T-NER: An All-Round Python Library for Transformer-based Named Entity Recognition

Asahi Ushio and **Jose Camacho-Collados**
School of Computer Science and Informatics
Cardiff University, United Kingdom
{ushioa,camachocolladosj}@cardiff.ac.uk

Abstract

Language model (LM) pretraining has led to consistent improvements in many NLP downstream tasks, including named entity recognition (NER). In this paper, we present **T-NER**[1] (**T**ransformer-based **N**amed **E**ntity **R**ecognition), a Python library for NER LM finetuning. In addition to its practical utility, T-NER facilitates the study and investigation of the cross-domain and cross-lingual generalization ability of LMs finetuned on NER. Our library also provides a web app where users can get model predictions interactively for arbitrary text, which facilitates qualitative model evaluation for non-expert programmers. We show the potential of the library by compiling nine public NER datasets into a unified format and evaluating the cross-domain and cross-lingual performance across the datasets. The results from our initial experiments show that in-domain performance is generally competitive across datasets. However, cross-domain generalization is challenging even with a large pretrained LM, which has nevertheless capacity to learn domain-specific features if finetuned on a combined dataset. To facilitate future research, we also release all our LM checkpoints via the Hugging Face model hub[2]

1 Introduction

Language model (LM) pretraining has become one of the most common strategies within the natural language processing (NLP) community to solve downstream tasks (Peters et al., 2018; Howard and Ruder, 2018; Radford et al., 2018, 2019; Devlin et al., 2019). LMs trained over large textual data only need to be finetuned on downstream tasks to outperform most of the task-specific designed models. Among the NLP tasks impacted by LM pretraining, named entity recognition (NER) is one of the most prevailing and practical applications. However, the availability of open-source NER libraries for LM training is limited.[3]

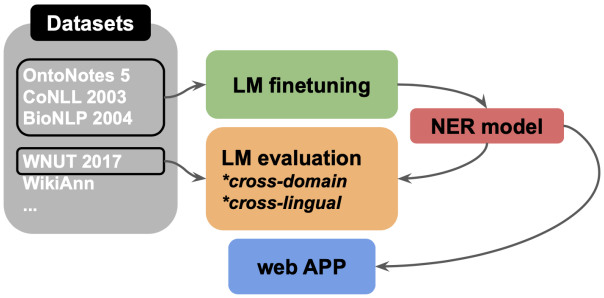

Figure 1: System overview of T-NER.

In this paper, we introduce **T-NER**, an open-source Python library for cross-domain analysis for NER with pretrained Transformer-based LMs. Figure 1 shows a brief overview of our library and its functionalities. The library facilitates NER experimental design including easy-to-use features such as model training and evaluation. Most notably, it enables to organize cross-domain analyses such as training a NER model and testing it on a different domain, with a small configuration. We also report initial experiment results, by which we show that although cross-domain NER is challenging, if it has an access to new domains, LM can successfully learn new domain knowledge. The results give us an insight that LM is capable to learn a variety of domain knowledge, but an ordinary finetuning scheme on single dataset most likely causes overfitting and results in poor domain generalization.

As a system design, T-NER is implemented in Pytorch (Paszke et al., 2019) on top of the Transformers library (Wolf et al., 2019). Moreover, the

[1] https://github.com/asahi417/tner
[2] https://huggingface.co/models?search=asahi417/tner.
[3] Recently, spaCy (https://spacy.io/) has released a general NLP pipeline with pretrained models including a NER feature. Although it provides a very efficient pipeline for processing text, it is not suitable for LM finetuning or evaluation on arbitrary NER data.

interfaces of our training and evaluation modules are highly inspired by Scikit-learn (Pedregosa et al., 2011), enabling an interoperability with recent models as well as integrating them in an intuitive way. In addition to the versatility of our toolkit for NER experimentation, we also include an online demo and robust pre-trained models trained across domains. In the following sections, we provide a brief overview about NER in Section 2, explain the system architecture of T-NER with a few basic usages in Section 3 and describe experiment results on cross-domain transfer with our library in Section 4.

2 Named Entity Recognition

Given an arbitrary text, the task of NER consists of detecting named entities and identifying their type. For example, given a sentence *"Dante was born in Florence."*, a NER model are would identify *"Dante"* as a person and *"Florence"* as a location. Traditionally, NER systems have relied on a classification model on top of hand-engineered feature sets extracted from corpora (Ratinov and Roth, 2009; Collobert et al., 2011), which was improved by carefully designed neural network approaches (Lample et al., 2016; Chiu and Nichols, 2016; Ma and Hovy, 2016). This paradigm shift was mainly due to its efficient access to contextual information and flexibility, as human-crafted feature sets were no longer required. Later, contextual representations produced by pretrained LMs have improved the generalization abilities of neural network architectures in many NLP tasks, including NER (Peters et al., 2018; Devlin et al., 2019).

In particular, LMs *see* millions of plain texts during pretraining, a knowledge that then can be leveraged in downstream NLP applications. This property has been studied in the recently literature by probing their generalization capacity (Hendrycks et al., 2020; Aharoni and Goldberg, 2020; Desai and Durrett, 2020; Gururangan et al., 2020). When it comes to LM generalization studies in NER, the literature is more limited and mainly restricted to in-domain (Agarwal et al., 2021) or multilingual settings (Pfeiffer et al., 2020a; Hu et al., 2020b). Our library facilitates future research in cross-domain and cross-lingual generalization by providing a unified benchmark for several languages and domain as well as a straightforward implementation of NER LM finetuning.

3 T-NER: An Overview

A key design goal was to create a self-contained universal system to train, evaluate, and utilize NER models in an easy way, not only for research purpose but also practical use cases in industry. Our package, T-NER, allows practitioners in NLP to get started working on NER with a few lines of code while diving into the recent progress in LM fine-tuning. We employ Python as our core implementation, as is one of the most prevailing languages in the machine learning and NLP communities. Our library enables Python users to access its various kinds of features such as model training, in- and cross-domain model evaluation, and an interface to get predictions from trained models with minimum effort. Moreover, we provide a demo web app (Figure 2) where users can get predictions from a trained model given a sentence interactively. This way, users (even those without programming experience) can conduct qualitative analyses on their own or existing pre-trained models. In the following we provide details on the technicalities of the package provided, including details on how to train and evaluate any LM-based architecture.

3.1 Datasets

For model training and evaluation, we compiled nine public NER datasets from different domains, unifying them into same format: OntoNotes5 (Hovy et al., 2006), CoNLL 2003 (Tjong Kim Sang and De Meulder, 2003), WNUT 2017 (Derczynski et al., 2017), WikiAnn (Pan et al., 2017), FIN (Salinas Alvarado et al., 2015), BioNLP 2004 (Collier and Kim, 2004), BioCreative V CDR[4] (Wei et al., 2015), MIT movie review semantic corpus,[5] and MIT restaurant review.[6] These unified datasets are also made available as part of our T-NER library.

Table 1 shows statistics of each dataset. Except for WikiAnn that contains 282 languages, all the datasets are in English, and only the MIT corpora are lowercased. As MIT corpora are commonly used for slot filling task in spoken language understanding (Liu and Lane, 2017), the characteristics

[4]The original dataset consists of long documents which cannot be fed on LM because of the length, so we split them into sentences to reduce their size.

[5]The movie corpus includes two datasets (*eng* and *trivia10k13*) coming from different data sources. While both have been integrated into our library, we only used the largest *trivia10k13* in our experiments.

[6]The original MIT NER corpora can be downloaded from https://groups.csail.mit.edu/sls/downloads/.

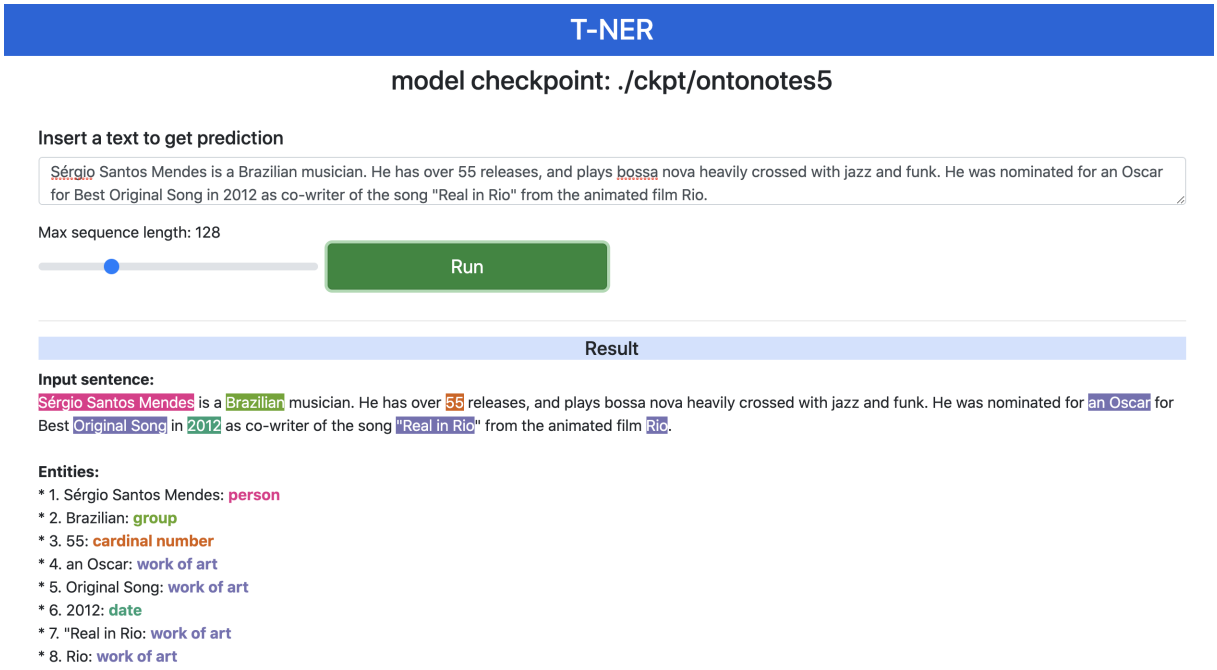

Figure 2: A screenshot from the demo web app. Here the model is trained on OntoNotes 5 and an example sentence is fetched from Wikipedia (https://en.wikipedia.org/wiki/Sergio_Mendes).

of the entities and annotation guidelines are quite different from the other datasets, but we included them for completeness and to analyze the differences across datasets. In Section 4, we train models on each dataset, and assess the in- and cross-domain accuracy over them.

Dataset format and customization. Users can utilize their own datasets for both model training and evaluation by formatting them into the IOB scheme (Tjong Kim Sang and De Meulder, 2003) which we used to unify all datasets. In the IOB format, all data files contain one word per line with empty lines representing sentence boundaries. At the end of each line there is a tag which states whether the current word is inside a named entity or not. The tag also encodes the type of named entity. Here is an example from CoNLL 2003:

```
EU B-ORG
rejects O
German B-MISC
call O
to O
boycott O
British B-MISC
lamb O
. O
```

3.2 Model Training

We provide modules to facilitate LM finetuning on any given NER dataset. Following Devlin et al. (2019), we add a linear layer on top of the last embedding layer in each token, and train all weights with cross-entropy loss. The model training component relies on the Huggingface transformers library (Wolf et al., 2019), one of the largest Python frameworks for distributing pretrained LM checkpoint files. Our library is therefore fully compatible with the Transformers framework: once new model was deployed on the Transformer hub, one can immediately try those models out with our library as a NER model. To reduce computational complexity, in addition to enabling multi-GPU support, we implement mixture precision during model training by using the apex library[7].

The instance of model training in a given dataset[8] can be used in an intuitive way as displayed below:

```
from tner import TrainTransformersNER
model = TrainTransformersNER(
        dataset="ontonotes5",
        transformer="roberta-base")
model.train()
```

In this sample code, we would finetune *RoBERTa*$_{BASE}$ (Liu et al., 2019) on the OntoNotes5 dataset. To train on multiple datasets at the same time, we provide an easy extension as follows:

[7] https://github.com/NVIDIA/apex

[8] To use custom datasets, the path to a custom dataset folder can simply be included in the dataset argument.

Name	Domain	Entity types	Data size
OntoNotes5	News, Blog, Dialogue	18	59,924/8,582/8,262
CoNLL 2003	News	4	14,041/3,250/3,453
WNUT 2017	SNS	6	1,000/1,008/1,287
WikiAnn	Wikipedia (282 languages)	3	20,000/10,000/10,000
FIN	Finance	4	1,164/-/303
BioNLP 2004	Biochemical	5	18,546/-/3,856
BioCreative V	Biomedical	5	5,228/5,330/5,865
MIT Restaurant	Restaurant review	8	7,660/-/1,521
MIT Movie	Movie review	12	7,816/-/1,953

Table 1: Overview of the NER datasets used in our evaluation and included in T-NER. Data size is the number of sentence in training/validation/test set.

```
TrainTransformersNER(
    dataset=[
        "ontonotes5", "wnut2017"
        ],
    transformer="roberta-base")
```

Once training is completed, checkpoint files with model weights and other statistics are generated. These are automatically organized for each configuration and can be easily uploaded to the Hugging Face model hub. Ready-to-use code samples can be found in our Google Colab notebook[9], and details for additional options and arguments are included in the github repository. Finally, our library supports Tensorboard[10] to visualize learning curves.

3.3 Model Evaluation

Once a NER model is trained, users may want to test the models in the same dataset or a different one to assess its general performance across domains. To this end, we implemented flexible evaluation modules to facilitate cross-domain evaluation comparison, which is also aided by the unification of datasets into the same format (see Section 3.1) with a unique label reference lookup.

The basic usage of the evaluation module is described below.

```
from tner import TrainTransformersNER
model = TrainTransformersNER(
    "path-to-model-checkpoint"
    )
model.test("ontonotes5")
```

Here, the model would be tested on OntoNotes5 dataset, and it could be evaluated on any other test set including custom dataset. As with the model training module, we prepared a Google Colab notebook[11] for an example use case, and further details can be found in our github repository.

4 Evaluation

In this section, we assess the reliability of T-NER with experiments in standard NER datasets.

4.1 Experimental Setting

4.1.1 Implementation details

Through the experiments, we use *XLM-R* (Liu et al., 2019), which has shown to be one of the most reliable multi-lingual pretrained LMs for discriminative tasks at the moment. In all experiments we make use of the default configuration and hyperpameters of Huggingface's *XLM-R* implementation. For WikiAnn/ja (Japanese), we convert the original character-level tokenization into proper morphological chunk by MeCab[12].

4.1.2 Evaluation metrics and protocols

As customary in the NER literature, we report *span micro-F1 score* computed by seqeval[13], a Python library to compute metrics for sequence prediction evaluation. We refer to this F1 score as *type-aware* F1 score to distinguish it from the the type-ignored metric used to assess the cross-domain performance, which we explain below.

In a cross-domain evaluation setting, the *type-aware* F1 score easily fails to represent the cross-domain performance if the granularity of entity types differ across datasets. For instance, the MIT restaurant corpus has entities such as *amenity* and

[9] https://colab.research.google.com/drive/1AlcTbEsp8W11yflT7SyT0L4C4HG6MXYr?usp=sharing

[10] www.tensorflow.org/tensorboard

[11] https://colab.research.google.com/drive/1jHVGnFN4AU8uS-ozWJIXXe2fV8HUj8NZ?usp=sharing

[12] https://pypi.org/project/mecab-python3/

[13] https://pypi.org/project/seqeval/

Dataset	*BASE*	*LARGE*	SoTA
OntoNotes5	89.0	89.1	92.1
CoNLL 2003	90.8	92.9	94.3
WNUT 2017	52.8	58.5	50.3
FIN	81.3	76.4	82.7
BioNLP 2004	73.4	74.3	77.4
BioCreative V	88.0	88.6	89.9
MIT Restaurant	79.4	79.6	-
MIT Movie	69.9	71.2	-
WikiAnn/en	82.7	84.0	84.8
WikiAnn/ja	83.8	86.5	73.3
WikiAnn/ru	88.6	90.0	91.4
WikiAnn/es	90.9	92.1	-
WikiAnn/ko	87.5	89.6	-
WikiAnn/ar	88.9	90.3	-

Table 2: In-domain *type-aware* F1 score for test set on each dataset with current SoTA. SoTA on each dataset is attained from the result of *BERT-MRC-DSC* (Li et al., 2019) for OntoNotes5, *LUKE* (**?**) for CoNLL 2003, *CrossWeigh* (Wang et al., 2019) for WNUT 2017, (Pfeiffer et al., 2020a) for WikiAnn (en, ja, ru, es, ko, ar), (Salinas Alvarado et al., 2015) for FIN, (Lee et al., 2020) for BioNLP 2004, (Nooralahzadeh et al., 2019) for BioCreative V and (Pfeiffer et al., 2020a) for WikiAnn/en.

rating, while *plot* and *actor* are entities from the MIT movie corpus. Thus, we report *type-ignored* F1 score for cross-domain analysis. In this *type-ignored* evaluation, the entity type from both of predictions and true labels is disregarded, reducing the task into a simpler entity span detection task. This evaluation protocol can be customized by the user at test time.

4.2 Results

We conduct three experiments on the nine datasets described in Table 1: (i) in-domain evaluation (Section 4.2.1), (ii) cross-domain evaluation (Section 4.2.2), and (iii) cross-lingual evaluation (Section 4.2.3). While the first experiment tests our implementation in standard datasets, the second experiment is aimed at investigating the cross-domain performance of transformer-based NER models. Finally, as a direct extension of our evaluation module, we show the zero-shot cross-lingual performance of NER models on the WikiAnn dataset.

4.2.1 In-domain results

The main results are displayed in Table 2, where we report the *type-aware* F1 score from *XLM-R*$_{BASE}$ and *XLM-R*$_{LARGE}$ models along with current state-of-the-art (SoTA). One can confirm that our framework with *XLM-R*$_{LARGE}$ achieves a comparable SoTA score, even surpassing it in the WNUT 2017 dataset. In general, *XLM-R*$_{LARGE}$ performs consistently better than *XLM-R*$_{BASE}$ but, interestingly, the base model performs better than large on the FIN dataset. This can be attributed to the limited training data in this dataset, which may have caused overfitting in the large model.

Generally, it can be expected to get better accuracy with domain-specific or larger language models that can be integrated into our library. Nonetheless, our goal for these experiments were not to achieve SoTA but rather to provide a competitive and easy-to-use framework. In the remaining experiments we report results for *XLM-R*$_{LARGE}$ only, but the results for *XLM-R*$_{BASE}$ can be found in the appendix.

4.2.2 Cross-domain results

In this section, we show cross-domain evaluation results on the English datasets[14]: OntoNotes5 (ontonotes), CoNLL 2003 (conll), WNUT 2017 (wnut), WikiAnn/en (wiki), BioNLP 2004 (bionlp), and BioCreative V (bc5cdr), FIN (fin). We also report the accuracy of the same XLM-R model trained over a combined dataset resulting from concatenation of all the above datasets.

In Table 3, we present the *type-ignored* F1 results across datasets. Overall cross-domain scores are not as competitive as in-domain results. This gap reveals the difficulty of transferring NER models into different domains, which may also be attributed to different annotation guidelines or data construction procedures across datasets. Especially, training on the bionlp and bc5cdr datasets lead to a null accuracy when they are evaluated on other datasets, as well as others evaluated on them. Those datasets are very domain specific dataset, as they have entities such as *DNA*, *Protein*, *Chemical*, and *Disease*, which results in a poor adaptation to other domains. On the other hand, there are datasets that are more easily transferable, such as wnut and conll. The wnut-trained model achieves 85.7 on the conll dataset and, surprisingly, the conll-trained model actually works better than the wnut-trained model when evaluated on the wnut test set. This could be also attributed to the data size, as wnut only has 1,000 sentences, while conll has 14,041. Nevertheless, the fact that ontonotes has 59,924

[14]We excluded the MIT datasets in this setting since they are all lowercased.

train\test	ontonotes	conll	wnut	wiki	bionlp	bc5cdr	fin	avg
ontonotes	**91.6**	65.4	53.6	47.5	0.0	0.0	18.3	40.8
conll	62.2	**96.0**	69.1	61.7	0.0	0.0	22.7	35.1
wnut	41.8	85.7	**68.3**	54.5	0.0	0.0	20.0	31.7
wiki	32.8	73.3	53.6	**93.4**	0.0	0.0	12.2	29.6
bionlp	0.0	0.0	0.0	0.0	**79.0**	0.0	0.0	8.7
bc5cdr	0.0	0.0	0.0	0.0	0.0	**88.8**	0.0	9.8
fin	48.2	73.2	60.9	58.9	0.0	0.0	**82.0**	38.1
all	90.9	93.8	60.9	91.3	78.3	84.6	75.5	81.7

Table 3: *Type-ignored* F1 score in cross-domain setting over non-lower-cased English datasets. We compute average of accuracy in each test set, named as **avg**. The model trained on all datasets listed here, is shown as **all**.

	test					
train	**en**	**ja**	**ru**	**ko**	**es**	**ar**
en	**84.0**	46.3	73.1	58.1	71.4	53.2
ja	53.0	**86.5**	45.7	57.1	74.5	55.4
ru	60.4	53.3	**90.0**	68.1	76.8	54.9
ko	57.8	62.0	68.6	**89.6**	66.2	57.2
es	70.5	50.6	75.8	61.8	**92.1**	62.1
ar	60.1	55.7	55.7	70.7	79.7	**90.3**

Table 4: Cross-lingual *type-aware* F1 results on various languages for the WikiAnn dataset.

sentences but does not perform better than conll on wnut reveals a certain domain similarity between conll and wnut.

Finally, the model trained on the training sets of all datasets achieves a *type-ignored* F1 score close to the in-domain baselines. This indicates that a LM is capable of learning representations of different domains. Moreover, leveraging domain similarity as explained above can lead to better results as, for example, distant datasets such as bionlp and bc5cdr surely cause performance drops. This is an example of the type of experiments that could be facilited by T-NER, which we leave for future work.

4.2.3 Cross-lingual results

Finally, we present some results for zero-shot cross-lingual NER over the WikiAnn dataset, where we include six distinct languages: English (en), Japanese (ja), Russian (ru), Korean (ko), Spanish (es), and Arabic (ar). In Table 4, we show the cross-lingual evaluation results. The diagonal includes the results of the model trained on the training data of the same target language. There are a few interesting findings. First, we observe a high correlation between Russian and Spanish, which are generally considered to be distant languages and do not share the alphabet. Second, Arabic also transfers well to Spanish which, despite the Arabic (lexical) influence on the Spanish language (Stewart et al., 1999), are still languages from distant families.

Clearly, this is a shallow cross-lingual analysis, but it highlights the possibilities of our library for research in cross-lingual NER. Recently, (Hu et al., 2020a) proposed a compilation of multilingual benchmark tasks including the WikiAnn datasets as a part of it, and *XLM-R* proved to be a strong baseline on multilingual NER. This is in line with the results of Conneau et al. (2020), which showed a high capacity of zero-shot cross-lingual transferability. On this respect, Pfeiffer et al. (2020b) proposed a language/task specific adapter module that can further improve cross-lingual adaptation in NER. Given the possibilities and recent advances in cross-lingual language models in recent years, we expect our library to help practitioners to experiment and test these advances in NER.

5 Conclusion

In this paper, we have presented a Python library to get started with Transformer-based NER models. This paper especially focuses on LM finetuning, and empirically shows the difficulty of cross-domain generalization in NER. Our framework is designed to be as simple as possible so that any level of users can start running experiments on NER on any given dataset. To this end, we have also facilitated the evaluation by unifying some of the most popular NER datasets in the literature, including languages other than English. We believe that our initial experiment results emphasize the importance of NER generalization analysis, for which we hope that our open-source library can help NLP community to convey relevant research in an efficient and accessible way.

Acknowledgements

We would like to thank Dimosthenis Antypas for testing our library and the anonymous reviewers for their useful comments.

References

Oshin Agarwal, Yinfei Yang, Byron C Wallace, and Ani Nenkova. 2021. Entity-switched datasets: An approach to auditing the in-domain robustness of named entity recognition models. *arXiv preprint arXiv:2004.04123*.

Roee Aharoni and Yoav Goldberg. 2020. Unsupervised domain clusters in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7747–7763, Online. Association for Computational Linguistics.

Jason P.C. Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics*, 4:357–370.

Nigel Collier and Jin-Dong Kim. 2004. Introduction to the bio-entity recognition task at JNLPBA. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP)*, pages 73–78, Geneva, Switzerland. COLING.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(ARTICLE):2493–2537.

Alexis Conneau, Shijie Wu, Haoran Li, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Emerging cross-lingual structure in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6022–6034, Online. Association for Computational Linguistics.

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.

Shrey Desai and Greg Durrett. 2020. Calibration of pre-trained transformers. *arXiv preprint arXiv:2003.07892*.

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.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*.

Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. 2020. Pretrained transformers improve out-of-distribution robustness. *arXiv preprint arXiv:2004.06100*.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. OntoNotes: The 90% solution. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 57–60, New York City, USA. Association for Computational Linguistics.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020a. XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *International Conference on Machine Learning (ICML)*.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020b. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization. *arXiv preprint arXiv:2003.11080*.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Xiaoya Li, Xiaofei Sun, Yuxian Meng, Junjun Liang, Fei Wu, and Jiwei Li. 2019. Dice loss for data-imbalanced nlp tasks. *arXiv preprint arXiv:1911.02855*.

Bing Liu and Ian Lane. 2017. Multi-domain adversarial learning for slot filling in spoken language understanding. *arXiv preprint arXiv:1711.11310*.

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. *arXiv preprint arXiv:1907.11692*.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany. Association for Computational Linguistics.

Farhad Nooralahzadeh, Jan Tore Lønning, and Lilja Øvrelid. 2019. Reinforcement-based denoising of distantly supervised ner with partial annotation. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 225–233.

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, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pages 8026–8037.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.

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.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020a. Mad-x: An adapter-based framework for multi-task cross-lingual transfer. *arXiv preprint arXiv:2005.00052*.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020b. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673, Online. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 147–155, Boulder, Colorado. Association for Computational Linguistics.

Julio Cesar Salinas Alvarado, Karin Verspoor, and Timothy Baldwin. 2015. Domain adaption of named entity recognition to support credit risk assessment. In *Proceedings of the Australasian Language Technology Association Workshop 2015*, pages 84–90, Parramatta, Australia.

Miranda Stewart et al. 1999. *The Spanish language today*. Psychology Press.

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.

Zihan Wang, Jingbo Shang, Liyuan Liu, Lihao Lu, Jiacheng Liu, and Jiawei Han. 2019. Crossweigh: Training named entity tagger from imperfect annotations. 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 5157–5166.

Chih-Hsuan Wei, Yifan Peng, Robert Leaman, Allan Peter Davis, Carolyn J Mattingly, Jiao Li, Thomas C Wiegers, and Zhiyong Lu. 2015. Overview of the biocreative v chemical disease relation (cdr) task. In *Proceedings of the fifth BioCreative challenge evaluation workshop*, volume 14.

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. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

A Appendices

In all experiments we make use of the default configuration and hyperpameters of Huggingface's *XLM-R* implementation.

A.1 Cross-lingual Results

In this section, we show cross-lingual analysis on *XLM-R*$_{BASE}$, where the result is shown in Table 5. For these cross-lingual results, we rely on the WikiAnn dataset where zero-shot cross-lingual NER over six distinct languages is conducted: English (en), Japanese (ja), Russian (ru), Korean (ko), Spanish (es), and Arabic (ar).

A.2 Cross-domain Results

In this section, we show a few more results on our cross-domain analysis, which is based on non-lowercased English datasets: OntoNotes5 (ontonotes), CoNLL 2003 (conll), WNUT 2017 (wnut), WikiAnn/en (wiki), BioNLP 2004 (bionlp), and BioCreative V (bc5cdr), and FIN (fin). Table 6 shows the type-aware F1 score of the *XLM-R*$_{LARGE}$ and *XLM-R*$_{BASE}$ models trained on all the datasets. Furthermore, Table 7 shows additional results for *XLM-R*$_{BASE}$ in the type-ignored evaluation.

	test					
train	**en**	**ja**	**ru**	**ko**	**es**	**ar**
en	**82.8**	38.6	65.7	50.4	73.8	44.5
ja	53.8	**83.9**	46.9	60.1	71.3	46.3
ru	51.9	39.9	**88.7**	51.9	66.8	51.0
ko	54.7	51.6	53.3	**87.5**	63.3	52.3
es	65.7	44.0	66.5	54.1	**90.9**	59.4
ar	53.1	49.2	49.4	59.7	73.6	**88.9**

Table 5: Cross-lingual *type-aware* F1 score over WikiAnn dataset with *XLM-R*$_{BASE}$.

	uppercase		**lowercase**	
Datasets	*BASE*	*LARGE*	*BASE*	*LARGE*
ontonotes	85.8	87.8	81.7	85.6
conll	87.2	90.3	82.8	87.6
wnut	49.6	55.1	43.7	51.3
wiki	79.1	82.7	75.2	80.8
bionlp	72.9	74.1	71.7	74.0
bc5cdr	79.4	85.0	78.0	84.2
fin	72.4	72.4	72.4	73.5
restaurant	-	-	76.8	80.9
movie	-	-	67.8	71.8

Table 6: *Type-aware* F1 score across different test sets of models trained on all **uppercase/lowercase** English datasets with *XLM-R*$_{BASE}$ or *XLM-R*$_{LARGE}$.

Cross-domain results with lowercased datasets. In this section, we show cross-domain results on the English datasets including lowercased corpora such as MIT Restaurant (restaurant) and MIT Movie (movie). Since those datasets are lowercasd, we converted all datasets into lowercase. Tables 8 and Table 9 show the *type-ignored* F1 score across models trained on different English datasets including lowercased corpora with *XLM-R*$_{LARGE}$ and *XLM-R*$_{BASE}$, respectively.

train\test	ontonotes	conll	wnut	wiki	bionlp	bc5cdr	fin	avg
ontonotes	**91.8**	62.2	51.7	44.7	0.0	0.0	31.8	40.3
conll	60.5	**95.7**	66.6	60.8	0.0	0.0	33.5	45.3
wnut	41.3	81.3	**63.0**	56.3	0.0	0.0	20.5	37.5
wiki	30.2	71.8	45.3	**92.6**	0.0	0.0	11.5	35.9
bionlp	0.0	0.0	0.0	0.0	**78.5**	0.0	0.0	11.2
bc5cdr	0.0	0.0	0.0	0.0	0.0	**87.5**	0.0	12.5
fin	49.0	73.5	62.2	60.7	0.0	0.0	**82.8**	46.9
all	89.7	92.4	55.8	89.3	78.2	80.0	74.8	80.0

Table 7: *Type-ignored* F1 score in cross-domain setting over non-lower-cased English datasets with *XLM-R*$_{BASE}$. We compute average of accuracy in each test set, named as **avg**. The model trained on all datasets listed here, is shown as **all**.

train\test	ontonotes	conll	wnut	wiki	bionlp	bc5cdr	fin	restaurant	movie	avg
ontonotes	89.3	59.9	50.1	44.7	0.0	0.0	15.1	4.5	88.6	39.1
conll	57.7	94.8	67.0	57.9	0.0	0.0	20.5	23.9	0.0	35.7
wnut	39.8	80.3	61.3	52.3	0.0	0.0	19.5	18.8	0.0	30.2
wiki	28.5	69.7	51.2	92.4	0.0	0.0	12.0	3.0	0.0	28.5
bionlp	0.0	0.0	0.0	0.0	79.0	0.0	0.0	0.0	0.0	8.7
bc5cdr	0.0	0.0	0.0	0.0	0.0	88.9	0.0	0.0	0.0	9.8
fin	46	72.0	61.5	54.8	0.0	0.0	83.0	24.5	0.0	37.9
restaurant	4.6	21.7	22.9	22.3	0.0	0.0	5.4	83.4	0.0	17.8
movie	10.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	73.1	9.3
all	88.5	92.1	58.0	90.0	79.0	84.6	74.5	85.3	74.1	80.7

Table 8: *Type-ignored* F1 score in cross-domain setting over lower-cased English datasets with *XLM-R*$_{LARGE}$. We compute average of accuracy in each test set, named as **avg**. The model trained on all datasets listed here, is shown as **all**.

train\test	ontonotes	conll	wnut	wiki	bionlp	bc5cdr	fin	restaurant	movie	avg
ontonotes	**88.3**	56.7	49.0	41.4	0.0	0.0	11.7	4.2	88.3	37.7
conll	55.1	**93.7**	60.5	56.8	0.0	0.0	20.4	21.9	0.0	34.3
wnut	38.1	73.0	**57.5**	49.1	0.0	0.0	21.1	20.4	0.0	28.8
wiki	26.3	66.5	41.4	**90.9**	0.0	0.0	9.7	7.6	0.0	26.9
bionlp	0.0	0.0	0.0	0.0	**78.7**	0.0	0.0	0.0	0.0	8.7
bc5cdr	0.0	0.0	0.0	0.0	0.0	**88.0**	0.0	0.0	0.0	9.8
fin	41.3	64.4	45.8	57.8	0.0	0.0	**81.5**	22.0	0.0	34.8
restaurant	8.1	19.1	19.6	19.1	0.0	0.0	13.5	**83.6**	0.0	18.1
movie	14.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**73.1**	9.7
all	86.1	89.5	49.9	86.2	76.9	78.8	75.4	82.4	72.2	77.5

Table 9: *Type-ignored* F1 score in cross-domain setting over lower-cased English datasets with *XLM-R*$_{BASE}$. We compute average of accuracy in each test set, named as **avg**. The model trained on all datasets listed here, is shown as **all**.