



SALT at SemEval-2025 Task 2: A SQL-based Approach for LLM-Free Entity-Aware-Translation

Tom Völker and Jan Pfister and Andreas Hotho

Center for Artificial Intelligence and Data Science (CAIDAS)
Data Science Chair, Julius-Maximilians-Universität Würzburg (JMU)
{voelker,pfister,hotho}@informatik.uni-wuerzburg.de

Abstract

Entity-aware machine translation faces significant challenges when translating culturally-adapted named entities that require knowledge beyond the source text. We present SALT  (SQL-based Approach for LLM-Free Entity-Aware-Translation), a parameter-efficient system for the SemEval-2025 Task 2. Our approach combines SQL-based entity retrieval with constrained neural translation via logit biasing and explicit entity annotations. Despite its simplicity, it achieves state-of-the-art performance (First Place) among approaches not using gold-standard data, while requiring far less computation than LLM-based methods. Our ablation studies show simple SQL-based retrieval rivals complex neural models, and strategic model refinement outperforms increased model complexity. SALT  offers an alternative to resource-intensive LLM-based approaches, achieving comparable results with only a fraction of the parameters.


1 Introduction

Despite ever-progressing language model capabilities, they continue to struggle with tasks requiring precise factual knowledge and cross-cultural understanding (Wang et al., 2024; Lin et al., 2022; Hu et al., 2024). One such challenge is named entity translation, where direct word-for-word approaches often miss cultural nuances (Díaz-Millón and Olvera-Lobo; Gaballo et al., 2012). Accurate entity-aware translation is essential for preserving meaning across languages. For example, Roald Dahl’s *The Witches* became *Hexen hexen* (“Witches bewitch”) in German – a choice no model could infer from the source alone.

Recent advances in machine translation, particularly with large language models (LLMs), have greatly improved translation quality (Team et al., 2022; Tang et al., 2020; Workshop et al., 2023; Zhu et al., 2024). However, translating cultur-

ally adapted entity names remains difficult (Hershcovich et al., 2022) due to: 1) The need for transcreation—creative adaptation beyond literal translation (Gaballo et al., 2012) 2) Constantly emerging entities, which frozen LLM weights cannot capture (Lazaridou et al., 2021; Hu et al., 2024) 3) Variability in translation based on cultural, geographical, or temporal context (Hershcovich et al., 2022)

The SemEval-2025 Task 2 on Entity-Aware Machine Translation (EA-MT) (Conia et al., 2025) addresses this challenge by requiring systems to translate English sentences with named entities into ten target languages, spanning both Latin (e.g., German, Spanish) and non-Latin scripts (e.g., Japanese, Arabic). For example, “What year did Roald Dahl release the novel *The Witches*?” should be translated into German as “In welchem Jahr veröffentlichte Roald Dahl den Roman *Hexen hexen*?” – using the localized title.

We present SALT ¹ (SQL-based Approach for LLM-Free Entity-Aware-Translation), a simple yet effective solution to this challenge. While based on neural machine translation (Team et al., 2022), SALT avoids the complexity of additional neural components for entity handling and the parametric overhead of modern LLMs. Instead, it leverages efficient SQL-based entity retrieval, constrained neural translation via logit biasing, and explicit entity annotations. Our system achieves state-of-the-art results among approaches without gold-standard data (e.g., Wikidata IDs) during testing, while maintaining significantly lower computational costs (Conia et al., 2025).


Our key contributions are:

- A parameter-efficient entity-aware translation approach that achieves competitive results without additional trainable components beyond the base model.
- Evidence that, for well-structured multilingual

¹github.com/LSX-UniWue/Semeval-2025-Task-2

knowledge bases, simple SQL-based retrieval can rival complex neural methods while being significantly more efficient.

- Comprehensive ablation studies comparing retrieval and integration strategies, showing that explicit knowledge integration via entity annotations and logit biasing outperforms added neural complexity or increased parameter counts (Zhang et al., 2018; Lewis et al., 2021).

While LLM-based approaches achieve marginally better results in our ablation studies, SALT  comes remarkably close with only a fraction of the parameters.

2 Related Work

Named Entity Linking. Named Entity Linking (NEL) maps entity mentions in text to knowledge base entries. Early methods used string matching and heuristics (Shen et al., 2015), while modern approaches usually employ neural models that encode mention context and entity representations via transformers and graph neural networks for better disambiguation (Kolitsas et al., 2018; Cao et al., 2018; Wu et al., 2020; Conia et al., 2024). Many state-of-the-art systems follow a two-stage process: efficient candidate generation followed by neural re-ranking (Lai et al.; Hebert et al.).

Augmented Neural Translation. Neural Machine Translation (NMT) often struggles with low-frequency or novel entities. Retrieval-augmented techniques incorporate external translations at inference time (Zhang et al., 2018), while lexically constrained decoding enforces correct entity translation (Hokamp and Liu, 2017). Other methods integrate external knowledge via data augmentation or explicit entity translation modules (Campolungo et al., 2022; Zeng et al.; Conia et al., 2024).

3 System Description

We propose a surprisingly simple yet effective two-stage pipeline for entity-aware machine translation, that focuses on parameter efficiency. Our approach consists of (3.1) a deterministic entity retrieval and translation lookup phase, followed by (3.2) a constrained neural translation step. Despite exploring more complex methods in our ablation studies (Section 6), this streamlined approach achieves highly competitive results with significantly lower computational cost.

3.1 Entity Retrieval and Translation Lookup

Given an English source sentence and a target language, our system first identifies relevant named entities and retrieves their translations from a knowledge base. This forms the first stage of our pipeline. With Wikidata containing over 71 million entities², efficient candidate filtering is essential. To this end, we implement a normalized string matching approach, leveraging SQL indexing for fast retrieval.

For an input sentence $x = \langle w_1, \dots, w_n \rangle$, we generate all possible n-grams³, normalize them (lowercasing and removing special characters), and query our database for exact matches with identically normalized entity names.⁴ Only entities with available translations in the target language are considered.

For each exact n-gram matched entity s in the database, we compute a relevance score prioritizing longer entity matches:

$$\text{score}(s, x) = 0.5 \cdot \frac{|\text{chars}(s)|}{|\text{chars}(x)|} + 0.5 \cdot \frac{|\text{words}(s)|}{|\text{words}(x)|}$$

This ensures multi-word entities are ranked higher (e.g., “The Lord of the Rings” would score higher than just “Rings”). This is based on our observation that longer, multi-word entities are more likely to have non-trivial translations requiring special handling, while shorter matches might simply be components of these larger entities. Ties are broken using entity popularity (measured by Wikidata history length, representing past edit activities).

For each input sentence x , this process yields a set of entity-translation pairs $\mathcal{E}_x = \{(e_1, t_1), \dots, (e_k, t_k)\}$, where e_i represents the source entity text and t_i its translation in the target language.

This approach offers excellent scalability advantages: it handles Wikidata’s massive entity collection efficiently through indexing, and unlike neural approaches, can be dynamically updated with new entities without retraining. This allows the system to remain current as new entities emerge in the real world.

3.2 Neural Translation with Knowledge Integration

The second stage of our pipeline adapts the knowledge integration approach introduced by Conia et al.

²wikidata.org/wiki/Wikidata:Statistics

³Up to $k = 15$, as this covers all entities in the datasets.

⁴The database construction scripts are available in our GitHub repository under the [/data/wikidata](#) directory.

(2024) to incorporate the retrieved entity translations into the translation process. Based on their findings, we use the encoder-decoder 600M NLLB-200 model (2022) as our base architecture.

Given a source text $x = \langle w_1, \dots, w_n \rangle$ and its corresponding entity-translation pairs \mathcal{E}_x , we construct an augmented input sequence:

$$x^+ = \langle w_1, \dots, w_n, \langle \text{meta} \rangle, e_1, \langle \text{translates_to} \rangle, t_1 \rangle,$$

where special tokens explicitly mark entity-translation pairs. This augmented sequence provides the model with direct access to the high-quality entity translation sourced from our knowledge base.

Unlike Conia et al., who provided the translation model with multiple translation candidates, our ablation studies (Section 6.3) show that selecting only the highest-scoring candidate significantly improves performance. When no matched entity is found in our SQL-based lookup, we simply refrain from amending entity-translation pairs to the input sentence. This defaulting-to-base strategy ensures that unmatched entities do not degrade overall translation quality. In principle, more sophisticated fallback methods (e.g., approximate string matching or partial re-ranking) could address near-miss matches, but we found our simpler approach to be sufficient for most test instances (Section 6.1).

To further encourage the model to use these curated translations in its output, we implement logit biasing during the beam search decoding process. This effectively creates a soft constraint that encourages the model to incorporate the retrieved entity translations while maintaining the flexibility to adapt to target language grammar and ensure overall translation fluency. To this end, we positively bias all logits corresponding to tokens present in the retrieved translation target (t_1), as these represent gold-standard entity renderings in the target language (shown to be effective by Zhang et al. (2018)):

$$p(y_t | y_{<t}, x^+) \propto \exp(\text{logits}(y_t) + b \cdot \mathbb{I}[y_t \in \text{tokens}(t_i)])$$

where b is the bias parameter and \mathbb{I} the indicator function. This mechanism allows us to guide the translation process without forcing rigid token copying that might result in grammatically incorrect output. Our approach improves upon Conia et al. (2024) in two ways: 1) selecting only one entity translation per instance, which enhances accu-

racy (Section 6), and 2) combining explicit knowledge integration with logit biasing, ensuring high entity translation accuracy without compromising overall fluency.

4 Experiments and Results

4.1 Experimental Setup

We evaluate our approach on the XC-Translate dataset (Conia et al., 2024), which includes 7,000 development and 50,000 test samples evenly distributed across ten target languages.

Following task guidelines, we train on both the development split and external data. For the latter, we use the Mintaka dataset (Sen et al.), a multilingual QA dataset with Wikidata annotations, suitable for entity-aware translation. To maintain quality comparable to XC-Translate, we apply strict filtering: 1) The translated entity must appear in the target sentence. 2) The Levenshtein distance (Miller et al., 2009) between source and target entity translation must exceed two characters to exclude trivial cases. This retains 40% of Mintaka while aligning it with XC-Translate’s sample characteristics.⁵ We combine these filtered samples with the development set in a 1:1 ratio, which our ablation studies (Section 6) show to strike a fair balance between performance and training time.

4.2 Training Configuration

We fine-tune the 600M NLLB-200 model (Team et al., 2022) using AdamW (Loshchilov and Hutter, 2019) with a 1e-5 learning rate. Training takes ≈ 1.5 hours on NVIDIA L40 GPUs, while evaluation, including COMET score computation, requires significant additional time. Full training parameters are in Appendix A.

4.3 Evaluation

The task employs two complementary metrics: M-ETA (Manual Entity Translation Accuracy) measures entity translation quality by checking for correct translations in system outputs via case-insensitive substring matching (Conia et al., 2024). COMET (Rei et al.) assesses overall translation quality using a neural model trained to predict human judgments of fluency and adequacy given the target translation. Systems are ranked using the

⁵Mintaka provides data for Arabic, French, German, Hindi, Italian, Japanese, Portuguese and Spanish, of which we use the six languages that overlap with XC-Translate (Arabic, German, Spanish, French, Italian and Japanese). This means we lack Mintaka data for Korean, Chinese, Thai and Turkish.

harmonic mean of both metrics, ensuring neither entity accuracy nor translation fluency is disproportionately favored.

5 Results

Table 1 compares our system’s performance to selected others across ten target languages. Our approach achieves an M-ETA score of 71.66% and a COMET score of 92.52, ranking highest among systems not accessing gold-standard Wikidata entity IDs in inference, with a HM-Score of 80.42.

Our SQL-based retrieval and constrained neural translation prove effective across all languages, outperforming both the top overall LLM-based system by FII-UAIC-SAI (78.17) and the next best non-LLM-based system by team Zero (47.79).⁶ The substantial 25-point gain over the baseline by Conia et al. (55.32) underscores the value of our small but substantial methodological refinements outlined above.

Performance is strongest on languages with substantial Mintaka training data (Arabic, German, Spanish, French, Italian, and Japanese), where M-ETA scores range from 72.20% to 81.72%. Chinese remains the most challenging (45.27% M-ETA), with FII-UAIC-SAI surpassing our system by 17.23 points.

For reference, we include the top-performing shared task submission (pingan_team) in gray, achieving a remarkable 91.79 overall using gold Wikidata annotations at inference. While this limits real-world applicability, it demonstrates the high potential upper bound of the dataset.

6 Ablations and Analyses

Having established the effectiveness of our minimal approach, we now investigate both the validity of our design choices and the potential gains in more complex alternatives, allowing us to quantify trade-offs while confirming our core architectural decisions.

6.1 What are the Limits of String Matching?

Our SQL-based approach achieves 83.05% Recall@1 for identifying the correct entity ID and 72.07% Rec@1 for retrieving the exact translation used in the target sentence (Table 2). The 83.05%

⁶We define LLM-based approaches as those utilizing decoder-only transformer models with billions of parameters, such as GPT (Brown et al.) or Llama (Grattafiori et al.) variants, as opposed to our encoder-decoder architecture with significantly fewer parameters.

entity identification performance is comparable to the 85.90% Rec@1 reported by Conia et al. using a neural retriever, suggesting our lightweight method offers a good efficiency-performance trade-off.

Dataset analysis highlights inherent retrieval limitations: “only” 97.74% of correct entities appear verbatim in the source, setting an upper bound, while another 0.46% differ slightly (edit distance ≤ 3). The remaining 2.26% vary significantly, where dense retrieval might help, but we felt the computational cost wouldn’t justify these minimal theoretical gains, making our string matching approach a reasonable compromise.

Even with gold-standard entity IDs, only 84.39% of Wikidata translations match target sentence renderings, with 10.52% differing substantially (edit distance > 3). This discrepancy imposes an 84.39% M-ETA ceiling, irrespective of retrieval method.

6.2 Should we use an LLM-based Reranker?

To assess how close we can get to the theoretical upper bounds identified above, we evaluate two well established neural reranking approaches beyond our SQL-based retrieval (Appendices B and C.1): a fine-tuned transformer-based cross-encoder and an LLM-based method (Table 3). The pre-trained transformer showed negligible gains in zero-shot settings, but fine-tuning improved Rec@1 by 6.22 points from 72.07% to 78.29%. For the LLM-based approach, we employ GPT-4o mini⁷ with a structured prompt that considers sentence context and entity metadata, achieving a slightly worse 77.26% Rec@1.

When integrated into the full translation pipeline (Table 4), it is able to retain almost all the improvement, boosting the M-ETA score by 5.47% to 77.13% without affecting translation quality. However, we chose to exclude neural rerankers to keep the pipeline simple, transparent, and computationally efficient and “explainable”.

6.3 How Many Entities to Provide for Translation?

Contrary to expectations based on Conia et al.’s (2024), who provided the top-3 candidates to the translation model (Section 3.2), we found that appending only one entity candidate performed significantly better (Figure 1). This may be due to reduced ambiguity: while 12.9% of correct entity translations appear only in positions 2-5 and are

⁷GPT-4o mini, used model with timestamp ‘gpt-4o-mini-2024-07-18’

System	AR		DE		ES		FR		IT		JA		KO		TH		TR		ZH		Avg		
	M	C	M	C	M	C	M	C	M	C	M	C	M	C	M	C	M	C	M	C	M	C	H
pingan_team	91.73	93.64	86.35	94.05	90.13	95.09	91.56	94.31	93.02	95.80	91.41	95.36	90.24	95.44	91.18	93.55	84.13	95.70	81.26	94.44	89.10	94.74	91.79
FII-UAIC-SAI	66.42	91.35	66.98	91.30	72.35	92.58	72.46	90.59	75.79	92.71	67.03	93.56	66.02	92.78	65.25	88.62	67.56	91.63	62.50	91.25	68.24	91.64	78.17
Zero	37.50	90.82	40.32	90.62	46.46	92.38	33.16	89.06	39.37	90.78	35.28	92.57	35.97	91.78	13.75	82.61	46.50	93.83	8.41	88.98	33.67	90.34	47.79
Conia et al.	50.60	-	36.50	-	47.80	-	39.80	-	47.50	-	42.20	-	47.10	-	39.60	-	49.70	-	10.60	-	41.10	84.60	55.32
NLLB-200	20.50	-	19.60	-	31.50	-	24.70	-	26.40	-	8.40	-	17.70	-	1.80	-	25.40	-	3.10	-	17.90	81.90	29.38
Our system	81.72	93.20	73.77	92.34	74.58	93.60	74.77	91.84	77.62	93.36	72.20	93.02	74.24	92.97	65.59	90.64	76.86	94.47	45.27	89.75	71.66	92.52	80.42

Table 1: Results across languages with (M)-ETA and (C)omet scores, along with (H)armonic Mean. Language codes: Arabic (AR), German (DE), Spanish (ES), French (FR), Italian (IT), Japanese (JA), Korean (KO), Thai (TH), Turkish (TR), and Chinese (ZH). Results in **bold** indicate highest scores among systems not using gold data. The pingan_team results were the best overall, but use gold data and are thus not directly comparable with our system. NLLB-200 represents the non-finetuned NLLB model with results as reported by Conia et al..

Metric	Test set retrieval performance		
	Recall@1	Recall@3	Recall@5
Entity ID	83.05%	91.65%	92.96%
Entity name	87.56%	93.79%	94.09%
Entity translation	72.07%	80.59%	81.60%
Conia et al.	85.90%	92.10%	-

Table 2: SQL-based retrieval performance on the development set, showing both entity identification (Entity retrieval) and correct translation retrieval (Entity transl.).

Reranking method	Translation retrieval performance		
	R@1	R@3	R@5
SQL-only (3.1)	72.07%	80.59%	81.60%
Transformer reranker	78.29%	81.43%	81.77%
LLM reranker	77.26%	79.81%	79.88%

Table 3: Comparison of entity reranking approaches.

omitted, the confusion from multiple candidates apparently outweighs this potential gain.

This insight shaped our system design, revealing that while the translation model effectively uses our amended syntax as a glossary, it struggles when simultaneously tasked with selecting between entity candidates. In our parameter-efficient neural translation pipeline, clarity in entity mapping proves more valuable than maximizing coverage.

6.4 Should we Augment the Training Data?

We evaluated four training data configurations Table 5): XC-Dev only (~7,000 samples, 78.90% HM-Score), filtered Mintaka only (~50,000 samples, 75.41% HM-Score), and two combinations in different ratios. While XC-Dev outperformed Mintaka alone, their combination yielded the best results (80.42% and 80.58% HM-Score for 1:1 and 7:1 ratios, respectively). The 1:1 ratio (our choice) balances performance and training efficiency, whereas the 7:1 ratio (full datasets) offers only marginal gains at much higher computational

Reranking method	Average across all languages		
	M-ETA	Comet	HM-Score
SQL-only (3.1)	71.66%	92.52%	80.42%
Transformer reranker	77.13%	92.75%	84.22%

Table 4: Reranking impact on translation performance.

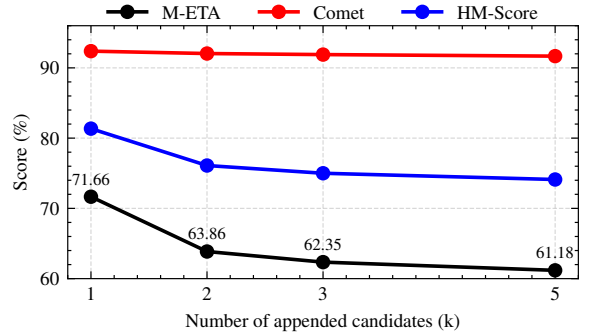


Figure 1: Impact of different Top-k append strategies.

cost. These findings highlight the datasets' complementary nature and the diminishing returns of increasing amounts of data beyond a certain point.

6.5 What's the Impact of our Logit Biasing?

As the last substantial difference between our approach and that of Conia et al. (2024), we analyze the impact of our logit biasing strategy and compare it to an additional constraining mechanism. Our main system employs logit biasing (Zhang et al., 2018), applying a positive bias to tokens from the retrieved entity translation during generation, guiding output without adding model parameters.

As an alternative, we evaluate a pointer-generator mechanism (See et al.), which augments the model with a trainable copy component. Like logit biasing, it aids token selection but is learnable. Instead of direct copying, it computes an attention-based probability distribution over input tokens alongside the vocabulary distribution, combining them via a learnable gate to balance generation and

Training data	Average across all languages		
	M-ETA	Comet	HM-Score
XC-Dev only	69.62%	91.03%	78.90%
Mintaka only	64.87%	90.03%	75.41%
Mintaka + XC-Dev (1:1)	71.66%	92.52%	80.42%
Mintaka + XC-Dev (7:1)	71.78%	92.61%	80.58%

Table 5: Impact of different training data combinations.

copying. While successfully applied in translation (Zeng et al.), it increases architectural complexity and requires additional parameters.

Constraint method	Average across all languages		
	M-ETA	Comet	HM-Score
No constraint	68.30%	92.71%	78.65%
Logit bias (3.2)	71.66%	92.52%	80.42%
Pointer generator	69.10%	92.63%	79.16%
PG + Logit bias	73.47%	92.31%	81.81%

Table 6: Performance of different constraint methods.

As shown in Table 6, our logit biasing approach improves M-ETA by 3.36 percentage points over the unconstrained baseline with minimal impact on translation quality, justifying its inclusion in our main pipeline. The pointer generator also enhances entity translation but less effectively than logit biasing, despite adding parameters.

Interestingly, combining both methods achieves the best results (M-ETA 73.47%, HM-Score 81.81%), suggesting a synergistic effect between the biasing and the pointer generator’s mechanism.

For our final system, we retain the simpler logit biasing approach, balancing performance and parameter efficiency to maintain a lightweight yet effective model.

6.6 “Why didn’t you just use an LLM?”

LLM approach	Average across all languages		
	M-ETA	Comet	HM-Score
SALT (no LLM)	71.66%	92.52%	80.42%
+ Reranker&Pointer	77.77%	92.63%	84.55%
LLM direct translation	78.77%	93.42%	85.17%
Vanilla NLLB + LLM	72.35%	87.48%	78.87%
Finetuned NLLB + LLM	77.13%	91.81%	83.63%

Table 7: Comparisons to our best non-LLM configuration (+ Reranker&Pointer) which combines transformer reranking (Section 6.2) with pointer generator and logit biasing (Section 6.5).

Lastly, we explore LLMs in our translation

pipeline, testing three GPT-4o mini⁷-based approaches (Table 7). For comparison, we include our baseline SALT (80.42% HM-Score) and our best non-LLM system (84.55%), which integrates a transformer reranker, pointer generator, and logit biasing (Sections 6.2 and 6.5).

First, we assess whether an LLM can refine finetuned NLLB translations by correcting entity mistranslations while preserving overall quality (Appendix C.4), as unlike our neural translation model (Section 6.3), LLMs should excel at disambiguation tasks like this. Providing the translation and top-5 entity candidates yields an 83.63% HM-Score, slightly below our best non-LLM system. A similar approach using vanilla (non-finetuned) NLLB translations performs worse (78.87%), suggesting LLMs struggle with lower-quality base translations (Appendix C.3).

Surprisingly, our best result (85.17% HM-Score) comes from direct LLM translation, using only the source text and entity candidates (Appendix C.2). However, the modest 0.62-point gain over our best non-LLM system comes at a cost: higher computation, API expenses, latency (>2s per sample), and reliance on closed-source models. This narrow performance gap validates our parameter-efficient pipeline as a competitive alternative that avoids these limitations.

7 Conclusion

We introduced SALT, a parameter-efficient, entity-aware machine translation approach that achieves first place among models not using gold data during translation. Our ablation studies challenge several intuitive assumptions: simpler retrieval methods often outperform complex ones; clarity trumps coverage when providing entity candidates; and lightweight techniques like logit biasing can match parameter-heavy approaches. The narrow gap between our system and LLM-based alternatives (0.62 pp) demonstrates that parameter efficiency need not sacrifice translation quality. Our work counters the prevailing trend toward ever-larger models, suggesting that targeted knowledge integration can be more effective than simply scaling parameters for specialized translation tasks.

Whilst achieving state-of-the-art results without gold-standard data, challenges remain, particularly in our Chinese translations and in generalizing to more diverse real-world translation scenarios.

Limitations

Despite SALT’s strong performance, a few limitations remain: (1) our approach is bounded by Wikidata coverage, with a theoretical M-ETA ceiling of 84.39% (Section 6.1) far from the top submissions in the task, which use gold data; (2) performance varies across languages, with Chinese translations presenting a particular challenge (Section 5) – a limitation we were unable to address due to language barriers; (3) our single-entity selection strategy (Section 6.3) works well for the benchmark at hand but would likely struggle with more diverse texts containing multiple complex entities per sentence; (4) approximately 2.26% of cases with substantial entity transformations remain problematic (Section 6.1) – a percentage likely to increase in less curated texts; and (5) though our parameter-efficient approach comes within 0.62 percentage points of LLM-based alternatives (Section 6.6), more sophisticated LLM implementations could potentially widen this gap.

Acknowledgments

The authors gratefully acknowledge the HPC resources provided by the JuliaV2 cluster at the Universität Würzburg (JMU), which was funded as DFG project as “Forschungsgroßgerät nach Art 91b GG” under INST 93/1145-1 FUGG. The project staff is partially funded by the DFG – 529659926. The data science chair is part of the CAIDAS, the Center for Artificial Intelligence and Data Science, and is supported by the Bavarian High-Tech Agenda, which made this research possible.

References

- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. [Language Models are Few-Shot Learners](#).
- Niccolò Campolungo, Tommaso Pasini, Denis Emelin, and Roberto Navigli. 2022. [Reducing disambiguation biases in NMT by leveraging explicit word sense information](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4824–4838, Seattle, United States. Association for Computational Linguistics.
- Yixin Cao, Lei Hou, Juanzi Li, and Zhiyuan Liu. 2018. [Neural collective entity linking](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 675–686, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Simone Conia, Daniel Lee, Min Li, Umar Farooq Minhas, Saloni Potdar, and Yunyao Li. 2024. [Towards Cross-Cultural Machine Translation with Retrieval-Augmented Generation from Multilingual Knowledge Graphs](#). *Preprint*, arXiv:2410.14057.
- Simone Conia, Min Li, Roberto Navigli, and Saloni Potdar. 2025. SemEval-2025 task 2: Entity-aware machine translation. In *Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025)*. Association for Computational Linguistics.
- Mar Díaz-Millón and María Dolores Olvera-Lobo. [Towards a definition of transcreation: A systematic literature review](#). 31(2):347–364.
- Viviana Gaballo et al. 2012. Exploring the boundaries of transcreation in specialized translation. *ESP across Cultures*, 9:95–113.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Alonsoius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Is-han Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, prefix=van der useprefix=false family=Linde, given=Jelmer, Jennifer Billelock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz

Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yearly, prefix=van der useprefix=false family=Maaten, given=Laurens, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, prefix=de useprefix=false family=Oliveira, given=Luke, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Celebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Rapparthi, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gouget, Virginie Do, Vish Vogeti, Vitor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyan Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuwei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yunying Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testugine, Delia David, Devi Parikh, Diana Liskovich,

Didem Foss, Dingkan Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabza, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangarabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim

- Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. [The Llama 3 Herd of Models](#). *Preprint*, arXiv:2407.21783.
- Liam Hebert, Raheleh Makki, Shubhanshu Mishra, Hamidreza Saghir, Anusha Kamath, and Yuval Merhav. [Robust Candidate Generation for Entity Linking on Short Social Media Texts](#). In *Proceedings of the Eighth Workshop on Noisy User-generated Text (W-NUT 2022)*, pages 83–89. Association for Computational Linguistics.
- Daniel Hershcovich, Stella Frank, Heather Lent, Miryam de Lhoneux, Mostafa Abdou, Stephanie Brandl, Emanuele Bugliarello, Laura Cabello Piqueras, Ilias Chalkidis, Ruixiang Cui, Constanza Fierro, Katerina Margatina, Phillip Rust, and Anders Søgaard. 2022. [Challenges and strategies in cross-cultural NLP](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6997–7013, Dublin, Ireland. Association for Computational Linguistics.
- Chris Hokamp and Qun Liu. 2017. [Lexically constrained decoding for sequence generation using grid beam search](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1535–1546, Vancouver, Canada. Association for Computational Linguistics.
- Xuming Hu, Junzhe Chen, Xiaochuan Li, Yufei Guo, Lijie Wen, Philip S. Yu, and Zhijiang Guo. 2024. [Towards understanding factual knowledge of large language models](#). In *The Twelfth International Conference on Learning Representations*.
- Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. [DSPy: Compiling Declarative Language Model Calls into Self-Improving Pipelines](#). *Preprint*, arXiv:2310.03714.
- Nikolaos Kolitsas, Octavian-Eugen Ganea, and Thomas Hofmann. 2018. [End-to-end neural entity linking](#). In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 519–529, Brussels, Belgium. Association for Computational Linguistics.
- Tuan Lai, Heng Ji, and ChengXiang Zhai. [Improving Candidate Retrieval with Entity Profile Generation for Wikidata Entity Linking](#). In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3696–3711. Association for Computational Linguistics.
- Angeliki Lazaridou, Adhiguna Kuncoro, Elena Gribovskaya, Devang Agrawal, Adam Liska, Tayfun Terzi, Mai Gimenez, Cyprien de Masson d’Autume, Tomas Kocisky, Sebastian Ruder, Dani Yogatama, Kris Cao, Susannah Young, and Phil Blunsom. 2021. [Mind the gap: Assessing temporal generalization in neural language models](#). *Preprint*, arXiv:2102.01951.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. [Retrieval-augmented generation for knowledge-intensive nlp tasks](#). *Preprint*, arXiv:2005.11401.
- Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. [Truthfulqa: Measuring how models mimic human falsehoods](#). *Preprint*, arXiv:2109.07958.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [Roberta: A robustly optimized bert pretraining approach](#). *Preprint*, arXiv:1907.11692.
- Ilya Loshchilov and Frank Hutter. 2019. [Decoupled weight decay regularization](#). *Preprint*, arXiv:1711.05101.
- Frederic P. Miller, Agnes F. Vandome, and John McBrewster. 2009. *Levenshtein Distance: Information theory, Computer science, String (computer science), String metric, Damerau-Levenshtein distance, Spell checker, Hamming distance*. Alpha Press.
- Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. [COMET: A Neural Framework for MT Evaluation](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702. Association for Computational Linguistics.
- Abigail See, Peter J. Liu, and Christopher D. Manning. [Get To The Point: Summarization with Pointer-Generator Networks](#). *Preprint*, arXiv:1704.04368.
- Priyanka Sen, Alham Fikri Aji, and Amir Saffari. [Mintaka: A Complex, Natural, and Multilingual Dataset for End-to-End Question Answering](#). In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1604–1619. International Committee on Computational Linguistics.
- Wei Shen, Jianyong Wang, and Jiawei Han. 2015. [Entity linking with a knowledge base: Issues, techniques, and solutions](#). *IEEE Transactions on Knowledge and Data Engineering*, 27(2):443–460.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. [Multilingual translation with extensible multilingual pretraining and finetuning](#). *Preprint*, arXiv:2008.00401.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Hefernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barraud, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. [No language left behind: Scaling human-centered machine translation](#). *Preprint*, arXiv:2207.04672.

Yuxia Wang, Minghan Wang, Muhammad Arslan Manzoor, Fei Liu, Georgi Georgiev, Rocktim Jyoti Das, and Preslav Nakov. 2024. [Factuality of large language models: A survey](#). *Preprint*, arXiv:2402.02420.

BigScience Workshop, :, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Lucioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, Dragomir Radev, Eduardo González Ponferrada, Efrat Levkovizh, Ethan Kim, Eyal Bar Natan, Francesco De Toni, Gérard Dupont, Germán Kruszewski, Giada Pistilli, Hady Elsahar, Hamza Benyamina, Hieu Tran, Ian Yu, Idris Abdulmumin, Isaac Johnson, Itziar Gonzalez-Dios, Javier de la Rosa, Jenny Chim, Jesse Dodge, Jian Zhu, Jonathan Chang, Jörg Froberg, Joseph Tobing, Joydeep Bhattacharjee, Khalid Almubarak, Kimbo Chen, Kyle Lo, Leandro Von Werra, Leon Weber, Long Phan, Loubna Ben allal, Ludovic Tanguy, Manan Dey, Manuel Romero Muñoz, Maraim Masoud, María Grandury, Mario Šaško, Max Huang, Maximin Coavoux, Mayank Singh, Mike Tian-Jian Jiang, Minh Chien Vu, Mohammad A. Jauhar, Mustafa Ghaleb, Nishant Subramani, Nora Kassner, Nurulaqilla Khamis, Olivier Nguyen, Omar Espejel, Ona de Gibert, Paulo Villegas, Peter Henderson, Pierre Colombo, Priscilla Amuok,

Quentin Lhoest, Rheza Harliman, Rishi Bommasani, Roberto Luis López, Rui Ribeiro, Salomey Osei, Sampo Pyysalo, Sebastian Nagel, Shamik Bose, Shamsuddeen Hassan Muhammad, Shanya Sharma, Shayne Longpre, Somaieh Nikpoor, Stanislav Silberberg, Suhas Pai, Sydney Zink, Tiago Timponi Torrent, Timo Schick, Tristan Thrush, Valentin Danchev, Vassilina Nikoulina, Veronika Laippala, Violette Lepercq, Vrinda Prabhu, Zaid Alyafeai, Zeerak Talat, Arun Raja, Benjamin Heinzerling, Chenglei Si, Davut Emre Taşar, Elizabeth Salesky, Sabrina J. Mielke, Wilson Y. Lee, Abheesht Sharma, Andrea Santilli, Antoine Chaffin, Arnaud Stiegler, Debajyoti Datta, Eliza Szczechla, Gunjan Chhablani, Han Wang, Harshit Pandey, Hendrik Strobelt, Jason Alan Fries, Jos Rozen, Leo Gao, Lintang Sutawika, M Saiful Bari, Maged S. Al-shaibani, Matteo Manica, Nihal Nayak, Ryan Teehan, Samuel Albanie, Sheng Shen, Srulik Ben-David, Stephen H. Bach, Taewoon Kim, Tali Bers, Thibault Fevry, Trishala Neeraj, Urmish Thakker, Vikas Raunak, Xiangru Tang, Zheng-Xin Yong, Zhiqing Sun, Shaked Brody, Yallow Uri, Hadar Tojarieh, Adam Roberts, Hyung Won Chung, Jaesung Tae, Jason Phang, Ofir Press, Conglong Li, Deepak Narayanan, Hatim Bourfoune, Jared Casper, Jeff Rasley, Max Ryabinin, Mayank Mishra, Minjia Zhang, Mohammad Shoeybi, Myriam Peyrounette, Nicolas Patry, Nouamane Tazi, Omar Sanseviero, Patrick von Platen, Pierre Cornette, Pierre François Lavallée, Rémi Lacroix, Samyam Rajbhandari, Sanchit Gandhi, Shaden Smith, Stéphane Requena, Suraj Patil, Tim Dettmers, Ahmed Baruwaa, Amanpreet Singh, Anastasia Cheveleva, Anne-Laure Ligozat, Arjun Subramonian, Aurélie Névél, Charles Lovering, Dan Garrette, Deepak Tunuguntla, Ehud Reiter, Ekaterina Taktasheva, Ekaterina Voloshina, Eli Bogdanov, Genta Indra Winata, Hailey Schoelkopf, Jan-Christoph Kalo, Jekaterina Novikova, Jessica Zosa Forde, Jordan Clive, Jungo Kasai, Ken Kawamura, Liam Hazan, Marine Carpuat, Miruna Clinciu, Na-young Kim, Newton Cheng, Oleg Serikov, Omer Antverg, Oskar van der Wal, Rui Zhang, Ruochen Zhang, Sebastian Gehrmann, Shachar Mirkin, Shani Pais, Tatiana Shavrina, Thomas Scialom, Tian Yun, Tomasz Limisiewicz, Verena Rieser, Vitaly Protasov, Vladislav Mikhailov, Yada Pruksachatkun, Yonatan Belinkov, Zachary Bamberger, Zdeněk Kasner, Alice Rueda, Amanda Pestana, Amir Feizpour, Ammar Khan, Amy Faranak, Ana Santos, Anthony Hevia, Antigona Urdreaj, Arash Aghagol, Arezoo Abdollahi, Aycha Tammour, Azadeh HajiHosseini, Bahareh Behroozi, Benjamin Ajibade, Bharat Saxena, Carlos Muñoz Ferrandis, Daniel McDuff, Danish Contractor, David Lansky, Davis David, Douwe Kiela, Duong A. Nguyen, Edward Tan, Emi Baylor, Ezinwanne Ozoani, Fatima Mirza, Frankline Onon-iwu, Habib Rezanejad, Hessie Jones, Indrani Bhattacharya, Irene Solaiman, Irina Sedenko, Isar Nadjdholi, Jesse Passmore, Josh Seltzer, Julio Bonis Sanz, Livia Dutra, Mairon Samagaio, Maraim Elbadri, Margot Mieskes, Marissa Gerchick, Martha Akinlolu, Michael McKenna, Mike Qiu, Muhammed Ghauri, Mykola Burynok, Nafis Abrar, Nazneen Rajani, Nour Elkott, Nour Fahmy, Olanrewaju Samuel,

Ran An, Rasmus Kromann, Ryan Hao, Samira Alizadeh, Sarmad Shubber, Silas Wang, Sourav Roy, Sylvain Viguier, Thanh Le, Tobi Oyeade, Trieu Le, Yoyo Yang, Zach Nguyen, Abhinav Ramesh Kashyap, Alfredo Palasciano, Alison Callahan, Anima Shukla, Antonio Miranda-Escalada, Ayush Singh, Benjamin Beilharz, Bo Wang, Caio Brito, Chenxi Zhou, Chirag Jain, Chuxin Xu, Cl  mentine Fourrier, Daniel Le  n Perin  n, Daniel Molano, Dian Yu, Enrique Manjavacas, Fabio Barth, Florian Fuhrmann, Gabriel Altay, Giyaseddin Bayrak, Gully Burns, Helena U. Vrabec, Imane Bello, Ishani Dash, Jihyun Kang, John Giorgi, Jonas Golde, Jose David Posada, Karthik Rangasai Sivaraman, Lokesh Bulchandani, Lu Liu, Luisa Shinzato, Madeleine Hahn de Bykhovetz, Maiko Takeuchi, Marc P  mies, Maria A Castillo, Marianna Nezhurina, Mario S  nger, Matthias Samwald, Michael Cullan, Michael Weinberg, Michiel De Wolf, Mina Mihaljcic, Minna Liu, Moritz Freidank, Myungsun Kang, Natasha Seelam, Nathan Dahlberg, Nicholas Michio Broad, Nikolaus Muellner, Pascale Fung, Patrick Haller, Ramya Chandrasekhar, Renata Eisenberg, Robert Martin, Rodrigo Canalli, Rosaline Su, Ruisi Su, Samuel Cahyawijaya, Samuele Garda, Shlok S Deshmukh, Shubhanshu Mishra, Sid Kiblawi, Simon Ott, Since Sang-aaroonsiri, Srishti Kumar, Stefan Schweter, Sushil Bharati, Tanmay Laud, Th  o Gigant, Tomoya Kainuma, Wojciech Kusa, Yannis Labrak, Yash Shailesh Bajaj, Yash Venkatraman, Yifan Xu, Yingxin Xu, Yu Xu, Zhe Tan, Zhongli Xie, Zifan Ye, Mathilde Bras, Younes Belkada, and Thomas Wolf. 2023. [Bloom: A 176b-parameter open-access multilingual language model](#). *Preprint*, arXiv:2211.05100.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2020. [Scalable zero-shot entity linking with dense entity retrieval](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6397–6407, Online. Association for Computational Linguistics.

Zixin Zeng, Rui Wang, Yichong Leng, Junliang Guo, Shufang Xie, Xu Tan, Tao Qin, and Tie-Yan Liu. [Extract and Attend: Improving Entity Translation in Neural Machine Translation](#). In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1697–1710. Association for Computational Linguistics.

Jingyi Zhang, Masao Utiyama, Eiichiro Sumita, Graham Neubig, and Satoshi Nakamura. 2018. [Guiding Neural Machine Translation with Retrieved Translation Pieces](#). 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 1325–1335. Association for Computational Linguistics.

Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. 2024. [Multilingual machine translation with large language models: Empirical results and analysis](#). In *Findings of the Association for Computa-*

tional Linguistics: NAACL 2024, pages 2765–2781, Mexico City, Mexico. Association for Computational Linguistics.

A Training

All experiments were conducted on NVIDIA L40 GPUs. A complete training and evaluation run took approximately 3 hours, with a significant portion dedicated to evaluation, including the computation of COMET scores.

We used the facebook/nllb-200-distilled-600M⁸ model (Team et al., 2022) as our base architecture. The hyperparameters used for the subsequent fine-tuning are listed in Table A1. For a complete list of hyperparameters, we refer to our configuration file at [src/conf/translation_config.yaml](#).

Parameter	Value
Number of epochs	10
Batch size	16
Gradient accumulation steps	4
Effective batch size	64
Optimizer	AdamW
Learning rate	1e-5
Loss function	Cross-entropy
Max sequence length (In & Out)	512
Precision	bfloat16
Logit bias parameter (b)	5.0
Beam search	5 beams

Table A1: Training hyperparameters used in our experiments.

B Transformer Reranker Details

For our transformer-based reranking approach, we utilized the Roberta (Liu et al., 2019) based pre-trained jina-reranker-v2-base-multilingual⁹ cross-encoder model that takes a query-document pair as input and produces a relevance score. The model operates as follows:

Given a source text x and a candidate entity e_i with associated metadata (including name, description, and available translations), we represent the relevance score as:

$$s(e_i, x) = f_{\theta}(x, r(e_i))$$

where f_{θ} is our transformer model with parameters θ , and $r(e_i)$ is a textual representation of the entity that concatenates its title, description, and

⁸<https://huggingface.co/facebook/nllb-200-distilled-600M>

⁹<https://huggingface.co/jinaai/jina-reranker-v2-base-multilingual>

translation. We fine-tune the model using margin ranking loss:

$$\mathcal{L} = \max(0, s(e^-, x) - s(e^+, x) + \gamma)$$

where e^+ is the correct entity, e^- is an incorrect entity, and γ is the margin (set to 0.3).

The model is fine-tuned using a learning rate of $2e-5$ with the AdamW (Loshchilov and Hutter, 2019) optimizer. Training samples are created by using the correct entity as the positive example and sampling hard negatives from the top 5 results of our SQL-based retrieval. Training converged rapidly, requiring less than one epoch on a quarter of the development set to achieve optimal performance.

C LLM prompt

In all LLM interactions, the DsPy¹⁰ framework (Khattab et al.) is used to automatically parse the input and output of the LLM with the prompts being defined via “Signatures” as outlined

C.1 Reranking prompt

The following prompt is provided to the LLM to rerank the retrieved entities.

Retrieve all distinct entity candidates from a provided context that might be relevant for disambiguation in a machine-translation task.

Requirements:

1. High Recall:
 - Include every candidate that could be the correct reference, knowing that the correct one is almost always among the list.
2. Translation Quality:
 - Do not add candidates with ambiguous translations; if unsure, include them and let later stages decide.
3. Handle Ambiguity:
 - When entities share names, include all with potential relevance based on their descriptions.
4. Ranking:
 - Return a sorted list of candidate identifiers prioritizing:
 - * Contextual clues from the input sample,
 - * Popularity and provided score,
 - * Clear descriptive evidence matching the candidate’s role in the sentence.

Objective:

Ensure that the correct candidate is positioned at the top of the candidate list.

Input Fields:

- context (str):

Original input sample that provides the context for disambiguation.

- candidates (list of EntityCandidate):

A list of candidate entities, each with detailed metadata obtained via fuzzy matching.

Output Field:

- selected_candidates (list of str):

Disambiguated list of relevant candidate identifiers (e.g., wikidata_ids), sorted by contextual relevance.

C.2 Self Translation prompt

The following prompt is provided to the LLM to generate the translations by itself, only provided with the input sentence, target language and entity candidates.

Generate a high-quality translation by accurately rendering named entities from candidate data.

Given only the original source sentence, the model should generate the translation on its own, while using the candidate entity information—each with a high likelihood of containing the correct translation—to incorporate the appropriate named entities. Not every provided candidate is relevant, so selectively apply those that enhance the contextual accuracy of the translation.

Input Fields:

- source_sentence (str):

The original input sentence in English that requires translation into the target language.

- target_locale (str):

The target language locale, e.g. 'de' for German.

- candidates (list of EntityCandidate):

A list of candidate entities with their potential translations and associated metadata. Each candidate has a high likelihood of being correct, but not every candidate is necessarily relevant.

Output Field:

- final_translation (str):

The final translation generated by the model, with accurately rendered named entities based on the candidate evidence.

C.3 Vanilla NLLB Translation Refiner prompt

The following prompt is provided to the LLM to refine translations from the vanilla NLLB model.

¹⁰<https://github.com/stanfordnlp/dspy>

Refine a vanilla NLLB translation by selectively incorporating named entities from candidate data.

The vanilla translation produced by the NLLB model might contain errors or omissions in the rendering of named entities.

Utilize the provided candidate entity information—each holding a high probability of correctness—to adjust the translation, ensuring accurate rendering of those entities while recognizing that not all candidates are relevant to the context.

Input Fields:

- `nllb_translation (str)`:

The initial translation produced by the vanilla NLLB model, which may contain omissions or errors in the depiction of named entities.

- `target_locale (str)`:

The target language locale, e.g. 'de' for German.

- `candidates (list of EntityCandidate)`:

A list of candidate entities with their potential translations and related metadata. Although these candidates are highly likely to include the correct entity translations, not every candidate may be applicable in the specific context.

Output Field:

- `final_translation (str)`:

The final translation where the named entities have been refined to accurately align with the most relevant candidate data.

German.

- `candidates (list of EntityCandidate)`:

A list of candidate entities with their expected translations and metadata.

These candidate translations are considered correct and should be applied to fix or supplement the named entity translations.

Output Field:

- `refined_translation (str)`:

The final translation where named entity translations are corrected to match the gold standard candidate information.

C.4 Finetuned Translation Refiner prompt

The following prompt is provided to the LLM to refine translations from the finetuned NLLB model.

Refine a finetuned translation by ensuring that all named entity translations align with the candidate data, which is considered correct.

Occasionally, the finetuned nllb model may apply a completely wrong candidate or miss additional relevant candidates for named entities.

Use the provided candidate details, whose translations are considered the gold standard, to fix any wrong entity translations and to add any missing ones. Only adjust the named entity expressions, preserving the overall translation quality.

Input Fields:

- `finetuned_translation (str)`:

The high-quality translation from the finetuned nllb model, which may contain errors in named entity translations.

- `target_locale (str)`:

The target language locale, e.g. 'de' for