PTST-UoM at SemEval-2021 Task 10:
Parsimonious Transfer for Sequence Tagging

Kemal Kurniawan[1] **Lea Frermann**[1] **Philip Schulz**[2*] **Trevor Cohn**[1]
[1]School of Computing and Information Systems, University of Melbourne
[2]Amazon Research
kemal.kurniawan@student.unimelb.edu.au
{lea.frermann,tcohn}@unimelb.edu.au
phschulz@amazon.com

Abstract

This paper describes PTST, a source-free unsupervised domain adaptation technique for sequence tagging, and its application to the SemEval-2021 Task 10 on time expression recognition. PTST is an extension of the cross-lingual parsimonious parser transfer framework (Kurniawan et al., 2021), which uses high-probability predictions of the source model as a supervision signal in self-training. We extend the framework to a sequence prediction setting, and demonstrate its applicability to unsupervised domain adaptation. PTST achieves F_1 score of 79.6 % on the official test set, with the precision of 90.1 %, the highest out of 14 submissions.[1]

1 Introduction

SemEval-2021 Task 10 presents source-free unsupervised domain adaptation (SFUDA) for semantic processing.[2] The goal of *unsupervised domain adaptation* is to transfer a model from a source domain to another, different domain — called target domain — using only unlabelled data in the target domain. *Source-free* unsupervised domain adaptation additionally assumes no access to source domain data: only the source model pre-trained on the source domain data is available. This situation may occur when the source domain data contains protected information that cannot be shared, or even if it can, requires signing a complex data use agreement. While there are numerous works on SFUDA outside NLP (Hou and Zheng, 2020; Kim et al., 2020; Liang et al., 2020; Yang et al., 2020), SFUDA research for NLP is severely lacking in spite of its importance in, for example, clinical NLP (Laparra et al., 2020). There are two tasks involved in SemEval-2021 Task 10: negation detection and time expression recognition. We participate only in the latter.

Our approach is an extension of the parsimonious parser transer framework (PPT; Kurniawan et al. (2021)). PPT allows cross-lingual transfer of dependency parsers in a source-free manner, requiring only unlabelled data in the target side. It leverages the output distribution of the source model to build a chart containing high probability trees for each sentence in the target data. We extend this work by (1) formulating PPT for chain structures and evaluating it on a semantic sequence tagging task; and (2) demonstrating its effectiveness in a domain adaptation setting. We call our method **P**arsimonious **T**ransfer for **S**equence **T**agging (PTST).

We find PTST effective for improving the precision of the system in the target domain. It ranks 7th out of 14 submissions in the official leaderboard in terms of F_1 score, but 1st in precision, with a gap of 3 points from the second best. Drawing on the model calibration literature, we provide a way to combat the problem of model overconfidence which is key to make PTST outperform a simple transfer of a source model to the target domain. However, we also find that PTST struggles in improving recall. In conclusion, our results suggest that PTST can be used for SFUDA, but further work is required to improve the precision-recall trade-off in the target domain.

2 Background

In the SemEval-2021 Task 10 time expression recognition task, the input is a single sentence, and the output is a sequence of tags indicating the time entity type of a word (if any). There are 32 time entities in total (e.g., `Year`, `Hour-of-Day`), and

[*] Work done outside Amazon.

[1] Our code is available at https://github.com/kmkurn/ptst-semeval2021.

[2] https://competitions.codalab.org/competitions/26152

Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pages 445–451
Bangkok, Thailand (online), August 5–6, 2021. ©2021 Association for Computational Linguistics

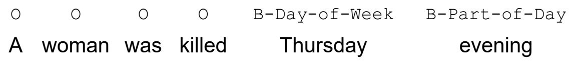
O O O O B-Day-of-Week B-Part-of-Day
A woman was killed Thursday evening

Figure 1: An example input sentence and its output sequence of tags for the time expression recognition task.

the tags are coded in BIO format. Fig. 1 shows an example input sentence and its output. The task organisers provided a pre-trained source model for the task. This source model is RoBERTa-base (Liu et al., 2019) that is pre-trained on more than 25K time expressions in English clinical notes from Mayo Clinic in SemEval-2018 Task 6. The model is distributed online via HuggingFace Models,[3] which can be obtained with HuggingFace Transformers library.[4] The organisers also released trial data for the practice phase containing 99 annotated English articles from the news domain. The official test data released by the organisers in the evaluation phase contains 47 articles that are in a different domain from the source and development data.

The time expression recognition task is formalised as a sequence tagging task. The literature on sequence tagging in NLP is massive (Jiang et al., 2020; He and Choi, 2020; Rahimi et al., 2019; He et al., 2019; Xie et al., 2018; Clark et al., 2018, *inter alia*). One closely related task is named-entity recognition (NER) whose goal is to detect mentions of named entities such as a `Person` or `Organisation` in an input sentence. Lample et al. (2016) introduced a now widely adopted neural architecture for this task, where input word embeddings are encoded with a bidirectional LSTM (Hochreiter and Schmidhuber, 1997) before they are passed through per-word softmax layers. In recent times, it is common to replace the LSTM with a Transformer (Vaswani et al., 2017). With the advancements of large pre-trained language models, the standard is to use a model such as BERT (Devlin et al., 2019) as the encoder and fine-tune the model on labelled data. The source model of the time expression recognition task provided by the organisers was also trained in this manner.

In the context of unsupervised domain adaptation, a popular approach is domain adversarial training. Introduced by Ganin et al. (2016), it leverages multi-task learning in which the model is optimised not only on the main task objective but also a domain prediction objective. The learning signal for the latter is passed through a gradient reversal layer, ensuring that the learnt parameters are predictive for the main task, but general across domains. With pre-trained language models, Han and Eisenstein (2019) proposed to continue pre-training on the unlabelled target domain data, prior to finally fine-tuning on the labelled source domain data. Unfortunately, these approaches require access to the source domain data.

There is relatively little work on SFUDA in NLP, however, some works on source-free cross-lingual transfer exist. Wu et al. (2020) employ teacher-student learning for source-free cross-lingual NER. A teacher model trained on the source side predicts soft labels on the unlabelled target side data, and a student model is trained on those soft labels. Their method outperforms a simple direct transfer method where the source model is directly applied on the target side. More recently, a method for source-free cross-lingual transfer of dependency parsers was introduced by Kurniawan et al. (2021). The key idea is to build a chart of high probability trees based on arc marginal probabilities for each unlabelled sentence on the target side, and treat all those trees as a weak supervision signal for training. Their method outperforms direct transfer as well as a variety of recent cross-lingual transfer methods that are not source-free. That said, the effectiveness of their method on (a) semantic (sequence labelling) tasks and (b) in a domain adaptation setting is unexplored, which is what we aim to address in this work.

3 System Description

We first describe our sequence tagging model (Section 3.1), before we present parsimonious transfer for sequence tagging (PTST) in Section 3.2.

3.1 Model

Our model is a linear-chain conditional random field (CRF) over tag sequences. It assigns a score $s(\boldsymbol{x}, \boldsymbol{y})$ to a pair of input sentence $\boldsymbol{x}$ and output tag sequence $\boldsymbol{y}$, which can be expressed as

$$s(\boldsymbol{x}, \boldsymbol{y}) = \sum_{j} \pi(\boldsymbol{x}, j, y_j) + \phi(y_j, y_{j+1}) \qquad (1)$$

where $\pi(\boldsymbol{x}, j, t)$ is the *emission* score of word x_j having tag t and $\phi(t, t')$ is the *transition* score of having tag t followed by tag t'. The probability of

[3] https://huggingface.co/clulab/roberta-timex-semeval

[4] https://github.com/huggingface/transformers

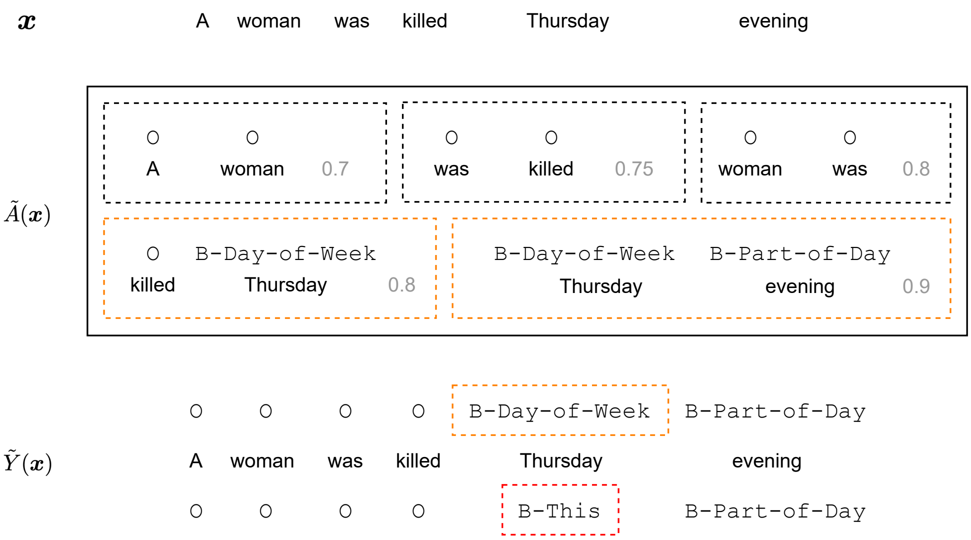
$\boldsymbol{x}$
A woman was killed Thursday evening
$\tilde{A}(\boldsymbol{x})$
O O
A woman 0.7
O O
was killed 0.75
O O
woman was 0.8
O B-Day-of-Week
killed Thursday 0.8
B-Day-of-Week B-Part-of-Day
Thursday evening 0.9
$\tilde{Y}(\boldsymbol{x})$
O O O O B-Day-of-Week B-Part-of-Day
A woman was killed Thursday evening
O O O O B-This B-Part-of-Day

Figure 2: Illustration of our method. Given an unlabelled sentence $\boldsymbol{x}$ from the target domain, we build the set of high probability tag pairs $\tilde{A}(\boldsymbol{x})$ using the source model, which may contain correct tag pairs that do not occur in the predicted tag sequence (in orange). From these tag pairs, we build the chart $\tilde{Y}(\boldsymbol{x})$ containing tag sequences that can be assembled from tag pairs in $\tilde{A}(\boldsymbol{x})$. The single predicted tag sequence from the source model (bottom) is also included in the chart, but it may contain an incorrect tag (in red) as it is noisy.

$\boldsymbol{y}$ given $\boldsymbol{x}$ is then

$$P(\boldsymbol{y} \mid \boldsymbol{x}) \propto \exp s(\boldsymbol{x}, \boldsymbol{y}). \tag{2}$$

The emission score function π is parameterised by a neural network whose parameters are initialised with the source model. Specifically, the emission score function is the RoBERTa model provided by the task organisers. The transition score function ϕ is a $T \times T$ parameter matrix that is learned during training, where T is the number of tags. Note that with dynamic programming, we can efficiently compute quantities such as the marginal probabilities of tag pairs $P((j, y_j, y_{j+1}) \mid \boldsymbol{x})$ or the partition function $Z(\boldsymbol{x}) = \sum_{\boldsymbol{y} \in \mathcal{Y}(\boldsymbol{x})} \exp s(\boldsymbol{x}, \boldsymbol{y})$ where $\mathcal{Y}(\boldsymbol{x})$ is the set of all possible tag sequences for $\boldsymbol{x}$.

3.2 Unsupervised Adaptation

To perform unsupervised adaptation on the unlabelled target domain data, we extend PPT, our past work for source-free cross-lingual parsing (Kurniawan et al., 2021), to chain structures. Given a set of unlabelled sentences D in the target domain, we build a chart of high probability tag sequences $\tilde{Y}(\boldsymbol{x})$ by leveraging the output distribution in the source model. The model then treats all sequences with sufficiently high predicted probability as possible tag sequences for $\boldsymbol{x}$ for training. Concretely, it minimises the loss:

$$\ell(\boldsymbol{\theta}) = -\sum_{\boldsymbol{x} \in D} \log \sum_{\boldsymbol{y} \in \tilde{Y}(\boldsymbol{x})} P_{\boldsymbol{\theta}}(\boldsymbol{y} \mid \boldsymbol{x}) \tag{3}$$

where $\boldsymbol{\theta}$ denotes the target model parameters. The set $\tilde{Y}(\boldsymbol{x})$ is defined formally as

$$\tilde{Y}(\boldsymbol{x}) = \{\boldsymbol{y} | \boldsymbol{y} \in \mathcal{Y}(\boldsymbol{x}) \wedge A(\boldsymbol{y}) \subseteq \tilde{A}(\boldsymbol{x})\} \tag{4}$$

where $\mathcal{Y}(\boldsymbol{x})$ is the set of all possible tag sequences for $\boldsymbol{x}$, $A(\boldsymbol{y}) = \{(j, y_j, y_{j+1}) | 1 \leq j < |\boldsymbol{y}|\}$ is the set of consecutive tag pairs in tag sequence $\boldsymbol{y}$, and $\tilde{A}(\boldsymbol{x}) = \bigcup_j \tilde{A}(\boldsymbol{x}, j)$ where $\tilde{A}(\boldsymbol{x}, j)$ is the set of high probability consecutive tag pairs (j, t_j, t_{j+1}) for words x_j and x_{j+1} (see Fig. 2 for illustration). Analogous to PPT, this set is constructed by adding tag pairs (t_j, t_{j+1}) in order of decreasing marginal probability until their cumulative probability exceeds a threshold σ. The predicted tag sequence from the source model is also included in $\tilde{Y}(\boldsymbol{x})$ so the chart is never empty.

Note that the method above is very similar to self-training where the predictions from the source model are used as supervision signal for training. In contrast to self-training, however, we build a chart of high probability predictions for each sample instead of just a single best prediction. We expect these predictions to be more useful than a single best prediction because it is more likely for the correct tag sequence to be in the chart than equal to the single best prediction. Even when this is not the case, we expect the partially correct tag sequences occur frequently enough in the chart so the model is still able to learn what the correct tag sequence is.

Team	F_1	P	R
BCLUFIGHT-1	**81.5**	84.7	78.5
Self-Adapter-1	81.1	87.3	75.7
BLCUFIGHT-2	81.0	83.4	78.7
Baseline-2	80.4	82.7	78.2
YNU-HPCC-2	80.3	81.7	79.1
Self-Adapter-2	79.7	83.9	76.0
PTST-UoM-1 (ours)	79.6	**90.1**	71.3
Boom-1	79.5	86.9	73.2
UArizona-1	79.5	78.6	80.4
UArizona-2	79.5	78.3	**80.7**
Baseline-1	79.4	84.9	74.6
KISNLP-1	79.3	81.0	77.7
KISNLP-2	78.1	79.8	76.4
YNU-HPCC-1	74.8	87.2	65.5

Table 1: TIMEX leaderboard on the test data.

In our preliminary experiments, we find that it is crucial to introduce temperature scaling to the emission scoring function in order to achieve good performance. Thus, for our main result, we define a new emission scoring function π' as

$$\pi'(\boldsymbol{x}, j, y_j) = \pi(\boldsymbol{x}, j, y_j)/\tau \qquad (5)$$

where τ is the temperature scale hyperparameter. We discuss and provide an analysis of this temperature scaling in Section 5.

4 Experimental Setup

We use the pre-trained source model and data provided for SemEval-2021 Task 10. We only use the model and data for the time expression recognition task (TIMEX hereinafter) as we only participate in that task. We use the practice data as the development set to tune the hyperparameters of our model with random search.[5] We set the threshold $\sigma = 0.95$ following the setup of Kurniawan et al. (2021). We do not use any data sets other than those provided by the task organisers for TIMEX.

We train PTST on the unlabelled test data for 5 epochs. As described in Section 3.1, we initialise the neural network for the emission scoring with the source RoBERTa model provided by the task organisers. We enforce the BIO constraints by initialising the transition matrix ϕ with $-\infty$ for entries corresponding to illegal transitions, and zero otherwise.[6] We use the linear CRF implementation provided in the Torch-Struct library (Rush, 2020). To avoid out-of-memory error, we discard sentences longer than 30 tokens. Additionally, we find that it is useful to freeze the embedding and the first few layers of the RoBERTa encoder. Thus, in our main result, we freeze the embedding layer and the first 6 (out of 12) layers of the encoder. An analysis is provided in Section 5.

5 Results and Discussion

Table 1 shows the TIMEX leaderboard on the official test data in the evaluation phase.[7] Our model PTST is ranked 7th out of 14 submissions in terms of F_1 score, below the baseline model submission by the task organisers which ranks 4th. This baseline model is the pre-trained source model fine-tuned on the labelled development data. Despite the relatively low F_1 score, PTST achieves 90.1 % precision, which is the highest among all submissions, and markedly above the second highest precision of 87.3 % achieved by the second best performing model. Looking at recall, our model has the second lowest score of 71.3 %, which is fairly below the third lowest one of 73.2 %. This result suggests that our model is sacrificing recall in favour of precision, which may be a desirable property for downstream tasks where making the right prediction is more critical.

Model overconfidence As mentioned in Section 3.2, in our preliminary experiments, we find that the source model is extremely confident about its predictions, making the marginal probability distribution of tag pairs at any position j very sharp. This sharpness results in $\tilde{Y}(\boldsymbol{x})$ containing mostly just a single tag sequence, which is the predicted sequence from the source model, rendering the whole approach no different from simple self-training. To remedy this problem, we introduce temperature scaling in the emission score, which has been shown to be a simple but effective trick in model calibration (Geman and Geman, 1984; Guo et al., 2017). We define the new emission scoring function as shown in Eq. (5). Table 2 shows how the performance of PTST changes when τ is varied. We see that as τ increases, precision does too, but recall decreases, although in a relatively slower rate so the F_1 score tends to increase as well. Also reported in Table 2 is the median number of tag sequences and the fraction of gold tag sequences

[5] Best learning rate and τ are 9×10^{-6} and 2.56 respectively.

[6] The constraints require an inside tag be always preceded by an inside or beginning tag of the same entity.

[7] Also available on https://machine-learning-for-medical-language.github.io/source-free-domain-adaptation/leaderboard.

τ	F_1	P	R	n	p (%)
1.0	77.0 ± 0.4	76.5 ± 1.0	77.5 ± 0.3	1	54.1
1.5	77.4 ± 0.2	77.5 ± 0.1	77.4 ± 0.5	1	56.9
2.0	77.6 ± 0.1	79.1 ± 0.5	76.1 ± 0.4	1	64.0
2.5	77.8 ± 0.5	81.2 ± 0.2	74.7 ± 1.0	9.2×10^{13}	79.7
3.0	26.7 ± 7.9	96.3 ± 2.9	15.7 ± 5.3	2.2×10^{17}	86.9
SRC	77.1	77.5	76.8	—	—

Table 2: Model performance on the development data as τ changes. SRC is the pre-trained source model directly applied on the development data. F_1, precision (P), and recall (R) scores are averages (± std) over 3 runs. n is the median number of high probability tag sequences in the chart $\tilde{Y}(\boldsymbol{x})$. p is the fraction of gold tag sequences contained in the chart.

	F_1	P	R
SRC	77.1	77.5	76.8
No freezing	77.8 ± 0.5	81.2 ± 0.2	74.7 ± 1.0
Freeze emb	77.7 ± 0.2	81.0 ± 0.4	74.7 ± 0.6
Freeze emb + 6 layers	78.2 ± 0.2	80.3 ± 0.3	76.2 ± 0.0
Freeze emb + 12 (all) layers	77.2 ± 0.0	77.7 ± 0.0	76.7 ± 0.0

Table 3: Model performance on the development data when the RoBERTa embedding and encoder layers are frozen during training. SRC is the pre-trained source model directly applied on the development data. Scores are averages (± std) over 3 runs.

contained in the chart. The two quantities grow as τ does, which indicates that increasing τ indeed allows the chart to contain more tag sequences, and thus increasing the coverage of correct tag sequences in the chart. However, when τ is too large ($\tau = 3.0$), the model breaks down, presumably because $\tilde{Y}(\boldsymbol{x})$ contains too many noisy tag sequences to be useful.

The decline in recall might be explained by the nature of the task, where in a single sentence most of the words are not time entities. When τ grows, the number of high probability tag sequences in $\tilde{Y}(\boldsymbol{x})$ does too. In the majority of these tag sequences, a word in $\boldsymbol{x}$ is likely to be tagged as a non-entity because time entities are naturally rare. Since tag sequences are treated uniformly (i.e. no tag sequence weighs more than the others), this provides a strong signal for the model that the word is a non-entity. Therefore, the model's capability of recognising entities is reduced. Conversely, a similar argument may explain the rise in precision. When the model predicts a word as an entity, it is likely that in the majority of tag sequences in $\tilde{Y}(\boldsymbol{x})$, the word is tagged as the same entity, providing a strong signal that the word is indeed that entity. In other words, if the model predicts an entity, the model is very confident about it. When confidence is high, it is more likely that the prediction is correct, thus resulting in higher precision.

Freezing layers We also find that it is helpful to freeze the embedding layer and the first few layers of the RoBERTa model's encoder during training, presumably because they encode low-level linguistic information that is invariant across domains. Table 3 reports how the model performance changes with varying numbers of layers frozen (τ is fixed to 2.5). We observe that freezing the embedding and first several encoder layers gives a small boost to performance, with best performance reached with 6 frozen layers (the setting adopted in the model reported in the main results).

Error analysis To better understand the errors of PTST, we present the confusion matrix of the model on the test data in Fig. 3. We see that the majority of the errors arise from the model not recognising actual time entities, consistent with the relatively low recall. The model has serious difficulties in recognising `Season-Of-Year`, for example, in fragments like:

*The increase in food aid beneficiaries is partly attributed to **Meher** harvest loss [...]* (1)

*The increase, which follows a **seasonal** trend, is seen in all regions except Tigray.* (2)

The model also seems to struggle with recognising `Between` and `This`. Example sentence fragments where the model wrongly predicts a

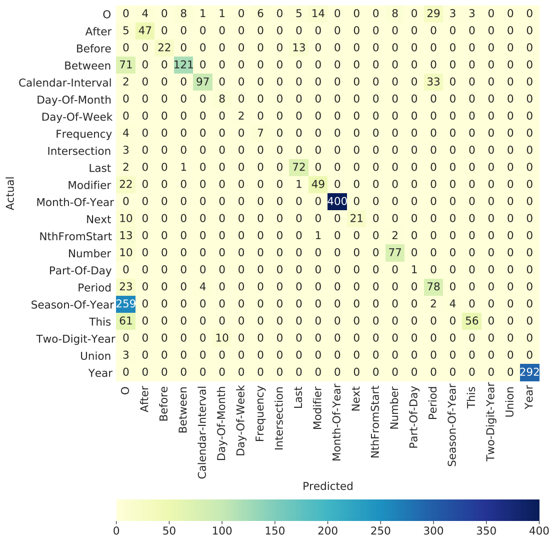

Figure 3: Confusion matrix on the test data.

non-entity are:

*In Gambella region, **between** 1 and 12 April, 3,346 South Sudanese refugees arrived [...]* (3)

where the model fails to recognise Between, and

*[...] to prevent a potential national outbreak of AWD during **this** rainy season* (4)

where the model fails to recognise This. Although the model has good precision, we still see that it misclassifies non-entities and Calendar-Interval as Period relatively often. For example, in the fragment

*[...] the malnutrition situation is expected to aggravate in the coming **months*** (5)

the word *months* is a Calendar-Interval but the model predicts it as Period. Another example, the model predicts the word *period* in the sentence fragment

*During the reporting **period**, an estimated 1,000 south Sudanese arrived [...]* (6)

as Period, while the word is actually not a time entity.

6 Conclusions

We present PTST, our submission to the time expression recognition task of SemEval-2021 Task 10. We describe our sequence tagging model as a CRF over chain structures, parameterised by a neural network. Our domain adaptation approach leverages the output distribution of the source model to build a chart of high probability tag sequences for every sentence in the unlabelled target domain data. PTST ranks 7th in terms of F_1 score in the official leaderboard, but achieves the highest precision out of 14 submissions. We provide analyses on the importance of temperature scaling to mitigate model overconfidence and the patterns of errors.

Acknowledgments

A graduate research scholarship is provided by Melbourne School of Engineering to Kemal Kurniawan.

References

Kevin Clark, Minh-Thang Luong, Christopher D. Manning, and Quoc V. Le. 2018. Semi-supervised sequence modeling with cross-view training. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1914–1925.

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.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030.

S. Geman and D. Geman. 1984. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):721–741.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330.

Xiaochuang Han and Jacob Eisenstein. 2019. Unsupervised domain adaptation of contextualized embeddings for sequence labeling. 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 4238–4248.

Han He and Jinho Choi. 2020. Establishing strong baselines for the new decade: Sequence tagging, syntactic and semantic parsing with BERT. In *The Thirty-Third International Flairs Conference*.

Junxian He, Zhisong Zhang, Taylor Berg-Kirkpatrick, and Graham Neubig. 2019. Cross-lingual syntactic transfer through unsupervised adaptation of invertible projections. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3211–3223.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Yunzhong Hou and Liang Zheng. 2020. Source free domain adaptation with image translation. *arXiv:2008.07514 [cs, eess]*.

Zhengbao Jiang, Wei Xu, Jun Araki, and Graham Neubig. 2020. Generalizing natural language analysis through span-relation representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2120–2133.

Youngeun Kim, Sungeun Hong, Donghyeon Cho, Hyoungseob Park, and Priyadarshini Panda. 2020. Domain adaptation without source data. *arXiv:2007.01524 [cs, eess]*.

Kemal Kurniawan, Lea Frermann, Philip Schulz, and Trevor Cohn. 2021. PPT: Parsimonious parser transfer for unsupervised cross-lingual adaptation. *arXiv:2101.11216 [cs]*.

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.

Egoitz Laparra, Steven Bethard, and Timothy A. Miller. 2020. Rethinking domain adaptation for machine learning over clinical language. *JAMIA Open*, 3(2):146–150.

Jian Liang, Dapeng Hu, and Jiashi Feng. 2020. Do we really need to access the source data? Source hypothesis transfer for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 6028–6039.

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:1907.11692 [cs]*.

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

Alexander Rush. 2020. Torch-Struct: Deep structured prediction library. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 335–342.

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*, volume 30.

Qianhui Wu, Zijia Lin, Börje Karlsson, Jian-Guang LOU, and Biqing Huang. 2020. Single-/multi-source cross-lingual NER via teacher-student learning on unlabeled data in target language. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6505–6514.

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

Shiqi Yang, Yaxing Wang, Joost van de Weijer, Luis Herranz, and Shangling Jui. 2020. Unsupervised domain adaptation without source data by casting a BAIT. *arXiv:2010.12427 [cs]*.