Towards Effective Disambiguation for Machine Translation with Large Language Models

Vivek Iyer **Pinzhen Chen** **Alexandra Birch**
School of Informatics, University of Edinburgh
{vivek.iyer, pinzhen.chen, a.birch}@ed.ac.uk

Abstract

Resolving semantic ambiguity has long been recognised as a central challenge in the field of Machine Translation. Recent work on benchmarking translation performance on ambiguous sentences has exposed the limitations of conventional Neural Machine Translation (NMT) systems, which fail to handle many such cases. Large language models (LLMs) have emerged as a promising alternative, demonstrating comparable performance to traditional NMT models while introducing new paradigms for controlling the target outputs. In this paper, we study the capabilities of LLMs to translate "ambiguous sentences" - i.e. those containing highly polysemous words and/or rare word senses. We also propose two ways to improve their disambiguation capabilities, through a) in-context learning and b) fine-tuning on carefully curated ambiguous datasets. Experiments show that our methods can match or outperform state-of-the-art systems such as DeepL and NLLB in four out of five language directions. Our research provides valuable insights into effectively adapting LLMs to become better disambiguators during Machine Translation. We release our curated disambiguation corpora and resources at https://data.statmt.org/ambiguous-europarl.

1 Introduction

While the field of NMT has advanced rapidly in recent times, the disambiguation and translation of ambiguous words still remain an open challenge. Notably, Campolungo et al. (2022) created a benchmark named DiBiMT to study the behaviour of state-of-the-art (SOTA) NMT systems when translating sentences with ambiguous words.[1] They reported that even the best-performing commercial NMT systems yielded accurate translations only 50-60% of the time,[2] while other open-source multilingual models like mBART50 (Tang et al., 2021) and M2M100 (Fan et al., 2021) performed much worse. This was found to be due to biases against rare and polysemous word senses inherited during pretraining. Table 1 shows an example from the DiBiMT benchmark where DeepL[3] mistranslates an ambiguous word while the LLM BLOOMZ resolves the word to its correct in-context meaning.

Source	The horse had a blaze between its eyes.
DeepL	那匹马的两眼之间有一团火焰。 (There is a flame between the horse's eyes.)
BLOOMZ (176B)	这匹马的眼睛之间有一道白线。 (There is a white line between the horse's eyes.)

Table 1: An example of English-to-Chinese translation involving an ambiguous term "blaze". For BLOOMZ, we use 1-shot prompting to obtain the translation.

In this paper, we explore whether LLMs can indeed perform better at translating "ambiguous sentences" – i.e. those containing highly polysemous and/or rare word senses. The motivation behind this is that while NMT models can potentially learn biases from noisy or narrow domain parallel data, hurting their ability to detect and translate rare word senses, LLMs can potentially be pretrained on a wider variety of monolingual text – though they might also prefer fluency over accuracy. Still, LLMs have shown many emergent abilities due to scale (Brown et al., 2020; Chowdhery et al., 2022; Wei et al., 2022a) and moreover, have demonstrated great potential for Machine Translation (MT) (Vilar et al., 2023; Zhang et al., 2023).

We comprehensively examine how these trends extend to the specific task of translating ambiguous sentences. We select a diverse set of foundational and instruction-tuned LLMs, of different

[1] https://nlp.uniroma1.it/dibimt/public/leaderboard

[2] Subsequent iterations of these commercial models have improved, but large margins still remain.

[3] https://deepl.com/en/translator

Proceedings of the Eighth Conference on Machine Translation (WMT), pages 482–495
December 6–7, 2023.

sizes and with varying combinations of languages in the pre-training data. We then compare how these LLMs match up against several widely used NMT models on the DiBiMT test set, which covers translation from English to five languages: Spanish, Italian, German, Russian and Chinese. We find that, with only 1-shot in-context learning (Brown et al., 2020), LLMs – in particular, BLOOMZ 176B (Muennighoff et al., 2023) and LLaMA 65B (Touvron et al., 2023) – match or outperform top-performing open-source and commercial MT systems, and set a new SOTA in two of the five languages we tested. Furthermore, we propose two methods for adapting LLMs for ambiguous translation: 1) in-context learning with sentences having the same word sense, and 2) fine-tuning on curated ambiguous parallel corpora. We show that these methods are highly effective and can further improve performance by up to 15 points in DiBiMT accuracy in the best case.

Our work thus makes three key contributions:

1. We evaluate the performance of LLMs compared to top-performing NMT systems in the challenging task of translating ambiguous sentences. We report SOTA scores on 2 of the 5 languages tested, and comparable performance otherwise.

2. We also show that our suggested techniques of similar sentence in-context learning and targeted disambiguation fine-tuning significantly outperform naive few-shot prompting

3. We conclude our work by evaluating LLMs on the FLORES200 test sets, and confirm that improvements in disambiguation accuracy correlate strongly with those in overall MT quality.

2 Background

2.1 Ambiguity in machine translation

Resolving ambiguity in the source sentence was historically framed as one of the most fundamental challenges in MT (Weaver, 1952). In an effort to address this challenge, traditional works integrating Word Sense Disambiguation in Statistical Machine Translation (Carpuat and Wu, 2007; Chan et al., 2007) were followed by those integrating it in NMT architectures in various ad-hoc ways (Choi et al., 2017; Liu et al., 2018; Pu et al., 2018). Later, with the introduction of the Transformer (Vaswani et al., 2017), it was shown that higher layer encoder representations are robust enough to handle disambiguation (Tang et al., 2019) without any explicit handling of word senses.

However, more recent research creating challenging evaluation benchmarks has called the purported abilities of NMT systems into question once again. Following the proposal of the MuCoW benchmark for testing WMT19 (Raganato et al., 2019) and WMT20 (Scherrer et al., 2020) systems, Raganato et al. (2020a) showed how Transformer-based NMT models, in general, underperform when translating rare word senses. Campolungo et al. (2022), who experimented with SOTA commercial (Google Translate, DeepL) and open-source systems (mBART50, M2M100, OPUS-NMT (Tiedemann and Thottingal, 2020), etc.), arrived at the same conclusion when they proposed the DiBiMT benchmark for evaluating MT systems between English and 5 languages (Spanish, Italian, German, Russian, and Chinese). They found similar biases against low-frequency and highly polysemous word senses. They also noted the accuracies of these systems were much lower than the then SOTA WSD system, ESCHER (Barba et al., 2021) – indicating significant room for improvement. In this work, we explored whether foundational and instruction-tuned LLMs could bridge this gap with minimal supervision (i.e. few-shot prompting).

2.2 LLMs and machine translation

Previous research has found that LLMs can perform machine translation without being specifically fine-tuned (Radford et al., 2019). In order to elicit a translation, research in this direction follows the paradigm of LLM prompting:

1. Zero-shot prompting, where an LLM is directly asked to translate a source input into the target language (Radford et al., 2019).

2. Few-shot prompting, also called in-context learning, where an LLM is supplied with demonstrations of input and output pairs from the same task it is performing, before being queried an input (Brown et al., 2020).

3. Chain-of-thought (CoT), where an LLM is prompted to reason to gain relevant knowledge about the input before producing an output (Wei et al., 2022b; Kojima et al., 2022).

Besides training-free approaches, another route is instruction tuning, which optimizes an LLM on a

mixed range of downstream tasks and fine-tunes the model to understand and respond to user intention through natural language (Wei et al., 2021).

It was observed that LLMs might not surpass Transformer models solely trained to translate, especially for non-English and low-resource translation directions (Vilar et al., 2023; Hendy et al., 2023). Nevertheless, LLMs have been shown to achieve superiority in tasks requiring in-depth understanding and manipulation of text, primarily due to them being pretrained on very large corpora. For example, without fine-tuning, LLMs are good at adapting to word alignments (Moslem et al., 2023), translation evaluation (Kocmi and Federmann, 2023), idiom translation (Raunak et al., 2023), iterative refinement (Chen et al., 2023), and interactive translation via CoT (Pilault et al., 2023; He et al., 2023). Related to our work is Pilault et al. (2023)'s proposal of using interactive question answering as a CoT process for LLMs to disambiguate source words. As an alternative approach, we aim to generate translations in a single pass by leveraging SOTA WSD systems to provide contexts that guide LLMs to disambiguate better.

3 Methodology

3.1 Preliminaries

A word sense is a concept in a Knowledge Base (in this work, BabelNet by Navigli et al. (2021)) that denotes a distinct meaning of a word in the context of a sentence. The polysemy degree of an ambiguous word is defined as the total count of all possible senses that a particular word can have. The sense frequency is defined as the occurrence count of that particular sense in a disambiguated training corpus.

In this work, we define an ambiguous word as a polysemous term with multiple possible, and likely related, meanings – with the correct sense inferable only from the sentence-level context. We then refer to a sentence with an ambiguous word as an "ambiguous sentence" for brevity and ease of explanation. By definition, the DiBiMT test set (Campolungo et al., 2022) contains only one ambiguous word per sentence.

Word Sense Disambiguation (WSD) is the process of linking an ambiguous word in a sentence to its appropriate word sense in the Knowledge Base. We use ESCHER-WSD (Barba et al., 2021) in this work, a high-performing WSD system that had achieved the SOTA for English.

3.2 *K*-shot prompting

Given a test sentence X and a Large Language Model to prompt for translations, we construct a query with k demonstrations, i.e. parallel sentence pairs $\{(X_1, Y_1), (X_2, Y_2) \dots (X_k, Y_k)\}$ as examples, followed by the test sentence. As shown in Figure 1, for foundation LLMs, we frame the prompt as a text completion task, while for instruction-tuned LLMs (like BLOOMZ) we structure the last phrase as a question, in order to conform to the latter's question answering format. In the naive setting, we choose our demonstrations randomly from the development set.

3.3 In-context learning with similar ambiguous contexts

LLMs can effectively gain knowledge relevant to the test domain through prompting, and this process is named in-context learning (ICL). We leverage ICL to help LLMs ingest information on translation of ambiguous sentences, by providing related sense translations as examples in the prompt. To achieve this, we first identify the most polysemous word in the input sentence by disambiguating it with a WSD system, and then calculate the polysemy degree of all disambiguated senses with respect to a large development set. We choose the most polysemous word sense[4] and search for other occurrences of the same sense in the same development set. Finally, we randomly sample k source-target pairs including such a sense to use as demonstrations in k-shot prompting, instead of using random pairs. This technique seemed to return enough examples for our purposes in most cases – for 5-shot prompting, given a corpus of 1.8M sentences, we observed that we got all 5 matches 92.5% of the time.

3.4 Low-rank fine-tuning

Apart from providing relevant examples through prompting, another conventional approach is to optimize the model parameters in a domain adaptation fashion for disambiguation. Considering the computational cost, our work experiments with instruction fine-tuning via low-rank adaptation (LoRA). This technique appends trainable lower-rank decomposition matrices to giant matrices in an LLM

[4]Currently, we only explore the case of one ambiguous word per sentence, due to the nature of the benchmark. One could extend our approach to multiple ambiguous words by separately sampling examples for each polysemous word and conducting higher-shot prompting - but further research would be needed to find the optimal way to combine these examples.

```
Translate the following sentence from {src_lang} to {tgt_lang}: {src_demo_1}
The translation in {tgt_lang} is: {tgt_demo_1}
                              ⋮
                              k demonstrations
                              ⋮
Translate the following sentence from {src_lang} to {tgt_lang}: {src_demo_k}
The translation in {tgt_lang} is: {tgt_demo_k}
[FOUNDATION LLM]
Translate the following sentence from {src_lang} to {tgt_lang}: {src_test}
The translation in {tgt_lang} is:
                                    OR
[INSTRUCTION-TUNED LLM]
Translate the following sentence from {src_lang} to {tgt_lang}: {src_test}
Can you translate the input sentence to {tgt_lang}?
```

Figure 1: Templates used for k-shot LLM prompting, with $k >= 0$.

that can remain frozen during fine-tuning (Hu et al., 2021). By sacrificing a little performance, this fine-tuning method achieves great parameter efficiency. We aim to adjust LLMs to perform the translation task specifically. In order to maximise an LLM's capability to disambiguate when translating, we follow a careful data selection procedure to identify the most ambiguous sentences in our corpus.

Given the size of LLMs, it would be infeasible to fine-tune them on a large parallel corpus, so we opt to curate a smaller dataset that suits the ambiguous translation task. We would like a balanced mix of sentences with highly polysemous words as well as those with rare senses of a given word. This is to ensure fine-tuning reduces both polysemy degree-related and sense frequency-related biases, as discovered by Campolungo et al. (2022) and consequently, maximises disambiguation performance. We, thus, sort our corpora in two ways: one, by the maximum polysemy degree (greatest first) and two, by the minimum sense frequency (rarest first) of all word senses in a given sentence, disambiguated with ESCHER-WSD. We take the top $N/2$ sentences from each set and interleave them to create our final fine-tuning corpus of size N. We release our fine-tuning corpus, along with the ESCHER-WSD disambiguation outputs for public use.[5]

Once the data is chosen, we follow the fine-tuning paradigm of Alpaca (Taori et al., 2023): the model is prompted with an instruction specifying the source and target languages, as well as the test sentence as an input, and the model is expected to respond with the translation.[6]

[5] https://data.statmt.org/ambiguous-europarl

[6] https://github.com/tatsu-lab/stanford_alpaca

4 Experiments

In this section, we seek to answer the following research questions:

1. **RQ1:** How do LLMs perform at translation of ambiguous sentences compared to traditional high-performing NMT systems? (Section 4.3)
2. **RQ2:** What methods could one use to adapt LLMs for this task and improve performance over naive few-shot prompting? (Section 4.4)
3. **RQ3:** How do these disambiguation-adapted LLMs fare in terms of overall translation quality? (Section 4.5)

4.1 Models

To ensure reproducibility, we pick four well-known and high-performing open-source LLMs,[7] of which we sample seven versions for experimentation:

- BLOOM (Scao et al., 2022): A fully open-source, multilingual, foundation LLM that supports 46 languages. To establish the range of its capabilities, we explore both the smallest (7.1B) and the largest (176B) versions.
- BLOOMZ (Muennighoff et al., 2023): BLOOM instruction-tuned on a multilingual prompting set. Again, we choose the smallest (7.1B) and the largest (176B) versions.
- LLaMA (Touvron et al., 2023): The popular LLM trained by Meta AI, on gigantic datasets ranging up to 1.5T tokens. We evaluate the smallest (7B) and the largest (65B) versions.

[7] at the time of experiment formulation

- Alpaca (Taori et al., 2023): A LLaMA model instruction-tuned on a 52K dataset generated using Self-Instruct (Wang et al., 2023).

To effectively position these open-source LLMs against traditional NMT systems, we compare them against the best-performing and the most widely used commercial and open-source models:

1. DeepL Translator[8]: a SOTA commercial NMT system (accessed on 24th July 2023).
2. Google Translate[9]: Probably the most widely used commercial NMT system (accessed on 24th July 2023).
3. OPUS (Tiedemann and Thottingal, 2020): Small, bilingual, Transformer-based NMT models trained on the OPUS parallel corpora.
4. mBART50 (Tang et al., 2021): Multilingual NMT models pretrained on monolingual corpora from 50 languages, and fine-tuned on the translation task. We report performances of both the English-to-many and many-to-many fine-tuned models.
5. M2M100 (Fan et al., 2021): A massive multilingual NMT model that was trained on 2200 translation directions to support many-to-many translation among 100 languages in total. We compare both the base (418M) and the large (1.2B) versions.
6. NLLB-200 (NLLB Team et al., 2022): It is the current SOTA in many low-resource pairs, scaling to 200 languages. We experiment with all its variants, where the largest is a mixture-of-experts (MoE) model with 54B parameters. We also benchmark its smaller checkpoints at 1.3B and 3.3B, as well as distilled versions at 0.6B and 1.3B.

We take the results for mBART50, M2M100, and OPUS directly from the DiBiMT leaderboard.[10] We use Hugging Face[11] for accessing and inferencing all other models – except for Google Translate and DeepL, which are accessed using their respective APIs. Despite their presence on the leaderboard, we re-evaluate these systems since they are being constantly updated.

[8] https://www.deepl.com/en/translator
[9] https://translate.google.com/
[10] https://nlp.uniroma1.it/dibimt/public/leaderboard
[11] https://huggingface.co/

System	En-Es	En-It
Similar contexts dev set	1.81M	1.73M
Fine-tuning corpus	100K	100K

Table 2: Statistics of data used in our experiments, in terms of parallel sentence count.

4.2 Experimental setup

Datasets In this study, we use the DiBiMT test set for evaluation and measure accuracy across all five translation directions: English to Spanish, Italian, Chinese, Russian, and German, respectively. For validation, we use the development set from FLORES 200 (NLLB Team et al., 2022) in our base setting. To search for similar ambiguous contexts (Section 3.3), we require a larger development set to find relevant examples and also to accurately estimate polysemy degree. Hence, we use the Europarl corpus (Koehn, 2005), disambiguated with ESCHER-WSD. We also use the same disambiguated corpus for fine-tuning, however, we first follow the filtering procedure described in Section 3.4 to create a small corpus full of ambiguous sentences. Validation during fine-tuning is done using 500 randomly sampled sentences from this corpus and the rest is used for training. We detail the data statistics used for these experiments in Table 2.

LLM prompting setup Due to memory constraints, and to compare all models fairly, we load LLMs in 8-bit and use a batch size of 1. For generation, we set both beam size and temperature to 1. To prevent repetition in LLM output, we set `no_repeat_ngram_size` to 4. From the LLM's response, we filter out the sentence before the first newline character as the output translation.

LoRA fine-tuning We inject LoRA modules into all query, key, and value matrices. We set rank to 8, alpha to 8, and dropout to 0.05. For training, we set the effective batch size to 32, the learning rate to 3e-4, and the maximum length to 256. The total training budget is 5 epochs, and we pick the best model checkpoint based on cross-entropy loss on the validation set. The training data is shuffled after every epoch. Inference is done with a beam size of 3, and a maximum generation length of 150.

4.3 LLMs vs NMT systems on DiBiMT

We show our results in Table 3. For the subsequent discussion, we note that LLaMA was not intentionally trained on Chinese and is, thus, an 'unseen'

System	# Params	Variant	En-Es	En-It	En-Zh	En-Ru	En-De	Average
Commercial systems								
DeepL	Unknown	July 2023	63.91	**65.47**	58.42	**67.53**	**76.64**	**66.39**
Google Translate	Unknown	July 2023	54.73	53.59	52.09	62.03	***67.35***	57.96
Open-source NMT systems								
OPUS	74M	Bilingual En-X models	36.79	29.93	25.94	28.71	27.04	29.68
mBART50	611M	One-to-Many	31.31	26.62	26.63	30.93	26.43	28.38
	611M	Many-to-Many	29.98	25.89	28.12	27.54	24.25	27.16
M2M100	418M	Base	22.35	17.27	12.34	17.01	15.62	16.92
	1.2B	Large	28.81	23.16	17.30	27.03	22.87	23.83
NLLB-200	0.6B	Distilled version	40.93	36.38	28.64	47.13	33.41	37.30
	1.3B	Distilled version	50.40	53.65	41.15	54.52	52.81	50.51
	1.3B	Original checkpoint	48.81	48.43	37.31	54.36	48.93	47.57
	3.3B	Original checkpoint	53.23	57.23	39.95	57.44	56.24	52.82
	54B	Mixture of Experts	61.33	**67.19**	48.02	**67.88**	**67.97**	**62.48**
LLaMA family LLMs								
LLaMA	7B	1-shot prompting	53.64	48.84	30.61†	60.65	57.41	50.23
		3-shot prompting	55.53	50.53	30.52†	57.31	55.34	49.85
		5-shot prompting	56.33	48.66	27.92†	56.83	55.26	49.00
	65B	1-shot prompting	56.57	60.22	44.73†	65.71	62.05	57.86
		3-shot prompting	59.83	60.18	42.77†	***67.45***	63.41	58.73
		5-shot prompting	60.78	***63.47***	42.49†	66.31	62.98	***59.21***
Alpaca	7B	0-shot prompting	49.75	45.24	29.63†	55.23	51.52	46.27
BLOOM family LLMs								
BLOOM	7.1B	1-shot prompting	55.69	28.79†	51.08	40.00†	29.67†	41.05
	176B	1-shot prompting	63.66	42.02†	60.30	43.22†	37.04†	49.25
		3-shot prompting	64.52	46.33†	61.20	44.30†	36.69†	50.61
		5-shot prompting	65.53	45.99†	61.73	42.92†	38.06†	50.85
BLOOMZ	7.1B	0-shot prompting	56.89	33.91†	53.2	33.33†	21.67†	39.80
		1-shot prompting	60.87	40.68†	52.37	33.33†	30.65†	43.58
	176B	0-shot prompting	62.67	45.78†	61.87	47.98†	44.06†	52.47
		1-shot prompting	***64.35***	49.31†	**66.57**	51.88†	43.92†	55.21
		3-shot prompting	**67.31**	45.91†	**64.44**	53.42†	45.08†	55.23
		5-shot prompting	**68.55**	49.22†	***63.36***	52.60†	44.94†	55.73

Table 3: Accuracies on DiBiMT test for establish NMT systems and LLMs, using naive k-shot prompting. For Alpaca, we can only use 0-shot prompting due to its particular prompt template. We highlight the top three scores per language in bold, with the best underlined as well, the 2nd best as is, and the 3rd best italicized. We indicate scores for unseen languages (ie. not intentionally included in pretraining) with a †.

language. Similarly, for BLOOM, Chinese and Spanish are "seen" and the rest are "unseen". We share our key observations below:

1. **LLMs usually match or beat massive MT models on seen languages.** Except for the very rich-resourced En-De, where supervised MT systems appear to have an edge, LLaMA 65B mostly matches the SOTA NMT systems (namely DeepL and NLLB-200). Furthermore, BLOOMZ sets a new SOTA in its seen languages, Spanish and Chinese, and outperforms DeepL by margins of 7.3% and 12.2% respectively. These improvements against such strong, supervised massive NMT systems are particularly remarkable since our corresponding setup for inferencing the LLMs is quite cheap – as we noted previously, this is only naive few-shot prompting of an 8-bit quantized model, with a beam size of 1.

2. **LLMs perform relatively worse for unseen languages, but they can still be much better than some supervised MT models.** We note that relative to seen languages, LLaMA underperforms in translation to Chinese. Similarly,

BLOOM performs worse for its' unseen languages of German, Italian, and Russian. Still, LLMs yield reasonable performance here that is still much better than some supervised NMT systems. For example, BLOOMZ-7B achieves 40.68% accuracy in English-Italian, which is about 35.9% more than OPUS, 52.8% more than mBART50 and 75% more than M2M100-1.2B. While NLLB-200 does outperform BLOOMZ-7B, our results just highlight the power of pretraining at scale.

3. **Scale helps improve performance for ambiguity translation**. Continuing from the last point, similar to NMT models that improve with scale (e.g. NLLB-200), we observe that LLMs too perform consistently better at ambiguous translation on scaling up to their larger variants. This applies to the translation of both seen and unseen languages. That said, the lighter models, such as LLaMA 7B or BLOOM 7B, also perform quite well and in many cases, 1-shot prompting of these LLMs is almost as good as NLLB translations.

4. **LLM performance does improve on average with more demonstrations, but this is not uniform.** On average, we observe that 5-shot prompting works best, followed by 3-shot and then 1-shot, though some outliers exist for LLaMA 7B. Moreover, when looking at the performance of individual language pairs, we note that the improvement trend is not uniform, and it is possible a 3-shot translation outperforms a 5-shot one. This aligns with the finding of Zhang et al. (2023), who reach the same conclusion regarding overall MT quality. Nonetheless, as we show in Section 4.4.1, accuracy does significantly improve when we provide relevant and helpful examples – suggesting quality of demonstrations matters more than quantity.

5. **General-purpose instruction-tuned LLMs consistently outperform foundation LLMs.** Interestingly, we observe that 1-shot prompting of a general-purpose instruction-tuned LLM like BLOOMZ often significantly outperforms 5-shot prompting of BLOOM, even on the very specific task of ambiguity translation. In fact, even with 0-shot prompting, models like Alpaca 7B, BLOOMZ 7B and BLOOMZ 176B perform reasonably well, matching some supervised MT systems. We observed that this did not work for foundation LLMs like BLOOM 165B and LLaMA 7B, and 0-shot prompting of these models yielded hallucinations in many cases.

Lastly, we include a qualitative comparison of DeepL and BLOOMZ 176B translations for the En-Zh pair in the Appendix (see Table 8) – where we observe that BLOOMZ generates more contextual translations, relatively speaking, while its counterpart tends to translate literally in many cases.

4.4 Adapting LLMs for ambiguous MT

This section reports experiments with two proposed strategies to enable LLMs to disambiguate better and improve performance on the ambiguous translation task. While both methods are shown to significantly improve performance, we include a discussion of the relative tradeoffs between the techniques in Appendix A.2.

4.4.1 Improving In-Context Learning by leveraging similar ambiguous contexts

Rather than selecting our examples randomly as in our naive setting, we employ the data selection procedure described in Section 3.3 to discover other examples that contain the same word sense as the most polysemous sense in the input sentence. We report our scores in Table 4, and our findings below:

1. **Similar contexts yield more improvements as the example count increases** We observe that for 1-shot prompting, similar contexts perform comparably or slightly better than random examples. However, the gains increase substantially as we move towards 3-shot and 5-shot prompting. We can understand this from the intuition that 1-shot prompting likely just guides the LLM towards generating a reasonable translation, whereas with more relevant examples, it learns to disambiguate better and translate in context accordingly.

2. **Larger models observe greater and more consistent gains than smaller LLMs** Compared to LLaMA 7B, the other LLMs (LLaMA 65B, BLOOM 176B and BLOOMZ 176B) yield much larger accuracy improvements on a more uniform basis. This is probably because scaling up allows LLMs to model polysemous words better in their semantic space, facilitating effective in-context learning of disambiguation capabilities.

System	1-shot		3-shot		5-shot	
	Rand.	Sim.	Rand.	Sim.	Rand.	Sim.
DeepL	—63.91—					
NLLB-200 54B	—***61.33***—					
LLaMA 7B	53.64	**54.01**	**55.53**	52.52	**56.33**	54.45
LLaMA 65B	56.57	**59.38**	59.83	**62.44**	60.78	**63.74**
BLOOM 176B	**63.66**	62.44	64.52	**66.19**	65.53	**68.22**
BLOOMZ 176B	64.35	**69.57**	67.31	**71.15**	68.55	**71.33**

(a) English-Spanish

System	1-shot		3-shot		5-shot	
	Rand.	Sim.	Rand.	Sim.	Rand.	Sim.
DeepL	—65.47—					
NLLB-200 54B	—***67.19***—					
LLaMA 7B	48.84	**49.47**	50.53	**53.85**	48.66	**52.17**
LLaMA 65B	**60.22**	59.77	60.18	**64.94**	63.47	**65.33**
BLOOM 176B	42.02	**43.17**	46.33	**48.09**	45.99	**50.00**
BLOOMZ 176B	49.31	**49.60**	45.91	**50.73**	49.22	**50.53**

(b) English-Italian

Table 4: 1-shot, 3-shot and 5-shot results for En-Es and En-It prompting with randomised examples (`Rand.`) versus similar contexts (`Sim.`). The best-performing systems from Table 3, i.e. DeepL and NLLB-200 are chosen as baselines. For LLMs, for each setting, the better-performing baseline between `Rand.` and `Sim.` is highlighted in bold. The overall best score (among all LLMs) is underlined as well, while the best NMT system is also italicized.

4.4.2 Fine-tuning with ambiguous corpora

We fine-tune Alpaca 7B, BLOOM 7B and BLOOMZ 7B in En-Es and En-It directions using the data described in Section 4.2. We show our results when prompting these fine-tuned LLMs in Table 5. We make the following observations:

1. **Fine-tuning generally improves performance.** We observe that fine-tuned LLMs significantly outperform their non-finetuned versions in most cases. The biggest improvement is observed for BLOOM 7B in En-It, where accuracy increases by as high as 47.73%, indicating the effectiveness of our method. The only exception to this is when the LLM is already strong, such as BLOOMZ 7B at En-Es, and then the improvements are marginal. But even so, strong instruction-tuned LLMs like BLOOMZ still gain significantly from fine-tuning on the En-It pair – where it was originally weaker due to Italian being an unseen language during pretraining.

2. **Best Cross Entropy does not necessarily translate to best disambiguation accuracy.** Looking at Table 5, we note that the checkpoints with the best cross-entropy fall short of the topline with the best DiBiMT accuracies, suggesting the former is not an optimal metric for this task. Future work could benefit from using disambiguation-specific metrics for validation, leveraging other ambiguous test sets like MuCoW (Raganato et al., 2020b).

3. **Fine-tuning for 2-3 epochs is sufficient.** We plot the DiBiMT accuracy versus epoch curves in Figure 2 where the performance is evaluated after each epoch. We observe that in all cases, accuracy peaks between the 1st and the 3rd epoch, after which it mostly plateaus or dips slightly - suggesting that one does not need to fine-tune these LLMs for too long.

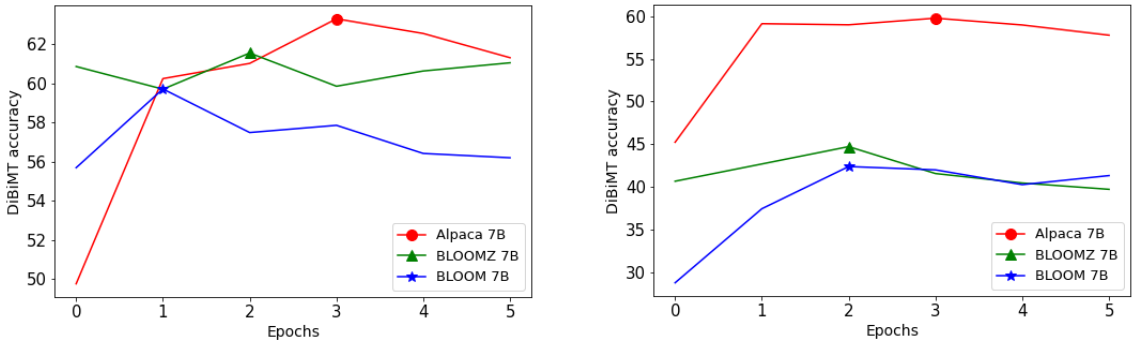
4. **Fine-tuning improves LLM performance until about 65K training samples.** We now try to answer the Research Question of how many training samples we need for fine-tuning these LLMs, to get optimal performance. We plot the Accuracy vs corpus size graph in Figure 3, where we indicate corpus size by the number of parallel sentences. We observe that accuracy increases non-monotonically with an increase in corpus size, but peaks anywhere between 36K-63K training samples, which seems to depend on the pre-existing capabilities of the LLM. For a raw foundation LLM like BLOOM 7B, relatively more fine-tuning data (54K-63K) appears to be beneficial. Alpaca 7B, which has been instruction-tuned on an English-only dataset, also seems to benefit from further fine-tuning—especially for En-Es, accuracy peaks after 63K training samples. However, for a powerful LLM like BLOOMZ that has been instruction-tuned on a large multilingual dataset like xP3 (Muennighoff et al., 2023), fine-tuning on smaller datasets (at most 36K sentences, in our case) appears to suffice.

4.5 Overall MT performance of disambiguation-adapted LLMs

Lastly, for completeness, we evaluate the overall translation quality of the key LLMs used in this work, since we are interested in noting how well the reported disambiguation accuracies extend to overall MT performance. For our test set, we want to choose one recently released (ideally within the

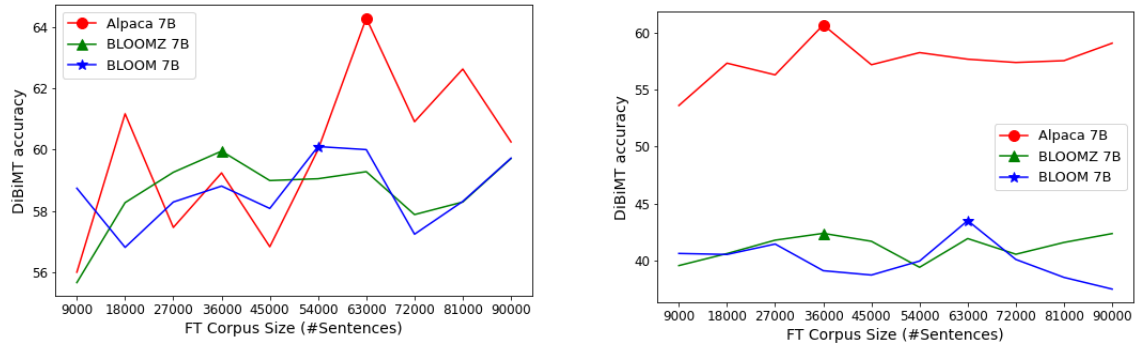
System	En-Es			En-It		
	Alpaca 7B	BLOOM 7B	BLOOMZ 7B	Alpaca 7B	BLOOM 7B	BLOOMZ 7B
w/o FT	49.75	55.69	60.87	45.24	28.79	40.68
FT (Best Cross-Entropy Loss)	63.27	57.86	60.39	59.62	37.72	39.73
FT (Best Attained Acc.)	63.31	59.72	61.56	59.77	42.40	44.73

Table 5: DiBiMT Accuracies after fine-tuning Alpaca 7B, BLOOM 7B, and BLOOMZ 7B on En-Es and En-It pairs. The second row indicates checkpoints with the best cross-entropy loss on the validation set, while the last row shows the one with the best attained DiBiMT accuracy when evaluating after each epoch, and serves as a "topline".



(a) English-Spanish (b) English-Italian

Figure 2: DiBiMT accuracy at the end of every epoch, for the LoRA fine-tuned LLMs



(a) English-Spanish (b) English-Italian

Figure 3: DiBiMT accuracy vs fine-tuning (FT) corpus size in terms of parallel sentence count. These results are obtained from evaluating checkpoints at every 300 steps in the 1st epoch - which roughly corresponds to about 9K sentences, since we use a batch size of 32.

last year) to minimize the chances of its inclusion in the pretraining corpora of LLMs. We, thus, use FLORES 200 (NLLB Team et al., 2022) as our test set since it satisfies this criterion and also supports all our languages of evaluation. We use spBLEU[12] (Goyal et al., 2022), chrF++[13] (Popović, 2017) and COMET22 (Rei et al., 2022) using the `wmt22-comet-da` model as metrics. In this setting, we evaluate Alpaca with 0-shot prompting, while LLaMA 7B, LLaMA 65B and BLOOM 176B use the 1-shot setup. NLLB-200 is our primary supervised NMT baseline. We also evaluate LoRA fine-tuned versions of Alpaca 7B and BLOOM 7B, from section 4.4.2, on the English-Spanish and English-Italian pairs. We exclude BLOOMZ from this evaluation since it is instruction-tuned on FLORES200. We report our results in Table 6.

We observe trends similar to those of our DiBiMT experiments. BLOOM 176B performs well in translation of seen languages, performing comparably to NLLB-200 in English-Spanish and outperforming it in English-Chinese. This is particularly the case for COMET22 scores, a metric which has shown high correlations with human

[12] `nrefs:1|case:mixed|eff:no|tok:flores101|smooth:exp|version:2.3.1`

[13] `nrefs:1|case:mixed|eff:yes|nc:6|nw:2|space:no|version:2.3.1`

System	En-Es			En-It		
	spBLEU	chrF++	COMET22	spBLEU	chrF++	COMET22
NLLB-200 54B	32.50	53.79	0.86	37.60	57.33	0.89
Alpaca 7B (0-shot)	23.90	47.30	0.83	23.30	46.40	0.83
LLaMA 7B (1-shot)	23.20	46.20	0.82	22.10	45.00	0.82
LLaMA 65B (1-shot)	27.20	49.70	0.83	28.50	50.50	0.85
BLOOM 7B (1-shot)	24.00	46.30	0.82	10.00†	33.40†	0.63†
BLOOM 176B (1-shot)	28.60	51.20	0.85	20.80†	45.20†	0.81†
Alpaca 7B (FT, 0-shot)	27.40	50.20	0.85	29.20	51.40	0.87
BLOOM 7B (FT, 0-shot)	28.70	51.00	0.86	20.90	45.80	0.80

System	En-Zh			En-Ru			En-De		
	spBLEU	chrF++	COMET22	spBLEU	chrF++	COMET22	spBLEU	chrF++	COMET22
NLLB-200 54B	23.10	22.83	0.82	38.00	56.34	0.90	44.80	62.79	0.88
Alpaca 7B (0-shot)	4.80†	10.40†	0.62†	21.80	42.60	0.82	27.30	50.30	0.82
LLaMA 7B (1-shot)	5.60†	10.80†	0.66†	20.70	41.20	0.79	22.80	45.40	0.78
LLaMA 65B (1-shot)	13.80†	17.60†	0.77†	26.70	46.10	0.82	31.80	52.80	0.81
BLOOM 7B (1-shot)	19.00	19.50	0.83	3.70†	22.30†	0.46†	8.20†	31.70†	0.51†
BLOOM 176B (1-shot)	25.10	23.80	0.86	10.30†	31.80†	0.65†	19.90†	45.40†	0.74†

Table 6: FLORES 200 results for k-shot prompting of some key LLMs used in this work, compared with the NLLB-200 baseline. We also include results for the LoRA fine-tuned models, for the En-Es and En-It pairs. Same as the previous notation, we indicate all unseen language results with a †. We observe similar trends in all standard MT metrics, as those observed with DiBiMT accuracy.

	spBLEU w/ acc.	ChrF++ w/ acc.	COMET22 w/ acc.
ρ	0.83	0.56	0.76
p-value	0.0001	0.0039	0.0010

Table 7: Pearson's correlation ρ (Benesty et al., 2009) between DiBiMT accuracy and spBLEU, chrF++, and COMET22 respectively, together with p-values.

evaluation, ranking second in the WMT22 Metrics shared task (Freitag et al., 2022). For the other languages, LLaMA 65B usually performs better than BLOOMZ, but in the 1-shot prompting setup, it is unable to beat the NLLB-200 54B MOE. We also notice that the fine-tuned versions of Alpaca 7B and BLOOM 7B consistently outperform their vanilla counterparts – suggesting our techniques to improve disambiguation performance also boost overall translation quality.

Thus, while we evaluate key LLMs to verify consistency in trends, we avoid re-running all our baselines on FLORES200. Instead, we try to answer a broader question: how well does DiBiMT disambiguation accuracy correlate with standard MT metrics? We conduct a Pearson's correlation test (Benesty et al., 2009) between the accuracy metric and spBLEU, chrF++, and COMET22 respectively. We report our results in Table 7, and find that all MT quality metrics correlate positively with accuracy—with p-values of the two-sided alternative hypothesis being much lesser than 0.05 in all cases. We discover that spBLEU and COMET22 exhibit higher correlations than chrF++. We hypothesize that this could be due to the character-level chrF++ being less sensitive to word-level senses. Overall, the results of Tables 6 and 7 suggest that the significant accuracy improvements noted earlier are not at the cost of translation quality, and in turn, could yield improvements in overall MT scores too.

5 Conclusion

In this work, we studied the capabilities of LLMs to handle ambiguity during machine translation. We choose seven of the most widely used foundation and instruction-tuned LLMs and compare accuracy with SOTA commercial and open-source NMT systems on the DiBiMT translation benchmark. Out of 5 language directions, we report scores comparable to the SOTA on two (En-Ru, En-It) and set a new SOTA on two others (En-Zh, En-Es). We then present two techniques that significantly improve disambiguation accuracy: in-context learning with similar contexts, and fine-tuning on an ambiguous corpus. We end the paper with an evaluation of overall MT quality. We hope the methods and findings shared in this work could guide future researchers studying ambiguity in translation.

Limitations

In this work, we attempt to note overall trends in LLM performance as compared to conventional NMT systems and, based on our results, suggest methods that generally improve performance. That said, there are exceptions to these trends - prompting with similar contexts can, at times, degrade performance and so can increasing the number of demonstrations (see Table 4). But there is some consistency here too that these observations mostly apply to smaller LLMs (such as LLaMA 7B) while the larger LLMs benefit more significantly. Also, as noted in Section 4.4.1, in a small percentage of cases (7.5%), we are unable to find 5 matches when attempting 5-shot prompting with similar contexts. In such cases, it might be worthwhile, from a performance perspective, to use random demonstrations; nonetheless, since we are interested in verifying the utility of similar contexts and also since there are only a few cases where it might be pertinent, we do not explore this.

Acknowledgements

This work has received funding from UK Research and Innovation under the UK government's Horizon Europe funding guarantee [grant numbers 10039436 and 10052546].

The computations described in this research were performed using the Baskerville Tier 2 HPC service (https://www.baskerville.ac.uk/). Baskerville was funded by the EPSRC and UKRI through the World Class Labs scheme (EP/T022221/1) and the Digital Research Infrastructure programme (EP/W032244/1) and is operated by Advanced Research Computing at the University of Birmingham.

References

Edoardo Barba, Tommaso Pasini, and Roberto Navigli. 2021. ESC: Redesigning WSD with extractive sense comprehension. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4661–4672.

Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. 2009. Pearson correlation coefficient. In *Noise reduction in speech processing*, pages 37–40. Springer.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Niccolò Campolungo, Federico Martelli, Francesco Saina, and Roberto Navigli. 2022. DiBiMT: A novel benchmark for measuring Word Sense Disambiguation biases in Machine Translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4331–4352, Dublin, Ireland. Association for Computational Linguistics.

Marine Carpuat and Dekai Wu. 2007. Improving statistical machine translation using word sense disambiguation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 61–72, Prague, Czech Republic. Association for Computational Linguistics.

Yee Seng Chan, Hwee Tou Ng, and David Chiang. 2007. Word sense disambiguation improves statistical machine translation. In *Proceedings of the 45th annual meeting of the association of computational linguistics*, pages 33–40.

Pinzhen Chen, Zhicheng Guo, Barry Haddow, and Kenneth Heafield. 2023. Iterative translation refinement with large language models. *arXiv preprint*.

Heeyoul Choi, Kyunghyun Cho, and Yoshua Bengio. 2017. Context-dependent word representation for neural machine translation. *Computer Speech & Language*, 45:149–160.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. PaLM: Scaling language modeling with pathways. *arXiv preprint*.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, et al. 2021. Beyond english-centric multilingual machine translation. *The Journal of Machine Learning Research*, 22(1):4839–4886.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, Eleftherios Avramidis, Tom Kocmi, George Foster, Alon Lavie, and André F. T. Martins. 2022. Results of WMT22 metrics shared task: Stop using BLEU – neural metrics are better and more robust. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 46–68, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538.

Zhiwei He, Tian Liang, Wenxiang Jiao, Zhuosheng Zhang, Yujiu Yang, Rui Wang, Zhaopeng Tu, Shuming Shi, and Xing Wang. 2023. Exploring human-like translation strategy with large language models. *arXiv preprint*.

Amr Hendy, Mohamed Gomaa Abdelrehim, Amr Sharaf, Vikas Raunak, Mohamed Gabr, Hitokazu Matsushita, Young Jin Kim, Mohamed Afify, and Hany Hassan Awadalla. 2023. How good are GPT models at machine translation? a comprehensive evaluation. *arXiv preprint*.

J. Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. LoRA: Low-rank adaptation of large language models. *arXiv preprint*.

Tom Kocmi and Christian Federmann. 2023. Large language models are state-of-the-art evaluators of translation quality. *arXiv preprint*.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of machine translation summit x: papers*, pages 79–86.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Frederick Liu, Han Lu, and Graham Neubig. 2018. Handling homographs in neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1336–1345, New Orleans, Louisiana. Association for Computational Linguistics.

Yasmin Moslem, Rejwanul Haque, and Andy Way. 2023. Adaptive machine translation with large language models. *EAMT*.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual generalization through multitask finetuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15991–16111, Toronto, Canada. Association for Computational Linguistics.

Roberto Navigli, Michele Bevilacqua, Simone Conia, Dario Montagnini, and Francesco Cecconi. 2021. Ten years of BabelNet: A survey. In *IJCAI*, pages 4559–4567.

NLLB Team, Marta Ruiz Costa-jussà, James Cross, Onur cCelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Alison Youngblood, Bapi Akula, Loïc Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon L. Spruit, C. Tran, Pierre Yves Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzm'an, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation. *ArXiv*, abs/2207.04672.

Jonathan Pilault, Xavier Garcia, Arthur Bražinskas, and Orhan Firat. 2023. Interactive-chain-prompting: Ambiguity resolution for crosslingual conditional generation with interaction. *arXiv preprint*.

Maja Popović. 2017. chrf++: words helping character n-grams. In *Proceedings of the second conference on machine translation*, pages 612–618.

Xiao Pu, Nikolaos Pappas, James Henderson, and Andrei Popescu-Belis. 2018. Integrating weakly supervised word sense disambiguation into neural machine translation. *Transactions of the Association for Computational Linguistics*, 6:635–649.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Alessandro Raganato, Yves Scherrer, and Jörg Tiedemann. 2019. The mucow test suite at wmt 2019: Automatically harvested multilingual contrastive word sense disambiguation test sets for machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 470–480.

Alessandro Raganato, Yves Scherrer, and Jörg Tiedemann. 2020a. An evaluation benchmark for testing the word sense disambiguation capabilities of machine translation systems. In *Proceedings of The 12th Language Resources and Evaluation Conference*. European Language Resources Association (ELRA).

Alessandro Raganato, Yves Scherrer, and Jörg Tiedemann. 2020b. An evaluation benchmark for testing the word sense disambiguation capabilities of machine translation systems. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3668–3675, Marseille, France. European Language Resources Association.

Vikas Raunak, Arul Menezes, Matt Post, and Hany Hassan. 2023. Do GPTs produce less literal translations? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1041–1050, Toronto, Canada. Association for Computational Linguistics.

Ricardo Rei, José G. C. de Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André F. T. Martins. 2022. COMET-22: Unbabel-IST 2022 submission for the metrics shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 578–585, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint*.

Yves Scherrer, Alessandro Raganato, and Jörg Tiedemann. 2020. The MUCOW word sense disambiguation test suite at WMT 2020. In *Proceedings of the Fifth Conference on Machine Translation*, pages 365–370, Online. Association for Computational Linguistics.

Gongbo Tang, Rico Sennrich, and Joakim Nivre. 2019. Encoders help you disambiguate word senses in neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1429–1435, Hong Kong, China. Association for Computational Linguistics.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2021. Multilingual translation from denoising pre-training. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3450–3466, Online. Association for Computational Linguistics.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford Alpaca: An instruction-following LLaMA model. `https://github.com/tatsu-lab/stanford_alpaca`.

Jörg Tiedemann and Santhosh Thottingal. 2020. Opus-mt–building open translation services for the world. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*. European Association for Machine Translation.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. LLaMA: Open and efficient foundation language models. *arXiv preprint*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

David Vilar, Markus Freitag, Colin Cherry, Jiaming Luo, Viresh Ratnakar, and George Foster. 2023. Prompting PaLM for translation: Assessing strategies and performance. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15406–15427, Toronto, Canada. Association for Computational Linguistics.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Warren Weaver. 1952. Translation. In *Proceedings of the Conference on Mechanical Translation*.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022a. Emergent abilities of large language models. *Transactions on Machine Learning Research*. Survey Certification.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022b. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Biao Zhang, Barry Haddow, and Alexandra Birch. 2023. Prompting large language model for machine translation: A case study. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 41092–41110. PMLR.

Source	Tap the head of the drum for this roll.	
DeepL	敲击鼓的头部进行滚奏。	*head, literal*
BLOOMZ	敲击鼓面，发出这个鼓点。	*striking surface of a drum*
Source	they tracked him back toward the head of the stream.	
DeepL	他们跟踪他回到溪头.	*head (literal and unnatural)*
BLOOMZ	他们跟着他回到了河的上游。	*upstream*
Source	The expedition followed the river all the way to the head.	
DeepL	探险队沿着河流一直走到河头。	*head (literal and unnatural)*
BLOOMZ	探险队顺着河一直走到源头。	*source*
Source	How much head do you have at the Glens Falls feeder dam?	
DeepL	格伦瀑布支坝的水头有多大?	*hydraulic head*
BLOOMZ	你有多少头牛在格伦瀑布的蓄水池里?	*(a classifier word to express quantities (of cows))*
Source	the office was full of secret heads.	
DeepL	办公室里到处都是秘密头目.	*leader, ringleader*
BLOOMZ	办公室挤满了神秘的首脑。	*leader, head of state*

Table 8: Manual inspection on English-to-Chinese translation focused on the disambiguation of "head", corresponding to the first five test instances in DiBiMT. The baselines are DeepL and BLOOMZ 176B, the highest performing NMT system and LLM for this pair (from Table 3). The reported annotations are obtained from a native Chinese speaker who was invited to label the sense of the translated ambiguous word.

A Appendix

A.1 Qualitative comparison: BLOOMZ vs DeepL

We choose the best-performing LLM and the SOTA MT system from Table 3 – focusing on the En-Zh pair since LLMs seem to yield the highest gains there. With the help of a native Chinese speaker, we got hypotheses from these two systems annotated, for the first 5 sentences of the DiBiMT test set. We observe that although there are cases where DeepL gets it right over BLOOMZ (example 4) or where both are correct (Example 5), in many instances BLOOMZ appears to generate more contextual (and less literal) translations. We hypothesize that this could potentially be due to the former's powerful language modelling abilities

A.2 Trade-off between prompting and fine-tuning

We show in Section 4.4 that both prompting with similar contexts through In-Context Learning (ICL) and LoRA fine-tuning can significantly improve performance. However, depending on the use case, it might be better to favour one over the other. For instance, in production environments, LLMs that are LoRA fine-tuned on ambiguous text can provide powerful disambiguation performance, while also being more feasible to deploy and run at scale. In contrast, ICL with k-shot prompting, especially for higher values of k, can significantly increase query size and memory consumption, necessitating reduced batch size and thus, throughput.

However, conducting ICL with similar ambiguous contexts can be used to query LLMs as large as LLaMA 65B and BLOOMZ 176B and yield performance comparable to SOTA MT systems (see Table 4). The preprocessing cost overhead of such a method, namely disambiguating the test set, is also low - it took us about 13 seconds to disambiguate a test set of about 500 sentences on 1 Nvidia GeForce RTX 3090. In contrast, the one-time cost of fine-tuning can be quite expensive—for instance, it took us 44 hours to fine-tune an Alpaca 7B with LoRA on a single Nvidia Tesla A100 40G. Thus, in GPU-scarce settings where the costs of LoRA fine-tuning are prohibitive, it might be favourable to use ICL to query massive LLMs and obtain SOTA performances. In contrast, production environments are likely to prefer the fine-tuned LLMs, since the one-off fine-tuning costs can be amortized.