A Comparison of Strategies for Source-Free Domain Adaptation

Anonymous ACL submission

Abstract

Data sharing restrictions are common in NLP, especially in the clinical domain, but there is little research on adapting models to new domains without access to the original training data, a setting known as source-free domain adaptation. We take algorithms that traditionally assume access to the source-domain training data—active learning, self-training, and data augmentation—and adapt them for source free domain adaptation. Then we systematically compare these different strategies across multiple tasks and domains. We find that while self-training and data augmentation are occasionally successful, only active learning yields consistent gains across all tasks and domains.

1 Introduction

Recent deep neural network models achieve high performance in various tasks, but typically require annotated training data for each new domain. Domain adaptation algorithms aim to take models trained on one domain (the “source domain”) and transfer the model’s knowledge to another domain (the “target domain”). They typically try to do this without a huge amount of annotated data in the target domain. Domain adaptation can be easy if the source and target domain have similar distributions, but domains often differ substantially (Wilson and Cook, 2020).

While there has been a lot of progress in domain adaptation methods (Kouw, 2018) and even in unsupervised domain adaptation where there are no target-domain labels (Ramponi and Plank, 2020), most methods assume access to the labeled source data. Yet this assumption is often not satisfied, especially in the clinical domain due to privacy concerns and the need for data compression (Laparra et al., 2020).

The recent SemEval 2021 Task 10 on source-free domain adaptation (Laparra et al., 2021) called attention to this challenging but more realistic scenario where labeled source data are not accessible, and only the model trained on the source domain data can be shared. Participants explored methods including self-learning, active-learning, and data augmentation (Laparra et al., 2021) but it is hard to make fair comparisons between algorithms since different teams varied in their base implementations.

We therefore conducted a series of experiments to provide a systematic comparison of algorithms for source-free domain adaptation. Our contributions are:

1. The first systematic comparison of self-training, active learning, and data augmentation for source-free domain adaptation, carried out across two different types of tasks (sentence classification and named entity recognition) and four different target domains.
2. We identify a specific formulation of source-free active learning that consistently improves performance of the source-domain model, and sometimes even outperforms the oracle of fine-tuning on a large set of labeled target domain data.
3. We perform an error analysis across tasks and domains and show that the selected formulation of active learning corrects several types of errors that self-training does not.

Upon publication, we will make our code publicly available.

2 Related Work

2.1 Source-free Domain Adaptation

Recently, there is rising interest in computer vision to develop methods for unsupervised source-free domain adaptation. Several works utilize a generative framework with a classifier trained on source data to generate labeled training examples (Kurmi et al., 2021; Li et al., 2020) or transfer the target examples to match the source style (Hou and Zheng, 2020; Sahoo et al., 2020). Other works use self-
supervised pseudo-labeling. Liang et al. (2020) proposes source hypothesis transfer that freezes the classifier of the source model domain but fine-tunes the encoding of the source model with a goal to reduce the entropy of individual output prediction while maintaining global diversity. They also augment the strategy by self-supervised pseudo labels via the nearest centroid classifier. Kim et al. (2020) select low self-entropy instances as class prototypes and pseudo-label the remaining target instances based on the distance to the class prototypes and progressively update the models on target data in the manner of self-training.

Despite of a growing number of computer vision studies on source-free domain adaptation, there is relatively little NLP research into this challenging but realistic scenario. The main effort is the recent SemEval 2021 Task 10 (Laparra et al., 2021), which asked participants to perform source-free domain adaptation on two tasks: negation detection and time expression recognition. A variety of techniques—including active learning, self-training, and data augmentation—were applied to this task. However, different techniques were applied by different participants, each with different baseline models, so the shared task results do not allow us to make fair comparisons between different techniques. In the current article, we implement and then systematically compare these different techniques.

2.2 Self-training

Self-training (Yarowsky, 1995; McClosky et al., 2006) is a technique that leverages a model trained on a labeled dataset $L$ to make predictions (“pseudo-labels”) on the unlabeled dataset $U$. Specifically, self-training first trains a model on $L$ and at each iteration the model pseudo-labels $U$. The examples in $U$ that the model labels with high confidence (“silver labels”) are then added to $L$, and the model is retrained on the new, larger $L$. This process is repeated until no more predictions are highly confident. Self-training has been applied to a variety of domain adaptation scenarios (Ruder and Plank, 2018; Yu et al., 2015; Cui and Bollegala, 2019), but always with the assumption that the original labeled data $L$ is available at each iteration. In source-free domain adaptation, $L$ is not available, so source-free self-training could train on only the pseudo-labels, and it is unclear whether that would yield a superior or inferior model.

2.3 Active Learning

Active learning is a technique that selects a small number of examples to be manually annotated. Active learning selection criteria are designed to select the examples that should most benefit the model. Various active-learning selection strategies have been developed (see the survey of Settles, 2009), and recent work has shown the benefits of active-learning even with pretrained transformer models (Ein-Dor et al., 2020). Active learning is also frequently used in domain adaptation. For example, Chan and Ng (2007) applied uncertainty sampling for domain adaptation of word sense disambiguation models, and Rai et al. (2010) combined model confidence and a domain discriminator to select target-domain examples for sentiment analysis. As with self-training, active-learning algorithms typically assume that the source-domain training data is available and can be combined with target-domain examples. Thus, the efficacy of source-free active learning is currently unclear.

2.4 Data Augmentation

Data Augmentation is a popular technique to enhance limited data by using existing resources (WordNet, similar datasets, etc.) and/or rule-based transformations of the training data to create new training examples. A variety of data augmentation techniques have been proposed (see the survey of Liu et al., 2020) including back-translation (Sennrich et al., 2016; Wang et al., 2021), lexical-substitution (Zhou et al., 2019; Arefyev et al., 2020; Wei and Zou, 2019; Miao et al., 2020), noise injection (Wei and Zou, 2019), conditional generation (Juuti et al., 2020; Malandrakis et al., 2019; Kobayashi, 2018), and data transformation with preset task-specific rules or templates (Şahin and Steedman, 2018; Wang et al., 2021; Xu et al., 2020). Data augmentation assumes access to the source-domain training data, so cannot be used by itself in source-free domain adaptation. It could be coupled with source-free self-training or source-free active learning, but researchers have not yet systematically explored such combinations.

3 Data

We base our experiments off of the data and source-domain models from the two tasks of SemEval 2021 Task 10: negation detection and time expression recognition. Each task has a model trained from a source domain and a test set for each of
two target domains. The source-domain models for both tasks are RoBERTa-base models (Liu et al., 2019) fine-tuned on source domain data sets, implemented via the Huggingface Transformers library (Wolf et al., 2020).

Negation detection is a “span-in-context” binary sentence classification task. The goal is to predict whether an event (denoted by two special tokens \(<e>\) and \(</e>\)) in the sentence is negated by its context. For example, given the sentence:

\[
\text{Has no } <e> \text{ diarrhea } </e> \text{ and no new lumps or masses}
\]

the goal is to predict that the event \text{diarrhea} is negated by its context. The source-domain negation detection model was trained on Mayo clinic clinical notes. The target domains are Partners HealthCare clinical notes from the i2b2 2010 Challenge and Beth Israel ICU progress notes from the MIMIC III corpus\(^1\).

Time expression recognition is a named entity recognition task. The goal is to identify the time entities in the document and label them with types defined by the SCATE schema (Bethard and Parker, 2016). For example, given the sentence:

\[
\text{the patient underwent appendicitis surgery on August 29, 2018,}
\]

the goal is to identify and label \text{August} as \text{Month-Of-Year}, 29 as \text{Day-Of-Month}, and 2018 as \text{Year}. The source-domain time expression recognition model was trained on the Mayo Clinic clinical notes of SemEval 2018 Task 6 (Laparra et al., 2018). The target domains are news articles (also from SemEval 2018 Task 6) and reports from food security warning systems including the UN World Food Programme\(^2\) and the Famine Early Warning Systems Network\(^3\).

We split each of the shared task’s test sets into 20% as a development set and 80% as a true test set. For active-learning, the development set is used to select examples for manual annotation. For self-training, the development set is where the model makes predictions to generate pseudo-labels. Note that the labels of the development set are only used in active learning experiments (and only for the small number of examples selected by active-learning) and are unused in all other experiments. Detailed data information is shown in table 1.

We use the same evaluation metrics as in SemEval 2021 Task 10 for both tasks: precision, recall, and F1 score.

### Research Questions

Three strategies popular in SemEval 2021 Task 10—self-training, active learning, and data augmentation—were implemented by participants with a variety of different algorithmic decisions. We aim for a systematic analysis of these algorithms, which leads us to the following questions:

1. How much can we gain from having human intervention (active learning) and not just the model alone (self-training)?

---

\(^1\)https://mimic.physionet.org/

\(^2\)https://www.wfp.org/

\(^3\)https://fews.net/.

| Domain | Data Source | #       |
|--------|-------------|---------|
| **Negation Detection Data** | | |
| Source | SHARP Seed | 10,259 Sentences |
| Target: development | i2b2 2010 | 1109 Sentences |
| Target: test | i2b2 2010 | 4436 Sentences |
| Target: development | MIMIC III | 1916 Sentences |
| Target: test | MIMIC III | 7664 Sentences |
| **Time Expression Detection Data** | | |
| Source | SemEval 2018 Task 6 clinical notes | 278 Documents |
| Target: development | SemEval 2018 Task 6 news articles | 20 Documents |
| Target: test | SemEval 2018 Task 6 news articles | 79 Documents |
| Target: development | Food security Reports | 4 Documents |
| Target: test | Food security Reports | 13 Documents |
2. For active learning, given a fixed annotation budget, is it better to do several iterations of selecting examples for annotation and retraining the model, or is it better to select and retrain just once?

3. For self-training, given a fixed confidence threshold, is it better to do several iterations of generating pseudo-labels and retraining the model, or is it better to generate and train only once?

4. In each iteration of active learning or self-training, should we use the training data from the previous iteration or start anew?

5. In each iteration of active learning or self-training, should we continue training the model from the previous iteration or the model from the source-domain?

6. Is it better to combine active learning or self-training with data augmentation, or do they work better alone?

## 5 Method

We design source-free variants of self-training, active learning, and data augmentation that incorporate the following parameters, allowing us to investigate the questions above.

- $T$ the maximum number of iterations for self-training or active learning
- $S_D$ the data construction strategy: $IData$ to keep the training data from the previous iteration, or $RData$ to start anew on each iteration.
- $S_M$ the model training strategy: $IModel$ to continue training the model from the previous iteration, or $RModel$ to continue training from the source-domain model.
- $S_A$ whether or not to use data augmentation.

### 5.1 Source-Free Self-training

Our algorithm for self-training is presented in algorithm 1. It follows standard self-training (Yarowsky, 1995) in using the model to add pseudo-labels to the unlabeled data (line 12). However, there is no source-domain labeled data, so the model can train only on the pseudo-labels. The remainder of the code ensures that models and/or data are retained, discarded, or augmented as per the selected strategies.

Self-training requires a measure of the model’s confidence on each prediction. In both tasks, we add pseudo-labeled training data a sentence at a time, so we measure confidence at the sentence level. In negation detection, we use the predicted probability at RoBERTa’s special sentence-initial token <s>. In time expression recognition, we use the average of the predicted probabilities of the most probable class of each token.

### 5.2 Source-Free Active Learning

Our algorithm for active-learning is presented in algorithm 2. It follows an active learning algorithm similar to Su et al. (2021). Like most active-learning algorithms, the core is to select examples the model is uncertain of (line 9) and then manually annotate them (line 10). Since our development sets are already annotated, we simulate annotation by simply revealing the (previously hidden) labels for the selected examples.

Active learning requires a measure of the model’s uncertainty on each prediction. In both tasks, we simulate manual annotation a sentence at a time, so we measure uncertainty at the sentence
level. In negation detection, we use the predicted entropy at RoBERTa’s special sentence-initial token, <s>. In time expression recognition, we use the average of the predicted entropies of the tokens in the sentence.

5.3 Data Augmentation

Inspired by Miao et al. (2020), we use a pool-based data augmentation method to automatically increase the size of the training set.

In negation detection, we extract all the events from the target domain test set to construct our data augmentation pool. For each example to be augmented, we substitute it with $n$ randomly-sampled events from the pool for its entity type. For example, if data augmentation is performed on the sentence: *Has no diarrhea* $<$e/$>$, we replace the *diarrhea* with random events from the pool, resulting in sentences like *Has no asthma* $<$e/$>$.

In time expression recognition, we manually generate a candidate pool for each time entity type from the guidelines of the SCATE annotation schema. We remove the entities from the pool that do not appear in the target domain test set. For each entity in the example to be augmented, we substitute it with $n$ randomly-sampled entities from the pool for its entity type. For example, in the sentence, *the patient underwent appendicitis surgery on August 29, 2018*, there are three time entities (August: Month-Of-Year, 29: Day-Of-Month, 2018: Year). Data augmentation can therefore generate up to $n \times 3$ sentences with different years, months, and days, e.g., *the patient underwent appendicitis surgery on September 1st, 2017*.

6 Experiments

For both tasks, we follow the conventional RoBERTa input format (insert special tokens $<$s$>$ and $<$e/$>$ at the beginning and end of the input sequence, respectively). We used the sentencizer from Spacy (Honnibal et al., 2020) to split the documents in time expression recognition data into sentences. The input to the source-domain models for both task is a sentence. The output for the negation detection model is a sentence label (negated or not negated), and the output for the time expression model is token labels (time entity types).

When we continue training the source-domain model on the target domain, we keep the same hyperparameters as were used when the shared task organizers trained the models on the source domains. In the source-free domain adaptation setting, there is no (or very little) labeled development data available, so it is not possible to tune hyperparameters. All hyperparameters are shown in tables 4 and 5 in appendix A.

For each target domain, we consider two models for comparison with our adapted models: the unadapted source-domain model (baseline model) and the source-domain model fine-tuned on all labeled data in the development set (oracle model).

In self-training, we set the threshold $\tau$ to 0.95, and experiment with running just a single iteration and with running 30 iterations with the different $S_D$ and $S_M$ strategies. In active learning, we set our annotation budget to 96 sentences, and experiment with spending these 96 sentences at once and in 8 iterations with the different $S_D$ and $S_M$ strategies. For all experiments, we run one version with data augmentation (with $n = 5$) and one without.

---

Training may run for fewer iterations when the stopping conditions are met.
### Table 2: The performance of different domain adaptation strategies on the negation detection target domains.

| #  | Strategy | Negation: MIMIC-III | Negation: i2b2 |
|----|----------|----------------------|----------------|
|    |          | F  | P  | R  | F  | P  | R  |
| 1  | SD (baseline) | 0.656 | 0.921 | 0.510 | 0.837 | 0.855 | 0.820 |
| 2  | SD + FT (Oracle) | 0.868 | 0.875 | 0.862 | 0.925 | 0.928 | 0.922 |

### Active Learning

| #  | Strategy          | F  | P  | R  | F  | P  | R  |
|----|-------------------|----|----|----|----|----|----|
| 3  | AL (96 x 1)       | 0.759 | 0.901 | 0.656 | 0.886 | 0.943 | 0.836 |
| 4  | AL (12 x 8) + Rmodel + Idata | 0.800 | 0.828 | 0.774 | 0.891 | 0.951 | 0.838 |
| 5  | AL (12 x 8) + Rmodel + Rdata | 0.618 | 0.842 | 0.489 | 0.778 | 0.972 | 0.649 |
| 6  | AL (12 x 8) + Imodel + Idata | 0.817 | 0.867 | 0.773 | 0.859 | 0.852 | 0.865 |
| 7  | AL (12 x 8) + Imodel + Rdata | 0.777 | 0.890 | 0.689 | 0.877 | 0.928 | 0.831 |

### Active Learning + Data Augmentation

| #  | Strategy          | F  | P  | R  | F  | P  | R  |
|----|-------------------|----|----|----|----|----|----|
| 8  | AL (90 x 1) + DA (5) | 0.708 | 0.652 | 0.773 | 0.883 | 0.937 | 0.834 |
| 9  | AL (12 x 8) + Rmodel + Idata + DA (5) | 0.805 | 0.803 | 0.806 | 0.891 | 0.960 | 0.831 |
| 10 | AL (12 x 8) + Rmodel + Rdata + DA (5) | 0.586 | 0.489 | 0.730 | 0.817 | 0.960 | 0.710 |
| 11 | AL (12 x 8) + Imodel + Idata + DA (5) | 0.805 | 0.878 | 0.744 | 0.881 | 0.925 | 0.841 |
| 12 | AL (12 x 8) + Imodel + Rdata + DA (5) | 0.745 | 0.882 | 0.645 | 0.889 | 0.929 | 0.852 |

### Self-training

| #  | Strategy          | F  | P  | R  | F  | P  | R  |
|----|-------------------|----|----|----|----|----|----|
| 13 | ST (1)            | 0.677 | 0.916 | 0.537 | 0.854 | 0.871 | 0.838 |
| 14 | ST (30) + Rmodel + Idata | 0.679 | 0.937 | 0.533 | 0.857 | 0.876 | 0.839 |
| 15 | ST (30) + Rmodel + Rdata | 0.695 | 0.912 | 0.562 | 0.861 | 0.880 | 0.843 |
| 16 | ST (30) + Imodel + Idata | 0.664 | 0.906 | 0.525 | 0.864 | 0.890 | 0.840 |
| 17 | ST (30) + Imodel + Rdata | 0.654 | 0.879 | 0.521 | 0.858 | 0.883 | 0.834 |

### Self-training + Data Augmentation

| #  | Strategy          | F  | P  | R  | F  | P  | R  |
|----|-------------------|----|----|----|----|----|----|
| 18 | ST (1) + DA (5)   | 0.654 | 0.943 | 0.501 | 0.863 | 0.894 | 0.833 |
| 19 | ST (30) + Rmodel + Idata + DA (5) | 0.000 | 0.000 | 0.000 | 0.861 | 0.887 | 0.838 |
| 20 | ST (30) + Rmodel + Rdata + DA (5) | 0.000 | 0.000 | 0.000 | 0.864 | 0.897 | 0.834 |
| 21 | ST (30) + Imodel + Idata + DA (5) | 0.000 | 0.000 | 0.000 | 0.854 | 0.869 | 0.839 |
| 22 | ST (30) + Imodel + Rdata + DA (5) | 0.000 | 0.000 | 0.000 | 0.855 | 0.885 | 0.827 |

**Discussion**

Tables 2 and 3 show the results of our experiments. As we discuss these tables, we are interested less in the best model for a particular configuration, but rather in which configurations are successful across multiple tasks and domains. This is important because in source-free domain adaptation, there is typically no (or very little) labeled target domain data available for hyperparameter tuning. Therefore, what we need is a universal strategy that does not require careful tuning.

For source-free active learning, IData models (rows 4, 6, 9, and 11 in tables 2 and 3) have higher F1s than the baseline source domain model across all tasks and domains. The active learning IData models are also at least as good as, and typically much better than, the self-training models (rows 13-22 in tables 2 and 3). The RModel+RData models always have the worst F1s of the active learning models (rows 5 and 10 in tables 2 and 3).

For source-free self-training, there is almost always a model that has a higher F1 than the baseline source domain model (the exception is negation on MIMIC III), but there is no single model configuration that outperforms the baseline in all tasks and domains. This means that to use self-training in source-free domain adaptation, you either need to get lucky (as some participants in the shared task did), or you need labeled data to tune the strategies to your domain. In the latter case, you would be better off using the labeled data for active learning.

Data augmentation was also inconsistent, helping in some cases (e.g., self-training time expression recognition on news), and hurting in others (e.g., self-training time expression recognition on food insecurity). As with self-training, this suggests that data augmentation (or at least the variants of it that we explored) is probably not viable for source-free domain adaptation where no labeled data for hyperparameter tuning is available.

It is worth noting that several active learning...
models achieve higher F1s than the “oracle” model that fine-tuned on the full labeled development set (row 6, 8, 9, 11, 12 in table 3 Time: News and row 6, 9, 12 in table 3 Time: Food). This emphasizes a challenge of source-free domain adaptation: more data is not always better data. Since we do not have access to the source domain training data, if we train on too much target domain data the model may start to forget what it learned on the source domain, i.e., “catastrophic forgetting” (McCloskey and Cohen, 1989). In these cases, the active learning models, by selecting a small set of just the most uncertain examples, reap the benefits of knowing something about the target domain without losing what they learned from the source domain.

To summarize and answer the questions we posed in section 4, we make the following suggestions for source-free domain adaptation:

1. If there is sufficient expertise to label the data, use active learning and iteratively adapt the model with the IModel+IData strategy instead of spending the annotation budget all at once. This is the best model without data augmentation in three of the four domains (Negation: MIMIC III, Time: News, Time: Food). Note that expertise is important: Su et al. (2021) found that active learning with non-experts in the face of a complex annotation scheme did not yield performance improvements.

2. Self-training and data augmentation, at least as implemented here, are not good choices for source-free domain adaptation: sometimes they led to gains, and sometimes they led to losses. While a good strategy could be found by labeling some target domain data and performing hyperparameter search, such annotation effort would have a higher payoff if used for active learning instead.

8 Error Analysis

We performed an error analysis to try to determine if different adaptation strategies resulted in differ-
ent types of errors being corrected (as compared to the source domain model). For negation detection we sampled and categorized around 200 errors of the source-domain model for each target domain. When the model failed to predict a negation, we manually categorized the error by the negation cue (*no*, *free*, *absent*, etc.). When the model predicted a negation it should not have, we manually categorized the error into “wrong cue” (there was a negation cue in the sentence but it did not apply to the target event) or “short sentence” (especially on the i2b2 domain, the model liked to predict all short sentences as negated). For time expression recognition, we categorized all errors of the source-domain model by entity type (inside–outside–beginning format) for each target domain.

For both tasks, we then calculated how many of these source-domain model errors the best adapted models continued to make. We plot these analyses as heatmaps. Figure 1 shows the analysis for i2b2 negation. Due to space constraints, the other analyses are in appendix A.2. Across all tasks and domains, we see that the best self-trained models correct errors roughly evenly across source-domain error categories, while the best active learning models correct different errors, more like the oracle (target-trained) model. For example, the oracle model and active learning adapted models correct many more “wrong cue” errors in the negation i2b2 domain, more *denies* and *none* errors in the negation MIMIC III domain, more B-Period and B-Month-Of-Year entities in the time news domain, and more B-Season-Of-Year, I-Season-Of-Year, and B-This entities in the time food domain.

Some error types appear to be only learnable with substantially more data. Only the oracle model is able to correct errors with the *non* and *afebrile* negation cues in the i2b2 domain and with the *hold* negation cue in MIMIC-III domain. This suggests that the source-domain model may be very confident in some types of wrong examples causing them not to be selected in active learning and generating poor pseudo-labels in self-training.

**9 Conclusion**

In this paper, we present a detailed comparison of the use of active learning, self-training and data augmentation to adapt a source-domain model on a target domain when the source-domain training data is unavailable. We identify a specific formulation of source-free active learning that consistently improves performance of the source-domain model. We believe our work highlights the interesting challenges of source-free domain adaptation, and its systematic comparison provides a solid base for future research in this area.
References

Nikolay Arefyev, Boris Sheludko, Alexander Podolskiy, and Alexander Panchenko. 2020. Always keep your target in mind: Studying semantics and improving performance of neural lexical substitution. In Proceedings of the 28th International Conference on Computational Linguistics, pages 1242–1255, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Steven Bethard and Jonathan Parker. 2016. A semantically compositional annotation scheme for time normalization. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16), pages 3779–3786, Portorož, Slovenia. European Language Resources Association (ELRA).

Yee Seng Chan and Hwee Tou Ng. 2007. Domain adaptation with active learning for word sense disambiguation. In Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pages 49–56, Prague, Czech Republic. Association for Computational Linguistics.

Xia Cui and Danushka Bollegala. 2019. Self-adaptation for unsupervised domain adaptation. In Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019), pages 213–222, Varna, Bulgaria. INCOMA Ltd.

Liat Ein-Dor, Alon Halfon, Ariel Gera, Eyal Shnarch, Lena Dankin, Leshem Choshen, Marina Danilevsky, Ranit Aharonov, Yoav Katz, and Noam Slonim. 2020. Active Learning for BERT: An Empirical Study. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 7949–7962, Online. Association for Computational Linguistics.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Yunzhong Hou and Liang Zheng. 2020. Source free domain adaptation with image translation.

Mika Juuti, Tommi Gröndahl, Adrian Flanagan, and N. Asokan. 2020. A little goes a long way: Improving toxic language classification despite data scarcity. In Findings of the Association for Computational Linguistics: EMNLP 2020, pages 2991–3009, Online. Association for Computational Linguistics.

Youngeun Kim, Sungeun Hong, Donghyeon Cho, Hyoungseob Park, and Priyadarshini Panda. 2020. Domain adaptation without source data. CoRR, abs/2007.01524.

Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pages 452–457, New Orleans, Louisiana. Association for Computational Linguistics.

Wouter M. Kouw. 2018. An introduction to domain adaptation and transfer learning. CoRR, abs/1812.11806.

Vinod K. Kurmi, Venkatesh K. Subramanian, and Vinay P. Namboodiri. 2021. Domain impression: A source data free domain adaptation method. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pages 615–625.

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

Egoitz Laparra, Xin Su, Yiyun Zhao, Ozlem Uzuner, Timothy Miller, and Steven Bethard. 2021. Semeval-2021 task 10: Source-free domain adaptation for semantic processing. In Proceedings of the 15th International Workshop on Semantic Evaluation (Semeval).

Egoitz Laparra, Dongfang Xu, Ahmed Elsayed, Steven Bethard, and Martha Palmer. 2018. SemEval 2018 task 6: Parsing time normalizations. In Proceedings of The 12th International Workshop on Semantic Evaluation, pages 88–96, New Orleans, Louisiana. Association for Computational Linguistics.

Rui Li, Qianfen Jiao, Wenming Cao, Hau-San Wong, and Si Wu. 2020. Model adaptation: Unsupervised domain adaptation without source data. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).

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 Proceedings of the 37th International Conference on Machine Learning, volume 119 of Proceedings of Machine Learning Research, pages 6028–6039. PMLR.

Pei Liu, Xuemin Wang, Chao Xiang, and Weiye Meng. 2020. A survey of text data augmentation. In 2020 International Conference on Computer Communication and Network Security (CCNS), pages 191–194.

Y. Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, M. Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. ArXiv, abs/1907.11692.

Nikolaos Malandrakis, Minmin Shen, Anuj Goyal, Shuyang Gao, Abhishek Sethi, and Angeliki Metallinou. 2019. Controlled text generation for data augmentation in intelligent artificial agents. In Proceedings of the 3rd Workshop on Neural Generation
and Translation, pages 98–102, Hong Kong. Association for Computational Linguistics.

Michael McCloskey and Neal J. Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. volume 24 of Psychology of Learning and Motivation, pages 109–165. Academic Press.

David McCloskey, Eugene Charniak, and Mark Johnson. 2006. Effective self-training for parsing. In Proceedings of the Human Language Technology Conference of the NAACL, Main Conference, pages 152–159, New York City, USA. Association for Computational Linguistics.

Zhengjie Miao, Yuliang Li, Xiaolan Wang, and Wang-Chiew Tan. 2020. Snippext: Semi-supervised opinion mining with augmented data. CoRR, abs/2002.03049.

Piyush Rai, Avishek Saha, Hal Daumé, and Suresh Venkatasubramanian. 2010. Domain adaptation meets active learning. In Proceedings of the NAACL HLT 2010 Workshop on Active Learning for Natural Language Processing, pages 27–32, Los Angeles, California. Association for Computational Linguistics.

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, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Sebastian Ruder and Barbara Plank. 2018. Strong bases for neural semi-supervised learning under domain shift. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1044–1054, Melbourne, Australia. Association for Computational Linguistics.

Gözde Gül Şahin and Mark Steedman. 2018. Data augmentation via dependency tree morphing for low-resource languages. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 5004–5009, Brussels, Belgium. Association for Computational Linguistics.

Roshni Sahoo, Divya Shanmugam, and John V. Guttag. 2020. Unsupervised domain adaptation in the absence of source data. CoRR, abs/2007.10233.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Burr Settles. 2009. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences.

Xin Su, Yiyun Zhao, and Steven Bethard. 2021. The university of arizona at semeval-2021 task 10: Applying self-training, active learning and data augmentation to source-free domain adaptation. In Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval).

Bailin Wang, Wenheng Yin, Xi Victoria Lin, and Caiming Xiong. 2021. Learning to synthesize data for semantic parsing. CoRR, abs/2104.05827.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. 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 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Garrett Wilson and Diane J. Cook. 2020. A survey of unsupervised deep domain adaptation. ACM Trans. Intell. Syst. Technol., 11(5).

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierrick Cistac, Tim Rault, Remi 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 Rush. 2020. Transformers: State-of-the-art natural language processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 38–45, Online. Association for Computational Linguistics.

Silei Xu, Sina Semnani, Giovanni Campagna, and Monica Lam. 2020. AutoQA: From databases to QA semantic parsers with only synthetic training data. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 422–434, Online. Association for Computational Linguistics.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In 33rd Annual Meeting of the Association for Computational Linguistics, pages 189–196, Cambridge, Massachusetts, USA. Association for Computational Linguistics.

Juntao Yu, Mohab Elkaref, and Bernd Bohnet. 2015. Domain adaptation for dependency parsing via self-training. In Proceedings of the 14th International Conference on Parsing Technologies, pages 1–10, Bilbao, Spain. Association for Computational Linguistics.

Wangchunshu Zhou, Tao Ge, Ke Xu, Furu Wei, and Ming Zhou. 2019. BERT-based lexical substitution. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 3368–3373, Florence, Italy. Association for Computational Linguistics.
A Appendix

A.1 Hyperparameters

| Hyperparameter                        | Value |
|---------------------------------------|-------|
| maximum sequence length               | 128   |
| batch size                            | 8     |
| epochs                                | 10    |
| gradient accumulation steps           | 4     |
| learning rate warm up steps           | 0     |
| weight decay                          | 0.0   |
| learning rate                         | 5e-5  |
| adam epsilon                          | 1e-08 |
| maximum gradient norm                 | 1.0   |

Table 4: Hyperparameters for negation detection systems.

| Hyperparameter                        | Value |
|---------------------------------------|-------|
| maximum sequence length               | 271   |
| batch size                            | 2     |
| epochs                                | 3     |
| gradient accumulation steps           | 1     |
| learning rate warm up steps           | 500   |
| weight decay                          | 0.01  |
| learning rate                         | 5e-5  |
| adam epsilon                          | 1e-08 |
| maximum gradient norm                 | 1.0   |

Table 5: Hyperparameters for time expression recognition systems.

A.2 Heat Maps for Error Analysis

Figure 2: Negation i2b2 target domain error heat map. Source is source-domain model. Oracle is oracle model. AL is the best performing active learning model. ALDA is the best performing active learning with data augmentation model. ST is the best self-training model. STDA is the best self-training with data augmentation model. The numbers in parentheses are the F1 scores of the models.
Figure 3: Negation MIMIC-III target domain error heat map. Source is source-domain model. Oracle is oracle model. ALDA is the best performing active learning model. AL is the best performing active learning with data augmentation model. ST is the best self-training model. STDA is the best self-training with data augmentation model. The numbers in parentheses are the F1 scores of the models.

Figure 4: Time news target domain error heat map. Source is source-domain model. Oracle is oracle model. AL is the best performing active learning model. ALDA is the best performing active learning with data augmentation model. ST is the best self-training model. STDA is the best self-training with data augmentation model. The numbers in parentheses are the F1 scores of the models.
Figure 5: Time food security target domain error heat map. Source is source-domain model. Oracle is oracle model. AL is the best performing active learning model. ALDA is the best performing active learning with data augmentation model. ST is the best self-training model. STDA is the best self-training with data augmentation model. The numbers in parentheses are the F1 scores of the models.