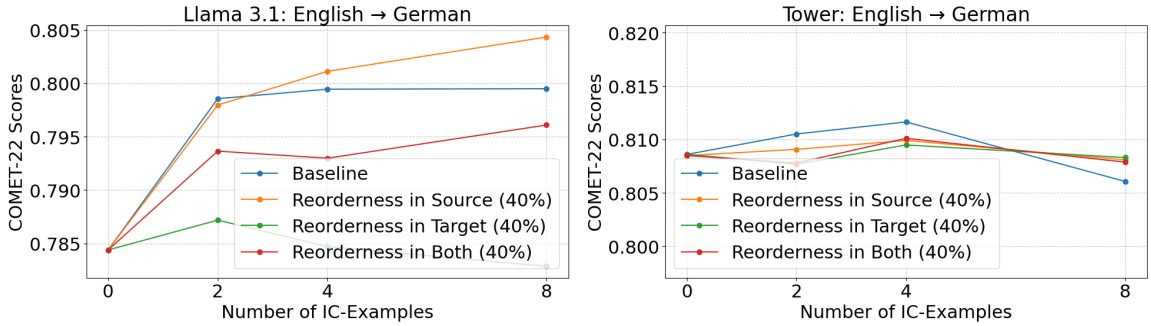

(c) Untrained → Untrained

Figure A.2.: Average COMET-22 scores for translations involving non-fine-tuned languages in the Reordered-Words setting with 20% noise level. 20% of the words in each sentence are randomly repositioned within the same sentence. Results are shown for three conditions: reordering applied to the source sentences, the target translations, or both. Left: Llama 3.1; Right: Tower. *Untrained* refers to languages neither model was fine-tuned on (Czech, Ukrainian, Nepali). Y-axis scales are consistent across models (offsets may differ) for comparability.

### A.1.2. Reordered Words with 40% noise level

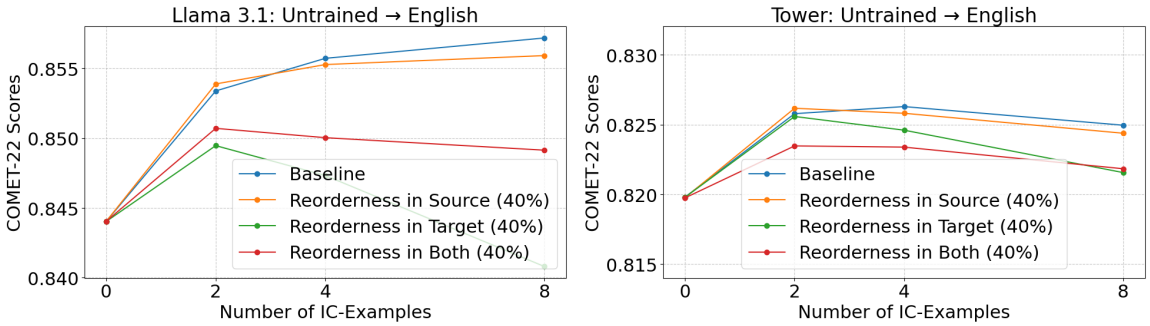



(a) German → English

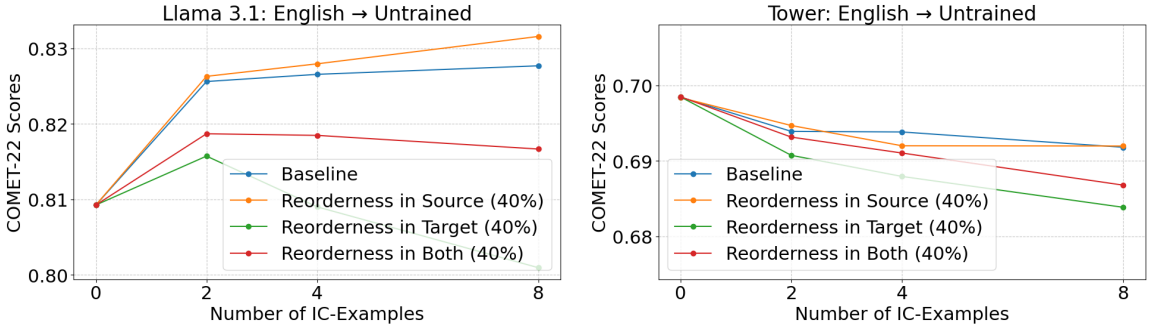



(b) English → German

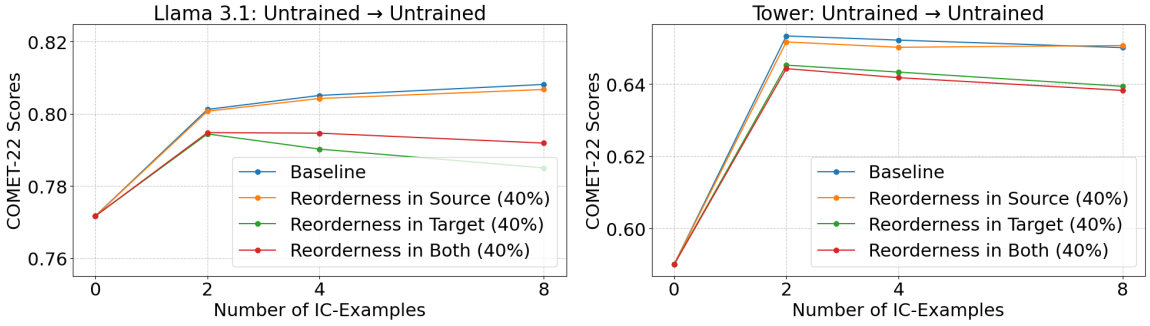
Figure A.3.: Average COMET-22 scores for translations involving fine-tuned languages in the Reordered-Words setting with 40% noise level. 40% of the words in each sentence are randomly repositioned within the same sentence. Results are shown for three conditions: reordering applied to the source sentences, the target translations, or both. Left: Llama 3.1; Right: Tower. Y-axis scales are consistent across models (offsets may differ) for comparability.




(a) Untrained → English



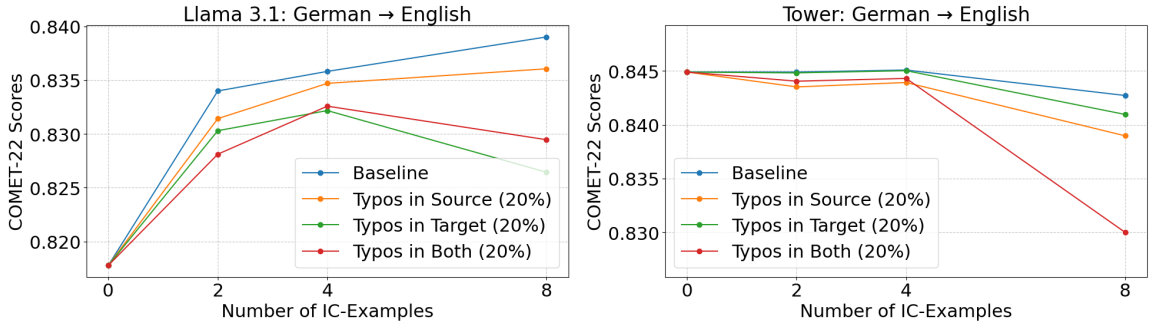

(b) English → Untrained



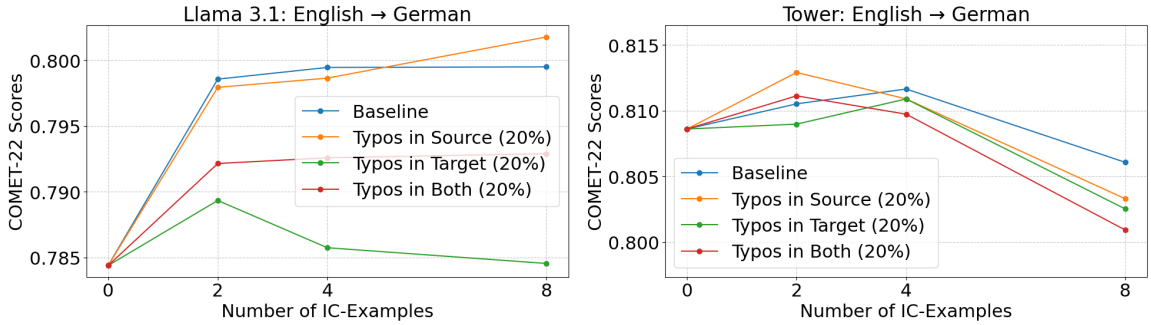

(c) Untrained → Untrained

Figure A.4.: Average COMET-22 scores for translations involving non-fine-tuned languages in the Reordered-Words setting with 40% noise level. 40% of the words in each sentence are randomly repositioned within the same sentence. Results are shown for three conditions: reordering applied to the source sentences, the target translations, or both. Left: Llama 3.1; Right: Tower. *Untrained* refers to languages neither model was fine-tuned on (Czech, Ukrainian, Nepali). Y-axis scales are consistent across models (offsets may differ) for comparability.

### A.1.3. Typos with 20% noise level

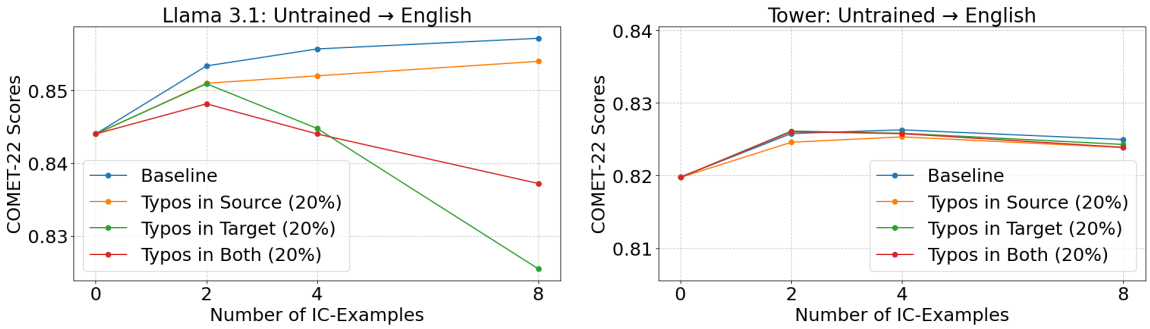



(a) German → English

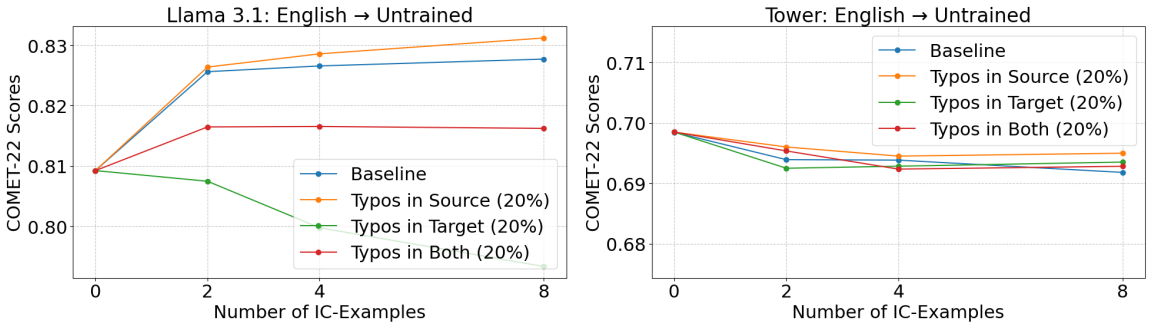



(b) English → German

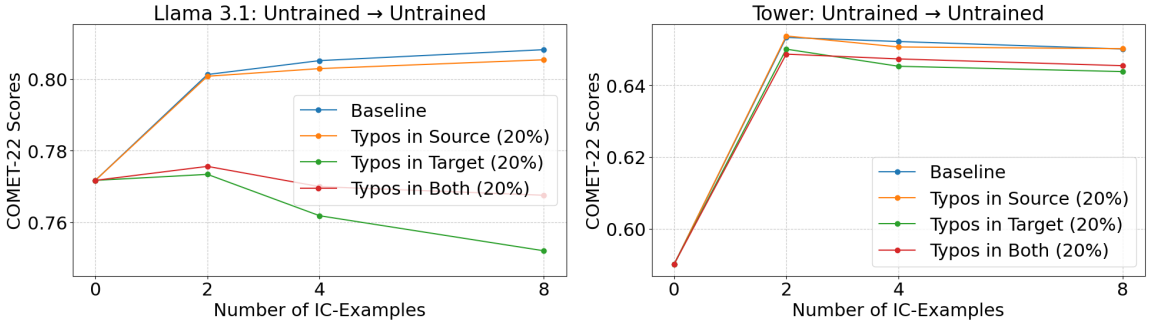
Figure A.5.: Average COMET-22 scores for translations involving fine-tuned languages in the Typos setting with 20% noise level. 20% of character pairs are switched in each sentence. Results are shown for three conditions: typos applied to the source sentences, the target translations, or both. Left: Llama 3.1; Right: Tower. Y-axis scales are consistent across models (offsets may differ) for comparability.




(a) Untrained → English



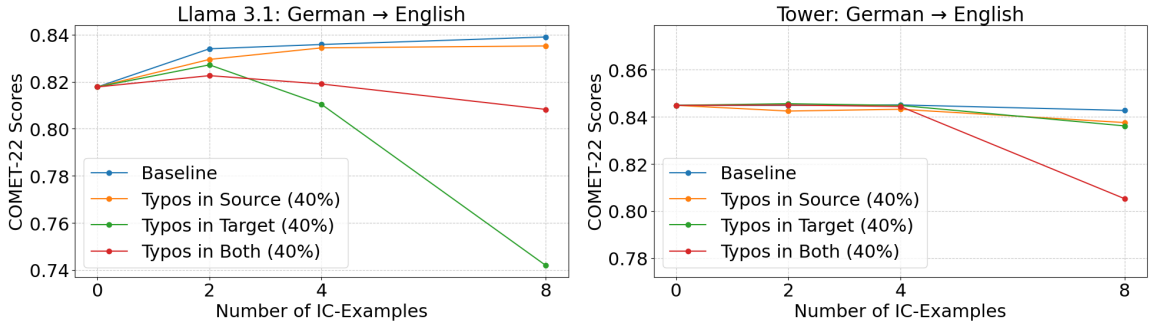

(b) English → Untrained



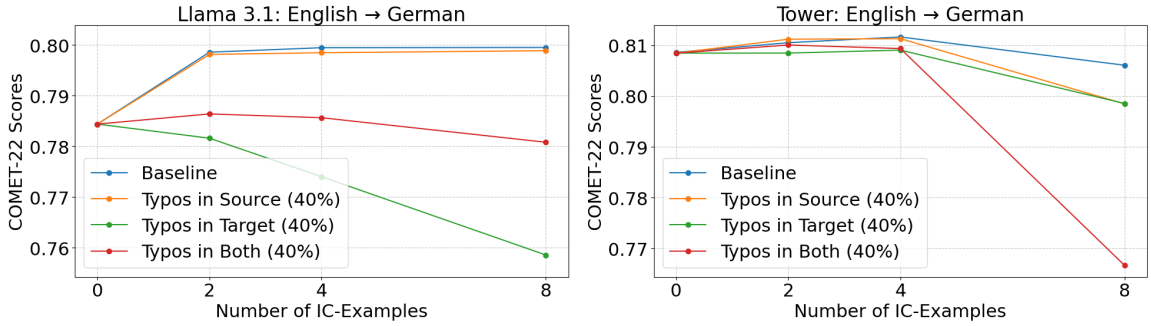

(c) Untrained → Untrained

Figure A.6.: Average COMET-22 scores for translations involving non-fine-tuned languages in the Reordered-Words setting with 20% noise level. 20% of character pairs are switched in each sentence. Results are shown for three conditions: typos applied to the source sentences, the target translations, or both. Left: Llama 3.1; Right: Tower. *Untrained* refers to languages neither model was fine-tuned on (Czech, Ukrainian, Nepali). Y-axis scales are consistent across models (offsets may differ) for comparability.

### A.1.4. Typos with 40% noise level

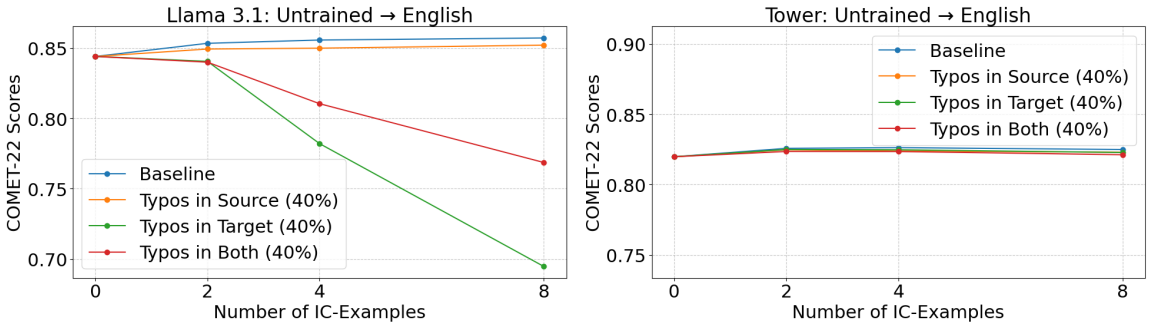



(a) German → English

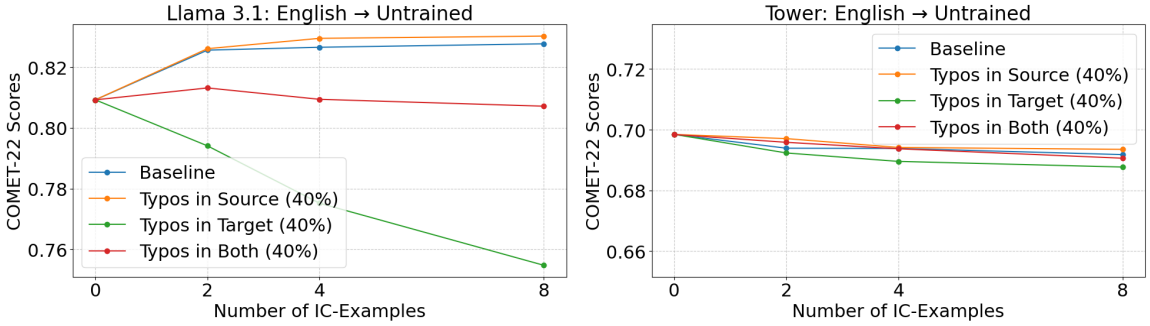



(b) English → German

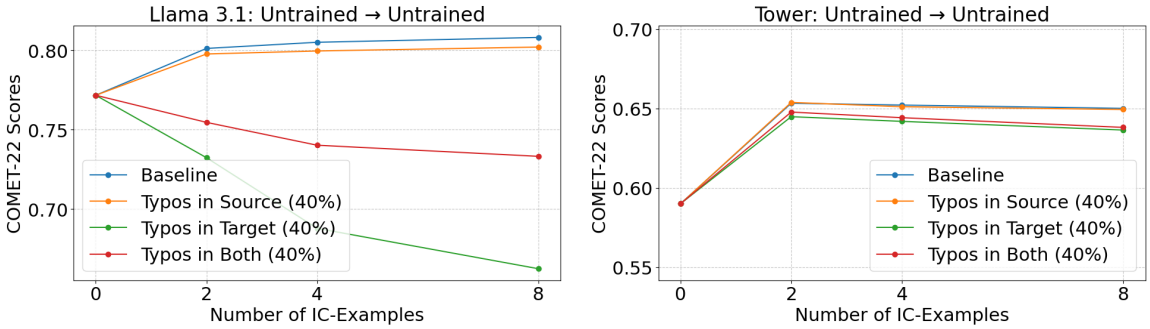
Figure A.7.: Average COMET-22 scores for translations involving fine-tuned languages in the Typos setting with 40% noise level. 40% of character pairs are switched in each sentence. Results are shown for three conditions: typos applied to the source sentences, the target translations, or both. Left: Llama 3.1; Right: Tower. Y-axis scales are consistent across models (offsets may differ) for comparability.




(a) Untrained → English




(b) English → Untrained




(c) Untrained → Untrained

Figure A.8.: Average COMET-22 scores for translations involving non-fine-tuned languages in the Reordered-Words setting with 40% noise level. 40% of character pairs are switched in each sentence. Results are shown for three conditions: typos applied to the source sentences, the target translations, or both. Left: Llama 3.1; Right: Tower. *Untrained* refers to languages neither model was fine-tuned on (Czech, Ukrainian, Nepali). Y-axis scales are consistent across models (offsets may differ) for comparability.