A Comparison of Strategies for Source-Free Domain Adaptation

Xin Su Yiyun Zhao Steven Bethard
University of Arizona
Tucson, AZ, USA
{xinsu, yiyunzhao, bethard}@email.arizona.edu

Abstract

Data sharing restrictions are common in NLP, especially in the clinical domain, but there is limited 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 active learning yields consistent gains across all SemEval 2021 Task 10 tasks and domains, but though the shared task saw successful self-trained and data augmented models, our systematic comparison finds these strategies to be unreliable for source-free domain adaptation.

1 Introduction

Deep neural networks achieve high performance in many 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 much 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 (Laparra et al., 2020).

SemEval 2021 Task 10 (Laparra et al., 2021), on source-free domain adaptation, called attention to this challenging but more realistic scenario where labeled source data are not accessible, only the model trained on the source domain data can be shared[1], and little or no labeled target data are available. Participants explored methods including self-training, 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 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 multiple tasks and domains.
2. We identify a formulation of source-free active learning that consistently improves performance of the source-domain model, and sometimes even outperforms 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.

Our code is publicly available.[2]

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 ex-

[1] In general, it is easier to distribute models than raw data. For example, Lehman et al. (2021) found that none of the algorithms they tried could effectively recover protected health information from a pre-trained language model.

[2] github.com/xinsu626/SourceFreeDomainAdaptation

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 8352 - 8367
May 22-27, 2022 ©2022 Association for Computational Linguistics

amples 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 limited NLP research into this challenging but realistic scenario. Though there is partially related research on continual learning (de Masson d'Autume et al., 2019; Sun et al., 2020) and generalization of pre-trained models (Hendrycks et al., 2020), the only work to explicitly test source-free domain adaptation is SemEval 2021 Task 10 (Laparra et al., 2021), which asked participants to perform source-free domain adaptation on negation detection and time expression recognition. A variety of techniques were applied to this task, including active learning, self-training, and data augmentation. However, different techniques were applied by different participants 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) trains a model on a labeled dataset L and then iteratively makes predictions ("pseudo-labels") on an unlabeled dataset U and re-trains. On each iteration, the examples in U that the model labels with high confidence ("silver labels") are 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 selects a small number of examples to be manually annotated, using strategies 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 pre-trained 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 enhances 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 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 tasks of SemEval 2021 Task 10: negation detection and time expression

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

Table 1: Data summary for negation detection and time expression recognition tasks.

recognition. We select these tasks because:

1. They represent real-world data-sharing problems: the negation source-domain data "cannot currently be distributed" and the time expression source-domain data is "difficult to gain access to due to the complex data use agreements" (Laparra et al., 2021). Only the task organizers had access to the data and permission to distribute models trained on the (de-identified) data.
2. The annotation schemes are complex enough that the problem cannot be easily solved by manually annotating the target domain. Su et al. (2021) found that annotations from annotators given only the time annotation guidelines yielded no gains to models, while annotations from heavily trained annotators did yield gains.
3. These two tasks suffer a large performance loss under domain shift: the source-trained model is 15+ points of F1 lower on the target test set than on the source test set (Laparra et al., 2021).

The popular Amazon reviews sentiment analysis dataset (Blitzer et al., 2007) violates the points above: labeled source and target data are easily available, the annotation scheme is easy (it is artificially balanced and removes reviews with neutral labels, as others have noted (He et al., 2018; Miller, 2019)), and the source domain model performs well on the target domain (within 0-4 points of F1). We nonetheless include some experiments on this dataset in appendix A.3. We find that with simple data preprocessing and source-domain hyperparameter tuning, the source-domain model alone outperforms all domain adaptation models from Ye et al. (2020) and Ben-David et al. (2020).

SemEval 2021 Task 10 negation detection is a "span-in-context" 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:

> *Has no <e> diarrhea </e> and no new lumps or masses*

the goal is to predict that *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.

SemEval 2021 Task 10 time expression recognition is a sequence-tagging task. The goal is to identify the time entities in the document and label them with SCATE types (Bethard and Parker, 2016). For example, given the sentence:

> *the patient underwent appendicitis surgery on August 29, 2018*,

the goal is to label *August* as *Month-Of-Year*, *29* as *Day-Of-Month*, and *2018* as *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 and the Famine Early Warning Systems Network.

Each task has a model trained from a source domain and a test set for each of two target domains. For each target domain, we split the data into 20% as a development set and 80% as a test set. Detailed data information is shown in table 1.

Source data We do not use source domain data. We use only the English RoBERTa-base models (Liu et al., 2019) (approx. 125M parameters) that the task organizers fine-tuned on the source domain data sets via the Huggingface Transformers library v3.5.1 (Wolf et al., 2020).

Target development data We use the development data for fine-tuning the model. For active learning, to simulate manual annotation, we fine-tune on a small number of automatically selected labeled examples. For self-training, no labels are used; we fine-tune on predictions (pseudo-labels) generated by the model on the development data. For oracle experiments, we fine-tune the model on all labeled examples in the development set.

Target test data We evaluate on the test data. No fine-tuning is performed. Models always treat this data as unlabeled[3]. Its labels are used only during evaluation. We use the same evaluation metrics as in SemEval 2021 Task 10: precision, recall, and F1 score.

4 Research Questions

We aim for a systematic analysis of three strategies with many different implementations in SemEval 2021 Task 10: self-training, active learning, and data augmentation. Our research questions are:

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

[3]The data augmentation strategies assume that the target test data represents all available unlabeled data, and therefore deterministically restrict their lexicons to words in this data.

Algorithm 1: Source-Free Self-training Algorithm

Input:
M: the source-domain model
D: the unlabeled target domain data
τ: the self-training threshold
T: the maximum number of iterations
S_D: the data construction strategy
S_M: the model training strategy
S_A: the data augmentation strategy

1 $M_0 \leftarrow Copy(M)$
2 $D_0 \leftarrow Copy(D)$
3 $L \leftarrow \emptyset$
4 **for** $i \leftarrow 0$ **to** T **do**
5   **if** $D = \emptyset$ **then**
6   Stop training
7   **if** $S_D = ResetData$ **then**
8         $L = \emptyset$
9         $D = D_0$
10     $L_{C_i} \leftarrow$ $\{(d, M(d))$ for $d \in D$ if $M(d)$ confidence $> \tau\}$
11   **if** $L_{C_i} = \emptyset$ *or* $L_{C_i} = L_{C_{i-1}}$ **then**
12   Stop training
13     $L = L \cup L_{C_i}$
14   **if** $S_D = KeepData$ **then**
15         $D \leftarrow D - \{d$ for $(d, l) \in L_{C_i}\}$
16   **if** $S_A = Augment$ **then**
17         $L \leftarrow L \cup Augment(L_{C_i})$;
18   **if** $S_M = ResetModel$ **then**
19         $M \leftarrow M_0$;
20   Fine-tune M on L;

T the maximum number of iterations for self-training or active learning

S_D the data construction strategy: $KeepData$ to keep the training data from the previous iteration, or $ResetData$ to start anew on each iteration.

S_M the model training strategy: $KeepModel$ to continue training the model from the previous iteration, or $ResetModel$ to continue training from the source-domain model.

S_A whether or not to use data augmentation.

5.1 Source-Free Self-training

Algorithm 1 presents our self-training algorithm. It follows standard self-training (Yarowsky, 1995) in using the model to add pseudo-labels to the unlabeled data (line 10). However, there is no source-domain labeled data, so the model can fine-tune only on the pseudo-labels. The remainder of the code ensures that models and/or data are kept, reset, or augmented as per the selected strategies.

Self-training requires a measure of model 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

Algorithm 2: Source-Free Active Learning Algorithm

Input:
M: the source-domain model
D: the development set of the target domain
T: the maximum number of iterations
K: the number of annotations per iteration
S_D: the data construction strategy
S_M: the model training strategy
S_A: the data augmentation strategy

1 $M_0 \leftarrow Copy(M)$
2 $D_0 \leftarrow Copy(D)$
3 $L \leftarrow \emptyset$
4 **for** $i \leftarrow 0$ **to** T **do**
5   **if** $S_D = ResetData$ **then**
6         $L = \emptyset$
7         $D = D_0$
8     $D_U \leftarrow$ [d for $d \in D$ sorted by uncertainty of $M(d)$]
9     $L_U \leftarrow$ $\{(d, Annotate(d))$ for $d \in$ top K of $D_U\}$
10     $L \leftarrow L \cup L_U$
11   **if** $S_D = KeepData$ **then**
12         $D \leftarrow D - \{d$ for $(d, l) \in L_U\}$
13   **if** $S_A = Augment$ **then**
14         $L \leftarrow L \cup Augment(L_U)$;
15   **if** $S_M = ResetModel$ **then**
16         $M \leftarrow M_0$
17   Fine-tune M on L;

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

Algorithm 2 presents our active learning algorithm. It follows an approach similar to Su et al. (2021). Like most active learning algorithms, the core is to select examples the model is uncertain of (line 8) and then manually annotate them (line 9). 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 model uncertainty on each prediction. In both tasks, we add annotations 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 construct a pool of all event words in the unlabeled target domain test data. For each development data example to be augmented, we substitute its event with n randomly-sampled words from the pool. For example, if data augmentation is performed on the sentence: *Has no <e> diarrhea </e>*, we replace the *diarrhea* with random words from the pool, resulting in sentences like *Has no <e> asthma </e>*.

In time expression recognition, we construct a pool of words for each time entity type using the guidelines of the SCATE annotation schema, excluding words that do not appear in the unlabeled target domain test data. For each entity in a development data example to be augmented, we substitute it with n randomly-sampled words 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

The input to the source-domain models for both tasks is a sentence. The output for the negation detection model is a sentence label (negated or not negated). The output for the time expression model is one label per token (its time entity type). For both tasks, we use the conventional RoBERTa input format, surrounding the sentence with the special tokens <s> and </s>. The negation detection data is already split into sentences. For the time recognition data, we split it into sentences using the English sentencizer from Spacy v2.3.2 (Honnibal et al., 2020).

When we fine-tune the source-domain model on the target domain, we keep the same training hyperparameters as were used when the shared task organizers trained the models on the source domains. In source-free domain adaptation, there is no (or very little) labeled development data available, so it is not possible to tune hyperparameters. All hyperparameters are given in appendix A.1. All experiments are run on a single Nvidia P100 GPU. The total approximate GPU hours are 70 hours.

In self-training, we set the threshold τ 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. The threshold and the number of iterations are adapted from Su et al. (2021). Training may run for fewer iterations when the stopping conditions are met. 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.

For each source and target domain pair, we compare our adapted model with the following models.

1. **Source-Domain Model**: The baseline. It is unadapted, trained only on the source domain.
2. **Fine-Tuned Source-Domain Model**: The oracle. It is fine-tuned on the target domain using the entire labeled development set.
3. **Self-Distilled Model**: A RoBERTa-base model fine-tuned on the development set using pseudo labels generated by the source-domain model.
4. **Passive Learning Model**: The source-domain model fine-tuned on 96 randomly sampled examples from the labeled development set.

7 Discussion

Tables 2 and 3 show the results of our experiments. 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 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, we find that even small amounts of annotated data are useful, and that smart data selection (e.g., using uncertainty scores) is usually helpful. The active learning KeepData models (rows 6, 8, 11, and 13 in tables 2 and 3) have higher F1s than the baseline source domain models across all tasks and domains (0.054 F1 higher on average). Active learning KeepData models also outperform passive learning models (that randomly select data) in 14 out of 16 cases, and are at least as good as, and typically much better than, the self-training models (rows 15-24 in tables 2 and 3). The ResetModel+ResetData models always have the worst F1s of the active learning models (rows 7 and 12 in tables 2 and 3).

Several active learning models achieve higher F1s than the "oracle" model that fine-tuned on the full labeled development set (row 8, 10, 11, 13, 14 in table 3 Time: News and row 8, 11, 14 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 fine-tune 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.

For source-free self-training, we find that iteratively updating both model and data is slightly above baseline, and that it is better to start from the source-domain model than from RoBERTa without fine-tuning. The KeepModel+KeepData (without data augmentation) is slightly above the source-domain model across all tasks and domains (0.013 F1 higher on average). Every other configuration, even if they outperform KeepModel+KeepData in one task or domain, is below the source-domain baseline in another. All self-trained models without data augmentation (which start from the source-domain model) do at least outperform self-distilled models (which start from the RoBERTa model without fine-tuning; row 3 in tables 2 and 3). The small gains from the only self-training configuration that consistently outperformed the source-domain model suggest that self-training may not be worthwhile for source-free domain adaptation.

Data augmentation helped in some cases (e.g., self-training time expression recognition on news), and hurt in others (e.g., self-training time expression recognition on food security). Data augmentation sometimes led to ill-behaving models: on the negation MIMIC-III dataset, data augmentation made the self-trained model predict all examples as not negated resulting in 0.000 F1 (rows 21 -24 in table 2: Negation-MIMIC-III). 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 tuning strategies is available.

We thus 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 KeepModel+KeepData strategy instead of spending the annotation budget all at

#	Strategy	Negation: MIMIC-III F	P	R	Negation: i2b2 F	P	R
1	Source-Domain Model (baseline)	0.656	0.921	0.510	0.837	0.855	0.820
2	Fine-Tuned Source-Domain Model (oracle)	0.868	0.875	0.862	0.925	0.928	0.922
3	Self-Distilled Model	0.623	0.825	0.501	0.846	0.849	0.842
4	Passive Learning Model	0.722	0.792	0.663	0.882	0.914	0.853
Active Learning							
5	AL (96 × 1)	0.759	0.901	0.656	0.886	0.943	0.836
6	AL (12 × 8) + ResetModel + KeepData	0.800	0.828	0.774	**0.891**	0.951	0.838
7	AL (12 × 8) + ResetModel + ResetData	0.618	0.842	0.489	0.778	0.972	0.649
8	AL (12 × 8) + KeepModel + KeepData	**0.817**	0.867	0.773	0.859	0.852	0.865
9	AL (12 × 8) + KeepModel + ResetData	0.777	0.890	0.689	0.877	0.928	0.831
Active Learning + Data Augmentation							
10	AL (96 × 1) + DA (5)	0.708	0.652	0.773	0.883	0.937	0.834
11	AL (12 × 8) + ResetModel + KeepData + DA (5)	**0.805**	0.803	0.806	**0.891**	0.960	0.831
12	AL (12 × 8) + ResetModel + ResetData + DA (5)	0.586	0.489	0.730	0.817	0.960	0.710
13	AL (12 × 8) + KeepModel + KeepData + DA (5)	**0.805**	0.878	0.744	0.881	0.925	0.841
14	AL (12 × 8) + KeepModel + ResetData + DA (5)	0.745	0.882	0.645	0.889	0.929	0.852
Self-training							
15	ST (1)	0.677	0.916	0.537	0.854	0.871	0.838
16	ST (30) + ResetModel + KeepData	0.679	0.937	0.533	0.857	0.876	0.839
17	ST (30) + ResetModel + ResetData	**0.695**	0.912	0.562	0.861	0.880	0.843
18	ST (30) + KeepModel + KeepData	0.664	0.906	0.525	**0.864**	0.890	0.840
19	ST (30) + KeepModel + ResetData	0.654	0.879	0.521	0.858	0.883	0.834
Self-training + Data Augmentation							
20	ST (1) + DA (5)	**0.654**	0.943	0.501	0.863	0.894	0.833
21	ST (30) + ResetModel + KeepData + DA (5)	0.000	0.000	0.000	0.861	0.887	0.838
22	ST (30) + ResetModel + ResetData + DA (5)	0.000	0.000	0.000	**0.864**	0.897	0.834
23	ST (30) + KeepModel + KeepData + DA (5)	0.000	0.000	0.000	0.854	0.869	0.839
24	ST (30) + KeepModel + ResetData + DA (5)	0.000	0.000	0.000	0.855	0.885	0.827

Table 2: Performance of domain adaptation strategies on the negation detection target domains. AL ($k \times i$) is active learning with k samples and i iterations. ST (i) is self-training up to i iterations. DA (n) is augmenting each example with up to n new examples. The best scores are in bold and the worst scores are underlined.

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.
3. Active learning is better than passive learning: smart example selection is better than random example selection.
4. Self-training is better than self-distillation: the models benefit from the task knowledge learned from the source-domain.

Our systematic analysis allowed us to make the above more specific suggestions than the shared task's main suggestion that "the best performing [systems] incorporated. . . active-learning, handcrafted heuristics or semiautomatically building a training set" (Laparra et al., 2021).

8 Error Analysis

We performed an error analysis to try to determine if different adaptation strategies resulted in different 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 cate-

#	Strategy	Time: News F	Time: News P	Time: News R	Time: Food F	Time: Food P	Time: Food R
1	Source-Domain Model (baseline)	0.771	0.772	0.770	0.781	0.834	0.734
2	Fine-Tuned Source-Domain Model (oracle)	0.844	0.826	0.864	0.851	0.841	0.861
3	Self-Distilled Model	0.572	0.590	0.555	0.766	0.831	0.711
4	Passive Learning Model	0.796	0.783	0.809	0.770	0.755	0.785
Active Learning							
5	AL (96 × 1)	0.812	0.800	0.825	0.819	0.821	0.818
6	AL (12 × 8) + ResetModel + KeepData	0.812	0.794	0.830	0.842	0.844	0.840
7	AL (12 × 8) + ResetModel + ResetData	0.771	0.771	0.770	0.781	0.832	0.737
8	AL (12 × 8) + KeepModel + KeepData	**0.861**	0.844	0.879	**0.872**	0.866	0.879
9	AL (12 × 8) + KeepModel + ResetData	0.772	0.758	0.787	0.781	0.797	0.765
Active Learning + Data Augmentation							
10	AL (96 × 1) + DA (5)	0.856	0.829	0.884	0.840	0.824	0.855
11	AL (12 × 8) + ResetModel + KeepData + DA (5)	**0.860**	0.830	0.893	**0.856**	0.840	0.873
12	AL (12 × 8) + ResetModel + ResetData + DA (5)	0.790	0.748	0.836	0.793	0.782	0.805
13	AL (12 × 8) + KeepModel + KeepData + DA (5)	0.849	0.820	0.881	0.841	0.821	0.863
14	AL (12 × 8) + KeepModel + ResetData + DA (5)	0.853	0.828	0.879	**0.856**	0.831	0.881
Self-training							
15	ST (1)	0.753	0.733	0.774	0.777	0.807	0.750
16	ST (30) + ResetModel + KeepData	**0.786**	0.791	0.782	0.780	0.815	0.747
17	ST (30) + ResetModel + ResetData	0.727	0.688	0.770	0.787	0.815	0.761
18	ST (30) + KeepModel + KeepData	0.784	0.777	0.792	0.786	0.832	0.745
19	ST (30) + KeepModel + ResetData	0.633	0.551	0.743	**0.789**	0.829	0.752
Self-training + Data Augmentation							
20	ST (1) + DA (5)	**0.800**	0.794	0.805	0.756	0.787	0.726
21	ST (30) + ResetModel + KeepData + DA (5)	0.789	0.790	0.788	0.754	0.780	0.730
22	ST (30) + ResetModel + ResetData + DA (5)	0.795	0.792	0.798	0.765	0.788	0.744
23	ST (30) + KeepModel + KeepData + DA (5)	0.794	0.801	0.788	**0.759**	0.786	0.734
24	ST (30) + KeepModel + ResetData + DA (5)	0.797	0.791	0.802	0.747	0.771	0.724

Table 3: Performance of domain adaptation strategies on the time expression recognition target domains. AL ($k \times i$) is active learning with k samples and i iterations. ST (i) is self-training up to i iterations. DA (n) is augmenting each time entity with up to n new examples. The best scores are in bold and the worst scores are underlined.

gorized 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. Heatmaps of these analyses are plotted 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-fine-tuned) 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.

Acknowledgements

Research reported in this publication was supported by the National Library of Medicine of the National Institutes of Health under Award Numbers R01LM012918 and R01LM010090. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

Ethical Considerations

Our comparison experiments and proposed formulation are intended to encourage model sharing in source-free domain adaptation while avoiding the risk of privacy leakage caused by direct data sharing. The data we use in this experiment are publicly available and from a shared task, however some of that data is from health institutions and requires a data use agreement to work with the data. Though recent research has found it difficult to recover protected information from trained models (Lehman et al., 2021), there is still some small risk that more complex models may be able to do so. However, as our research is a comparative study, we are not directly releasing models, and thus not risking any release of protected health information.

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.

Eyal Ben-David, Carmel Rabinovitz, and Roi Reichart. 2020. PERL: Pivot-based domain adaptation for pre-trained deep contextualized embedding models. *Transactions of the Association for Computational Linguistics*, 8:504–521.

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).

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, Bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 440–447, Prague, Czech Republic. Association for Computational Linguistics.

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.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

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.

Cyprien de Masson d'Autume, Sebastian Ruder, Lingpeng Kong, and Dani Yogatama. 2019. Episodic memory in lifelong language learning. In *NeurIPS*.

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.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2018. Adaptive semi-supervised learning for cross-domain sentiment classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3467–3476, Brussels, Belgium. Association for Computational Linguistics.

Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. 2020. Pretrained transformers improve out-of-distribution robustness. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2744–2751, 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, Özlem 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-2021)*, pages 348–356, Online. Association for Computational Linguistics.

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.

Eric Lehman, Sarthak Jain, Karl Pichotta, Yoav Goldberg, and Byron Wallace. 2021. Does BERT pretrained on clinical notes reveal sensitive data? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 946–959, Online. 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 191–195.

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 90–98, Hong Kong. Association for Computational Linguistics.

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

David McClosky, 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.

Timothy Miller. 2019. Simplified neural unsupervised domain adaptation. 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 414–419, Minneapolis, Minnesota. Association for Computational Linguistics.

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 baselines 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-2021)*, pages 458–466, Online. Association for Computational Linguistics.

Fan-Keng Sun, Cheng-Hao Ho, and Hung-Yi Lee. 2020. LAMAL: LAnguage modeling is all you need for lifelong language learning. In *International Conference on Learning Representations*.

Bailin Wang, Wenpeng 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, Pierric 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.

Hai Ye, Qingyu Tan, Ruidan He, Juntao Li, Hwee Tou Ng, and Lidong Bing. 2020. Feature adaptation of pre-trained language models across languages and domains with robust self-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7386–7399, Online. 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.

Yftah Ziser and Roi Reichart. 2017. Neural structural correspondence learning for domain adaptation. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 400–410, Vancouver, Canada. Association for Computational Linguistics.

A Appendix

A.1 Hyperparameters

For both tasks, when we continue training the source-domain model on the target domain, we keep the same training hyperparameters as were used when the shared task organizers trained the models on the source domains. Those hyperparameters are shown in tables A1 and A2.

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 A1: 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 A2: Hyperparameters for time expression recognition systems.

A.2 Heat Maps for Error Analysis

For both tasks, we calculated how many source-domain model errors the best adapted models continued to make, and plotted them as heatmaps, where the rows are types of errors, and the columns are different models. Figures A1 to A4 show these analyses.

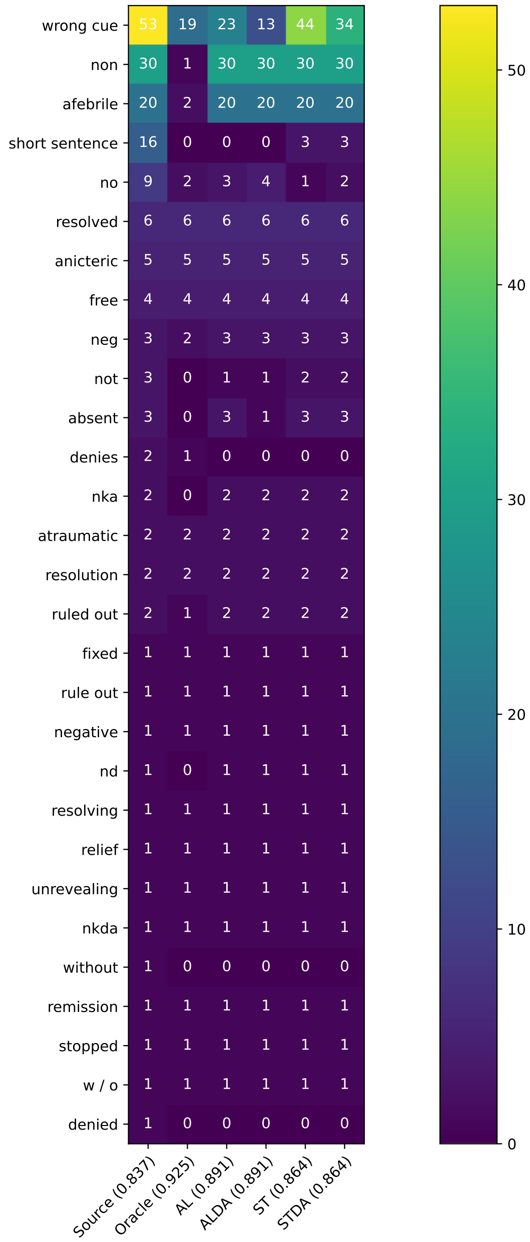

Figure A1: Negation i2b2 target domain error heat map. Source is source-domain model. Oracle is oracle model. AL is the best performing active learning model. ALDA is the best performing active learning with data augmentation model. ST is the best self-training model. STDA is the best self-training with data augmentation model. The numbers in parentheses are the F1 scores of the models.

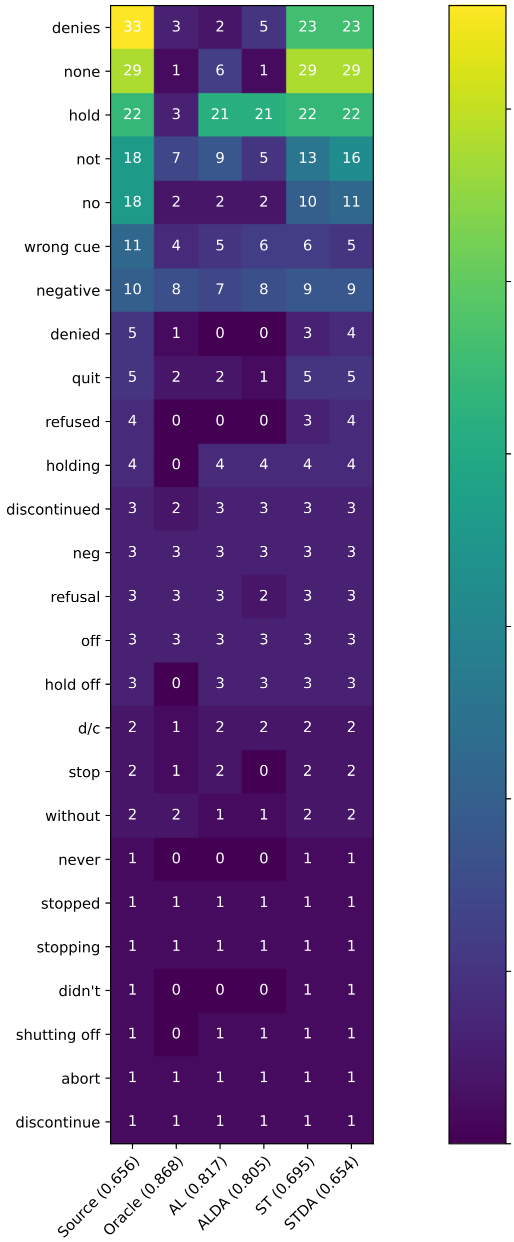

Figure A2: Negation MIMIC-III target domain error heat map. Source is source-domain model. Oracle is oracle model. AL is the best performing active learning model. ALDA is the best performing active learning with data augmentation model. ST is the best self-training model. STDA is the best self-training with data augmentation model. The numbers in parentheses are the F1 scores of the models.

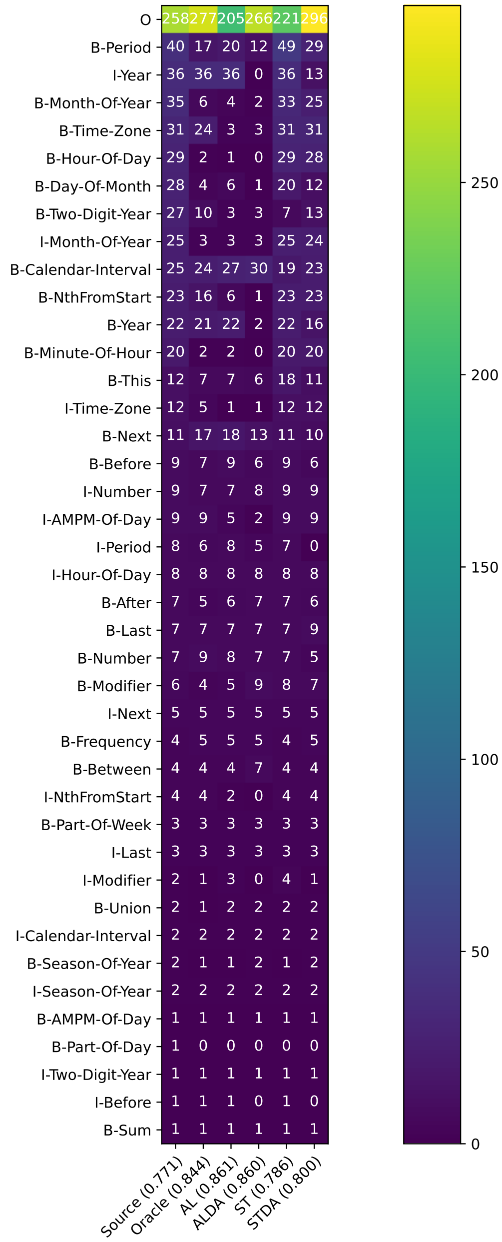

Figure A3: Time news target domain error heat map. Source is source-domain model. Oracle is oracle model. AL is the best performing active learning model. ALDA is the best performing active learning with data augmentation model. ST is the best self-training model. STDA is the best self-training with data augmentation model. The numbers in parentheses are the F1 scores of the models.

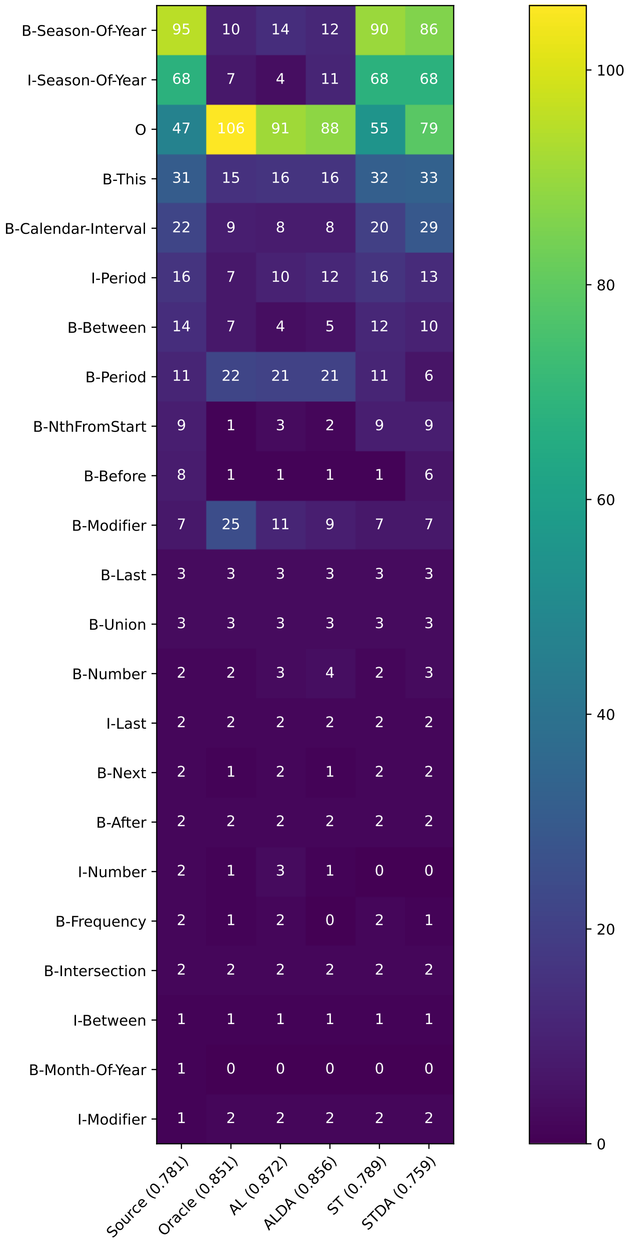

Figure A4: Time food security target domain error heat map. Source is source-domain model. Oracle is oracle model. AL is the best performing active learning model. ALDA is the best performing active learning with data augmentation model. ST is the best self-training model. STDA is the best self-training with data augmentation model. The numbers in parentheses are the F1 scores of the models.

Strategy	B→D	B→E	B→K	D→B	D→E	D→K	E→B	E→D	E→K	K→B	K→D	K→E
Source-Domain Model (baseline)	88.5	92.0	93.8	90.2	91.7	90.7	89.0	89.2	93.5	92.0	90.5	94.8
Fine-Tuned Source-Domain Model (oracle)	89.7	**93.0**	94.5	91.5	**93.5**	94.3	**93.2**	91.0	94.0	92.2	90.5	94.3
Self-Distilled Model	88.0	91.7	**95.5**	**92.5**	90.5	93.0	89.2	90.5	94.0	90.5	90.0	92.5
Passive Learning Model	86.5	92.5	92.5	91.5	89.2	91.2	90.0	90.2	93.2	91.5	89.7	91.2
Best model from Ye et al. (2020)	87.9	91.3	92.5	91.5	91.6	92.5	88.7	88.2	93.6	89.8	87.9	92.6
Active Learning												
AL (96 x 1)	87.7	90.2	92.7	90.7	91.0	93.0	90.2	90.7	93.2	91.7	90.0	93.8
AL (12 X 8) + KeepModel + KeepData	88.2	90.0	91.0	90.2	90.5	**94.8**	91.0	88.2	94.0	89.7	91.0	92.7
AL (12 X 8) + KeepModel + ResetData	87.5	**93.0**	79.0	83.5	90.5	91.0	86.8	78.5	89.0	85.3	83.8	89.5
AL (12 X 8) + ResetModel + KeepData	87.5	92.2	93.5	**92.5**	91.2	94.0	91.2	89.0	94.5	91.0	89.2	**94.8**
AL (12 X 8) + ResetModel + ResetData	75.0	84.0	67.2	91.7	62.5	90.0	89.2	87.5	91.0	**93.0**	69.0	94.5
Self-training												
ST (1)	87.5	91.7	94.3	91.5	90.5	92.5	90.2	91.7	92.5	91.5	91.5	94.3
ST (30) + KeepModel + KeepData	87.5	92.5	94.0	90.5	91.0	92.0	89.5	89.5	94.5	90.2	89.7	93.2
ST (30) + KeepModel + ResetData	**90.0**	91.2	94.3	91.2	90.2	92.7	90.7	90.5	94.5	91.2	90.5	93.5
ST (30) + ResetModel + KeepData	88.2	91.0	94.3	91.7	91.0	91.7	90.7	**92.2**	**95.3**	91.0	**92.0**	92.7
ST (30) + ResetModel + ResetData	89.0	92.5	94.0	90.7	90.5	92.2	90.0	90.7	94.8	91.5	91.2	94.3

Table A3: Accuracy on the Amazon benchmark dataset from Ye et al. (2020). B is Books. D is DVDs. E is Electronics. K is Kitchen. The bolded score is the highest score for the entire column. The underlined score is the worst score for the entire column.

A.3 Results on Amazon Benchmark

The Amazon Sentiment Analysis dataset has been used as a domain adaptation benchmark dataset by a large number of previous works (Blitzer et al., 2007; Ziser and Reichart, 2017; He et al., 2018; Ye et al., 2020; Ben-David et al., 2020). The data consists of reviews of four different product types (domains): Books, DVDs, Electronics, and Kitchen appliances. For the labeled portion, there are 1000 positive reviews and 1000 negative reviews for each domain. From these 4 domains, we construct 12 source-free domain adaptation tasks. For better comparison we directly use the data and split from the software release of Ye et al. (2020). The data of each source domain is split into 80% as source-domain training set and 20% as source-domain development set. The source-domain model is trained on the source-domain training set and its hyperparameters are tuned using the source-domain development set. The data of each target domain is split into 80% as target-domain development set and 20% as target-domain test set. The use of target-domain development set and target-domain test set is the same as in section 3.

When training the source-domain model, we used RoBERTa-base as a starting point and used grid search to tune the hyperparameters within the space of:

Learning Rate (Adam): 1e-5, 2e-5, 3e-5
Batch Size: 8
Gradient Accumulation Steps: 2, 4
Epochs: 10

Table A3 shows the results of these 12 source-free domain adaptations. In 9 of 12 cases, our unadapted source-domain models score higher than the best adaptation model from Ye et al. (2020) [4] . The gap between these unadapted source-domain models and the fully target-domain adapted (oracle) models is also very small: the average difference is only 1.3 points, much smaller than the 11.1 point average difference in tables 2 and 3. In essence, no domain adaptation is needed for this data, so it is a poor dataset for evaluating source-free domain adaptation. Unsurprisingly, we thus see no source-free domain adaptation models that consistently improve performance, though we do see that the active learning ResetData models are typically poor, as they were in tables 2 and 3.

To make sure that it is not a specific split or a smaller test set that leads to good source-domain models, we also use the data from Ben-David et al. (2020) to train and test the source-domain models again. The source-domain data split and usage here is the same as before. The only difference is that there is no target-domain development set and the entire target domain is used as a test set. We show the results in table A4. All source-domain models outperform the best adapted models from Ben-David et al. (2020). It is worth noting that when we

[4] The model used in Ye et al. (2020) is XLM-R(Conneau et al., 2020).

Strategy	B→D	B→E	B→K	D→B	D→E	D→K	E→B	E→D	E→K	K→B	K→D	K→E
SD	**91.8**	**93.5**	**95.0**	**93.0**	**93.0**	**94.6**	**92.8**	**90.8**	**94.7**	**92.1**	**90.2**	**94.4**
Best model from Ben-David et al. (2020)	87.8	87.2	90.2	85.6	89.3	90.4	84.3	85.0	91.2	83.0	85.6	91.2

Table A4: Accuracy on the Amazon benchmark dataset from Ben-David et al. (2020). B is Books. D is DVDs. E is Electronics. K is Kitchen. The bolded score is the highest score for the entire column. The underlined score is the worst score for the entire column.

trained the source-domain model, we found that a large number of punctuation and special symbols included in the data from Ben-David et al. (2020) caused severe overfitting of the model (accuracy is 1 on the source-domain development set). After removing these symbols, the problem was resolved.

A.4 Other Experimented Methods

We also tried to adapt the source-domain model by continuing to pre-train it with masked language modeling on the target domain. We removed the classification layer of the source-domain model, replaced it with a randomly initialized masked language modeling layer, then trained the language model on the unlabeled target-domain data, and then replaced the masked language modeling layer with the original classification layer. The hope was that this would bring the internal representations of the source-domain model closer to the target domain. However, despite a number of attempts at pre-training both all layers and selected layers, performance of this model was always much worse than the source-domain model. In the future, we plan to experiment with different initialization methods for the masked language model layer.