T2NER: Transformers based Transfer Learning Framework for Named Entity Recognition

Saadullah Amin   Günter Neumann

Department of Language Science and Technology, Saarland University, Saarbrücken
Multilinguality and Language Technology Lab, DFKI GmbH, Saarbrücken
{saadullah.amin, guenter.neumann}@dfki.de

Abstract

Recent advances in deep transformer models have achieved state-of-the-art in several natural language processing (NLP) tasks, whereas named entity recognition (NER) has traditionally benefited from long-short term memory (LSTM) networks. In this work, we present a **T**ransformers based **T**ransfer Learning framework for **N**amed **E**ntity **R**ecognition (T2NER) created in PyTorch for the task of NER with deep transformer models. The framework is built upon the *Transformers* library as the core modeling engine and supports several transfer learning scenarios from sequential transfer to domain adaptation, multi-task learning, and semi-supervised learning. It aims to bridge the gap between the algorithmic advances in these areas by combining them with the state-of-the-art in transformer models to provide a unified platform that is readily extensible and can be used for both the transfer learning research in NER, and for real-world applications. The framework is available at: https://github.com/suamin/t2ner.

1 Introduction

Named entity recognition (NER) is an important task in information extraction, benefiting the downstream applications such as entity linking (Cucerzan, 2007), relation extraction (Culotta and Sorensen, 2004) and question answering (Krishnamurthy and Mitchell, 2015). NER has been a challenging task in NLP due to large variations in entity names and flexibility in how entities are mentioned. These challenges are further enhanced in cross-lingual and cross-domain NER settings, where the added difficulty comes from the difference in text genre and entity names across languages and domains (Jia et al., 2019).

Furthermore, NER models have shown relatively high variance even when trained on the same data (Reimers and Gurevych, 2017). These models generalize poorly when tested on data from different domains and languages, and even more so when they contain unseen entity mentions (Augenstein et al., 2017; Agarwal et al., 2020; Wang et al., 2020). These challenges make transfer learning research an important and well studied area in NER.

Recent successes in transfer learning have mainly come from pre-trained language models (Devlin et al., 2019; Radford et al., 2019) with contextualized word embeddings based on deep transformer models (Vaswani et al., 2017). These models achieve state-of-the-art in several NLP tasks such as named entity recognition, document classification, and question answering. Due to their wide success and the community adoption, successful frameworks like *Transformers* have emerged. In NER, the existing frameworks like NCRF++ (Yang and Zhang, 2018) lack the core infrastructure to support such models directly with state-of-the-art transfer learning algorithms.

In this paper, we present an adaptable and user-friendly development framework for growing research in transfer learning with deep transformer models for NER, with underexplored areas such as semi-supervised learning. This is in contrast to the standard LSTM based approaches which have largely and successfully dominated the NER research. Our framework is aimed to bridge several gaps with core design principles that are discussed in next section.

2 Design Principles

T2NER is divided into several components as shown in Figure 1. The core design principle is to seamlessly integrate the *Transformers* (Wolf et al., 2020) library as the backend for modeling, while extending it to support different transfer learning scenarios with a range of existing algorithms. *Trans-*

Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations, pages 212–220
April 19 - 23, 2021. ©2021 Association for Computational Linguistics

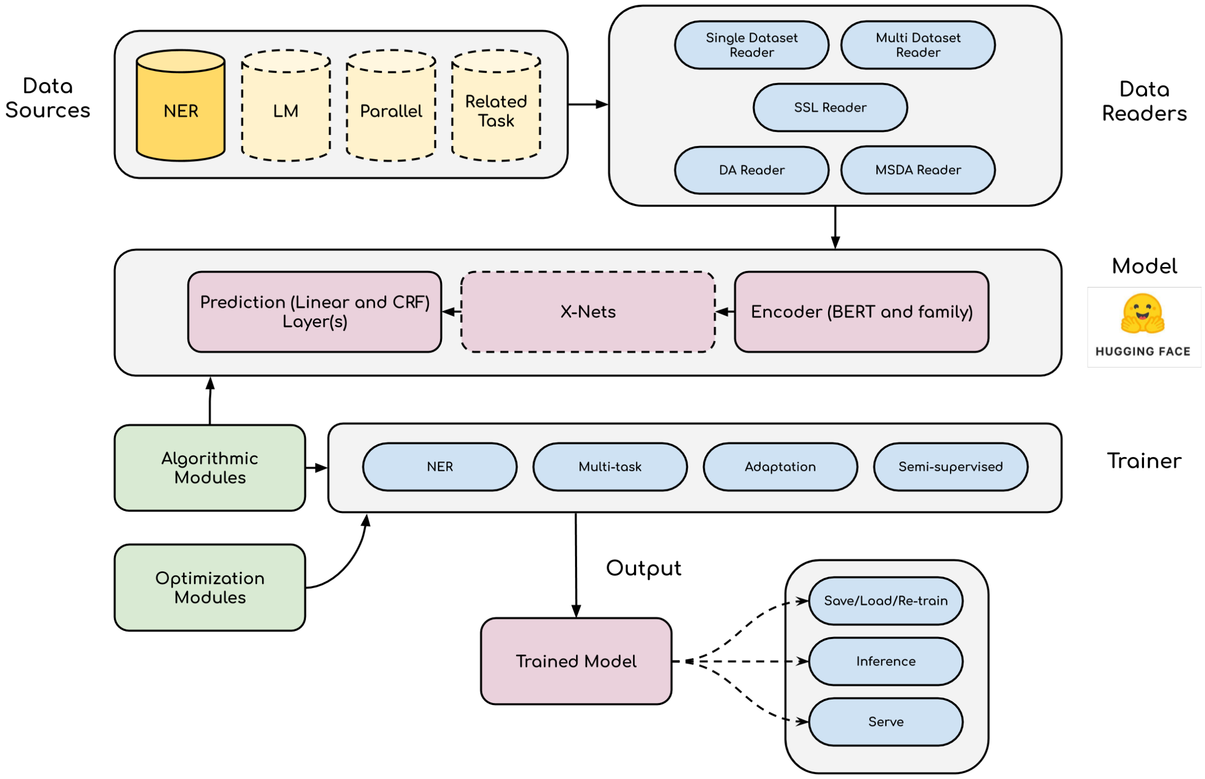


Figure 1: Overview of the T2NER framework.

formers offer optimized implementations of several deep transformer models, including BERT (Devlin et al., 2019), GPT (Radford et al., 2019), RoBERTa (Liu et al., 2019), and XLM (Conneau and Lample, 2019) among others, with multi-GPU, distributed, and mixed precision training.

The second design principle is inspired by previous pre-trained models in the computer vision: `Dassl.pytorch` (Zhou et al., 2020)[1] and `Trans-Learn` (Jiang et al., 2020)[2] that unify domain adaptation, domain generalization, and semi-supervised learning, thus allowing easy benchmarking, fair comparisons, and reproducibility. T2NER is the unification of these major algorithmic approaches to bridge the gap between the algorithms and advance transfer learning research in NER.

Lastly, the cross-lingual and cross-domain research in NER has itself proposed several advances, including multi-task and joint learning (Pan et al., 2017; Peng and Dredze, 2017; Lin et al., 2018; Jia et al., 2019; Wang et al., 2020), adversarial learning (Zhou et al., 2019; Keung et al., 2019), feature transfer (Daumé III, 2007; Kim et al., 2015; Wang et al., 2018), newer architectures (Lin et al., 2018; Jia and Zhang, 2020), parameter sharing (Lee et al., 2018; Yang et al., 2018; Lin and Lu, 2018), parameter generation (Jia et al., 2019), mixture-of-experts (Chen et al., 2018), and usage of external resources (Xie et al., 2018; Wang et al., 2019). Therefore, our final design principle aims to unify these researches and offer a framework to test them with deep transformer models, wherever such an algorithmic abstraction is possible, while exploring new paradigms.

3 The T2NER Framework

3.1 Data Sources

The main data source is the NER data, which is expected to be labeled or unlabeled in the CoNLL format. We adopt widely used BIO tagging scheme. In practice, the differences in results which arise due to different schemes are negligible (Ratinov and Roth, 2009). A simple preprocessing routine is provided to standardize the data files, along with the required metadata, that is used through-

[1] https://github.com/KaiyangZhou/Dassl.pytorch

[2] https://github.com/thuml/Transfer-Learning-Library

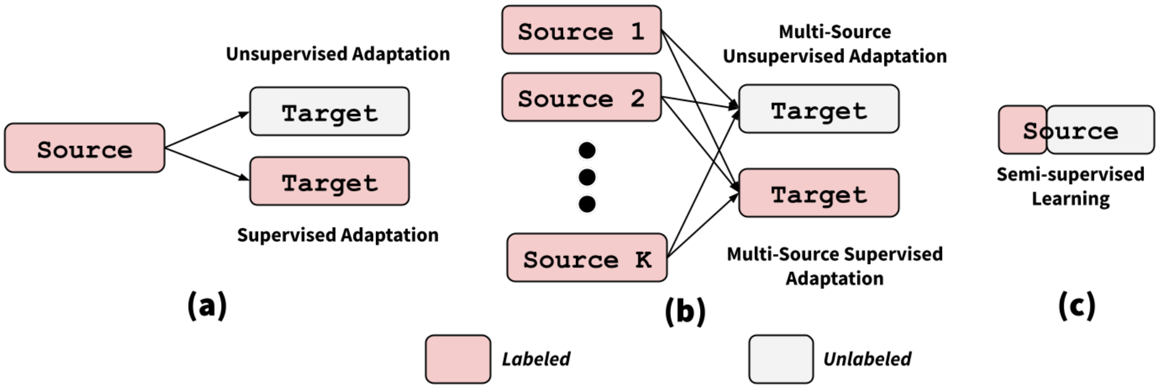


Figure 2: Transfer learning scenarios supported in T2NER. The adaptation scenarios apply to the *cross-domain*, *cross-lingual*, or a *mix* of both. These scenarios can further be complemented with multi-task learning. (a) Single source *supervised* or *unsupervised* domain or language adaptation (b) Multi-source *supervised* or *unsupervised* domain or language adaptation (c) Single source *semi-supervised* learning with partially labeled data. Further new directions in NER, such as multi-source adaptation with semi-supervised or few-shot learning of the target, are possible.

out the framework. In particular, for a given named collection as `domain.datasetname` (possibly split into train, development and test files), T2NER creates output data files named as `lang.domain.datasetname-split` and `lang.domain.datasetname.labels`, where language information is provided by the user. In case of missing metadata, a placeholder `xxx` can be used. For preprocessing, we tokenize via *Transformers* and split the sentences which are longer than the user-defined maximum length. An example output file could be `en.news.conll-train`, referring to the CoNLL 2003 data set (Tjong Kim Sang and De Meulder, 2003).

Besides NER data, additional task data can also be provided, such as that for language modeling, POS tagging, and alignment resources (e.g. bilingual dictionaries or parallel sentences).

3.2 Data Readers

These are classes that are designed to serve the data needs of a given transfer learning scenario in a modular and extensible way. The framework provides `SimpleData`, `SimpleAdaptationData`, `MultiData`, and `SemiSupervisedData` which are suitable for single dataset NER, cross- lingual and domain NER, multi-dataset NER, and single dataset semi-supervised NER, respectively. Each class is derived from a base class `BaseData` and can be extended for further scenarios. As a concrete example, consider a dataset reader class `SimpleAdaptationData` in T2NER, which can provide training data for *source* and *target* language or domain up to a requested number of copies.

3.3 Models

A model is composed of three main components: a base encoder from the *Transformers* (Wolf et al., 2020), any additional networks (X-nets) on top of the encoder, and the prediction layer(s).

Encoder is the main model component that takes as input tokenized text and returns hidden states such as those from BERT (Devlin et al., 2019) or RoBERTa (Liu et al., 2019). There are five encoder modes that we support:

- `finetune`: Fine-tunes the encoder and uses the last layer hidden states.
- `freeze`: Freezes the encoder and uses the last layer hidden states.
- `firstn`: Freezes only the first n layers of the encoder and uses the last layer hidden states (Wu and Dredze, 2019).
- `lastn`: Freezes the encoder and uses the aggregated hidden states by summing the outputs from the last n layers (Wang et al., 2019).
- `embedonly`: Uses and fine-tunes the embedding layer only of the encoder.

X-nets are additional neural architectures that can be used on top of the encoder to further function on the encoder hidden states. T2NER provides

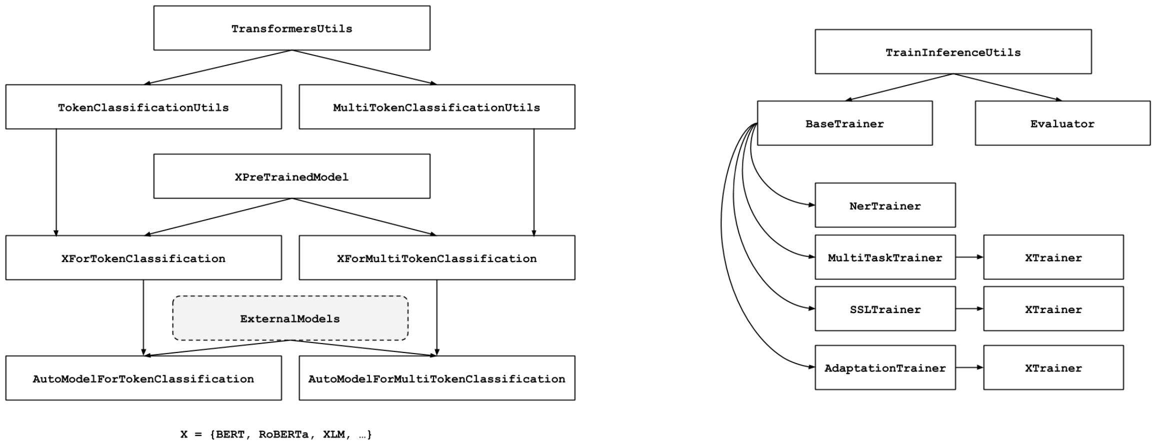


Figure 3: Class hierarchies in T2NER for two main class concepts: (**Left**) Main model architectures in single and multi-task settings with the adoption of Auto classes concepts from *Transformers* (Wolf et al., 2020), where customized functionality or new modeling concepts can easily be added. (**Right**) Main trainer classes that offer a particular transfer learning scenario and extend it to a specific transferring algorithm.

multi-layered Transformers and BiLSTM by default.

Prediction Layers offer the final classification layer for the sequence labeling. Following Devlin et al. (2019), the default prediction layer in T2NER is a linear layer, however support for linear-chain conditional random field (CRF) is included. In the multi-task setting, several output layers from different datasets in different domains or languages might be available with partial or exact entity types as outputs. To help the transfer across the tasks, **private** and **shared** prediction layers are also supported (Wang et al., 2020; Lin et al., 2018).

With these underlying components, models are mainly implemented as single or multi-task architectures. To support a wide range of encoders in a unified API, T2NER adopts the Auto classes design from the *Transformers*. Figure 3 shows the class hierarchies, outlining the customized extensions with further possibilities to extend with external model implementations.

3.4 Criterions

For a given sequence of length L with tokens $x = [x_1, x_2, ..., x_L]$, labels $y = [y_1, y_2, ..., y_L]$ with each $y_i \in \Delta^{\mathcal{C}}$ a one-hot entity type vector with C types, and the linear prediction layer, the NER loss is defined as:

$$\mathcal{L}(y;x) = -\sum_{i=1}^{C}\sum_{j=1}^{L} y_{ij} \log p(h_j = i|x_j)$$

where $p(h_j = i|x_j)$ is the probability of token x_j being labeled as entity type i and h_j is the model output. When p is softmax, this becomes cross-entropy loss. To tackle class-imbalance in real-world applications, T2NER also offers two-class sensitive loss functions:

- **Focal Loss** adds a modulating factor to the standard softmax which reduces the loss contribution from easy examples and extends the range in which an example receives low loss (Lin et al., 2017).
- **LDAM Loss** is the label-distribution-aware loss function that encourages the model to have the optimal trade-off between per-class margins by promoting the minority classes to have larger margins (Cao et al., 2019).

3.5 Auxiliary Tasks

Multi-task learning has greatly benefited transfer learning in NER (Lin et al., 2018; Wang et al., 2020; Jia et al., 2019; Jia and Zhang, 2020). Several auxiliary tasks are supported in a multi-task model by default:

- *Language Classification*: In the cross-lingual setting, this task provides an additional classification signal over the languages (e.g., English and Spanish) used in the training data (Keung et al., 2019).
- *Domain Classification*: In the cross-domain setting, this task provides an additional clas-

sification signal over the domains (e.g., News and Biomedical) used in the training data (Wang et al., 2020).

- *Adversarial Classification*: In the cross- lingual or domain setting, this task provides an additional adversarial classification signal over the languages or domains to learn invariant features used in the training data (Keung et al., 2019; Chen et al., 2018).
- *Language Modeling*: While pre-trained transformer models are already tuned on a specific corpora, additional causal language modeling signal is supported during fine-tuning over the raw texts (Rei, 2017; Jia et al., 2019; Jia and Zhang, 2020).
- *Entity Type Classification*: To better extract entity type knowledge, an additional linear classifier is added. This performs classification over entity types such as [PER, LOC, O, ...] without the segmentation tags such as B/I/E (Jia and Zhang, 2020).
- *Shared Tagging*: In NER settings where the entity types might differ, a shared prediction layer across all the entity types provides an additional signal to the base NER tasks.
- *All-Outside Classification*: This is a binary classification task which predicts if the sentence has entity types other than the outside (O) type.

3.6 Optimization Modules

T2NER provides thin wrappers around the optimizers and learning rate schedulers from the PyTorch (Paszke et al., 2019) and the *Transformers* (Wolf et al., 2020) libraries.

3.7 Trainers

Trainer is the main class concept that glues together all the components and provides a unified setup to develop, test, and benchmark the algorithms. Figure 3 shows the organization of trainer classes. Each transfer learning scenario inherits from the `BaseTrainer` class, where each scenario can further be extended to create an algorithm-specific training regime. This allows the researchers to focus mainly on the algorithms' logic while the framework fulfills the requirements of a chosen transfer scenario. Following (Zhou et al., 2020; Jiang et al., 2020), a few training algorithms are implemented by default which we briefly describe. In the following, a feature extractor is referred to as the base encoder with any X-nets. An optional pooling strategy {`mean`, `sum`, `max`, `attention`, ...} can be applied to aggregate the hidden states. In what follows, domain and language can be used interchangeably. For consistency, we use the word domain.

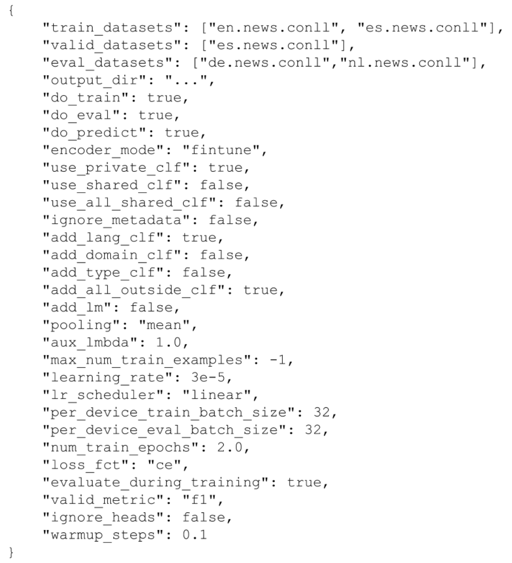
Gradient Reversal Layer (GRL) adds a domain classifier which is trained to discriminate whether input features come from the source or target domain, whereas the feature extractor is trained to deceive the domain classifier to match feature distributions.

Earth Mover Distance (EMD) adds a critic that maximizes the difference between unbounded scores of source and target features. This effectively returns the approximation of Wasserstein distance between source and target feature distributions (Arjovsky et al., 2017). The overall objective jointly minimizes NER cross-entropy loss and Wasserstein distance. Theoretically, GRL is effectively minimizing Jensen-Shannon (JS) divergence which suffers from discontinuities and thus provide poor gradients for feature extractor. In contrast Wasserstein distance is stable and less prone to hyperparamter selection (Chen et al., 2018). For stable training, the gradient penalty is also provided (Gulrajani et al., 2017).

Keung Adversarial is closely related to GRL but additionally uses the generator loss such that the features are difficult for the discriminator to classify correctly between source and target. The optimization is carried out in step-wise fashion for the feature extractor, discriminator, and generator (Keung et al., 2019).

Maximum Classifier Discrepancy (MCD) adds a second classifier to measure the discrepancy between the predictions of two classifiers on target samples. It is noted that the target samples outside the support of the source can be measured by two different classifiers. Overall, MCD solves a *minimax* problem in which the goal is to find two classifiers that maximize the discrepancy on the target sample, and a features generator that minimizes this discrepancy (Saito et al., 2018).

Minimax Entropy (MME) decreases the entropy on unlabeled target features in adversarial manner by using GRL to obtain high quality discriminative features (Saito et al., 2019). Besides unsupervised domain adaptation, the method can



```
{
    "train_datasets": ["en.news.conll", "es.news.conll"],
    "valid_datasets": ["es.news.conll"],
    "eval_datasets": ["de.news.conll","nl.news.conll"],
    "output_dir": "...",
    "do_train": true,
    "do_eval": true,
    "do_predict": true,
    "encoder_mode": "fintune",
    "use_private_clf": true,
    "use_shared_clf": false,
    "use_all_shared_clf": false,
    "ignore_metadata": false,
    "add_lang_clf": true,
    "add_domain_clf": false,
    "add_type_clf": false,
    "add_all_outside_clf": true,
    "add_lm": false,
    "pooling": "mean",
    "aux_lmbda": 1.0,
    "max_num_train_examples": -1,
    "learning_rate": 3e-5,
    "lr_scheduler": "linear",
    "per_device_train_batch_size": 32,
    "per_device_eval_batch_size": 32,
    "num_train_epochs": 2.0,
    "loss_fct": "ce",
    "evaluate_during_training": true,
    "valid_metric": "f1",
    "ignore_heads": false,
    "warmup_steps": 0.1
}
```

Figure 4: An example of the configuration file that allows the user to specify their choices. It shows an instantiation of the multi-task learning scenario.

additionally be used in semi-supervised and few-shot learning scenarios when some labeled target samples are available.

Further algorithms, such as classical conditional entropy minimization (CEM) for semi-supervised learning (Grandvalet and Bengio, 2004) or recent works based on maximum mean discrepancy (MMD) for multi-source domain adaptation (Peng et al., 2019), are provided. In general, extending T2NER for newer algorithms is simple and flexible.

4 Usage

T2NER offers a single entry point to the framework which relies on a base JSON configuration file, an experiment-specific JSON configuration file with an optional algorithm name to run. An example experiment-specific configuration file is shown in Figure 4. The command below shows an example run:

```
$ python t2ner/run.py \
    --exp_type unsup_adapt \
    --base_json configs/base.json \
    --exp_json configs/grl.json \
    --method grl
```

Like other frameworks, it can be further developed and used as a standard Python library.

5 Conclusion and Future Work

In this work we presented a transformer based framework for transfer learning research in named entity recognition (NER). We laid out the design principles, detailed out the architecture, and presented the transfer scenarios and some of the representative algorithms. T2NER offers to bridge the gap between growing research in deep transformer models, NER transfer learning, and domain adaptation. T2NER has the potential to serve as a unified benchmark for existing and newer algorithms with state-of-the-art models.

For future work, we consider the following:

- We would like to create a benchmark data and perform comparison of the transfer learning algorithms (Ramponi and Plank, 2020; Kashyap et al., 2020).

- We would like to investigate adding support for few-shot (Huang et al., 2020), nested (Jue et al., 2020) and document-level (Schweter and Akbik, 2020) NER.

- Assess the performance of framework in terms of speed and efficiency and compare with other tools[3].

- While we focused on the task of NER here, we would also like to add related tasks such as relation extraction, entity linking, and question answering.

Acknowledgments

The work was partially funded by the European Union's Horizon 2020 research and innovation programme under grant agreement No. 777107 through the project Precise4Q and by the German Federal Ministry of Education and Research (BMBF) through the project CoRA4NLP (01IW20010).

References

Oshin Agarwal, Yinfei Yang, Byron C Wallace, and Ani Nenkova. 2020. Interpretability analysis for named entity recognition to understand system predictions and how they can improve. *arXiv preprint arXiv:2004.04564*.

[3] https://github.com/JayYip/bert-multitask-learning

Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pages 214–223.

Isabelle Augenstein, Leon Derczynski, and Kalina Bontcheva. 2017. Generalisation in named entity recognition: A quantitative analysis. *Computer Speech & Language*, 44:61–83.

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. 2019. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems*, pages 1567–1578.

Xilun Chen, Yu Sun, Ben Athiwaratkun, Claire Cardie, and Kilian Weinberger. 2018. Adversarial deep averaging networks for cross-lingual sentiment classification. *Transactions of the Association for Computational Linguistics*, 6:557–570.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, pages 7059–7069.

Silviu Cucerzan. 2007. Large-scale named entity disambiguation based on wikipedia data. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, pages 708–716.

Aron Culotta and Jeffrey Sorensen. 2004. Dependency tree kernels for relation extraction. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 423–429.

Hal Daumé III. 2007. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 256–263.

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.

Yves Grandvalet and Yoshua Bengio. 2004. Semi-supervised learning by entropy minimization. *Advances in neural information processing systems*, 17:529–536.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. 2017. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pages 5767–5777.

Jiaxin Huang, Chunyuan Li, Krishan Subudhi, Damien Jose, Shobana Balakrishnan, Weizhu Chen, Baolin Peng, Jianfeng Gao, and Jiawei Han. 2020. Few-shot named entity recognition: A comprehensive study. *arXiv preprint arXiv:2012.14978*.

Chen Jia, Xiaobo Liang, and Yue Zhang. 2019. Cross-domain ner using cross-domain language modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2464–2474.

Chen Jia and Yue Zhang. 2020. Multi-cell compositional lstm for ner domain adaptation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5906–5917.

Junguang Jiang, Bo Fu, and Mingsheng Long. 2020. Transfer-learning-library. `https://github.com/thuml/Transfer-Learning-Library`.

WANG Jue, Lidan Shou, Ke Chen, and Gang Chen. 2020. Pyramid: A layered model for nested named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5918–5928.

Abhinav Ramesh Kashyap, Devamanyu Hazarika, Min-Yen Kan, and Roger Zimmermann. 2020. Domain divergences: a survey and empirical analysis. *arXiv preprint arXiv:2010.12198*.

Phillip Keung, Vikas Bhardwaj, et al. 2019. Adversarial learning with contextual embeddings for zero-resource cross-lingual classification and ner. 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 1355–1360.

Young-Bum Kim, Karl Stratos, Ruhi Sarikaya, and Minwoo Jeong. 2015. New transfer learning techniques for disparate label sets. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 473–482.

Jayant Krishnamurthy and Tom M Mitchell. 2015. Learning a compositional semantics for freebase with an open predicate vocabulary. *Transactions of the Association for Computational Linguistics*, 3:257–270.

Ji Young Lee, Franck Dernoncourt, and Peter Szolovits. 2018. Transfer learning for named-entity recognition with neural networks. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Bill Yuchen Lin and Wei Lu. 2018. Neural adaptation layers for cross-domain named entity recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2012–2022.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

Ying Lin, Shengqi Yang, Veselin Stoyanov, and Heng Ji. 2018. A multi-lingual multi-task architecture for low-resource sequence labeling. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 799–809.

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*.

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.

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.

Nanyun Peng and Mark Dredze. 2017. Multi-task domain adaptation for sequence tagging. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 91–100.

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. 2019. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1406–1415.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Alan Ramponi and Barbara Plank. 2020. Neural unsupervised domain adaptation in nlp—a survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6838–6855.

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.

Marek Rei. 2017. Semi-supervised multitask learning for sequence labeling. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2121–2130.

Nils Reimers and Iryna Gurevych. 2017. Reporting score distributions makes a difference: Performance study of lstm-networks for sequence tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 338–348.

Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. 2019. Semi-supervised domain adaptation via minimax entropy. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8050–8058.

Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. 2018. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3723–3732.

Stefan Schweter and Alan Akbik. 2020. Flert: Document-level features for named entity recognition. *arXiv preprint arXiv:2011.06993*.

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-Volume 4*, pages 142–147.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Jing Wang, Mayank Kulkarni, and Daniel Preoţiuc-Pietro. 2020. Multi-domain named entity recognition with genre-aware and agnostic inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8476–8488.

Zhenghui Wang, Yanru Qu, Liheng Chen, Jian Shen, Weinan Zhang, Shaodian Zhang, Yimei Gao, Gen Gu, Ken Chen, and Yong Yu. 2018. Label-aware double transfer learning for cross-specialty medical named entity recognition. 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 1–15.

Zirui Wang, Jiateng Xie, Ruochen Xu, Yiming Yang, Graham Neubig, and Jaime G Carbonell. 2019. Cross-lingual alignment vs joint training: A comparative study and a simple unified framework. In *International Conference on Learning Representations*.

Thomas Wolf, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac, Morgan Funtowicz, Joe Davison, Sam Shleifer, et al. 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.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of bert. 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 833–844.

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

Jie Yang, Shuailong Liang, and Yue Zhang. 2018. Design challenges and misconceptions in neural sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3879–3889.

Jie Yang and Yue Zhang. 2018. Ncrf++: An open-source neural sequence labeling toolkit. In *Proceedings of ACL 2018, System Demonstrations*, pages 74–79.

Joey Tianyi Zhou, Hao Zhang, Di Jin, Hongyuan Zhu, Meng Fang, Rick Siow Mong Goh, and Kenneth Kwok. 2019. Dual adversarial neural transfer for low-resource named entity recognition. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3461–3471.

Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. 2020. Domain adaptive ensemble learning. *arXiv preprint arXiv:2003.07325*.