A Comparative Study of Pre-trained Encoders for Low-Resource Named Entity Recognition

Yuxuan Chen[1] **Jonas Mikkelsen**[1]
Arne Binder[1] **Christoph Alt**[2,3] **Leonhard Hennig**[1]
[1]German Research Center for Artificial Intelligence (DFKI)
[2]Humboldt Universität zu Berlin [3]Science of Intelligence
[1]{*yuxuan.chen, jonas.mikkelsen, arne.binder, leonhard.hennig*}*@dfki.de*
[2]*christoph.alt@posteo.de*

Abstract

Pre-trained language models (PLM) are effective components of few-shot named entity recognition (NER) approaches when augmented with continued pre-training on task-specific out-of-domain data or fine-tuning on in-domain data. However, their performance in low-resource scenarios, where such data is not available, remains an open question. We introduce an encoder evaluation framework, and use it to systematically compare the performance of state-of-the-art pre-trained representations on the task of low-resource NER. We analyze a wide range of encoders pre-trained with different strategies, model architectures, intermediate-task fine-tuning, and contrastive learning. Our experimental results across ten benchmark NER datasets in English and German show that encoder performance varies significantly, suggesting that the choice of encoder for a specific low-resource scenario needs to be carefully evaluated.

1 Introduction

Pre-trained language models (PLM) have been shown to be very effective few-shot learners for a wide range of natural language processing tasks (Brown et al., 2020; Gao et al., 2021), as they capture semantically and syntactically rich representations of text via self-supervised training on large-scale unlabeled datasets (Peters et al., 2018; Devlin et al., 2019). Recent research in few-shot named entity recognition (NER) has leveraged such representations, e.g. for metric learning on task-specific out-of-domain[1] data (Fritzler et al., 2019; Yang and Katiyar, 2020), optionally augmented by continued pre-training with distantly supervised, in-domain data (Huang et al., 2021). However, there has been no systematic comparison of the NER performance of such representations in low-resource scenarios without task-specific out-of-domain data and very limited in-domain data; a prevalent setting in many practical applications.

In this paper we conduct a comparative study to answer the following research questions: How well do representations learnt by different pre-trained models encode information that benefits these low-resource scenarios? What can we observe for different categories of encoders, such as encoders trained with masked language modeling, versus encoders that are additionally fine-tuned on downstream tasks, or optimized with contrastive learning? How do they perform across different datasets and languages? We present an evaluation framework inspired by few-shot learning to evaluate representations obtained via different pre-training strategies, model architectures, pre-training data, and intermediate-task fine-tuning in low-resource NER scenarios of varying difficulty (see Figure 1).

We find that the choice of encoder can have significant effects on low-resource NER performance, with F1 scores differing by up to 25% between encoders, and simply picking an encoder of the BERT family at random will usually not yield the best results for a given scenario. We observe that while BERT in general performs adequately, ALBERT and RoBERTa outperform BERT by a large margin in many cases, with ALBERT being especially strong in very low-resource settings with only one available labeled example per class.

The main contributions of this study are: (1) a systematic performance evaluation of a wide range of encoders pre-trained with different strategies, such as masked language modeling, task-specific fine-tuning, and contrastive learning on the task of low-resource named entity recognition; (2) an evaluation on ten benchmark NER datasets in two languages, English and German; (3) an encoder-readout evaluation framework that can be easily extended with additional scenarios, encoders, datasets, and readout approaches; which we release at `https://github.com/dfki-nlp/fewie`.

[1]Out-of-domain and in-domain refer to NER-specific data with disjoint label spaces, i.e. $\mathcal{Y}_{out} \neq \mathcal{Y}_{in}$.

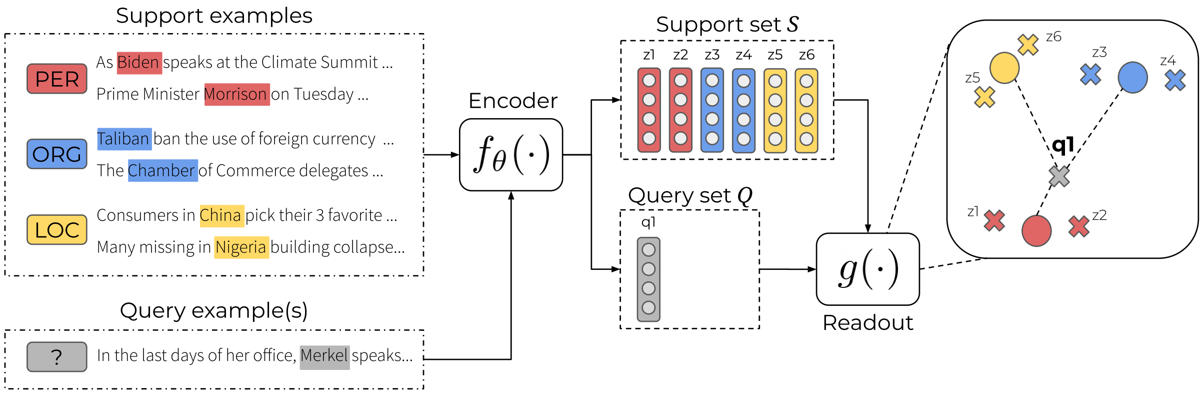

Figure 1: Encoder-readout evaluation framework. For each of the N classes, we randomly sample K support tokens including their sentence context, and an unlabeled query token with sentential context. The encoder $f_{\boldsymbol{\theta}}(\cdot)$ provides an embedding (or representation) for each token, and the readout module $g(\cdot)$ assigns a class to a query token by comparing its representation $\boldsymbol{q}_j$ to the representations $\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_{N \times K}\}$ of the support tokens. Depending on the readout approach, the c-th class in $\mathcal{S}$ is represented either by its prototype embedding (as shown in the example) or by its set of associated token embeddings, e.g. for nearest neighbor classification. In this example $\boldsymbol{q}_1$ representing *Merkel* would be assigned the class *PER* based on the closest class prototype embedding (red circle).

2 Encoder Evaluation Framework

To simulate low-resource NER scenarios of varying difficulty, we draw inspiration from the evaluation of few-shot learning methods. We first give a formal definition of the few-shot NER task, and then introduce the encoder evaluation framework itself.

2.1 Few-shot NER task definition

NER is typically formulated as a sequence labeling problem, where the input is a sequence of tokens $\mathbf{X} = \{x_1, x_2, \cdots, x_T\}$ and the output is the corresponding T-length sequence of entity type labels $\mathbf{Y} = \{y_1, y_2, \cdots, y_T\}$. In contrast, few-shot learning is cast as an episodic N-way K-shot problem, where in each episode, N classes are sampled with K examples each to construct a support set $\mathcal{S} = \{\mathbf{X}_i, \mathbf{Y}_i\}_{i=1}^{N \times K}$ for learning, and K' examples per class are sampled to create a query set $\mathcal{Q} = \{\mathbf{X}_j, \mathbf{Y}_j\}_{j=1}^{N \times K'}$ for evaluation ($\mathcal{S} \cap \mathcal{Q} = \emptyset$). In a sequence labeling problem like NER, samples are typically sentences, due to the importance of contextual information for token classification, but care has to be taken to ensure that the sampled sentences contain no other entities. In particular, there should be no entity overlap between the support and the query sets (Ding et al., 2021).

2.2 Encoder-Readout Framework

Our framework consists of two modules, an encoder $f(\cdot)$ and a readout module $g(\cdot)$, as shown in Figure 1. The encoder provides an embedding $\boldsymbol{z} = f_{\boldsymbol{\theta}}(x)$ of a token x, where $\boldsymbol{\theta}$ denotes the parameters of the encoder. The readout module is responsible for assigning a class to each token x' in the query set $\mathcal{Q}$ given the support set $\mathcal{S}$. Depending on the readout approach, the c-th class in $\mathcal{S}$ is represented either by its prototype embedding or by its associated set of token embeddings, e.g. for nearest neighbor classification. The decision is made by comparing the embedding $\boldsymbol{q} = f_{\boldsymbol{\theta}}(x')$ with each of the N class prototypes built from the support set $\mathcal{S}$, or with each of the token-level embeddings.

3 Experiments

We illustrate the evaluation framework using a representative set of encoders pre-trained with different strategies. We then give details of the readout approaches, the datasets we used, and all other experimental settings.

3.1 Encoders

We group encoders into four categories, depending on their type of pre-training:

PLM These models are pre-trained on a large general corpus in a self-supervised manner without any task-specific fine-tuning. We consider six representative encoders for English: BERT cased and uncased (Devlin et al., 2019), SpanBERT (Joshi et al., 2020), XLNet (Yang et al., 2019), ALBERT (Lan et al., 2020) and RoBERTa (Liu et al., 2019), and three encoders for German: deepset's BERT, GottBERT (Scheible et al., 2020) and XLM-RoBERTa (Conneau et al., 2020).[2]

[2]HuggingFace model identifiers for these and all other

Language	Dataset	Domain	# Entity types	Entity tag set
English	CoNLL-2003$_{\text{EN}}$	News	4	LOC, MISC, ORG, PER
	OntoNotes 5.0	News, Dialogue	18	CARDINAL, DATE, EVENT, MONEY, ...
	Few-NERD$_{\text{coarse}}$	General	8	art, building, event, product, ...
	Few-NERD$_{\text{fine}}$	General	66	art-film, product-car, other-law, ...
	WNUT-17	Social Media	6	corporation, creative-work, group, ...
	WikiAnn	General	3	LOC, ORG, PER
	WikiGold	General	4	LOC, MISC, ORG, PER
	Zhang et al.	e-Commerce	4	ATTRIBUTE, BRAND, COMPONENT, PRODUCT
German	CoNLL-2003$_{\text{DE}}$	News	4	LOC, MISC, ORG, PER
	GermEval 2014	General	12	LOC, LOCderiv, LOCpart, ORG, ...
	Smartdata	News, General	16	DISASTER-TYPE, DISTANCE, LOCATION, ...

Table 1: Statistics of the evaluated datasets

Fine-tuned PLM Recent research has shown that intermediate-task training can result in significant performance gains on the target task even in low-resource settings (Vu et al., 2020; Poth et al., 2021). We evaluate three BERT encoders that are fine-tuned on token-level, sentence-level, and document-level intermediate tasks, respectively: BERT$_{\text{POS}}$ for part-of-speech tagging, BERT$_{\text{MNLI}}$, fine-tuned on the MultiNLI dataset (Williams et al., 2018), and BERT$_{\text{SQuAD}}$ for extractive question answering (Rajpurkar et al., 2016). Evaluating these encoders may allow us to observe whether the representation granularity induced by the tasks they were fine-tuned on has an effect on NER performance: While token-level part-of-speech tag information is a staple feature of classic NER approaches (Finkel et al., 2005), it is less clear if encoders trained on tasks that require conceptual representations (and possibly understanding) of sentence- and document-length context, learn entity representations useful for NER.

PLM fine-tuned on NER We also experiment with BERT$_{\text{CoNLL}}$, a BERT model fine-tuned on the CoNLL-2003 NER dataset. As this model's hidden representations have been adapted to NER, we expect it to exhibit better performance than the other representations. The most interesting question of using this model is whether its representations transfer to NER datasets with non-CoNLL tagsets.

PLM with contrastive learning For each of the English PLM encoders, we apply contrastive learning to learn representations with better separability. The idea of contrastive learning is to pull positives closer and push negatives away in the representation space during the pre-training phase (Rethmeier and Augenstein, 2021). We use the loss function proposed by Chopra et al. (2005):

$$\mathcal{L}_{CL}(x_i, x_j; \boldsymbol{\theta}) := \mathbb{1}_{y_i=y_j} \cdot \|f_{\boldsymbol{\theta}}(x_i) - f_{\boldsymbol{\theta}}(x_j)\| + \mathbb{1}_{y_i \neq y_j} \cdot \max\big(0, \epsilon - \|f_{\boldsymbol{\theta}}(x_i) - f_{\boldsymbol{\theta}}(x_j)\|\big).$$

To guarantee that this label-aware contrastive learning conforms to the few-shot setting, we construct positive/negative pairs from the support set: Given an N-way K-shot support set, for each of the N classes we construct 1 positive pair and K negative pairs.[3]

3.2 Readout approaches

We analyze three variants for the readout approach:[4] (1) **Logistic Regression (LR)**, a linear classification algorithm that can be extended to multinomial logistic regression to deal with multi-class (N-way) settings, such as the one discussed here. (2) **k-Nearest Neighbor (NN)**, a non-parametric classification method adopted in metric space. As proposed in STRUCTSHOT (Yang and Katiyar, 2020), we set $k = 1$ to find the exact nearest token in the support set. (3) **Nearest Centroid (NC)** works similar to **NN**, but instead of computing the distance between the query and every instance in the embedding space, we represent each class by the centroid of all token embeddings belonging to this class, and assign the query to the class with the nearest centroid.

3.3 Datasets

In order to provide a comprehensive evaluation, we evaluate all encoders on a range of

models are listed in Appendix A.

[3] One extra example per class is needed for $K = 1$ to build one positive pair for this class. This extra example is involved only in the contrastive learning phase and not introduced to the encoding and readout steps.

[4] Computational details of the readout approaches can be found in Appendix B.

datasets covering different languages and domains, including seven English benchmarks: CoNLL-2003 (Tjong Kim Sang and De Meulder, 2003), Few-NERD (Ding et al., 2021), OntoNotes 5.0 (Weischedel et al., 2013), WikiAnn (Pan et al., 2017), WNUT-17 (Derczynski et al., 2017), WikiGold (Balasuriya et al., 2009), and the dataset of Zhang et al. (2020). For German, we selected the following three datasets: CoNLL-2003 (Tjong Kim Sang and De Meulder, 2003), Smartdata (Schiersch et al., 2018) and GermEval 2014 (Benikova et al., 2014). Table 1 lists the domains and tagset details of each dataset.

3.4 Experimental settings / Hyperparameters

Datasets We use the BIO tagging schema by default and the IO schema only when BIO is not provided by the original dataset (in case of Few-NERD, OntoNotes 5.0 and WikiGold). WikiGold and the dataset of Zhang et al. (2020) do not provide train/test splits, we therefore use the full dataset to sample support and query sets. For all other datasets, test splits are used for sampling.[5]

General settings For each dataset, we evaluate our methods under three few-shot scenarios: 5-way 1-shot, 5-way 5-shot and 5-way 10-shot. To produce accurate performance estimates, we sample 600 episodes for each scenario and report the mean token-level micro-F1 score over all episodes, averaged over all positive classes, and excluding the 'O' class.

Encoders Max-length is fixed at 128. We use randomly initialized, static embeddings as the baseline encoder (*Random*). For contrastive learning, we use the Adam optimizer and set the learning rate to be 5×10^{-5} and the number of epochs to be 1 across all encoders.

Readout approaches We L2-normalize the encoder embeddings before feeding them to the readout model. For NN and NC classification, Euclidean distance serves as the similarity metric between tokens. For LR, an L2-penalty is applied to the coefficients. All reported results use LR as the default readout method, unless specified otherwise, as we found LR to perform best on average (see Section 4.4).

Framework implementation We implement our low-resource NER encoder evaluation framework using the HuggingFace Transformers library (Wolf et al., 2020), Hydra (Yadan, 2019), and PyTorch (Paszke et al., 2019). Additional scenarios, encoders, and datasets can be easily added simply by creating new experiment configurations. Adding new readout methods is also a simple matter of a few lines of code.

4 Results and Discussion

4.1 Comparison of PLM encoders

We first analyze PLM encoders which have not been fine-tuned on any task.

English results Table 2 presents the experimental results of English-language encoders for different scenarios and datasets. For all scenarios and datasets, the PLM encoders outperform the randomly initialized baseline by a large margin. As expected, the NER classification performance of the encoders increases with higher K, i.e. with more instances per class in the support set. Overall, the level of performance across various datasets of this encoder-only approach to low-resource NER is surprisingly good: We observe that ALBERT achieves a token-level F1 score of $F1 = 72.8$ on CoNLL-2003, XLNet a score of $F1 = 85.7$ on Few-NERD fine-grained, and RoBERTa a score of $F1 = 83.8$ on OntoNotes 5.0. While these results are not directly comparable to those of state-of-the-art, fully supervised approaches due to the differences in the evaluation setup, they are achieved essentially fine-tuning-free, and with much fewer labeled instances per class.

Encoder analysis The best-performing encoders, on average and across datasets, are ALBERT, RoBERTa, and BERT. ALBERT is by far the best encoder for $K = 1$, but the other encoders achieve comparable performance or outperform ALBERT for $K \geq 5$. Even though ALBERT is an order of magnitude smaller in terms of its number of parameters than either BERT or RoBERTa, it provides very competitive embeddings in our evaluation setup. As can be expected, $\text{BERT}_{\text{cased}}$ consistently outperforms $\text{BERT}_{\text{uncased}}$ for datasets with tag sets where casing provides useful information for NER (e.g. CoNLL, WikiGold), but does not necessarily perform better if the tag set contains entity types whose instances use lower-case spelling. XLNet achieves mixed results, mainly depending on the dataset – on CoNLL-2003, WikiAnn and WNUT-17, its F1 scores are significantly lower for all scenarios than those of the best encoder, while on Few-NERD fine-grained, XLNet achieves the

[5] For Few-NERD, we use the test data from the "supervised" split.

Dataset	K	Random	BERT↓	BERT↑	ALBERT↓	RoBERTa↑	SpanBERT↑	XLNet↑
CoNLL-2003$_{\mathrm{EN}}$	1	9.52	21.96	22.04	**33.03**†	21.71	18.39	18.49
	5	12.53	60.94	62.17	**68.33**†	64.49	43.22	44.82
	10	13.71	66.11	68.79	**72.76**	72.09	49.79	52.43
OntoNotes 5.0	1	18.66	42.71	45.09	**50.45**†	42.74	34.30	38.40
	5	19.73	74.68	77.70	77.66	**78.70**	65.64	72.60
	10	18.88	80.92	82.70	82.10	**83.80**†	74.14	78.38
Few-NERD$_{\mathrm{coarse}}$	1	12.12	25.99	28.52	**35.67**†	28.12	23.34	25.93
	5	15.59	53.85	56.04	**59.14**	58.66	45.50	52.32
	10	16.04	59.44	63.20	63.30	**65.52**†	52.65	61.94
Few-NERD$_{\mathrm{fine}}$	1	21.14	49.74	48.50	**54.27**†	51.27	39.13	47.02
	5	21.00	80.12	79.26	78.08	81.70	71.93	**82.73**
	10	20.62	84.07	83.21	81.17	84.95	78.39	**85.73**
WNUT-17	1	18.86	25.71	25.67	**28.47**†	25.43	23.14	24.36
	5	19.11	51.56	50.58	**55.12**	54.59	42.29	42.26
	10	18.52	58.77	60.37	60.41	**63.93**†	48.84	49.74
WikiAnn	1	12.07	24.53	25.92	**32.63**†	24.80	22.67	22.06
	5	15.64	48.33	52.29	**53.11**†	51.34	40.60	36.81
	10	16.95	54.84	59.48	59.10	**60.83**	46.44	44.19
WikiGold	1	3.71	18.40	21.30	**32.30**†	20.63	14.90	18.01
	5	10.02	49.19	55.54	55.87	**56.08**	41.07	45.44
	10	11.62	55.85	63.91	61.23	**64.84**	48.09	53.85
Zhang et al.	1	13.49	37.39	36.82	**41.23**†	38.79	25.83	31.25
	5	17.08	63.19	62.17	62.73	**66.44**†	49.08	57.69
	10	16.21	67.45	67.09	66.61	**70.16**†	54.80	63.79

Table 2: Token-level micro-F1 scores of PLM encoders and a random baseline for 5-way K-shot scenarios, with logistic regression readout. † denotes scores with significant difference to the next-best encoder's score ($\alpha = 0.05$). ↑ and ↓ indicate cased and uncased models.

Dataset	K	Random	BERT↑	Gott-BERT↑	XLM-R↑
CoNLL-2003$_{\mathrm{DE}}$	1	12.53	29.42	26.27	**30.65**
	5	15.38	**65.98**	58.37	65.22
	10	16.00	**71.43**	64.77	71.18
GermEval 2014	1	17.52	25.89	24.08	**27.24**
	5	20.70	**61.79**†	54.06	58.51
	10	18.33	**71.18**†	60.30	65.37
Smartdata	1	26.12	51.12	49.96	**53.17**
	5	23.52	**82.50**†	79.30	80.89
	10	21.55	**86.01**	83.10	85.66

Table 3: Token-level micro-F1 scores of German PLM encoders and a random baseline under 5-way K-shot scenarios, with logistic regression readout. † denotes scores with a significant difference to the next-best encoder's score ($\alpha = 0.05$). ↑ indicates cased models.

best score of all encoders. SpanBERT on average shows the worst performance of all encoders, with F1 scores in most scenarios several percentage points lower than even those of XLNet. This suggests that SpanBERT's span-level masking and training with a span boundary objective produce token-level embeddings that are less well separable by the logistic regression classifier.

Dataset analysis On a per-dataset basis, we can observe the following from Table 2: On CoNLL-2003, ALBERT outperforms the next-best encoder BERT$_{\mathrm{cased}}$ for $K = 1$ by 11% F1, and achieves a best score of $F1 = 72.8$ for $K = 10$, closely followed by RoBERTa. XLNet's and SpanBERT's F1 scores are more than 20% lower than those of ALBERT for $K = 5$ and $K = 10$. On Few-NERD with coarse labels, ALBERT is again the best encoder at $K = 1$. For $K = 10$, RoBERTa achieves $F1 = 65.5$, but the other encoders except for SpanBERT perform almost as well. Using the fine-grained labels of Few-NERD, all encoders achieve around 80% F1 score. The overall picture is similar for OntoNotes 5.0 and the dataset of Zhang et al., with ALBERT being the best encoder at $K = 1$ and RoBERTa outperforming the other encoders at $K = 10$. BERT and XLNet show competitive performance to ALBERT and RoBERTa, yielding slightly lower F1 scores in all scenarios. This trend is also confirmed for the remaining datasets, WikiAnn, WNUT-17 and WikiGold, with ALBERT and RoBERTa being the strongest contenders, and BERT often catching up

Dataset	K	BERT↓	B_{POS}↓	B_{MNLI}↓	B_{SQuAD}↓
CoNLL-2003$_{EN}$	1	21.96	**43.01**†	22.29	35.05
	5	60.94	65.72	61.34	**65.94**
	10	66.11	68.46	64.71	**68.50**
OntoNotes 5.0	1	42.71	**50.85**†	42.99	47.83
	5	74.68	66.17	75.29	**76.37**
	10	80.92	68.02	**80.94**	79.68
Few-NERD$_{coarse}$	1	25.99	34.70	26.08	**35.07**
	5	53.85	49.88	52.52	**59.77**†
	10	59.44	52.78	58.17	**63.09**†
Few-NERD$_{fine}$	1	49.74	43.97	46.71	**51.17**
	5	**80.12**†	63.08	77.14	78.58
	10	**84.07**†	66.43	81.26	81.58
WNUT-17	1	25.71	**32.04**†	25.12	29.04
	5	**51.56**	44.90	48.50	51.05
	10	**58.77**†	49.11	56.30	54.58
WikiAnn	1	24.53	32.92	23.35	**33.33**
	5	48.33	43.54	46.94	**55.93**†
	10	54.84	45.70	53.47	**63.37**†
WikiGold	1	18.40	**37.46**†	20.33	30.80
	5	49.19	**55.54**†	50.86	53.96
	10	55.85	55.62	55.81	**57.99**†
Zhang et al.	1	37.39	**45.67**†	37.29	40.90
	5	**63.19**	59.58	62.98	61.01
	10	**67.45**	60.61	66.23	61.95

(a) Micro-F1 scores of BERT, and fine-tuned $BERT_{POS}$, $BERT_{MNLI}$ and $BERT_{SQuAD}$.

Dataset	Overlap	K	BERT↓	B_{CoNLL}↓
CoNLL-2003$_{EN}$	1.00	1	21.96	**90.46**†
		5	60.94	**94.73**†
		10	66.11	**94.40**†
WikiGold	1.00	1	18.40	**68.83**†
		5	49.19	**81.40**†
		10	55.85	**84.68**†
WikiAnn	0.75	1	24.53	**55.15**†
		5	48.33	**67.22**†
		10	54.84	**71.34**†
Few-NERD$_{coarse}$	0.50	1	25.99	**53.25**†
		5	53.85	**70.04**†
		10	59.44	**72.66**†
WNUT-17	0.25	1	25.71	**44.96**†
		5	51.56	**63.99**†
		10	58.77	**69.76**†
OntoNotes 5.0	0.16	1	42.71	**58.99**†
		5	74.68	**76.21**†
		10	**80.92**†	77.75
Few-NERD$_{fine}$	0	1	49.74	**59.36**†
		5	**80.12**	79.70
		10	**84.07**†	82.00
Zhang et al.	0	1	37.39	**49.22**†
		5	63.19	**65.40**†
		10	**67.45**	66.13

(b) Micro-F1 scores of BERT and $BERT_{CoNLL}$. The datasets are listed in descending order of tag set overlap with CoNLL-2003, as measured by Jaccard Index.

Table 4: Token-level micro-F1 scores of fine-tuned encoders under 5-way K-shot scenarios, with LR readout. † denotes scores with significant difference to the next-best encoder's score ($\alpha = 0.05$). ↓ indicates uncased models.

in terms of F1 scores with increasing K.

German results Table 3 shows the results of German-language encoders and the random baseline on three evaluation datasets. Similar to the English results, we observe that: (i) BERT, GottBERT and XLM-RoBERTa all benefit from more support instances, i.e. achieve a better performance with a larger training set, and outperform the random baseline by a large margin. (ii) XLM-RoBERTa shows the best performance across datasets in one-shot settings, whereas BERT outperforms the other encoders for $K \geq 5$. (iii) GottBERT's encodings yield features that are less useful for low-resource NER, resulting in worse performance than the other two encoders in all scenarios.

On CoNLL-2003, BERT achieves a micro-F1 score of 71.4 at $K = 10$, XLM-R a competitive score of 71.2, while GottBERT only achieves $F1 = 64.8$. Similar performance differences between the three encoders can be observed for the other two datasets at $K = 5$ and $K = 10$. At $K = 1$, XLM-R consistently outperforms BERT and GottBert, with GottBERT showing the worst performance. The results show that BERT, a model trained with less, but likely quality training data (Wikipedia, OpenLegalData, News) produces representations that are more suited for low-resource NER in most of the evaluated settings, compared to GottBERT (145GB of unfiltered web text), and XLM-RoBERTa (≈100GB filtered CommonCrawl data for German).

4.2 Fine-tuned encoders

Fine-tuned PLM The next group of encoders we analyze are encoders fine-tuned on an intermediate task, in our case POS tagging, NLI, and QA. Results are shown in Table 4a. We can see that using a BERT encoder fine-tuned on POS tagging significantly improves F1 scores at $K = 1$ for all datasets except Few-NERD fine-grained, on average by about 9 points. However, for $K \geq 5$, $BERT_{POS}$'s performance is significantly worse than that of BERT for the majority of datasets, except CoNLL-2003 and WikiGold.

The $BERT_{MNLI}$ model's performance is competitive with the base BERT model's, with no statistically significant differences. Fine-tuning on this sentence-level task, which is rather unrelated to NER, hence seems to have neither negative nor positive effects on the resulting token embeddings.

Embeddings obtained from $\text{BERT}_{\text{SQuAD}}$, fine-tuned on document-level span extraction, outperform BERT in most settings, often with statistical significance. However, on some datasets (e.g. WNUT-17, $\text{Few-NERD}_{\text{fine}}$), $\text{BERT}_{\text{SQuAD}}$'s scores are lower than BERT's for $K \geq 5$. Compared to the other fine-tuned encoders, $\text{BERT}_{\text{SQuAD}}$ performs better in general for $K \geq 5$. Its good performance may be attributed to the fact that approximately 41.5% of the answers in the SQuAD dataset correspond to common entity types, and another 31.8% to common noun phrases (Rajpurkar et al., 2016).

The observations for these three encoders coincide with the intuition, that the more relevant the knowledge encoded by the intermediate task is w.r.t. the target task, the more likely an improvement on the target task becomes.

PLM fine-tuned on NER Table 4b shows the results obtained for $\text{BERT}_{\text{CoNLL}}$, an encoder that was fine-tuned on CoNLL-2003. As can be expected, this encoder performs very well on the CoNLL-2003 test set, with large F1 gains in all scenarios. For most of the other datasets, F1 scores are also significantly improved for all settings of K, especially with a large tagset overlap. These results coincide with the intuition that the higher the tagset overlap, the larger the improvement. However, we note that some of these datasets are constructed from other data sources, e.g. web and social media texts, which indicates some transferability of the CoNLL-2003-tuned representations. Even for datasets where there is little or no overlap (OntoNotes 5.0, Zhang et al.), there are at least some gains at $K = 1$. However, at $K = 10$, the performance of the embeddings obtained from $\text{BERT}_{\text{CoNLL}}$ is significantly worse than that of the base BERT model.

4.3 PLM with contrastive learning

Table 5 compares the results of English encoders before and after contrastive learning. In general, results are mixed: For ALBERT and SpanBERT, using CL improves F1 scores in most cases, often with significant differences, whereas for BERT, RoBERTa and XLNET, the base encoders mostly exhibit (marginally) better performance.

Encoder analysis We observe that ALBERT benefits the most from contrastive learning, with significant F1 gains in 5 out of 12 comparisons, followed by SpanBERT (3), XLNet (1), BERT (1) and RoBERTa (0). Surprisingly, it achieves slightly higher F1-scores on Few-NERD coarse-grained and significantly higher F1-scores on WikiGold in all three scenarios. For 1-shot scenario on CoNLL-2003, ALBERT also gets a large F1 increase by 3.68%, the best improvement among all encoders.

Dataset analysis Few-NERD coarse-grained and WikiGold show better compatibility with contrastive learning, with 11 and 8 F1 improvements out of 15 comparisons after contrastive learning, respectively, compared with CoNLL-2003 (6) and OntoNotes 5.0 (4). Specifically, all five encoders have F1 gains on Few-NERD dataset in the one-shot scenario.

4.4 Readout approaches

Finally, Table 6 compares the different readout approaches on the CoNLL-2003 and OntoNotes 5.0 datasets, using ALBERT. For $K >= 5$, Logistic Regression outperforms Nearest Centroid and Nearest Neighbor classification, while for one-shot scenarios Nearest Neighbor performs best. NC is outperformed by LR and NN in all scenarios but 5-shot on OntoNotes 5.0. This suggests that with very few samples, the raw token embedding information, as used by NN, is a better representation of a class than the averaged embeddings as produced by LR and CN, but with more samples, weighted embeddings obtained with LR are more useful.

5 Related Work

Few-shot NER Recent work on few-shot NER has primarily focused on integrating additional knowledge to support the classification process. Fritzler et al. (2019) are the first to use pre-trained word embeddings for this task. Yang and Katiyar (2020) extend a Nearest Neighbor token-level classifier with a Viterbi decoder for structured prediction over entire sentences. Huang et al (2021) propose to continue pre-training of a PLM encoder with distantly supervised, in-domain data, and to integrate self-training to create additional, soft-labeled training data. Recently, Gao et al. (2021) and Ma et al. (2021) investigate methods for making PLMs better few-shot learners via prompt-based fine-tuning. While these approaches extend standard few-shot learning algorithms in promising di-

Dataset	K	BERT↓		ALBERT↓		RoBERTa↑		SpanBERT↑		XLNet↑	
		w/o CL	CL	w/o CL	CL	w/o CL	CL	w/o CL	CL	w/o CL	CL
CoNLL-2003$_{EN}$	1	21.96	**23.87**†	33.03	**36.71**†	21.71	**22.57**	**18.39**	17.61	**18.49**	18.25
	5	**60.94**	60.55	**68.33**	66.85	**64.49**	62.45	43.22	**44.23**	44.82	**45.93**
	10	**66.11**	65.03	**72.76**	70.66	**72.09**	70.17	48.79	**49.82**	**52.43**	49.25
OntoNotes 5.0	1	42.71	**42.89**	50.45	**51.38**	**42.74**	41.66	**34.30**	32.95	38.40	**38.64**
	5	**74.68**	74.02	**77.66**	76.65	**78.70**	75.29	**65.64**	64.29	**72.60**	70.66
	10	**80.92**	80.36	**82.10**	81.47	**83.80**	82.51	74.14	**74.72**	**78.38**	75.99
Few-NERD$_{coarse}$	1	25.99	**27.42**	35.67	**38.16**†	28.12	**29.10**	23.34	**23.40**	25.93	**26.35**
	5	**53.85**	52.97	59.14	**59.71**	**58.66**	55.75	45.50	**46.03**	52.32	**54.91**†
	10	59.44	**59.89**	63.30	**64.53**	**65.52**	62.86	52.65	**55.47**†	**61.94**	61.45
WikiGold	1	**18.40**	16.85	32.30	**34.05**†	**20.63**	19.90	14.90	**15.39**	18.01	**19.13**
	5	49.19	49.19	55.87	**57.67**†	**56.08**	53.91	41.07	**42.92**†	**45.44**	44.21
	10	55.85	**56.87**	61.23	**62.68**†	**64.84**	63.05	48.09	**50.93**†	**53.85**	52.26

Table 5: Token-level micro F1-scores of PLM encoders without and with contrastive learning (CL) for 5-way K-shot scenarios, with logistic regression readout. † denotes scores with a significant ($\alpha = 0.05$) improvement after contrastive learning. ↑ and ↓ indicate cased and uncased models.

Dataset	K	LR	NC	NN
CoNLL-2003$_{EN}$	1	33.03	35.21	**40.76**†
	5	**68.33**†	61.53	62.24
	10	**72.76**†	62.65	67.79
OntoNotes 5.0	1	50.45	51.52	**52.72**
	5	**77.66**†	72.46	71.04
	10	**82.10**†	73.49	76.11

Table 6: Micro-F1 scores of ALBERT for 5-way K-shot scenarios, comparing Logistic Regression (LR), Nearest Centroid (NC) and Nearest Neighbor (NN) readout approaches.

rections, none of them directly investigate the contribution of different pre-trained representations. As such, our analysis complements these works. Das et al. (2021) present a contrastive pre-training approach for few-shot NER that uses in-domain data to fine-tune token embeddings before few-shot classification. In contrast, we only consider contrastive examples from the sampled few-shot set to conform to the low-resource setting.

Encoder comparisons In parallel to our work, Pearce et al. (2021) compare different Transformer models on extractive question answering and, similar to our results, find RoBERTa to perform best, outperforming BERT. However, they did not reproduce the strong performance we achieved with ALBERT and, unlike our results, found XLNet to be consistently outperforming BERT. Cortiz (2021) compare Transformer models for text-based emotion recognition and also found RoBERTa to perform best with XLNet being (shared) second, again outperforming BERT.

There are several studies that investigate the performance and transferability of PLM representations that have been fine-tuned with task-specific NER data (Pires et al., 2019; Wu and Dredze, 2020; Adelani et al., 2021; Ebrahimi and Kann, 2021; Ács et al., 2021). For example, Wu and Dredze (2020) analyze multilingual mBERT representations, with a focus on low-resource languages, i.e. languages that are not well represented in the original mBERT training data. They observe that mBERT's NER performance is worse for very high- and very low-resource languages, and that performance drops significantly with less pretraining and supervised data. Adelani et al. (2021) find that fine-tuned XLM-R-large representations outperform fine-tuned mBERT representations in 7 of 10 evaluated African languages, which they attribute to the larger pretraining data size of XLM-R. Ebrahimi and Kann (2021) find that continued pretraining with Bible data from over 1600 languages improves zero-shot NER performance of XLM-R representations.

Our work can also be viewed as a kind of probing task (Conneau et al., 2018; Belinkov and Glass, 2019; Tenney et al., 2019; Petroni et al., 2019; Kassner et al., 2021), since we analyze how much information about named entities is preserved in the pre-trained representations, as measured by a linear classifier.

6 Conclusion

We presented a systematic, comparative study of pre-trained encoders on the task of low-resource named entity recognition. We find that encoder performance varies significantly depending on the scenario and the mix of pre-training and fine-tuning strategies. This suggests that the choice of encoders for a particular setting in current state-of-the-art low-resource NER approaches may need to be carefully (re-)evaluated. We also find that PLM encoders achieve reasonably good token classification performance on many English and German NER datasets with as little as 10 examples per class, in a fine-tuning-free setting. In particular, ALBERT turned out to be a very strong contender in one-shot settings, whereas RoBERTa often outperforms other PLMs in settings with more examples. For German, BERT shows the best average performance across scenarios, with XLM-R being more useful in one-shot settings.

One obvious direction for future work is to evaluate additional encoders, in particular models that are pre-trained in an entity-aware manner (Peters et al., 2019; Zhang et al., 2019), and PLMs for low-resource languages that are trained on much smaller corpora or underrepresented in multilingual PLMs. While our analysis is limited to NER, another future direction would be to adapt the encoder-readout framework in order to evaluate other low-resource classification tasks.

Acknowledgments

We would like to thank Nils Feldhus, David Harbecke, and the anonymous reviewers for their valuable comments and feedback on the paper. This work has been supported by the German Federal Ministry for Economic Affairs and Climate Action as part of the project PLASS (01MD19003E), and by the German Federal Ministry of Education and Research as part of the project CORA4NLP (01IW20010). Christoph Alt is supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2002/1 "Science of Intelligence" – project number 390523135.

References

Judit Ács, Dániel Lévai, and Andras Kornai. 2021. Evaluating transferability of BERT models on uralic languages. In *Proceedings of the Seventh International Workshop on Computational Linguistics of Uralic Languages*, pages 8–17, Syktyvkar, Russia (Online). Association for Computational Linguistics.

David Ifeoluwa Adelani, Jade Abbott, Graham Neubig, Daniel D'souza, Julia Kreutzer, Constantine Lignos, Chester Palen-Michel, Happy Buzaaba, Shruti Rijhwani, Sebastian Ruder, Stephen Mayhew, Israel Abebe Azime, Shamsuddeen H. Muhammad, Chris Chinenye Emezue, Joyce Nakatumba-Nabende, Perez Ogayo, Aremu Anuoluwapo, Catherine Gitau, Derguene Mbaye, Jesujoba Alabi, Seid Muhie Yimam, Tajuddeen Rabiu Gwadabe, Ignatius Ezeani, Rubungo Andre Niyongabo, Jonathan Mukiibi, Verrah Otiende, Iroro Orife, Davis David, Samba Ngom, Tosin Adewumi, Paul Rayson, Mofetoluwa Adeyemi, Gerald Muriuki, Emmanuel Anebi, Chiamaka Chukwuneke, Nkiruka Odu, Eric Peter Wairagala, Samuel Oyerinde, Clemencia Siro, Tobius Saul Bateesa, Temilola Oloyede, Yvonne Wambui, Victor Akinode, Deborah Nabagereka, Maurice Katusiime, Ayodele Awokoya, Mouhamadane MBOUP, Dibora Gebreyohannes, Henok Tilaye, Kelechi Nwaike, Degaga Wolde, Abdoulaye Faye, Blessing Sibanda, Orevaoghene Ahia, Bonaventure F. P. Dossou, Kelechi Ogueji, Thierno Ibrahima DIOP, Abdoulaye Diallo, Adewale Akinfaderin, Tendai Marengereke, and Salomey Osei. 2021. MasakhaNER: Named entity recognition for African languages. *Transactions of the Association for Computational Linguistics*, 9:1116–1131.

Dominic Balasuriya, Nicky Ringland, Joel Nothman, Tara Murphy, and James R. Curran. 2009. Named entity recognition in Wikipedia. In *Proceedings of the 2009 Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources (People's Web)*, pages 10–18, Suntec, Singapore. Association for Computational Linguistics.

Yonatan Belinkov and James Glass. 2019. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72.

Darina Benikova, Chris Biemann, and Marc Reznicek. 2014. NoSta-D named entity annotation for German: Guidelines and dataset. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2524–2531, Reykjavik, Iceland. European Language Resources Association (ELRA).

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Sumit Chopra, Raia Hadsell, and Yann LeCun. 2005. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546. IEEE.

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.

Alexis Conneau, Germán Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single \backslash$&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

Diogo Cortiz. 2021. Exploring transformers in emotion recognition: a comparison of bert, distillbert, roberta, xlnet and electra. *CoRR*, abs/2104.02041.

Sarkar Snigdha Sarathi Das, Arzoo Katiyar, Rebecca J. Passonneau, and Rui Zhang. 2021. Container: Few-shot named entity recognition via contrastive learning. *CoRR*, abs/2109.07589.

Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. 2017. Results of the WNUT2017 shared task on novel and emerging entity recognition. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 140–147, Copenhagen, Denmark. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ning Ding, Guangwei Xu, Yulin Chen, Xiaobin Wang, Xu Han, Pengjun Xie, Haitao Zheng, and Zhiyuan Liu. 2021. Few-NERD: A few-shot named entity recognition dataset. 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 3198–3213, Online. Association for Computational Linguistics.

Abteen Ebrahimi and Katharina Kann. 2021. How to adapt your pretrained multilingual model to 1600 languages. 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 4555–4567, Online. Association for Computational Linguistics.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 363–370, Stroudsburg, PA, USA. Association for Computational Linguistics.

Alexander Fritzler, Varvara Logacheva, and Maksim Kretov. 2019. Few-shot classification in Named Entity Recognition Task. *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing - SAC '19*, pages 993–1000. ArXiv: 1812.06158.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. 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 3816–3830, Online. Association for Computational Linguistics.

Jiaxin Huang, Chunyuan Li, Krishan Subudhi, Damien Jose, Shobana Balakrishnan, Weizhu Chen, Baolin Peng, Jianfeng Gao, and Jiawei Han. 2021. Few-shot named entity recognition: An empirical baseline study. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10408–10423, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Nora Kassner, Philipp Dufter, and Hinrich Schütze. 2021. Multilingual LAMA: Investigating knowledge in multilingual pretrained language models. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3250–3258, Online. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

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. *CoRR*, abs/1907.11692.

Ruotian Ma, Xin Zhou, Tao Gui, Yiding Tan, Qi Zhang, and Xuanjing Huang. 2021. Template-free prompt tuning for few-shot ner. *CoRR*, abs/2109.13532.

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.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Kate Pearce, Tiffany Zhan, Aneesh Komanduri, and Justin Zhan. 2021. A comparative study of transformer-based language models on extractive question answering. *CoRR*, abs/2110.03142.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. 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 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge enhanced contextual word representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 43–54, Hong Kong, China. Association for Computational Linguistics.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Clifton Poth, Jonas Pfeiffer, Andreas Rücklé, and Iryna Gurevych. 2021. What to pre-train on? Efficient intermediate task selection. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10585–10605, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Nils Rethmeier and Isabelle Augenstein. 2021. A primer on contrastive pretraining in language processing: Methods, lessons learned and perspectives. *CoRR*, abs/2102.12982.

Raphael Scheible, Fabian Thomczyk, Patric Tippmann, Victor Jaravine, and Martin Boeker. 2020. Gottbert: a pure german language model. *CoRR*, abs/2012.02110.

Martin Schiersch, Veselina Mironova, Maximilian Schmitt, Philippe Thomas, Aleksandra Gabryszak, and Leonhard Hennig. 2018. A German Corpus for Fine-Grained Named Entity Recognition and Relation Extraction of Traffic and Industry Events. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R. Thomas McCoy, Najoung Kim, Benjamin Van Durme, Sam Bowman, Dipanjan Das, and Ellie Pavlick. 2019. What do you learn from context? Probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Tu Vu, Tong Wang, Tsendsuren Munkhdalai, Alessandro Sordoni, Adam Trischler, Andrew Mattarella-Micke, Subhransu Maji, and Mohit Iyyer. 2020. Exploring and predicting transferability across NLP tasks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7882–7926, Online. Association for Computational Linguistics.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. 2013. Ontonotes release 5.0 ldc2013t19. *Linguistic Data Consortium, Philadelphia, PA*, 23.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. 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 1112–1122. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Shijie Wu and Mark Dredze. 2020. Are all languages created equal in multilingual BERT? In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 120–130, Online. Association for Computational Linguistics.

Omry Yadan. 2019. Hydra - a framework for elegantly configuring complex applications. Github.

Yi Yang and Arzoo Katiyar. 2020. Simple and effective few-shot named entity recognition with structured nearest neighbor learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6365–6375, Online. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Hanchu Zhang, Leonhard Hennig, Christoph Alt, Changjian Hu, Yao Meng, and Chao Wang. 2020. Bootstrapping named entity recognition in E-commerce with positive unlabeled learning. In *Proceedings of The 3rd Workshop on e-Commerce and NLP*, pages 1–6, Seattle, WA, USA. Association for Computational Linguistics.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy. Association for Computational Linguistics.

A Additional Training Details

We used a single RTXA6000-GPU for all experiments. The average runtime per scenario (dataset, encoder) for 600 episodes was approximately 1 minute (1-shot), 3 minutes (5-shot) and 6 minutes (10-shot). Constrastive pre-training was also performed on the same single RTXA6000-GPU, and took approximately 1 hour of GPU-time, including hyperparameter search.

For contrastive pre-training, the following hyperparameters were manually tuned: learning rate in $[2 \times 10^{-5}, 5 \times 10^{-5}]$, the number of epochs in $[1, 2, 5]$. We used the most occurrences of F1-gains across all encoders and scenarios on CoNLL-2003 dataset as criterion for hyperparameter selection.

All pre-trained models evaluated in this study were used as they are available from HuggingFace's model hub, without any modifications. Table 7 lists the model identifiers. We used HuggingFace's dataset hub for all datasets except the dataset by Zhang et al. (2020), which is used here with the permission of the authors.

Model	HuggingFace ID
BERT↓	bert-base-uncased
BERT↑	bert-base-cased
ALBERT	albert-base-v2
RoBERTa	roberta-base
SpanBERT	SpanBERT/spanbert-base-cased
XLNET	xlnet-base-cased
BERT DE	bert-base-german-cased
GottBERT	uklfr/gottbert-base
XLM-R	xlm-roberta-base
BERT_{POS}	vblagoje/bert-english-uncased-finetuned-pos
$\text{BERT}_{\text{MNLI}}$	textattack/bert-base-uncased-MNLI
$\text{BERT}_{\text{SQuAD}}$	csarron/bert-base-uncased-squad-v1
$\text{BERT}_{\text{CoNLL}}$	dslim/bert-base-NER-uncased

Table 7: HuggingFace model identifiers of evaluated encoders

B Readout approaches

Logistic Regression (LR) is a linear classification algorithm that can be extended to multinomial logistic regression to deal with multi-class (N-way) settings, such as the one discussed here. The probability that query token x' belongs to the c-th class is given by:

$$\begin{aligned} \Pr(y = c) &= \frac{\text{score}(x', c)}{\sum_{i=1}^{N} \text{score}(x', i)} \\ \text{score}(x', i) &:= \exp(W_i \cdot f_{\boldsymbol{\theta}}(x')), \end{aligned} \tag{1}$$

where W is a matrix of N rows learned from the support set $\mathcal{S}$, and W_i denotes the i-th row of W. score$(\cdot)$ serves as the metric to measure the affinity between token x' and the prototype of class c, and the prediction is given by

$$y^* = \arg \max_{c \in \{1, \cdots, N\}} \text{score}(x', c).$$

k-Nearest Neighbor (NN) is a non-parametric classification method adopted in metric space. As proposed in StructShot (Yang and Katiyar, 2020), we set $k = 1$ to find the exact nearest token in the support set. Given a query token x',

$$\begin{aligned} y^* &= \arg \min_{c \in \{1, \cdots, N\}} d_c(x') \\ d_c(x') &:= \min_{x \in \mathcal{S}_c} d\big(f_{\boldsymbol{\theta}}(x'), f_{\boldsymbol{\theta}}(x)\big), \end{aligned} \tag{2}$$

where $\mathcal{S}_c$ is the set of support tokens whose tags are c, and d denotes the distance between two embeddings in the representation space.

Nearest Centroid (NC) works similar to **NN**. In contrast, for each query token x', instead of computing the distance between $f_{\boldsymbol{\theta}}(x')$ and every instance in the embedding space, we represent each class by the centroid $\boldsymbol{c}_c$ of all embeddings belonging to this class, and assign token x' to the class with the nearest centroid:

$$\begin{aligned} y^* &= \arg \min_{c \in \{1, \cdots, N\}} d\big(f_{\boldsymbol{\theta}}(x'), \boldsymbol{c}_c\big) \\ \boldsymbol{c}_c &= \frac{1}{|\mathcal{S}_c|} \sum_{x \in \mathcal{S}_c} f_{\boldsymbol{\theta}}(x). \end{aligned} \tag{3}$$

C Entity tag sets of English datasets

We list the full entity tag sets for all English benchmarks. Overlap entity tags with $\text{CoNLL-2003}_{\text{EN}}$ are highlighted with underline.

C.1 $\text{CoNLL-2003}_{\text{EN}}$

<u>LOC</u>, <u>MISC</u>, <u>ORG</u>, <u>PER</u>.

C.2 OntoNotes 5.0

CARDINAL, DATE, EVENT, FAC, GPE, LANGUAGE, LAW, <u>LOC</u>, MONEY, NORP, ORDINAL, <u>ORG</u>, PERCENT, <u>PERSON</u>, PRODUCT, QUANTITY, TIME, WORK_OF_ART.

C.3 $\text{Few-NERD}_{\text{coarse}}$

art, building, event, <u>location</u>, <u>organization</u>, <u>other</u>[6], <u>person</u>, product.

[6] $\text{Few-NERD}_{\text{coarse}}$ sets non-entity as 'O' and various entity types as 'other'. Therefore, we treat 'other' as 'MISC' in this case.

C.4 Few-NERD$_{\text{fine}}$

art-broadcastprogram, art-film, art-music, art-other, art-painting, art-writtenart, building-airport, building-hospital, building-hotel, building-library, building-other, building-restaurant, building-sportsfacility, building-theater, event-attack/battle/war/militaryconflict, event-disaster, event-election, event-other, event-protest, event-sportsevent, location-GPE, location-bodiesofwater, location-island, location-mountain, location-other, location-park, location-road/railway/highway/transit, organization-company, organization-education, organization-government/governmentagency, organization-media/newspaper, organization-other, organization-politicalparty, organization-religion, organization-showorganization, organization-sportsleague, organization-sportsteam, other-astronomything, other-award, other-biologything, other-chemicalthing, other-currency, other-disease, other-educationaldegree, other-god, other-language, other-law, other-livingthing, other-medical, person-actor, person-artist/author, person-athlete, person-director, person-other, person-politician, person-scholar, person-soldier, product-airplane, product-car, product-food, product-game, product-other, product-ship, product-software, product-train, product-weapon

C.5 WNUT-17

corporation, creative-work, group, location, person, product.

C.6 WikiAnn

LOC, ORG, PER.

C.7 WikiGold

LOC, MISC, ORG, PER.

C.8 Zhang et al.

ATTRIBUTE, BRAND, COMPONENT, PRODUCT.