arXiv:2402.15343v1 [cs.CL] 23 Feb 2024

NuNER: Entity Recognition Encoder Pre-training via LLM-Annotated Data

Sergei Bogdanov [1] **Alexandre Constantin** [1] **Timothée Bernard** [2] **Benoit Crabbé** [2] **Etienne Bernard** [1]

[1] NuMind [2] Université Paris Diderot

sergei@numind.ai, alexandre@numind.ai, timothee.bernard@u-paris.fr,
benoit.crabbe@u-paris.fr, etienne@numind.ai

Abstract

Large Language Models (LLMs) have shown impressive abilities in data annotation, opening the way for new approaches to solve classic NLP problems. In this paper, we show how to use LLMs to create NuNER, a compact language representation model specialized in the Named Entity Recognition (NER) task. NuNER can be fine-tuned to solve downstream NER problems in a data-efficient way, outperforming similar-sized foundation models in the few-shot regime and competing with much larger LLMs. We find that the size and entity-type diversity of the pre-training dataset are key to achieving good performance. We view NuNER as a member of the broader family of *task-specific foundation models*, recently unlocked by LLMs.

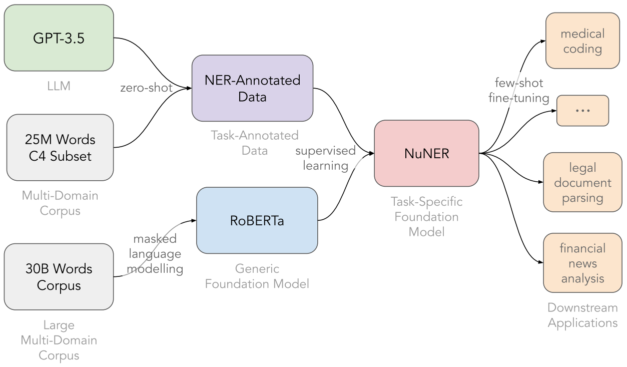


Figure 1. NuNER creation procedure. RoBERTa is further pre-trained on a subset of C4 automatically annotated by GPT-3.5. The resulting model can be fine-tuned on various downstream NER problems.

1. Introduction

Named Entity Recognition (NER) — the generic task of extracting and classifying entities from text — is a core component of natural language processing, present in a variety of applications such as medical coding, financial news analysis, or legal documents parsing (Francis et al., 2019; Dozier et al., 2010). Such application typically involves solving a particular NER problem, for a particular set of entity types, thus requiring the creation of a custom model.

For the last five years, the standard procedure for creating such custom model has consisted of using a transformer encoder (Vaswani et al., 2017) pre-trained in a self-supervised way to satisfy a masked language modeling (MLM) objective, such as models of the BERT family (Devlin et al., 2019; Liu et al., 2019; He et al., 2021). This *foundation model* is then fine-tuned in a supervised way on human-annotated data, either using a simple token-classification approach, or a more advanced strategy (Li et al., 2023; Zhang et al., 2023).

In the last few years, we have witnessed the emergence of generative large language models (LLMs) such as GPT-3 (Brown et al., 2020) and, more recently, GPT-4 (OpenAI, 2023), which typically have between 100 times and 10,000 times more parameters than BERT. These massive auto-regressive transformer models, trained via a next-word prediction objective, are language generators which can be prompted to perform a variety of tasks. For example, these models can be directly used through a well crafted prompt to tackle a particular NER problem with satisfying performance (Wang et al., 2023a). The main issue with this approach is the high inference cost due to the size of LLMs.

A simple solution to this inference-cost issue is to use a correctly-prompted LLM to annotate data for the particular NER problem, and then to train a smaller model on this data. LLMs have been shown to outperform crowdworkers (Gilardi et al., 2023) on some tasks for a fraction of the cost so this strategy is sound, but it has issues as well. First, crafting a good prompt — like delegating a task to someone else — is not easy; it requires multiple back-and-forth while validating the performance using human-annotated data. Second, LLMs are not perfect annotators either (Mao et al., 2023). Finally, the best LLMs are mainly hosted on private companies' servers and accessible through external APIs, opening the door to potential confidentiality and privacy leaks.

We propose an alternative approach that leverages LLMs to reduce the amount of human annotations needed to create custom models. Instead of using an LLM to directly annotate a particular single-domain dataset for a particular NER problem, our idea is to use this LLM to annotate a multi-domain dataset for variety of NER problems. We then further pre-train a small foundation model, such as BERT, on this annotated dataset. The resulting pre-trained model can then be fine-tuned to any downstream NER problem, just like any other foundation model, as depicted in Figure 1.

Because the resulting pre-trained model is specialized to a generic task but is still meant to be fine-tuned to a particular problem, we refer to such a model as a *task-specific foundation model*. Note that compact domain-specific foundation models like SciBERT (Beltagy et al., 2019) or BioBERT (Lee et al., 2019) are common, but task-specific foundation models of this kind are rare, mostly due to the lack of suitable datasets. Generative LLMs are the key to building such models.

In this paper, we apply the above idea to create NuNER, a task-specific foundation model for the generic task of NER.

In Section 3, we describe both the dataset creation and the training procedures. In a nutshell, we use GPT-3.5 to annotate a subset of C4 (Raffel et al., 2020), resulting in a 24.4M words dataset containing 4.38M annotations from 200k unique concepts. We then pre-train a base RoBERTa on this dataset via a contrastive-learning approach (Chen et al., 2020) to obtain NuNER.

In Section 4, we analyze the transfer learning performance of NuNER in an extended few-shot regime. We find that NuNER largely outperforms both its base model and the same base model further pre-trained on NER-BERT data (Liu et al., 2021), which is the largest and most diverse NER dataset we could find. These results demonstrate the validity of our approach.

In Section 5, we investigate the factors influencing NuNER's abilities. We find that the diversity of the annotations and the size of the pre-training dataset are the most influential factors. Surprisingly, the diversity of the text does not appear to be as influential.

In Section 6, for informational purposes, we compare fine-tuning NuNER with using GPT-3.5 and GPT-4 via in-context learning. We find that NuNER beats GPT-3.5 and competes with GPT-4 when more than a dozen entities of each type is seen during training. We also compare, via fine-tuning, NuNER with UniversalNER (Zhou et al., 2023), a recent LLM specialized in the NER task. We find that they exhibit similar transfer learning performance when fine-tuned, despite NuNER being 56 times smaller.

The contributions of our paper are as follows.

1. We introduce and demonstrate the validity of a procedure that consists of annotating raw data with an LLM in order to train a task-specific foundation model for NER.

2. We identify the factors that are likely to improve the performance of the resulting task-specific foundation model.

3. We provide and open-source NuNER[1], a compact encoder-based language representation model for NER. NuNER outperforms similar-sized models, competes with LLMs, and can be used as a drop-in replacement for RoBERTa.

4. We provide and open-source an LLM-annotated NER dataset[1], containing 4.38M annotations from 200k entity types, which is suitable for pre-training NER models.

2. Related Work

Early attempt to create NER-specific foundation models for low-resource NER problems focused on leveraging Wikipedia anchors (Mengge et al., 2020; Cao et al., 2019). In particular, Liu et al. (2021) (NER-BERT) combined these anchors and DBpedia Ontology to create a NER dataset containing 3.64M entities from 315 entity types. BERT was then pre-trained on this dataset, leading to improved few-shot performance. We demonstrate in Section 4.1 that our LLM-annotation procedure outperforms such approach.

Subsequent work has focused on pre-training generative LLMs on a large number of existing human-annotated NER datasets to achieve strong zero-shot capabilities (Wang et al., 2023b; Sainz et al., 2023). These works differ from ours in terms of type of data used, model employed, and objective (zero-shot vs. few-shot).

Recently, Zhou et al. (2023) proposed UniversalNER, an LLM with 7B and 13B parameters, also pre-trained on data annotated by GPT-3.5. Our work primarily differ in the model and training procedure: we found a way to train an encoder model with only 125M parameters on such data, making it substantially cheaper to use. In Section 6, we show that NuNER and UniversalNER have similar transfer-learning abilities when provided with more than a few training examples per entity types.

Even more recently, Zaratiana et al. (2023) proposed GLiNER, which uses UniversalNER's data to pre-train a small encoder model. Our work was done concurrently and independently from this work. The principal distinction is in the architecture used: GLiNER merges the text and concept

[1] https://huggingface.co/numind

encoders, whereas in our approach, they remain independent (see Figure 5). While GLiNER's integrated approach likely enhances performance, our decision to keep the encoders separate enables NuNER to function as a concept-agnostic language representation model. This gives the possibility to pre-compute text embeddings for a variety of applications such as information retrieval. This also makes NuNER a viable drop-in substitute for BERT or RoBERTa in standard NER methodologies.

3. NuNER

The creation of NuNER is a two-step process: dataset creation and model training.

3.1. Dataset Creation

We begin with a random sample of C4 (Raffel et al., 2020), an English web crawl corpus that contains text from a wide range of sources, including blog posts, news articles, and social media messages. We selected this dataset for its domain diversity.

We want to annotate this dataset with entities spanning a large and diverse set of types in order for our model to generalize to all kind of NER problems. To achieve this we opt for an unconstrained approach: we allow the LLM to extract any entity it identifies, and give it the freedom to assign any type it deems appropriate for each entity. This includes annotating with concepts more akin to topics than entity types (e.g., "wellness"). We refer to these entity types/topics as *concepts*. To resolve potential ambiguities, we also ask the LLM to provide descriptions for the concepts it identifies. However, we ultimately disregarded these descriptions as we found that they do not improve NuNER's performance. Our prompt is shown in Figure 2.

> The goal is to create a dataset for entity recognition. Label as many entities, concepts, and ideas as possible in the input text. Invent new entity types that may not exist in traditional NER Tasks such as more abstract concepts and ideas. Make sure the entity concept is not part of speech but something more meaningful. Avoid finding meaningless entities.
> Output format (separate entities with new line):
> entity from the text <> entity concept <> description of entity group/concept
> Input:
> [INPUT SENTENCE]

Figure 2. Prompt used to annotate NuNER's pre-training data.

Note that we do not ask to return the position of the entity in the text as LLMs are not good at counting. To train NuNER, we have to retrieve this position through an exact string match, which can lead to annotation errors in rare cases.

We use gpt-3.5-turbo-0301 with this prompt to annotate 1.35M sentences. Figure 3 shows one of these annotated sentences. We then apply a simple filter to remove sentences containing an annotation with the concept "concept", as we consider this too uninformative, and obtain a final dataset of 1M annotated sentences.

> "Steven Means has signed a one-year contract extension with the Falcons after making four starts in 2018."

ENTITY	CONCEPT	DESCRIPTION
Steven Means	NFL player	professional athlete
Falcons	NFL team	professional sports team
2018	year	unit of time

Figure 3. Sentence from C4 annotated with GPT-3.5.

We find that GPT-3.5 performs quite well in this task. From a manual review of 100 examples, we estimate that, in more than 95% of cases, the concept associated with the extracted entity is completely sensible, as shown in Figure 3. In fewer that 5% of cases, the extraction and associated concept are questionable, such as "cosmos seeds" identified as "plant variety" when it is actually a seed. However, many entities are missed. For instance, in the example of Figure 3, the LLM could have also identified "person". In a sense, this method has high precision but low recall. In Section 3.2, we propose a training procedure that mitigates the impact of these false negatives.

The resulting dataset comprises a total of 4.38M entity annotations, distributed across 200k unique concepts. These concepts cover a wide range of domains, as illustrated in Figure 6. The diversity of concepts in this dataset is far greater than what can be found in human-annotated NER datasets.

We observe a high imbalance in concept frequencies. Common concepts such as "person", "location", or "organization" each appears in more than 1% of the extracted entities, while over 100k concepts are seen only once in the dataset. This heavy-tailed distribution of concept frequencies is shown in Figure 4. We further investigate the importance of concept diversity in Section 5.2.

3.2. Model Training

We want to pre-train a language representation model using our annotated dataset, which contains 200k concepts and exhibits high concept-imbalance. Additionally, some concepts are similar to each other, such as "company" and "company name". Moreover, many potential entities in a sentence are not extracted. Due to these factors, training a conventional token classifier that only takes the text as input and returns a distribution over the entire set of concepts is

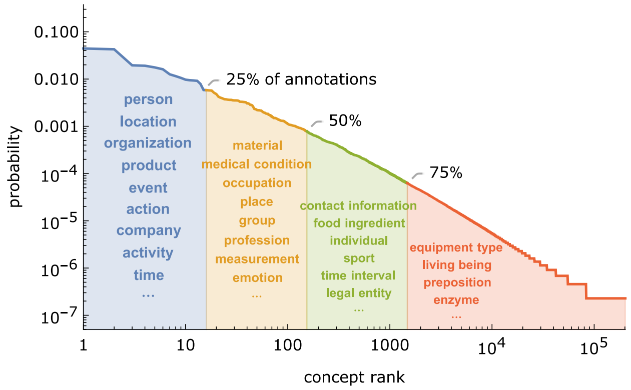


Figure 4. Frequency of each concept assigned by GPT-3.5, sorted from most to least common. We observe a heavy-tailed distribution.

not practical. We instead propose a training method based on the contrastive learning framework (Chen et al., 2020).

Our training network, depicted in Figure 5, consists of two separate sub-networks: The first is NuNER — the network of interest — which encodes the input text as a sequence of vectors. The second encodes a concept name as a unique vector. The text vectors are matrix-multiplied with the concept vector to obtain logits, which are then passed through a logistic sigmoid to yield probabilities. During training, this setup encourages each token embedding to align with the concept embedding if the token instantiates the concept, and to become opposite otherwise.

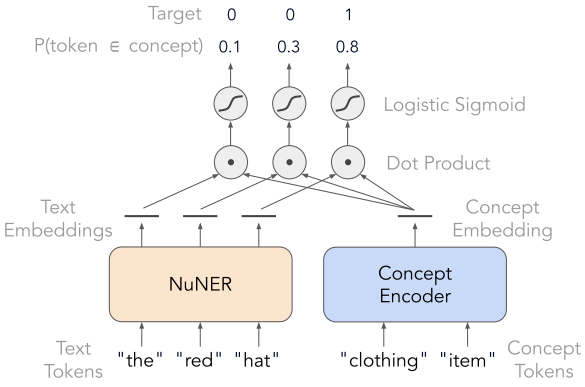


Figure 5. NuNER's pre-tranining procedure. The text and concept encoder are separated. Their embeddings are compared to obtain probabilities.

For each training batch, we collect all the concepts exemplified in the batch and construct a binary array of dimensions #sentences × #tokens × #concepts_in_batch. This array indicates the presence (1) or absence (0) of each concept present in the batch for every token in each sentence. We then use the binary cross-entropy loss to fit the network's probabilities to these target arrays.

Since we do not account for concepts not present in the batch, and because our dataset misses some concepts in some sentences, the probabilities generated by such a training network would need calibration for use in a zero-shot setting. However, in our case, this miscalibration is not a significant issue as we are only interested in using the text encoder NuNER.

3.3. Training Details

We use RoBERTa-base (Liu et al., 2019) for both the text encoder and the concept encoder. Our model is trained for 10 epochs on the full 1M sentence dataset where 90% of sentences are used for training and 10% for validation. We choose a learning rate $lr = 0.00003$, batch size$= 48$, and temperature before the sigmoid $\tau = 5$. We use AdamW optimizer (Kingma & Ba, 2015) with $\beta 1 = 0.9, \beta 2 = 0.999, \epsilon = 10^{-6}$, weight decay=0.01, and a linear scheduler with a warm-up for the first 10% of the training steps. The bottom 6 layers of the text encoder are frozen as we found it leads to better training stability. After training, we discard the concept encoder and keep the text encoder NuNER.

4. Transfer Learning Performance

NuNER is designed to be fine-tuned to downstream NER problems in a data-efficient way. We are mostly interested in the transfer learning performance of NuNER in an extended few-shot regime, typically from 1 to 100 annotations per entity types. We focus on this range because we believe it represents the level of annotation effort practitioners are willing to undertake without resorting to external annotation solutions, such as crowdsourcing.

4.1. Few-Shot with Frozen Foundation

This first experiment aims to demonstrate the benefits of further pre-training an MLM encoder on our LLM-annotated dataset. Also, we want to compare NuNER's pre-training with an alternative large-scale NER dataset. To this end, we compare NuNER with its base model, RoBERTa-base, as well as RoBERTa-base pre-trained on the dataset of NER-BERT (Liu et al., 2021), see Section 2.

We pre-train RoBERTa on NER-BERT data by replicating the training process of Liu et al. (2021), with the exception that we freeze the bottom half of the network — as when training NuNER — since we find it improves the final performance.

In order to compare these three foundation models, we transform them into token classifiers by attaching a linear layer on top of their final token representations. The entity

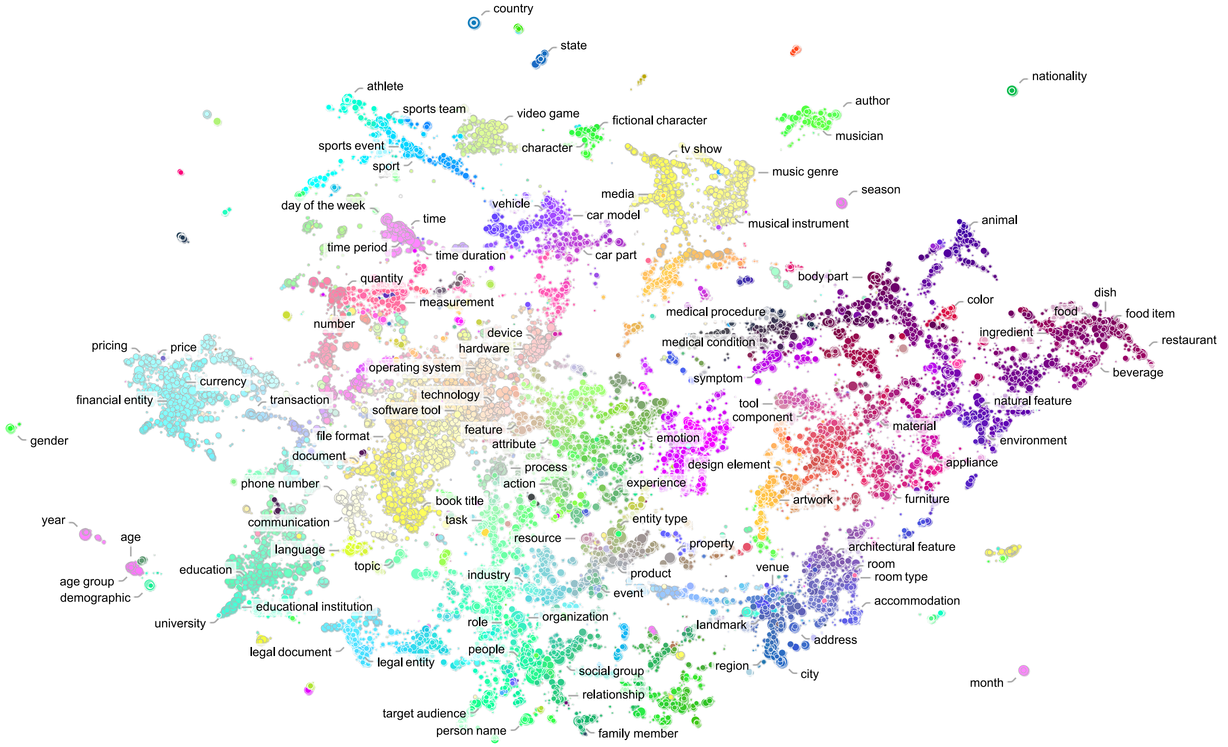


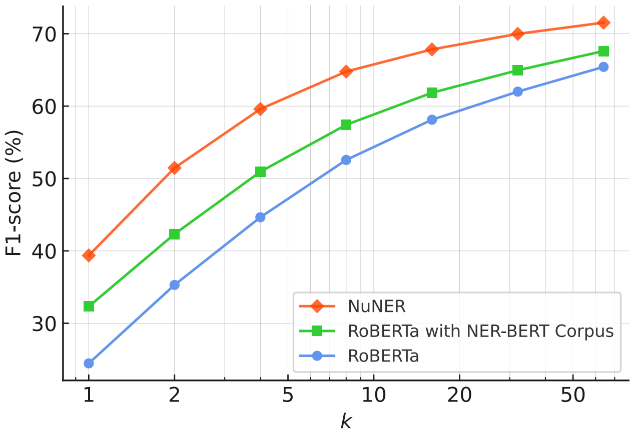
Figure 6. Feature map of the 50k most common concepts extracted by gpt-3.5-turbo-0301. Embeddings are obtained from the concept encoder (as depicted in Figure 5). UMAP(McInnes & Healy, 2018) is used to obtain 2D positions and 3D RGB color values. Disk size is proportional to log(concept frequency).

types returned by the classifiers are mutually exclusive, and a special "None" class is used to indicate the absence of entities. To simplify the few-shot training procedure — and because our focus is on the relative performance of the models — we train only the top layer while keeping the representation network frozen.

We use four datasets from different domains: OntoNotes 5.0 (Weischedel et al., 2013), BioNLP 2004 (Collier et al., 2004), MIT Restaurant (Liu et al., 2013), and MIT Movie (Liu et al., 2013). Performance is measured using the macro-averaged F1-Score of token classifications.

We adopt the $k \sim 2k$ mining procedure of (Ding et al., 2021) to obtain training examples that contain between k and $2k$ annotations per entity type. We measure performance averaged over 10 training sets for each value of $k \in \{1, 2, 4, 8, 16, 32, 64\}$. The reported performance for a given k is the average across all four datasets.

Performance is reported in Figure 7. As expected, NuNER largely outperforms pure RoBERTa. More surprisingly, NuNER outperforms RoBERTa trained on NER-BERT data by a large margin on all training sizes. We see this behavior in all four datasets, although the effect is stronger in some than others. This result demonstrates the benefits of pre-training using NuNER's dataset.



MODEL	1	4	16	64
ROBERTA	24.5	44.7	58.1	65.4
ROBERTA W. NER-BERT	32.3	50.9	61.9	67.6
NUNER	**39.4**	**59.6**	**67.8**	**71.5**

Figure 7. Transfer learning performance of NuNER, RoBERTa, and RoBERTa pre-trained on NER-BERT data as function of k. NuNER substantially outperforms both models for all training sizes. Full table and dataset-wise results are shown in Table 3 and Figure 13 in the appendix.

4.2. Few-Shot with TadNER on Few-NERD

To complement the previous analysis, we evaluate NuNER on Few-NERD (Ding et al., 2021), a challenging and widely recognized benchmark for few-shot NER. Our goal is to see whether NuNER can achieve new state-of-the-art performance.

Few-NERD is comprised of 188k Wikipedia sentences that are human-annotated from a set of 8 coarse-grained and 66 fine-grained entity types. In this benchmark, the tested solution is allowed to undergo pre-training on a large subset of Few-NERD before being evaluated using 5,000 few-shot train-test splits. In the INTRA setting, the entities seen during pre-training and evaluation belong to different coarse-grained types, while in the INTER setting, the entities share the same coarse-grained types.

The current state-of-the-art on Few-NERD is TadNER (Li et al., 2023), an advanced framework that employs a span-detection network, a type-classification network, as well as type-aware span filtering process and prototype construction. BERT is used for both networks. We adapt TadNER by replacing BERT with a NuNER based on the same BERT. Furthermore, since NuNER is designed to have all its entity-related knowledge in its last layer, we modify TadNER to only use this last layer instead of averaging over the last four layers.

Results are presented in Table 1. We observe that NuNER outperforms the original TadNER results in all settings and training sizes, thereby establishing it as the new state-of-the-art for this benchmark. This further hints at the benefits of using NuNER's pre-training procedure.

5. Ablation Studies

We aim to understand which factors in the pre-training procedure most significantly affect the performance of NuNER. To this end, we investigate the impact of text diversity, concept diversity, pre-training dataset size, and model size, using the same benchmark as in Section 4.1.

5.1. Effect of Text Diversity

In Section 4.1, we saw that NuNER's pre-training data, based on C4, leads to better performance than NER-BERT's pre-training data, based on Wikipedia. This might simply be because C4 is more diverse than Wikipedia.

To investigate this, we downsample both datasets to 50k sentences each, and annotate the Wikipedia subset with our LLM-annotation procedure. This way, the only distinguishing factor between these datasets is their original corpus: C4 vs. Wikipedia. We then pre-train RoBERTa on these two datasets and measure the transfer learning performance of the resulting models.

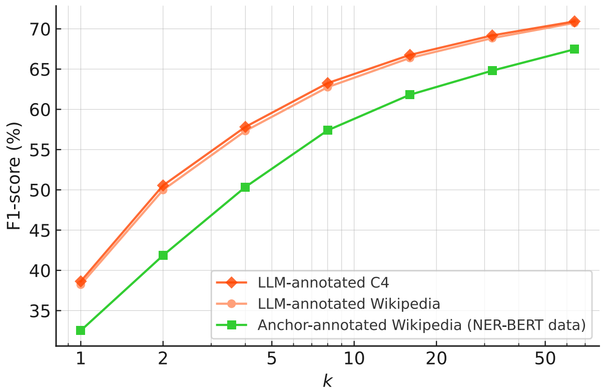


Figure 8. Effect of text diversity on NuNER's performance. Wikipedia and C4 lead to similar performance when they are both annotated by the LLM. Results table is shown in Table 4 in the appendix.

We see in Figure 8 that the LLM-annotated C4 subset and the LLM-annotated Wikipedia subset result in very similar model performance. This shows that the main reason for the performance gap is the LLM-annotation procedure rather than the underlying corpus.

5.2. Effect of Concept Diversity

To understand the effect of concept diversity, we first take a random sample of 100k annotated examples from NuNER's dataset, which includes approximately 80k unique concepts. We then retain only the annotations from the top-n most frequent concepts, simulating an annotation procedure that excludes rare concepts. We use $n = 4$, 16, 154, 1.5k, and 80k concepts, corresponding to 12.5%, 25%, 50%, 75%, and 100% of all annotations, respectively (see Figure 4). We pre-train NuNER on the resulting datasets and measure the transfer learning performance for $k = 8$.

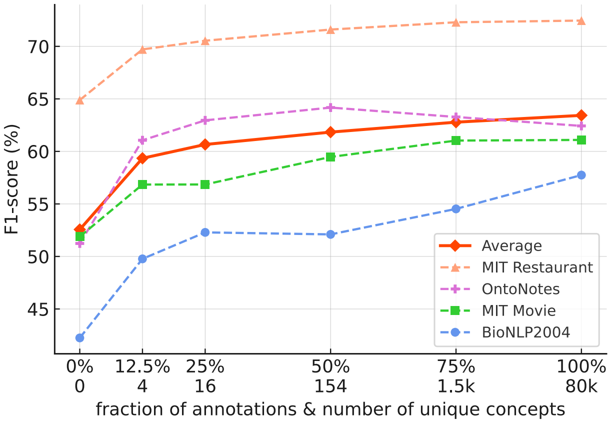


Figure 9. Effect of concept diversity on NuNER's performance. Results table is shown in Table 5 in the appendix.

Table 1. Few-NERD performance using TadNER (Li et al., 2023) and a modified TadNER using NuNER-BERT as the backbone.

MODEL	5-WAY 1 $\sim$ 2	10-WAY 1 $\sim$ 2	5-WAY 5 $\sim$ 10	10-WAY 5 $\sim$ 10	AVG
TADNER — INTRA	60.78±0.32	55.44±0.08	67.94±0.17	60.87±0.22	61.26
NUNER-BERT — INTRA	**62.48±0.28**	**57.63±0.38**	**69.16±0.28**	**62.99±0.27**	**63.07**
TADNER — INTER	64.83±0.14	64.06±0.19	72.12±0.12	69.94±0.15	67.74
NUNER-BERT — INTER	**67.37±0.31**	**66.54±0.40**	**73.50±0.09**	**71.04±0.14**	**69.61**

Results are shown in Figure 9. As expected, overall performance increases with concept diversity. However, there are variations across datasets. BioNLP appears to benefit the most from concept diversity while it seems to harm OntoNotes past 154 concepts. This difference is likely because BioNLP contains rarer concepts than OntoNotes. The performance degradation on OntoNotes may indicate the difficulty of encoding a large number of concepts into the 768-dimensional embedding vector.

Note that, in this experiment, the number of annotations grows with concept diversity, which might bring an additional effect. Results of Section 5.3 shows that such effect would account here for less than 1% of F1-score.

5.3. Effect of Dataset Size

We next investigate the effect of the pre-training dataset size, ranging from one 1k examples to 1M examples. We again pre-train NuNER on each dataset and measure its transfer learning performance for $k = 8$. Results are shown in Figure 10.

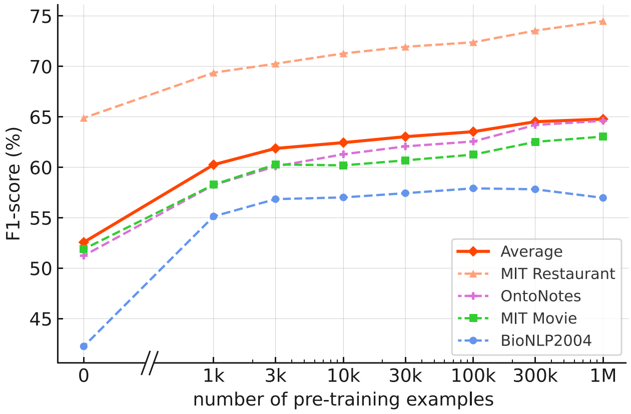


Figure 10. Effect of pre-training dataset size on NuNER's performance. Results table is shown in Table 6 in the appendix.

As expected, the overall performance increases with data size, and continues to improve slightly from 300k to 1M examples. Again, we see variability across datasets, with the performance for BioNLP decreasing after 100k examples, while all other datasets experience a monotonic increase. The reason for this discrepancy is unclear.

5.4. Effect of Model Size

Finally, we investigate the influence of model size. We pre-train a version of NuNER using RoBERTa large (355M parameters) and compare it to the original NuNER (155M parameters). We see in Figure 11 an overall increase of the F1-score of a few percent. This increase is more pronounced for smaller training sets ($k < 10$) than for larger ones. Combined with the positive impacts of concept diversity and dataset size, this result suggests that scaling up both models and data would lead to further performance improvements.

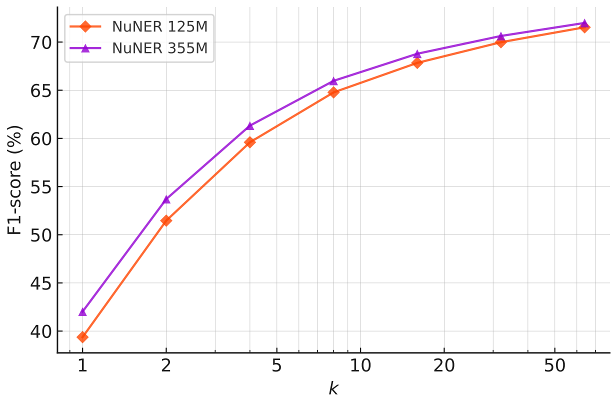


Figure 11. Effect of model size on NuNER's performance. Results table shown in Table 7 in the appendix.

6. Comparison with LLMs

Although the comparisons are indicative, we finally provide for information a comparison of NuNER with modern generative LLMs. We choose GPT-3.5 (gpt-3.5-turbo-16k-0613) for its popularity and longer context, GPT-4 (gpt-4-0613) for its high performance (Zheng et al., 2023), and UniversalNER (UniversalNER-7B-type) for its specialization in the NER task. UniversalNER has 56 times more parameters than NuNER, and GPT-3.5 and GPT-4 are likely to have around 1,000 and 10,000 times more parameters than NuNER, respectively.

GPT-4 and GPT-3.5 are used via in-context learning using Spacy's NER V3 prompt. This advanced prompt template allows to create a prompt for a particular NER problem by providing it the set of entity types and some training examples.

UniversalNER (Zhou et al., 2023) is trained to be used for zero-shot inferences, conducted through a conversation in which one sequentially prompt the model to identify each entity type. To adapt this model for a few-shot setting, we need to fine-tune it. We use the original training settings of UniversalNER but modify them to enhance few-shot learning performance, as detailed in Appendix A.2. For NuNER, we simply attach a two-layer fully-connected network regularized via dropout, and fine-tune the entire network for 30 epochs.

Because of financial constraints associated with GPT-4, we deviate from the extended few-shot learning protocol of Section 4.1. We only use the MIT Restaurant and BioNLP datasets, and downsample test sets to 1,000 examples. Also, we create training sets with a specific number of words belonging to a given entity type, that we call k_w, instead of using the $k \sim 2k$ entity-based mining method. We conduct several runs for each training-size (using the same training sets for all models) and average results, except for GPT-4 where we only perform one run. Results are presented in Figure 12.

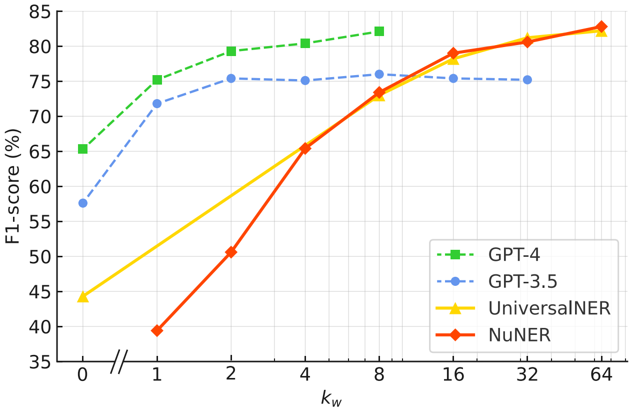


Figure 12. Comparison of NuNER with LLMs. Dashed curves indicate in-context learning and solid curves indicate fine-tuning. Results tables are shown in Table 10 and Table 11 in the appendix.

We find that GPT-3.5 and GPT-4 already perform well in the zero-shot regime, and show rapid improvement when examples are added to the prompt. However, for larger training sets, we see something unexpected: the performance of GPT-3.5 quickly plateaus, and it ends up being outperformed by both UniversalNER and NuNER when $k_w > 8$. The same thing might happen with GPT-4 but we cannot conclude with such noisy and incomplete results. UniversalNER starts lower than both GPT-3.5 and GPT-4, but steadily catches on to eventually surpass GPT-3.5. Clearly, fine-tuning does not suffer from this early saturation issue.

NuNER begins at a lower performance level than the others as it is not intended to be a zero-shot model. However, it quickly matches UniversalNER's performance, and eventually surpass GPT-3.5. It remains unclear whether it would also surpass GPT-4.

To compare NuNER and UniversalNER in a more standard and reproducible manner, we conduct an additional experiment using the $k \sim 2k$ setting on all four datasets mentioned in Section 4.1. In this experiment, we use the full test sets and measure entity-level micro F1-score, averaged over these datasets. We perform 3 runs for each k. We see in Table 2 that NuNER and UniversalNER have similar performance.

Table 2. NuNER vs. UniversalNER few-shot entity-level F1-score in the $k \sim 2k$ setting showing similar performance. Dataset-wise tables can be found in Table 8 and in Table 9 in the appendix.

MODEL	$8 \sim 16$	$64 \sim 128$
UNIVERSALNER	57.89 ± 4.34	71.02 ± 1.53
NUNER	58.75 ± 0.93	70.30 ± 0.35

The fact that NuNER surpasses GPT-3.5 and possibly also GPT-4 is likely attributable to the limitations of in-context learning. The situation might differ with proper, albeit challenging, few-shot fine-tuning for these LLMs. More surprising to us is that NuNER achieves performance comparable to UniversalNER despite being 56 times smaller and trained on similar data. This could be due to an inherent advantage of encoders over generative models for this task. Alternatively, it might be related to NuNER's pre-training procedure, which encourages human concepts to emerge in the last layers of the network, being easily accessible during few-shot training. Further experiments would be needed to explore this aspect.

7. Conclusion

Modern large language models are opening new possibilities for addressing traditional NLP tasks. We have introduced a procedure that uses these LLMs to create a compact, yet data-efficient, NER-specific foundation model. We foresee an increasing trend in the development of such task-specific foundation models, which will facilitate the creation of high-quality custom NLP models without requiring intensive human or computational resources.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Beltagy, I., Lo, K., and Cohan, A. Scibert: A pretrained language model for scientific text. In Inui, K., Jiang, J., Ng, V., and Wan, X. (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pp. 3613–3618. Association for Computational Linguistics, 2019. doi: 10.18653/V1/D19-1371. URL https://doi.org/10.18653/v1/D19-1371.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Cao, Y., Hu, Z., Chua, T.-s., Liu, Z., and Ji, H. Low-resource name tagging learned with weakly labeled data. In Inui, K., Jiang, J., Ng, V., and Wan, X. (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 261–270, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1025. URL https://aclanthology.org/D19-1025.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A Simple Framework for Contrastive Learning of Visual Representations, November 2020. URL https://proceedings.mlr.press/v119/chen20j.html.

Collier, N., Ohta, T., Tsuruoka, Y., Tateisi, Y., and Kim, J.-D. Introduction to the bio-entity recognition task at JNLPBA. In Collier, N., Ruch, P., and Nazarenko, A. (eds.), *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP)*, pp. 73–78, Geneva, Switzerland, August 28th and 29th 2004. COLING. URL https://aclanthology.org/W04-1213.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. BERT: pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T. (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/V1/N19-1423. URL https://doi.org/10.18653/v1/n19-1423.

Ding, N., Xu, G., Chen, Y., Wang, X., Han, X., Xie, P., Zheng, H., and Liu, Z. Few-nerd: A few-shot named entity recognition dataset. In Zong, C., Xia, F., Li, W., and Navigli, R. (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pp. 3198–3213. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.ACL-LONG.248. URL https://doi.org/10.18653/v1/2021.acl-long.248.

Dozier, C., Kondadadi, R., Light, M., Vachher, A., Veeramachaneni, S., and Wudali, R. Named entity recognition and resolution in legal text. In Francesconi, E., Montemagni, S., Peters, W., and Tiscornia, D. (eds.), *Semantic Processing of Legal Texts: Where the Language of Law Meets the Law of Language*, volume 6036 of *Lecture Notes in Computer Science*, pp. 27–43. Springer, 2010. doi: 10.1007/978-3-642-12837-0_2. URL https://doi.org/10.1007/978-3-642-12837-0_2.

Francis, S., Landeghem, J. V., and Moens, M. Transfer learning for named entity recognition in financial and biomedical documents. *Inf.*, 10(8):248, 2019. doi: 10.3390/INFO10080248. URL https://doi.org/10.3390/info10080248.

Gilardi, F., Alizadeh, M., and Kubli, M. ChatGPT outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30):e2305016120, July 2023. URL https://www.pnas.org/doi/10.1073/pnas.2305016120.

He, P., Liu, X., Gao, J., and Chen, W. Deberta: decoding-enhanced bert with disentangled attention. In *9th International Conference on Learning Representations, ICLR*

2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net, 2021. URL https://openreview.net/forum?id=XPZIaotutsD.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Kingma, D. P. and Ba, J. Adam: A Method for Stochastic Optimization. In Bengio, Y. and LeCun, Y. (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., and Kang, J. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, September 2019. ISSN 1367-4811. doi: 10.1093/bioinformatics/btz682. URL http://dx.doi.org/10.1093/bioinformatics/btz682.

Li, Y., Yu, Y., and Qian, T. Type-aware decomposed framework for few-shot named entity recognition. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 8911–8927, Singapore, December 2023. Association for Computational Linguistics. URL https://aclanthology.org/2023.findings-emnlp.598.

Liu, J., Pasupat, P., Cyphers, S., and Glass, J. R. Asgard: A portable architecture for multilingual dialogue systems. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, BC, Canada, May 26-31, 2013*, pp. 8386–8390. IEEE, 2013. doi: 10.1109/ICASSP.2013.6639301. URL https://doi.org/10.1109/ICASSP.2013.6639301.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019. URL http://arxiv.org/abs/1907.11692.

Liu, Z., Jiang, F., Hu, Y., Shi, C., and Fung, P. NER-BERT: A pre-trained model for low-resource entity tagging. *CoRR*, abs/2112.00405, 2021. URL https://arxiv.org/abs/2112.00405.

Mao, R., Chen, G., Zhang, X., Guerin, F., and Cambria, E. Gpteval: A survey on assessments of chatgpt and GPT-4. *CoRR*, abs/2308.12488, 2023. doi: 10.48550/ARXIV.2308.12488. URL https://doi.org/10.48550/arXiv.2308.12488.

McInnes, L. and Healy, J. UMAP: uniform manifold approximation and projection for dimension reduction. *CoRR*, abs/1802.03426, 2018. URL http://arxiv.org/abs/1802.03426.

Mengge, X., Yu, B., Zhang, Z., Liu, T., Zhang, Y., and Wang, B. Coarse-to-Fine Pre-training for Named Entity Recognition. In Webber, B., Cohn, T., He, Y., and Liu, Y. (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6345–6354, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.514. URL https://aclanthology.org/2020.emnlp-main.514.

OpenAI. GPT-4 technical report. *CoRR*, abs/2303.08774, 2023. doi: 10.48550/ARXIV.2303.08774. URL https://doi.org/10.48550/arXiv.2303.08774.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Sainz, O., García-Ferrero, I., Agerri, R., de Lacalle, O. L., Rigau, G., and Agirre, E. Gollie: Annotation guidelines improve zero-shot information-extraction. *CoRR*, abs/2310.03668, 2023. doi: 10.48550/ARXIV.2310.03668. URL https://doi.org/10.48550/arXiv.2310.03668.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is All you Need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30*, pp. 5998–6008. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7181-attention-is-all-you-need.

Wang, S., Sun, X., Li, X., Ouyang, R., Wu, F., Zhang, T., Li, J., and Wang, G. GPT-NER: named entity recognition via large language models. *CoRR*, abs/2304.10428, 2023a. doi: 10.48550/ARXIV.2304.10428. URL https://doi.org/10.48550/arXiv.2304.10428.

Wang, X., Zhou, W., Zu, C., Xia, H., Chen, T., Zhang, Y., Zheng, R., Ye, J., Zhang, Q., Gui, T., Kang, J., Yang, J., Li, S., and Du, C. Instructuie: Multi-task instruction tuning for unified information extraction. *CoRR*, abs/2304.08085, 2023b. doi: 10.48550/ARXIV.2304.08085. URL https://doi.org/10.48550/arXiv.2304.08085.

Weischedel, R., Palmer, M., Marcus, M., Hovy, E., Pradhan, S., Ramshaw, L., Xue, N., Taylor, A., Kaufman, J., Franchini, M., El-Bachouti, M., Belvin, R., and Houston, A. OntoNotes Release 5.0, 2013. URL https://hdl.handle.net/11272.1/AB2/MKJJ2R.

Zaratiana, U., Tomeh, N., Holat, P., and Charnois, T. Gliner: Generalist model for named entity recognition using bidirectional transformer. *CoRR*, abs/2311.08526, 2023. doi: 10.48550/ARXIV.2311.08526. URL https://doi.org/10.48550/arXiv.2311.08526.

Zhang, S., Cheng, H., Gao, J., and Poon, H. Optimizing bi-encoder for named entity recognition via contrastive learning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/pdf?id=9EAQVEINuum.

Zheng, L., Chiang, W., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E. P., Zhang, H., Gonzalez, J. E., and Stoica, I. Judging llm-as-a-judge with mt-bench and chatbot arena. *CoRR*, abs/2306.05685, 2023. doi: 10.48550/ARXIV.2306.05685. URL https://doi.org/10.48550/arXiv.2306.05685.

Zhou, W., Zhang, S., Gu, Y., Chen, M., and Poon, H. Universalner: Targeted distillation from large language models for open named entity recognition. *CoRR*, abs/2308.03279, 2023. doi: 10.48550/ARXIV.2308.03279. URL https://doi.org/10.48550/arXiv.2308.03279.

A. Appendix

A.1. Comparison with LLMs - Details

In the experiment of Figure 12 we only conduct one training run for GPT-4 for cost reasons. For all the other models we conduct 16 training runs for $k_w = 1$, 8 training runs for $k_w = 2$, 4 training runs for $k_w = 4$, 2 training runs for $k_w = 8$, and a unique training run for higher training sizes. These choices were made to reduce GPT-3.5 costs and to for the training data to be kept consistent across all models for each training size.

A.2. UniversalNER Training Details

We create UniversalNER's training data following Zhou et al. (2023)'s procedure: We use the dataset-specific instruction tuning template and populate it to extract all entity types. We don't need to add any negative sampling here since all possible entity types are queried, therefore having an empty list when no entities of a given type are present in the example.

We fine-tune UniversalNER using authors' fine-tuning code, only changing the number of training epochs and the number of gradient accumulation steps to obtain better few-shot results. After experimenting with external datasets, we found that using 20 epochs for $k_w = 8$ and $k = 8$, 15 epochs for $k_w = 16$ and $k_w = 32$, and 10 epoch for $k_w = 64$ and $k = 64$ worked well. We used these values and, for $k_w = 8$ and $k = 8$, also reduced the number of gradient accumulation steps to 4. We didn't train UniversalNER on $k_w < 8$ as regularization was difficult to tune.

We also tried adopting LORA (Hu et al., 2022) for fine-tuning UniversalNER. Although, this methodology was more stable thanks to its implied regularization, it consistently led to worse results than full fine-tuning and required more time to converge.

We believe our heuristics allow to train UniversalNER pretty well in a few-shot setting, and are enough to make a rough comparison with NuNER. However, we should note that fine-tuning such large model on such small amount of data is not easy, and there are certainly better automatic machine learning procedures for this model.

A.3. Extra Tables and Figures

In this section we present the tables that were partially cut or just illustrated as plots in the body of the paper. We also show dataset-wise results that were shown as averages in the body of the paper.

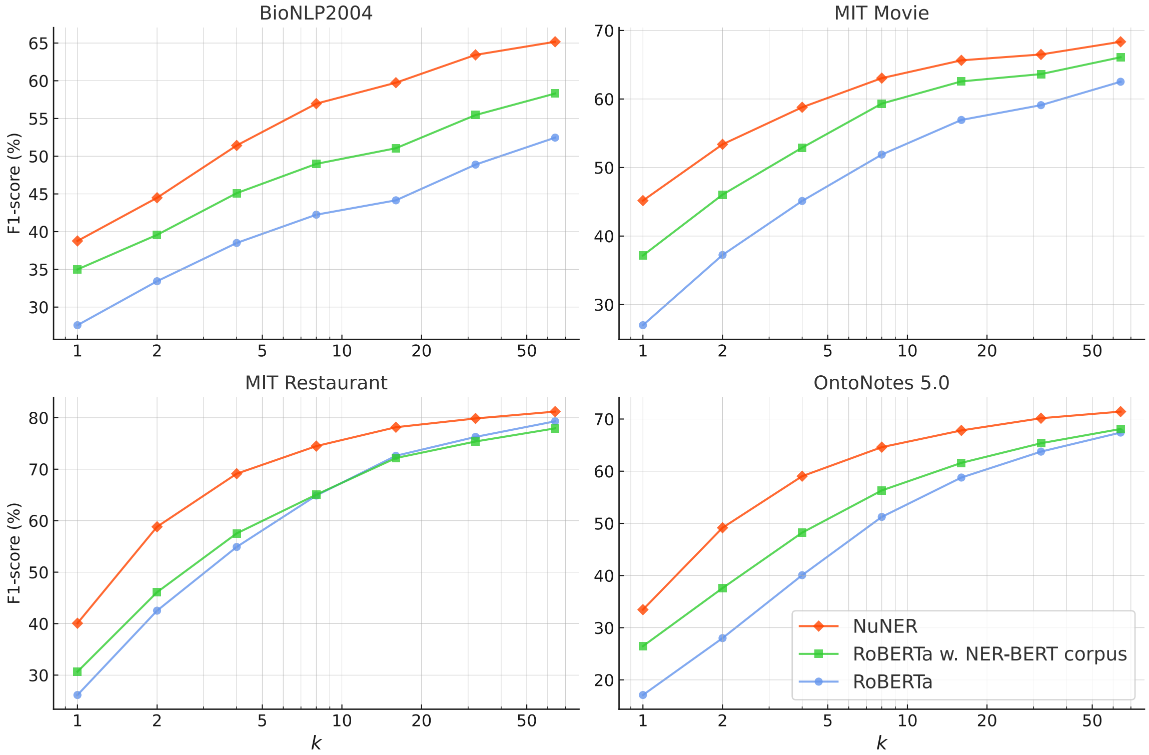


Figure 13. Dataset-wise results of Figure 7. Transfer learning performance of NuNER, RoBERTa, and RoBERTa pre-trained on NER-BERT data as function of k. NuNER substantially outperforms both models for all training sizes. We see similar behaviors for every dataset.

Table 3. Results table of Figure 7. Transfer learning performance of NuNER, RoBERTa, and RoBERTa pre-trained on NER-BERT data as function of k. NuNER substantially outperforms both models for all training sizes.

k	RoBERTa	RoBERTa w. NER-BERT	NuNER
1	24.46	32.32	**39.37**
2	35.30	42.33	**51.46**
4	44.65	50.94	**59.60**
8	52.56	57.42	**64.78**
16	58.12	61.85	**67.84**
32	62.00	64.96	**69.98**
64	65.42	67.61	**71.53**

Table 4. Results table of Figure 8. Effect of text diversity on NuNER's performance. Wikipedia and C4 lead to similar performance when they are both annotated by the LLM.

k	Anch-An. Wiki	LLM-an. Wiki	LLM-an. C4
1	32.52	38.23	**38.64**
2	41.86	49.99	**50.55**
4	50.35	57.32	**57.82**
8	57.40	62.76	**63.26**
16	61.82	66.39	**66.76**
32	64.81	68.85	**69.18**
64	67.46	70.74	**70.93**

Table 5. Results table of Figure 9. Effect of concept diversity on NuNER's performance.

k	4	16	154	1500	80K
	12.5%	25%	50%	75%	100%
1	32.78	36.95	**39.56**	39.45	38.83
2	44.40	47.33	49.75	50.53	**50.69**
4	52.57	54.75	56.60	57.58	**57.97**
8	59.35	60.66	61.84	62.78	**63.43**
16	63.74	65.05	65.49	66.38	**66.83**
32	66.76	67.67	67.96	68.92	**69.27**
64	69.31	70.02	70.03	70.76	**71.04**

Table 6. Full results table of Figure 10 for all training sizes. Effect of pre-training dataset size on NuNER's performance.

k	1K	3K	10K	30K	100K	300K	1M
1	32.8	35.6	37.3	38.3	38.9	39.3	**39.4**
2	45.5	48.0	49.3	50.2	50.8	**51.6**	51.5
4	54.0	56.2	56.9	57.6	58.1	59.1	**59.6**
8	60.3	61.9	62.5	63.0	63.5	64.5	**64.8**
16	64.3	65.4	66.1	66.6	66.9	67.7	**67.8**
32	66.5	67.8	68.4	69.0	69.3	69.8	**70.0**
64	68.3	69.4	70.0	70.7	71.1	**71.6**	71.5

Table 7. Full results table of Figure 11 for all training sizes. Effect of model size on NuNER's performance.

k	NuNER	NuNER-large
1	39.37	42.02
2	51.46	53.70
4	59.60	61.33
8	64.78	65.97
16	67.84	68.79
32	69.98	70.63
64	71.53	71.99

Table 8. Dataset-wise results of Table 2 for $k = 8$.

DATASET	NuNER	UniversalNER
BIONLP	43.84 ± 2.18	41.79 ± 17.08
MIT MOVIE	58.69 ± 1.16	61.57 ± 2.26
MIT RESTAURANT	62.66 ± 2.28	63.4 ± 1.69
ONTONOTES	69.82 ± 1.64	64.8 ± 1.17
AVERAGE	58.75 ± 0.93	57.89 ± 4.34

Table 9. Dataset-wise results of Table 2 for $k = 64$

DATASET	NuNER	UniversalNER
BIONLP	61.18 ± 1.15	59.94 ± 6.03
MIT MOVIE	65.56 ± 0.29	69.50 ± 0.57
MIT RESTAURANT	73.66 ± 0.65	75.84 ± 0.97
ONTONOTES	80.81 ± 0.41	78.78 ± 0.26
AVERAGE	70.30 ± 0.35	71.02 ± 1.53

Table 10. Results table of Figure 12 for MIT Restaurant. Comparison of NuNER with LLMs. F1-macro token classification metric.

k_w	GPT-3.5	GPT-4	UniversalNER	NuNER
0	51.70	**69.54**	28.35	-
1	77.26	**79.66**	-	38.48
2	78.83	**81.48**	-	51.08
4	80.01	**84.54**	-	74.47
8	81.00	**85.35**	76.56	78.67
16	**82.00**	-	80.30	81.58
32	81.34	-	82.36	**82.53**
64	-	-	83.40	**84.79**

Table 11. Results table of Figure 12 for BioNLP. Comparison of NuNER with LLMs. F1-macro token classification metric.

k_w	GPT-3.5	GPT-4	UniversalNER	NuNER
0	**63.48**	61.09	60.23	-
1	67.62	**70.82**	-	40.39
2	70.70	**77.10**	-	50.02
4	70.22	**76.32**	-	56.31
8	70.86	**78.80**	69.52	68.01
16	68.88	-	76.09	**76.31**
32	69.12	-	**80.09**	78.78
64	-	-	**80.96**	80.85