BLCUFIGHT at SemEval-2021 Task 10: Novel Unsupervised Frameworks For Source-Free Domain Adaptation

Weikang Wang, Yi Wu, Yixiang Liu, Pengyuan Liu
Beijing Language and Culture University, Beijing, China
{978955719wwk, wyclover51}@gmail.com
lyx_blcu@163.com, liupengyuan@pku.edu.cn

Abstract

Domain adaptation assumes that samples from source and target domains are freely accessible during a training phase. However, such assumption is rarely plausible in the real-world and may causes data-privacy issues, especially when the label of the source domain can be a sensitive attribute as an identifier. SemEval-2021 task 10 focuses on these issues. We participate in the task and propose novel frameworks based on self-training method. In our systems, two different frameworks are designed to solve text classification and sequence labeling. These approaches are tested to be effective which ranks the third among all system in subtask A, and ranks the first among all system in subtask B.

1 Introduction

Deep neural networks have achieved remarkable success in a variety of applications across different fields while with huge expense of laborious large-scale training data annotation. To avoid expensive data labeling, domain adaptation(DA) methods was proposed to fully utilize previously labeled datasets and unlabeled data on hand in a transductive manner, which obtained promising results in sentiment analysis, part-of-speech tagging, machine translation, etc. (Glorot et al., 2011; Yang and Eisenstein, 2014; Chu and Wang, 2018)

Unsupervised Domain Adaptation(UDA) aims to reduce the domain shift between labeled and unlabeled target domains. Early works (Blitzer et al., 2006; Pan et al., 2010) learnt domain-invariant features to link the target domain to the source domain. Along with the growing popularity of deep learning, plenty of works benefited from its powerful representation learning ability for domain adaptation. Those methods typically minimized the distribution discrepancy between two domains (Plank et al., 2014), or deployed adversarial training (Ganin and Lempitsky, 2015; Bousmalis et al., 2016; Li et al., 2018).

However, a crucial requirement in the methodology of these methods is that all samples from both domains are freely available during the training process, which is inefficient in data transmission and may violate the data privacy policy. For example, it is not allowed to share tweet texts according to Twitter policies, though tweet IDs can be shared. The situation is even more common in clinical NLP, where patient health information must be protected, and annotations over health text, when released at all, often require the signing of complex data use agreements.

SemEval 2021 task 10 focuses on the problem of source-free domain adaptation for semantic processing. Subtask A of task 10 is negation detection which aims to classify clinical event mentions (e.g., diseases, symptoms, procedures) for whether they are being negated by their context. Traditional systems, such as one of the first algorithms NegEx (Chapman et al., 2001) was based on rules. Subsequently, syntax-based methods were developed (Huang and Lowe, 2007; Mehrabi et al., 2015). In recent years, some researchers explored new generation of transfer learning models (BERT) to solve this task (Khandelwal and Sawant, 2019), outperforming the previous state-of-the-art systems by a significant margin.

Subtask B of task 10 is time expression recognition which aims to find time expressions in text. It is a sequence labeling task as (Laparra et al., 2018) described in their work. A few of works combined traditional machine learning with rules achieved good performances (Olex et al., 2018). Some studies got character-level contextual embeddings (Xu et al., 2019) and applied to this task, yielding major performance improvements over the previous state-of-the-art.

In this paper, we propose two different unsupervised frameworks for each subtask in source-free setting. For negation detection task, we design a framework which obtain pseudo labels with high

Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pages 357–363
Bangkok, Thailand (online), August 5–6, 2021.

confidence by using reliable pseudo labels as prototypes. For time expression recognition, we design an unsupervised teacher-student framework with Mean Teacher.

2 System description

For subtask A, we used a pseudo-labeling training method. To reduce the uncertainty from the pseudo labels, we only chose those with high confidence to fine-tune the model. Finally, we ensembled 5 models to make the model have better robustness and results. For subtask B, we started by data preprocessing. Then, we enlarged the training set with pseudo-labeled sentences, which were predicted on the test set by teacher model. In addition, Mean Teacher helps to generate better pseudo labels. Finally, we used the ensemble model to make predictions and add manual expressions. Each module will be introduced in detail in the following sections.

2.1 subtask A: Negation detection

2.1.1 Pre-processing

Samples in the test data was split by punctuation to a single sentence which included the entity being detected.This was done to avoid the impact of the irrelevant context. All white spaces were removed.

2.1.2 Architecture

For negation detection, we utilized the RoBERTa-base (Liu et al., 2019) pretrained model fine-tuned on the 10,259 instances (902 negated) in the SHARP Seed dataset which is different from the target domain. To adapt the source domain to the target domain, we kept the feature extractor of the source model fixed and trained the classifier module by using pseudo labels with high confidence (He and Zhou, 2011). It aims to learn a domain-specific classifier learning module.

Our model is composed of two parts, the first part is Adaptive Prototype Memory (APM) (Kim et al., 2020), which provides pseudo labels with high confidence for the target model. The second part is the target model where parameters of the feature extractor are fixed, i.e., does not participate in training.The overall architecture of our model is shown in Fig.1

Pseudo Labeling: Pseudo labeling (Lee et al., 2013) was originally proposed for semi-supervised learning. Since Pseudo labeling is a simple and efficient method, it gains popularity in other transductive learning problems like **Domain Adaptation**. The main idea is to label unlabeled data with the maximum predicted probability and perform fine-tuning together with labeled data. For this task, we don't have labeled training data, so our method uses a more reliable pseudo-labels to fine-tune the model.

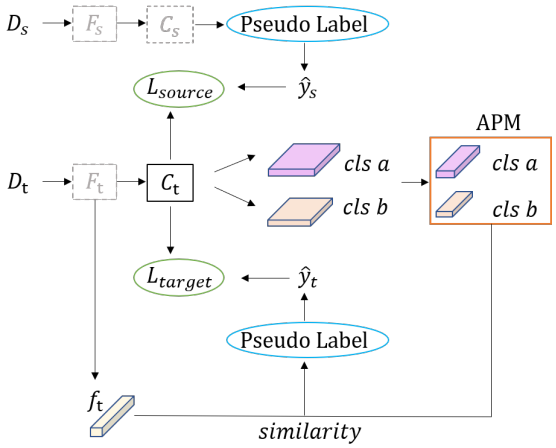

Figure 1: Overall flow of subtask A framework. In the figure, **D**, **F**, **C** and **L** represent Data, Feature extractor Classifier and Loss, respectively. The subscripts **s** and **t** indicate whether they come from the source domain or the target domain. Dashed lines indicate fixed model parameters.

APM: To obtain reliable pseudo labels, prediction uncertainty is measured by self-entropy, i.e., $H(x) = -\sum p(x)log(p(x))$. The smaller the entropy is, the more confidence of the prediction is. First of all, we calculated the normalized self-entropy of target samples.

$$H(x_t) = -\frac{1}{logN_c}(x_t)log(l(x_t))$$

where N_c refers to the number of classes, $l(x_t)$ is the output of the target classifier, x_t represents the samples from the target domain. The next step is to select the reliable part among all target samples, i.e., the part with smaller entropy. In order to minimize the influence of incorrect pseudo labels, we chose 20% as a threshold to get reliable samples. So the top 20% target samples of the smaller entropy are stored in the APM.

Based on prototypes from APM module which can represent each class, we can assign labels to unlabeled target data according to similarity score:

$$S(x_t) = \frac{1}{|M_c|}\sum_{p_c \in M_c}\frac{p_c^T f_t}{||p_c||_2||f_t||_2}$$

where c represents two classes, i.e., "*negated*" or "*not negated*", f_t and p_c stand for the embedded feature of target data and prototype respectively.

Loss Function: Pseudo labels generated by the first part are used to train the classifier of the target model. During the training process, to avoid the influence of unstable pseudo labels, our loss consists of two parts. One is the loss of the source classifier, and the other is the loss of the trainable target classifier.

$$L_{total}(D_t) = (1-\alpha)L_{source}(D_t) + \alpha L_{target}(D_t)$$

At the beginning, loss of the source classifier accounts for a large proportion, it is added for regularization, because the generated label may be unstable. With the increase of training steps, the proportion of source decreases gradually, while the proportion of the loss of the target classifier increases.

2.1.3 Ensembling

To obtain a more robust model, we trained five models by changing the hyper parameters, and integrated the five models by voting ensemble method. Test data were passed through the ensembled model as the final output of the system.

2.2 Subtask B: Time expression recognition

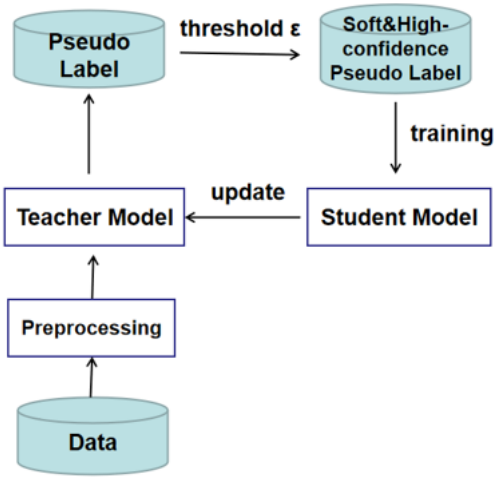

Figure 2: Overall flow of subtask B framework

2.2.1 Pre-processing

- Denoising: Since the first two lines of the development set text are the descriptions of the file name and would not appear in the test set, we removed the first two lines of all verification set texts.
- Training set: Development set and randomly selected partial test set.

2.2.2 Teacher and Student architecture

The main part of our system for subtask B is the teacher-student framework (Liang et al., 2020), which is an unsupervised method. Specifically, The teacher model is initialized by student model. To avoid losing too much information of other classes, we proposed to use soft labels. Recall that for the *n*-th token in the *m*-th sentence, the output probability simplex over *C* classes is denoted as:

$$\left[f_{n,1}\left(X_m;\theta\right),\ldots,f_{n,C}\left(X_m;\theta\right)\right].$$

After the teacher model generated soft labels from training set (let denote $\left\{S_m=[s_{m,n}]_{n=1}^{N}\right\}_{m=1}^{M}$ the soft pseudo-labels generated from teacher model), in order to further address the uncertainty in the data, we selected tokens based on the prediction confidence. That is to say, we selected a set of high confidence tokens from the *m*-th sentence by

$$H_m = \left\{n : \max_c s_{m,n,c} > \epsilon\right\},$$

where $\epsilon \in (0,1)$ is a tuning threshold. The high confidence selection essentially enforces the student model to better fit tokens with high confidence, and therefore is able to improve the model robustness against low-confidence tokens. Loss1 is denoted as:

$$Loss1 = \frac{1}{M}\sum_{m=1}^{M}\ell_{\mathrm{KL}}\left(S_m^{(t)}, f\left(X_m;\theta\right)\right)$$

where $\ell_{KL}(\cdot,\cdot)$ denotes the KL-divergence-based loss:

$$\ell_{kl}(S_m, f(X_m;\theta)) = \frac{1}{|H_m|}\sum_{n\in H_m}\sum_{c=1}^{C} -s_{m,n,c} log f_{n,c}(X_m;\theta).$$

2.2.3 Mean Teacher

In our architecture, we also added Mean Teacher loss to update student model. Mean Teacher (Tarvainen and Valpola, 2017) is a simple but effective method to improve teacher model performance. After the weights of the student model have been updated with gradient descent, the teacher model weights are updated as an moving average of the student weights as follows:

$$\theta'_t = \alpha\theta'_{t-1} + (1-\alpha)\theta_t,$$

where α is a smoothing coefficient hyperparameter. Loss2 denote Mean Teacher loss (same as the formula in 2.2.2), but $\ell_{KL}(\cdot,\cdot)$ denotes:

$$\ell_{kl}(S_m, f(X_m;\theta)) = \frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C} -s_{m,n,c} log f_{n,c}(X_m;\theta).$$

Accordingly, the student model can be optimized by minimizing the loss (consists of Loss1 and Loss2).

The process described above is repeated periodically to train the student model. Eventually, early stopping is adapted to prevent student model from overfitting.

2.2.4 Post-processing

- Ensembling: Ensemble has shown its power on effectively improving the robustness and accuracy of each individual prediction (Opitz and Maclin, 1999; Rokach, 2010). By ensembling predictions from models with different hyper-parameters or architectures, we can get better results than each individual model. In our system, we set different hyper-parameters on learning rate and random seed. In this case, two ensembled model generate predictions individually and we take the union of the two independent predictions as model predictions.

- Manual rules: Through observation on the data, we obtained a set of feature words which appear frequently. Specifically, we labeled feature words "daily" and "annual" with "Calendar-Interval", label "minutes" and "decades" with "Period", label "ago" and "before" with "Before" etc.

3 Experiments

3.1 Data

SemEval 2021 Task 10 released the training, development and test dataset.

For subtaskA, the development data is the i2b2 2010 Challenge Dataset, a de-identified dataset of notes from Partners HealthCare, containing 2886 unlabeled train instances (entities in sentence context), and 5545 dev instances with a corresponding labeling for with negation status. The test data is from the MIMIC III corpus v1.4, which is much messier than the development data. The detailed statistics are shown in Table 1.

	Train	Dev	Test
Positive	-	1115	958
Negative	-	4430	8622
Total	2886	5545	9580

Table 1: Data distribution of subtask A.

For subtask B, we found that the category labels were severely imbalanced. Specifically, the training set and the dev set have label types which are not mutually exclusive. In addition, the dev set labels is mainly distributed in Month-Of-Year, Day-Of-Month, Period, etc., while the test set labels are mainly distributed in Month-Of-Year, Season-Of-Year, Year, etc.

3.2 Evaluation Metrics

F1, Precision and Recall were used to evaluate the performance of both subtask A and subtask B. The evaluation will verify whether the predicted "label" is the same as the desired "label" which is annotated by human workers, and then calculate its F1 scores, precision and recall.

3.3 Experimental Details

Hyper-Parameters of subtask A. Since we were training the model with unlabeled data, we added the same amount of dev and test data as train data to fine-tune the model to get better results. We use an Adam optimizer to tune the parameters with learning rate = 5e-5, max seq length = 128, batch size = 32, seed = 40 and we trained each model for 2 epochs. Then we used the APM module to get $M = 400$ prototypes which represents each class. By computing the similarity between each target sample and all prototypes in APM, we obtained 8658 pseudo-labels with high confidence.

Hyper-Parameters of subtask B. For subtask B, we trained our two ensemble model (each with three models) on unlabeled data with seed = 32,42, learning rate = 2e-5,2.5e-5,3e-5, we trained each model for 5 epochs with early stopping. We also used an AdamW optimizer to tune the parameters with epsilon = 1e-6, batch_size = 16.

4 Results

The performance of our system and the task baselines for both subtasks are shown in following tables.

Model	F1	Precision	Recall
Baseline	0.660	0.917	0.516
Baseline(fine-tuned)	0.730	0.908	0.611
SFDA w/o L_{source}	0.674	0.874	0.548
SFDA w/o APM	0.686	0.927	0.545
SFDA	0.717	**0.936**	0.581
SFDA+ensemble	**0.736**	0.913	**0.616**

Table 2: Results of different ablation experiments for subtask A. All the models are trained on training data, development data and test data.

Table 2 shows the results of several ablation experiments. Compared with models without APM or L_{source}, we found that adding both together improved the performance of the model. The voting ensemble model of single models trained with extra development and test data outperforms all other models and achieves the highest F1 score.

Model	F1(dev)	F1(test)
SFDA(t) w/o L_{source}	0.838	0.661
SFDA(t) w/o APM	0.814	0.717
SFDA(t)	0.859	0.707
SFDA(t)+ensemble	**0.873**	0.720
SFDA(t+d) w/o L_{source}	0.864	0.689
SFDA(t+d) w/o APM	0.851	0.668
SFDA(t+d)	0.868	0.718
SFDA(t+d)+ensemble	0.870	**0.725**

Table 3: Results of models trained on different data sets. SFDA(t) refers to the model trained on train data and SFDA(t+d) represents the model trained on both train data and development data.

Table 3 shows the results of models trained on different data sets. Since we don't need labeled data to train the model, we added development and test data to train the model. Compared with our final model which was trained on three data sets, models trained on fewer data sets, i.e., only on train data or on both train data and development data perform less well.

Figures 3 and 4 show the confusion matrix of the classification results of baseline model and our model on the test dataset. This corresponds to *Baseline* and *SFDA+ensemble* in Table2, respectively. Compared with baseline, we predicted more true positive samples. However, the false prediction of negative samples as positive has increased. As a result, the recall and F1 score of our model have been improved, but the precision has decreased a little.

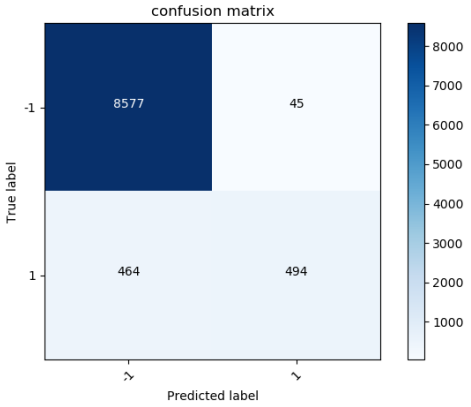

Figure 3: The confusion matrix of the baseline model

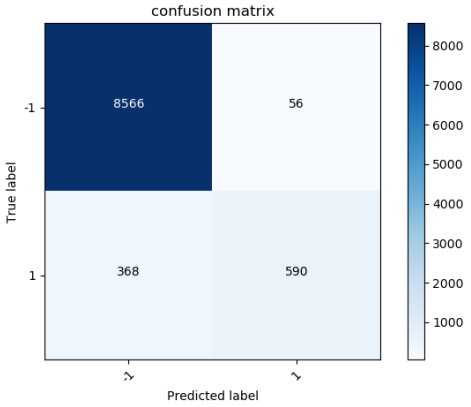

Figure 4: The confusion matrix of our model

Table 4 shows the results of several ablation experiments. Without soft labels, we can find that F1 score drop significantly. A possible explanation is that the soft labels preserve more information and generate better labels. Based on soft labels, MT and manual rules marginally improve the F1 scores. Finally, the ensemble model (MT+soft+rules) outperforms all other models and achieves the highest F1 score.

Table 5 shows the results of our model trained on dev set. Compared with the dev set, the F1 of the test set dropped by an average of 2%.

Error analysis. For subtask A, we conducted statistics and analysis on the classification results of Baseline model and our best model. There are 474 sentences that both Baseline model and our best model predict correctly. The entity being detected usually follow the word that express negative

[1]Model pre-trained on only the source data (official provided).

[2]Model pre-trained on the source data and then fine-tuned on the dev set (official provided).

[3]Here MT refer to Mean Teacher.

Model	F1	Precision	Recall
Baseline[1]	0.794	0.849	0.746
Baseline(fine-tuned)[2]	0.804	0.827	0.782
MT[3]	0.755	0.747	0.763
soft	0.807	0.859	0.761
MT+soft	0.801	0.854	0.754
MT+soft+rules	0.812	**0.863**	0.767
MT+soft+rules(ensemble)	**0.815**	0.847	**0.785**

Table 4: Results of different ablation experiments for subtask B. Our models are trained on training set.

Model	F1(dev)	F1(test)
Baseline	0.746	0.794
Baseline(fine-tuned)	-	0.804
MT	0.767	0.747
soft	0.815	0.791
MT+soft	0.813	0.799
MT+soft(ensemble)	**0.832**	**0.814**

Table 5: Results of subtask B. Our models are trained on dev set.

meanings in these sentences closely. e.g. *"... no <e>erythema </e>...", "... denies <e>chest pain </e>..."* ... There are 116 sentences that our model predicts correctly but the baseline predicts incorrectly. In this part, there are some long-distance keywords or parallel phrases. e.g. *"... No tobacco, EtOH, or <e>IV drug use </e>"* There are 348 sentences that are not predicted correctly by both models. For these, we consider to add some handcraft rules to improve the results of the model.

For subtask B, we conducted a manual error analysis. For the raw text *"during the harvest season"*, both our model and baseline model incorrectly labeled "harvest" with "Season-Of-Year" instead of "harvest season", "harvest" is just the activity, though if it instead said "harvest season", we would annotate that whole thing as a "Season-Of-Year". For the raw text *"February (27,661)"*, baseline model incorrectly label "27" with "Day-Of-Month" while our model didn't, which proves the effectiveness of our architecture. In addition, we list a detailed description of the recall of subtask B in Table 6.

5 Conclusion

We introduce two different frameworks which are both based on self-training method for text classification and sequence labeling in SemEval 2021 task 10, in order to address the problems of source-free, labeled training data scarcity. In subtask A, we used a metric learning method, combining pseudo labeling with prototype network and achieve good results. In subtask B, we employed teacher-student framework, and then we propose to use high-confidence soft labels to further improve the self-training. Our system takes third place in subtask A and first place in subtask B.

In future, we would like to introduce adversarial training and more data augmentation approaches in our model to further facilitate source-free domain adaptation.

Acknowledgments

Thanks for Shucheng Zhu's suggestions on writing this paper. Pengyuan Liu is the corresponding author.

References

John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 conference on empirical methods in natural language processing*, pages 120–128.

Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. 2016. Domain separation networks. *arXiv preprint arXiv:1608.06019*.

Wendy W Chapman, Will Bridewell, Paul Hanbury, Gregory F Cooper, and Bruce G Buchanan. 2001. A simple algorithm for identifying negated findings and diseases in discharge summaries. *Journal of biomedical informatics*, 34(5):301–310.

Chenhui Chu and Rui Wang. 2018. A survey of domain adaptation for neural machine translation. *arXiv preprint arXiv:1806.00258*.

Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *ICML*.

Yulan He and Deyu Zhou. 2011. Self-training from labeled features for sentiment analysis. *Information Processing & Management*, 47(4):606–616.

Yang Huang and Henry J Lowe. 2007. A novel hybrid approach to automated negation detection in clinical radiology reports. *Journal of the American medical informatics association*, 14(3):304–311.

Aditya Khandelwal and Suraj Sawant. 2019. Negbert: A transfer learning approach for negation detection and scope resolution. *arXiv preprint arXiv:1911.04211*.

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

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.

Dong-Hyun Lee et al. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3.

Yitong Li, Timothy Baldwin, and Trevor Cohn. 2018. What's in a domain? learning domain-robust text representations using adversarial training. *arXiv preprint arXiv:1805.06088*.

Chen Liang, Yue Yu, Haoming Jiang, Siawpeng Er, Ruijia Wang, Tuo Zhao, and Chao Zhang. 2020. Bond: Bert-assisted open-domain named entity recognition with distant supervision. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1054–1064.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Saeed Mehrabi, Anand Krishnan, Sunghwan Sohn, Alexandra M Roch, Heidi Schmidt, Joe Kesterson, Chris Beesley, Paul Dexter, C Max Schmidt, Hongfang Liu, et al. 2015. Deepen: A negation detection system for clinical text incorporating dependency relation into negex. *Journal of biomedical informatics*, 54:213–219.

Amy Olex, Luke Maffey, Nicholas Morgan, and Bridget McInnes. 2018. Chrono at semeval-2018 task 6: a system for normalizing temporal expressions. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 97–101.

David Opitz and Richard Maclin. 1999. Popular ensemble methods: An empirical study. *Journal of artificial intelligence research*, 11:169–198.

Sinno Jialin Pan, Xiaochuan Ni, Jian-Tao Sun, Qiang Yang, and Zheng Chen. 2010. Cross-domain sentiment classification via spectral feature alignment. In *Proceedings of the 19th international conference on World wide web*, pages 751–760.

Barbara Plank, Anders Johannsen, and Anders Søgaard. 2014. Importance weighting and unsupervised domain adaptation of pos taggers: a negative result. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 968–973.

Lior Rokach. 2010. Ensemble-based classifiers. *Artificial intelligence review*, 33(1):1–39.

Antti Tarvainen and Harri Valpola. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *arXiv preprint arXiv:1703.01780*.

Dongfang Xu, Egoitz Laparra, and Steven Bethard. 2019. Pre-trained contextualized character embeddings lead to major improvements in time normalization: A detailed analysis. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (* SEM 2019)*, pages 68–74.

Yi Yang and Jacob Eisenstein. 2014. Fast easy unsupervised domain adaptation with marginalized structured dropout. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 538–544.

Type	**R(Baseline)**	**R(Ours)**	**Type**	**R(Baseline)**	**R(Ours)**
Month-Of-Year	0.99	0.993	Between	0.812	0.79
Season-Of-Year	0.055	0.348	Two-Digit-Year	0.1	0
Year	0.988	0.988	Before	0.667	0.818
Number	0.88	0.855	After	0.915	0.872
Last	0.986	0.986	Frequency	0.6	0.6
Calendar-Interval	0.702	0.740	Next	0.839	0.839
Day-Of-Month	1.0	0.875	Intersection	0	0
This	0.528	0.537	Day-Of-Week	1.0	1.0
Period	0.816	0.827	NthFromStart	0	0
Modifier	0.826	0.855	Union	0	0
Part-Of-Day	1.0	1.0			

Table 6: Recall of each type of subtask B.