Does Synthetic Data Help Named Entity Recognition for Low-Resource Languages?

Gaurav Kamath[*1, 2] **Sowmya Vajjala**[3]

[1]McGill University, Canada [2]Mila - Quebec AI Institute, Canada
[3]National Research Council, Canada

gaurav.kamath@mail.mcgill.ca sowmya.vajjala@nrc-cnrc.gc.ca

Abstract

We explore whether synthetic datasets generated by large language models using a few high quality seed samples are useful for low-resource named entity recognition, considering 11 languages from three language families. Our results suggest that synthetic data created with such seed data is a reasonable choice when there is no available labeled data, and is better than using entirely automatically labeled data. However, a small amount of high-quality data, coupled with cross-lingual transfer from a related language, always offers better performance.[1]

1 Introduction

Named Entity Recognition (NER) for low-resource languages aims to produce robust systems for languages with limited labeled training data available, and has been an area of increasing interest within natural language processing (NLP) over the past decade. Two common approaches to address this data scarcity are cross-lingual transfer and data augmentation/synthesis; recent research has in particular explored the usefulness of large language models (LLMs) for such data augmentation and synthetic data creation in NLP (Whitehouse et al., 2023; Li et al., 2023), while their use for NER is also emerging (Bogdanov et al., 2024; Dao et al., 2025).

In this background, we propose LLM-based synthetic data generation using a small amount of gold examples (Figure 1) as an alternative to relying on automatically created datasets for low-resource NER. With experiments covering 11 languages from 3 language families—Danish, Swedish and Slovak from the Indo-European language family; Swahili, Kinyarwanda, Yoruba and Igbo from the Niger-Congo language family, and Kannada, Malayalam, Tamil, and Telugu from the Dravidian language family, we show that:

1. Even a small amount of human annotated data can yield far better performance than much larger amounts of synthetic data.
2. Zero-shot transfer from a related language can provide high baselines for low-resource language NER.
3. Synthetic data generated by prompting an LLM with a few high quality (generally human labeled) examples (Figure 1) could

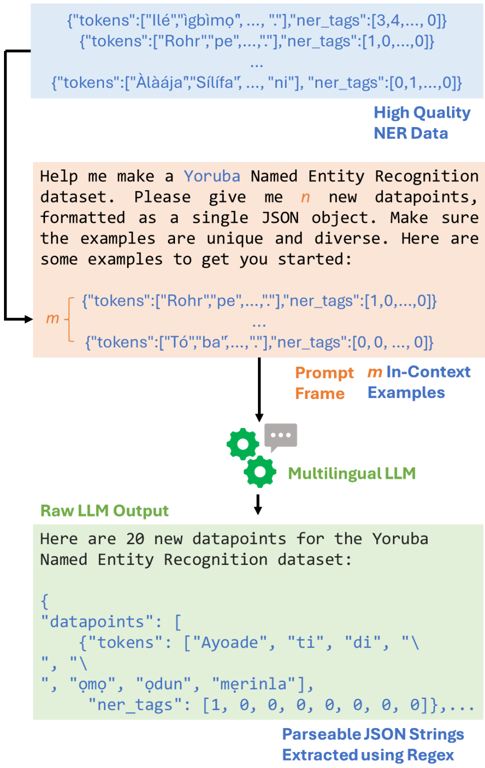


Figure 1: High-level overview of our data generation process. We use multilingual large language models to generate new NER data points on the basis of a handful of high quality human labeled data points. See Section 3.1 for more.

[*]Work done during an internship at the National Research Council, Canada.

[1]Data and code available at: https://github.com/grvkamath/low-resource-syn-ner.

Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 159–167
December 20-24, 2025

be better than using automatically labeled datasets when training low-resource NER models.

We start with a review of related literature (Section 2) and describe our data generation approach and experimental setup in Section 3, followed by a discussion of the results (Section 4), limitations (Section 6) and broader impact (Section 7).

2 Related Work

NER in low resource settings has long been a topic of interest in NLP. Significant research examines cross-lingual transfer from a high resource source language to a lower-resource target language for the task (Rahimi et al., 2019; Mueller et al., 2020; Zeng et al., 2022; Zhao et al., 2022; Yang et al., 2022; Zhou et al., 2022), while other approaches have explored the creation of synthetic datasets through e.g. parallel corpora or machine translation (Mayhew et al., 2017; Ni et al., 2017; Pan et al., 2017; Xie et al., 2018; Liu et al., 2021; Yang et al., 2022; Fetahu et al., 2022). There are also large existing automatically constructed multilingual NER datasets that rely on sources such as Wikipedia (Pan et al., 2017; Krishnan et al., 2021; Malmasi et al., 2022), some of which have become a part of large multilingual benchmarks (Asai et al., 2024).

More recent work has explored using LLMs as data generators for NER (Bogdanov et al., 2024; Heng et al., 2024; Evuru et al., 2024). We build on such work, but differ from their methods. Our data generation process uses high quality, human validated examples as seeds, and we not only evaluate different LLMs (both open and closed-source) as synthetic data generators, but also experiment with 11 languages covering three language families and five base scripts. To our knowledge, this is the first attempt to explore using large language models for synthetic data generation in low-resource NER, and the first to cover > 10 languages.

3 Our Approach

At a high level, our approach involves two steps:

1. Using the train split of a high quality (usually manually annotated) NER dataset for a target language to generate synthetic data for that language with the help of an LLM (Section 3.1); and then

2. Comparing the performance of an NER model on the test split of the high quality dataset when trained on synthetic data from Step 1 and another model trained on the train split of the same high quality dataset (Section 3.2).

3.1 Synthetic Data Generation:

Our synthetic data generation process (shown in Figure 1) involves using LLMs to generate new synthetic data points on the basis of existing, high quality NER annotations as described below:

- First, we randomly sample m data points from the train split of an organic (i.e. non-synthetic) NER dataset.

- Next, we format and append these data points to a prompt asking the model to produce n new, unique data points on the basis of the m data points in the prompt.

- We submit this prompt as input to the LLM, and extract the correctly-formatted data points from its response;

- We repeat steps (1)-(3) k times, with each call to the model choosing a different random sample of organic data points.

In our experiments, we set m to 10, n to 20, and k to 250. This sets an upper cap of 5000 synthetic training data points, if every model response contains perfectly formatted data points. We present and solicit data structured as JSON strings to the LLMs, and extract well-formatted samples from model responses using regular expressions. Appendix A provides further details about this process.

We compare three LLMs as our source of synthetic data: `GPT-4.1`[2] (OpenAI, 2025), which we assume to be the state of the art; `Llama-3.1-8B-Instruct` (Dubey et al., 2024), as a much smaller, open-source instruction-tuned model; and finally, `aya-expanse-32b` (Dang et al., 2024), as a larger open source multilingual LLM.

3.2 Training NER models:

For all experiments, we use the pre-trained version of `XLM-RoBERTa-large` (Conneau et al., 2020) as our base model and fine-tune it on our synthetic and organic training sets in two distinct settings.

[2]We use `gpt-4.1-2025-04-14`. Note that in an earlier draft of this work, we used `gpt-4-turbo` (Achiam et al., 2023), when it represented the state-of-the-art; surprisingly, `gpt-4-turbo` yielded slightly better results. Nevertheless, here we report results on `GPT-4.1`, to better represent currently available models.

LLM Response Quality	Examples
Well-Formatted	{"data": [{ "tokens": ["Lars", "Løkke", "Rasmussen", "besøgte", "firmaet", "i", "Odense", "."], "ner_tags": [1, 2, 2, 0, 0, 0, 5, 0]},...
Unequal Token & Tag Lengths	{"id": "4123", "tokens": ["Wananchi", "wamekunja", "mashitaka", "."], "ner_tags": [0, 0, 0, 0, 0, 0]}
Run-On & Incomplete Data	{"id": "9000", "tokens": ["Olorun", "lẹ̀ẹ̀", ..., "ò", "ò", "ò", "ò", "ò", "ò", "ò", "ò", "ò", "ò", "ò", "ò", "ò" {"id":"4617","tokens":["Ọ́ḍǹ", "ọ̀sè", "lè", "ọ̀rị́", "-", "èdè", "Ọ́ḅf́emi", "àfú", "fùn", "àwọn", "gb", "."] "ner_tags":[8,0,0,0,0
Empty Responses & Prompt Continuations	<EOS_TOKEN> <EOS_TOKEN>include a mix of names, locations, organizations...

Table 1: Examples of different types of responses from the synthetic data-generating LLMs tested, across languages

1. In the first setting, we use our synthetic data to train an NER model from scratch, by fine-tuning XLM-RoBERTa-large on target language NER data.

2. In the second setting, we first fine-tune the model on the high quality NER data in a *related* source language[3], and then further fine-tune this NER model using our synthetic or organic target language data.

While the first setting—which we name NER FROM SCRATCH—aims to shed light on the relative utility of synthetic data for training an NER model (largely) from the ground up, the latter —which we name NER FINE-TUNING—simulates a common setting, when a lower resource language lacks adequate NER data, but is related to a higher-resource language with existing NER systems. In both settings, we modulate the amount of data (both synthetic and organic) used, so as to compare model performance when trained on smaller or larger amounts of each type of data.

Languages & Datasets: We focus on 11 languages from three distinct language families: Tamil, Kannada, Malayalam, Telugu (Dravidian), Kinyarwanda, Swahili, Igbo, Yoruba (Niger-Congo), Swedish, Danish and Slovak (Indo-European). Of these, Igbo, Yoruba, and Kinyarwanda are not among the 100 languages in the XLM-Roberta pre-training corpus. We use the Universal NER dataset (Mayhew et al., 2024) as our high quality, manually annotated dataset for Swedish, Danish and Slovak; MasakhaNER2 (Adelani et al., 2022) for Kinyarwanda, Swahili, Igbo and Yoruba; and the Naamapadam dataset (Mhaske et al., 2023) for Tamil, Kannada, Malayalam and Telugu.

While the first two datasets are completely manually annotated, the train and validation splits of the Naamapadam dataset are constructed using parallel corpora, and thus contain some noise. Nevertheless, we choose it as our organic dataset, as (i) its test sets, which contain 500-1000 datapoints per language, are completely manually annotated, and (ii) it remains the largest NER resource for these four languages. Crucially, all of these datasets cover largely identical NER categories, allowing for comparisons between them. The Universal NER and Naamapadam datasets cover persons, locations and organizations as categories; the MasakhaNER2 data covers these three categories, as well as dates.

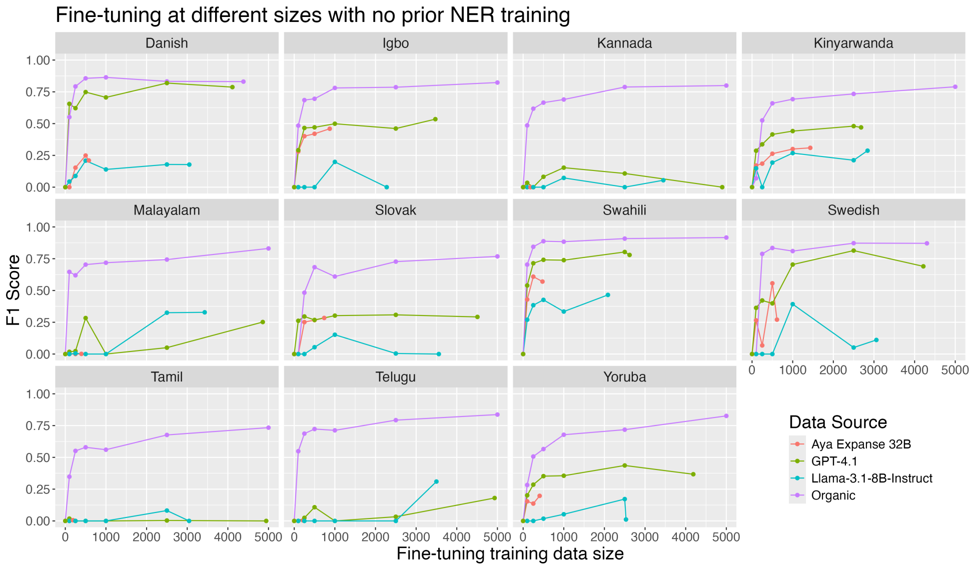
Additionally, we compare models trained entirely on LLM-generated data with those trained using WikiANN (Pan et al., 2017; Rahimi et al., 2019), a large, automatically created NER dataset based on Wikipedia cross-linking, as it covers the 11 languages we study. This dataset represents a different form of synthetic data—one generated not from LLMs, but instead from scraping knowledge bases without any seed data. Although the dataset has no manual annotations, it is frequently used as a standard low-resource NER benchmark (Schmidt et al., 2022; Asai et al., 2024).

4 Results

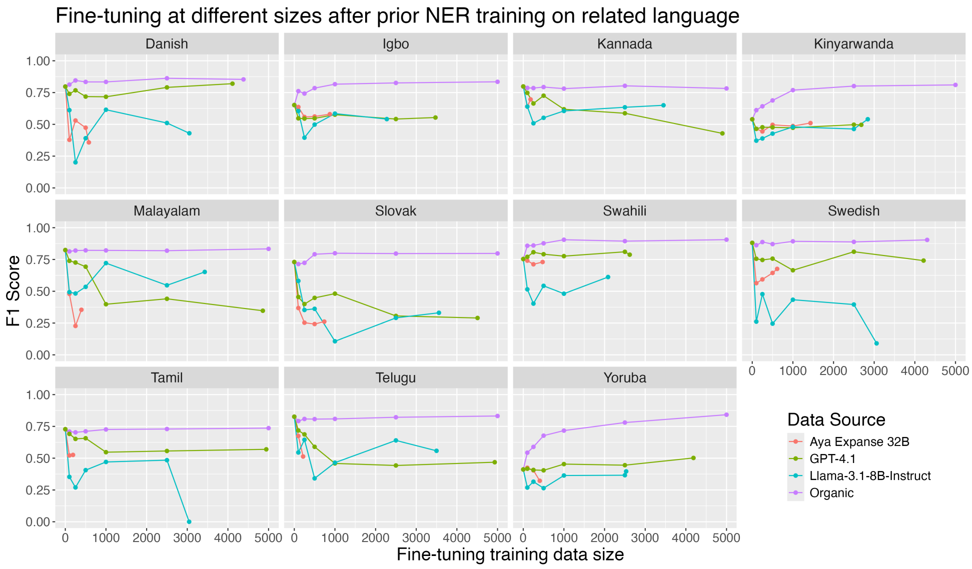
4.1 Synthetic Data Generation

We generate the synthetic datasets following the process described in Section 3.1. While model responses from GPT-4.1 are generally usable, we found more recurring errors in responses from the other two models. Some of these errors are described in Table 1; we discard such instances when compiling our synthetic datasets from model responses. The average percentage of usable training datapoints from GPT-4.1, Llama-3.1 and aya-expanse are 82.6%, 59.7% and 11.1% respectively. We assess the overall quality and viability of this synthetic data by measuring the performance of an NER model on a high quality, manually-annotated test set, when trained on the synthetic

[3]See Table 2 in Appendix B for the full list of chosen related languages for all the target languages.



(a) NER FROM SCRATCH Setting



(b) NER FINE-TUNING Setting

Figure 2: NER model performance when trained on increasingly large subsets of training data. `aya-expanse-32b` and `Llama-3.1-8B-Instruct` produced lower amounts of usable data; this is why they do often not extend as far as organic or `GPT-4.1`-produced data in fine-tuning data size. In the NER FINE-TUNING setting, performance at `Fine-tuning Training Data Size = 0` indicates zero-shot performance of a related-language NER model.

data.

4.2 Training on Synthetic Data

Figure 2 shows our results when using synthetic data from different models, in both the NER FROM SCRATCH and NER FINE-TUNING settings. While the models trained on organic data in the NER FROM SCRATCH setting always perform better than synthetic data based models, we find that the models trained on `GPT-4.1`-generated data often come the closest to models trained on organic data compared to the other synthetic data sources. Best results with synthetic data based training are seen for Danish, followed by Swahili and Swedish. We also find that more synthetic data is not necessarily useful; for most languages, we see relative saturation after 1000 data points, and in the case of Kannada, we actually notice a drop in performance with more data.

Perhaps more surprisingly, in the NER FINE-TUNING setting, we notice that zero-shot transfer from a related language usually outperforms the same models after they have been further fine-tuned on synthetic target language data. Fine-tuning the related language NER model with organic data from the target language helped for only Kinyarwanda and Yoruba. This suggests that in some cases where an NER model for a related language exists, synthetic data in target languages may actually be detrimental to overall performance. Models trained on synthetic data from `GPT-4.1` do better than those trained on synthetic data from `Llama-3.1-8B-Instruct` only about half of the time; on the other hand, there are often too few usable `aya-expanse-32b` datapoints for a fair comparison.

Comparison with WikiAnn: In addition to comparing models fine-tuned on synthetic data versus organic data, we also looked into the question of whether our synthetic data generation approach offers any benefits over automatically labeled datasets, taking WikiAnn as an example. Training models on WikiANN data leads to higher performance than training on `GPT-4.1`-generated data only for the four Dravidian languages in our data, but generally leads to significantly worse performance than training on synthetic data from `GPT-4.1` for the remaining languages (see Table 3 in Appendix C for detailed results). This holds in both the NER FROM SCRATCH and NER FINE-TUNING settings, when data size is comparable; and, in the case of Danish and Swedish, training on WikiANN leads to worse performance even when there is several times more WikiANN data than `GPT-4.1`-generated data. Overall, we can conclude that using synthetic data following our approach appears to be better than relying on WikiAnn for most languages. This echoes the findings by Lignos et al. (2022), who arrive at similarly negative findings around the data quality of WikiANN, and calls for not considering results on WikiANN as a benchmark for multilingual NER comparisons in the future.

5 Conclusions and Discussion

Our results lead us to three main conclusions around the utility of LLM-generated synthetic data for low resource language NER.

1. A small amount of carefully annotated data yields better performance than a large amount of synthetic data. As is evident in Figure 2, even 100 manually annotated data points can yield NER models that cannot be matched by models trained on much larger amounts of synthetic data.

2. In many cases, zero-shot transfer from a related-language NER model is a high baseline, and that further training such a model on synthetic data may even lower the performance. We find this to be true in the case of all languages tested except the Yoruba and Swahili. For these two languages, it is worth noting that the overall baselines are lower, presumably because these languages are all lower resource than the others tested. This may explain why synthetic data yields performance gains over the zero-shot baseline, though it does not change the trend of a small amount of manually annotated data yielding far better performance.

3. Despite the fact that it falls short of manually annotated data, LLM-generated data often still yields better model performance than WikiANN, which is automatically extracted from Wikipedia texts.

Overall, while showing how synthetic data from LLMs can help train NER models from scratch for low resource languages, our results reinforce the need for manually annotated gold test sets in benchmarking NER for lower resource languages.

6 Limitations

Although we experimented with many languages, the nature of the NER datasets used is relatively simple, containing only three or four entity categories (persons, locations, organizations and dates). Thus, we don't know if the general conclusions, especially about the quality of synthetic data, will extend to scenarios where there are many entity categories. While we did study datasets covering more than one language family, the selection of language is far from extensive, and is also constrained by the availability of human labeled test data. The observations need not necessarily hold true across all language families, naturally. Finally, to keep the experiments under control, we explored a limited set of methods for fine-tuning and synthetic data generation. Our findings should be viewed after taking these aspects into consideration.

7 Ethics and Broader Impact

We used publicly available datasets with human-annotated and automatically labeled data, and also created synthetically generated datasets as a part of this work. The models built using such artificially created datasets should always be validated with a human-labeled data. We did not involve any human participants in this study. All the code and generated datasets is provided at this GitHub repository to support reproducible research: `https://github.com/grvkamath/low-resource-syn-ner`.

Acknowledgments

This research was conducted at the National Research Council of Canada, thereby establishing a copyright belonging to the Crown in Right of Canada, that is, to the Government of Canada. Gaurav Kamath is supported by a Doctoral Training Award from the *Fonds de Recherche du Québec—Société et Culture*.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

David Adelani, Graham Neubig, Sebastian Ruder, Shruti Rijhwani, Michael Beukman, Chester Palen-Michel, Constantine Lignos, Jesujoba Alabi, Shamsuddeen Muhammad, Peter Nabende, et al. 2022. Masakhaner 2.0: Africa-centric transfer learning for named entity recognition. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4488–4508.

Akari Asai, Sneha Kudugunta, Xinyan Yu, Terra Blevins, Hila Gonen, Machel Reid, Yulia Tsvetkov, Sebastian Ruder, and Hannaneh Hajishirzi. 2024. BUFFET: Benchmarking large language models for few-shot cross-lingual transfer. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1771–1800, Mexico City, Mexico. Association for Computational Linguistics.

Sergei Bogdanov, Alexandre Constantin, Timothée Bernard, Benoit Crabbé, and Etienne Bernard. 2024. Nuner: Entity recognition encoder pre-training via llm-annotated data. *arXiv e-prints*, pages arXiv–2402.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

John Dang, Shivalika Singh, Daniel D'souza, Arash Ahmadian, Alejandro Salamanca, Madeline Smith, Aidan Peppin, Sungjin Hong, Manoj Govindassamy, Terrence Zhao, et al. 2024. Aya expanse: Combining research breakthroughs for a new multilingual frontier. *arXiv preprint arXiv:2412.04261*.

An Dao, Hiroki Teranishi, Yuji Matsumoto, Florian Boudin, and Akiko Aizawa. 2025. Overcoming data scarcity in named entity recognition: Synthetic data generation with large language models. In *Proceedings of the 24th Workshop on Biomedical Language Processing*, pages 328–340, Viena, Austria. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Chandra Kiran Evuru, Sreyan Ghosh, Sonal Kumar, Ramaneswaran S, Utkarsh Tyagi, and Dinesh Manocha. 2024. CoDa: Constrained generation based data augmentation for low-resource NLP. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3754–3769, Mexico City, Mexico. Association for Computational Linguistics.

Besnik Fetahu, Anjie Fang, Oleg Rokhlenko, and Shervin Malmasi. 2022. Dynamic gazetteer integration in multilingual models for cross-lingual and cross-domain named entity recognition. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2777–2790, Seattle, United States. Association for Computational Linguistics.

Yuzhao Heng, Chunyuan Deng, Yitong Li, Yue Yu, Yinghao Li, Rongzhi Zhang, and Chao Zhang. 2024. ProgGen: Generating named entity recognition datasets step-by-step with self-reflexive large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 15992–16030, Bangkok, Thailand. Association for Computational Linguistics.

Aravind Krishnan, Stefan Ziehe, Franziska Pannach, and Caroline Sporleder. 2021. Employing wikipedia as a resource for named entity recognition in morphologically complex under-resourced languages. In *Proceedings of the 14th Workshop on Building and Using Comparable Corpora (BUCC 2021)*, pages 28–39.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pages 611–626.

Zhuoyan Li, Hangxiao Zhu, Zhuoran Lu, and Ming Yin. 2023. Synthetic data generation with large language models for text classification: Potential and limitations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10443–10461.

Constantine Lignos, Nolan Holley, Chester Palen-Michel, and Jonne Sälevä. 2022. Toward more meaningful resources for lower-resourced languages. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 523–532, Dublin, Ireland. Association for Computational Linguistics.

Linlin Liu, Bosheng Ding, Lidong Bing, Shafiq Joty, Luo Si, and Chunyan Miao. 2021. MulDA: A multilingual data augmentation framework for low-resource cross-lingual NER. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5834–5846, Online. Association for Computational Linguistics.

Shervin Malmasi, Anjie Fang, Besnik Fetahu, Sudipta Kar, and Oleg Rokhlenko. 2022. Multiconer: A large-scale multilingual dataset for complex named entity recognition. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3798–3809.

Stephen Mayhew, Terra Blevins, Shuheng Liu, Marek Šuppa, Hila Gonen, Joseph Marvin Imperial, Börje Karlsson, Peiqin Lin, Nikola Ljubešić, Lester James Miranda, et al. 2024. Universal ner: A gold-standard multilingual named entity recognition benchmark. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4322–4337.

Stephen Mayhew, Chen-Tse Tsai, and Dan Roth. 2017. Cheap translation for cross-lingual named entity recognition. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2536–2545, Copenhagen, Denmark. Association for Computational Linguistics.

Arnav Mhaske, Harshit Kedia, Sumanth Doddapaneni, Mitesh M. Khapra, Pratyush Kumar, Rudra Murthy, and Anoop Kunchukuttan. 2023. Naamapadam: A large-scale named entity annotated data for Indic languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10441–10456, Toronto, Canada. Association for Computational Linguistics.

David Mueller, Nicholas Andrews, and Mark Dredze. 2020. Sources of transfer in multilingual named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8093–8104, Online. Association for Computational Linguistics.

Jian Ni, Georgiana Dinu, and Radu Florian. 2017. Weakly supervised cross-lingual named entity recognition via effective annotation and representation projection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1470–1480, Vancouver, Canada. Association for Computational Linguistics.

OpenAI. 2025. Introducing GPT-4.1 in the api. `https://openai.com/index/gpt-4-1/`. Accessed: 2025-11.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1946–1958, Vancouver, Canada. Association for Computational Linguistics.

Afshin Rahimi, Yuan Li, and Trevor Cohn. 2019. Massively multilingual transfer for NER. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 151–164, Florence, Italy. Association for Computational Linguistics.

Fabian David Schmidt, Ivan Vulić, and Goran Glavaš. 2022. SLICER: Sliced fine-tuning for low-resource cross-lingual transfer for named entity recognition. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10775–10785, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Chenxi Whitehouse, Monojit Choudhury, and Alham Fikri Aji. 2023. LLM-powered data augmentation for enhanced cross-lingual performance. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 671–686, Singapore. Association for Computational Linguistics.

Jiateng Xie, Zhilin Yang, Graham Neubig, Noah A. Smith, and Jaime Carbonell. 2018. Neural cross-lingual named entity recognition with minimal resources. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 369–379, Brussels, Belgium. Association for Computational Linguistics.

Jian Yang, Shaohan Huang, Shuming Ma, Yuwei Yin, Li Dong, Dongdong Zhang, Hongcheng Guo, Zhoujun Li, and Furu Wei. 2022. CROP: Zero-shot cross-lingual named entity recognition with multilingual labeled sequence translation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 486–496, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jiali Zeng, Yufan Jiang, Yongjing Yin, Xu Wang, Binghuai Lin, and Yunbo Cao. 2022. DualNER: A dual-teaching framework for zero-shot cross-lingual named entity recognition. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1837–1843, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yichun Zhao, Jintao Du, Gongshen Liu, and Huijia Zhu. 2022. TransAdv: A translation-based adversarial learning framework for zero-resource cross-lingual named entity recognition. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 742–749, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ran Zhou, Xin Li, Lidong Bing, Erik Cambria, Luo Si, and Chunyan Miao. 2022. ConNER: Consistency training for cross-lingual named entity recognition. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8438–8449, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

A Synthetic Data Generation

As shown in Figure 1, we present the following prompt to the LLM in the data generation process:

```
Help me make a {language} Named Entity Recognition dataset. Please give me {n} new datapoints, formatted as a single JSON object. Make sure the examples are unique and diverse. Here are some examples to get you started:
{m examples}
```

We prompt `GPT-4.1` using OpenAI's batch API functionality[4]; for the open-source models, we use the `vLLM` library (Kwon et al., 2023) to run inference.

For `GPT-4.1`, we used the OpenAI API's functionalities for structured outputs to ensure that outputs were formatted as JSON strings. For the open-sourced models, we experimented with using `transformers`-compatible libraries for obtaining structured outputs from LLMs, but ultimately found better results simply specifying the JSON requirement in the model and system prompt. For the open-sourced models, we used the following system prompt:

```
You are a helpful model that helps build text-based datasets, but does not produce any conversation besides the text it is asked to produce. You only output JSON strings.
```

[4]https://platform.openai.com/docs/guides/batch

For `GPT-4.1`, we used the following (minimally different) system prompt, on the assumption that specifying output mode in the system prompt was less important on account of the API's structured output functionalities:

```
You are a helpful model that helps build text-based datasets, but does not produce any conversation besides the text it is asked to produce.
```

We ran both open-sourced models with a temperature setting of 0.8, and nucleus sampling value of 0.8. We initially used a maximum new token limit of 4096 for both models. However, noticing that some of `Llama-3.1-8B-Instruct`'s unusable datapoints were specifically due to hitting new token limits, we regenerated data from this model with a maximum new token limit of 8192. Calls to `GPT-4.1` were made using default hyperparameters.

Table 1 shows some of the examples of the different types of responses to these prompts.

B Related-Language Model Details

In the NER FINE-TUNING setting, we first train an NER model on a language related to the target language, before fine-tuning it further on the target language NER data. Below is the list of related languages chosen to build a base NER model for each target language.

B.1 NER-fine tuning: Implementation Details

We source the pre-trained `XLM-RoBERTa-large` weights from Huggingface using the `transformers` library; fine-tuning is implemented using training pipelines from the same library. In the NER FROM SCRATCH setting, we train on the the target language data for 10 epochs; in the NER FINE-TUNING setting, we train on the related language data for 5 epochs, and then the target language data for 10 epochs. In all cases, we use a learning rate of 2e-05, and a batch size of 16.

C Full Results of WikiANN Comparison

The WikiANN dataset is a massively multilingual NER benchmark, comprising data from 176 lan-

Target Language	Related Language Chosen
Kannada	Telugu
Tamil	Telugu
Telugu	Kannada
Malayalam	Tamil
Kinyarwanda	Swahili
Swahili	Kinyarwanda
Yoruba	Igbo
Igbo	Yoruba
Swedish	Danish
Danish	Swedish
Slovak	English*

Table 2: List of related languages used in the NER FINE-TUNING setting for each target language. *English is not closely related to Slovak, but given the absence of another closely related language among the 11 target languages, it was chosen as the language for the base NER model to be fine-tuned.

guages (Pan et al., 2017; Rahimi et al., 2019).[5] Table 3 shows the full list of comparisons between NER model performance when trained on organic data, GPT-4.1-produced data, and WikiANN data. The sizes of the WikiANN train sets vary significantly between different languages, meaning we often cannot assess the quality of the data in the context of training sets containing over 1000 datapoints (e.g. Kannada and Yoruba, whose WikiANN train sets contain only 100 datapoints). In such cases, however, we compare model performance when trained on equally small amounts of organic or LLM-produced synthetic data.

Language		N.F.S. F1	N.F.T. F1	Data Size
Kannada	WikiANN	4.5e-3	0.77	100
	GPT-4.1	0.03	0.75	100
	GPT-4.1	0.00	0.43	4899
	Naamapadam	0.49	0.79	100
	Naamapadam	**0.80**	**0.78**	5000
Telugu	WikiANN	0.67	0.74	1000
	GPT-4.1	0.00	0.40	1000
	GPT-4.1	0.18	0.47	4931
	Naamapadam	0.71	0.81	1000
	Naamapadam	**0.84**	**0.83**	5000
Tamil	WikiANN	0.55	0.62	15000
	GPT-4.1	0.00	0.57	4944
	Naamapadam	**0.73**	**0.74**	5000
Malayalam	WikiANN	0.65	0.74	10000
	GPT-4.1	0.25	0.35	4859
	Naamapadam	**0.83**	**0.83**	5000
Yoruba	WikiANN	0.07	0.21	100
	GPT-4.1	0.20	0.42	100
	GPT-4.1	0.37	0.50	4187
	MasakhaNER 2	0.28	0.54	100
	MasakhaNER 2	**0.83**	**0.84**	5000
Swahili	WikiANN	0.50	0.59	1000
	GPT-4.1	0.74	0.78	1000
	GPT-4.1	0.78	0.79	2619
	MasakhaNER 2	0.88	0.91	1000
	MasakhaNER 2	**0.92**	**0.91**	5000
Kinyarwanda	WikiANN	7.9e-4	0.35	100
	GPT-4.1	0.29	0.46	100
	GPT-4.1	0.47	0.50	2683
	MasakhaNER 2	0.07	0.61	100
	MasakhaNER 2	**0.79**	**0.81**	5000
Igbo	WikiANN	7.7e-3	0.39	100
	GPT-4.1	0.29	0.55	100
	GPT-4.1	0.53	0.55	3473
	MasakhaNER 2	0.48	0.76	100
	MasakhaNER 2	**0.82**	**0.83**	5000
Danish	WikiANN	0.72	0.71	20000
	GPT-4.1	0.79	0.82	4112
	Universal NER	**0.83**	**0.85**	4383
Swedish	WikiANN	0.36	0.29	20000
	GPT-4.1	0.69	0.74	4215
	Universal NER	**0.87**	**0.90**	4303
Slovak	WikiANN	0.57	0.55	20000
	GPT-4.1	0.29	0.29	4508
	Universal NER	**0.77**	**0.80**	5000

Table 3: Performance of NER models trained on WikiANN, synthetic data from GPT-4.1, and high quality 'organic' data, for all 11 languages. N.F.S: NER FROM SCRATCH setting; N.F.T: NER FINE-TUNING setting.

[5]As Lignos et al. (2022) also note, strictly speaking, the original version of WikiANN put together by Pan et al. (2017) contains data from 282 languages; the version of the dataset commonly downloaded from Huggingface, however, and put together by Rahimi et al. (2019), contains data from 176 languages. In this work, we refer to the latter when referring to the WikiANN dataset.