Adaptive Unsupervised Self-training for Disfluency Detection

Zhongyuan Wang, Yixuan Wang, Shaolei Wang, Wanxiang Che*
Research Center for Social Computing and Information Retrieval
Harbin Institute of Technology, China
{zywang, yixuanwang, slwang, car}@ir.hit.edu.cn

Abstract

Supervised methods have achieved remarkable results in disfluency detection. However, in real-world scenarios, human-annotated data is difficult to obtain. Recent works try to handle disfluency detection with unsupervised self-training, which can exploit existing large-scale unlabeled data efficiently. However, their self-training-based methods suffer from the problems of selection bias and error accumulation. To tackle these problems, we propose an adaptive unsupervised self-training method for disfluency detection. Specifically, we re-weight the importance of each training example according to its grammatical feature and prediction confidence. Experiments on the Switchboard dataset show that our method improves 2.3 points over the current SOTA unsupervised method. Moreover, our method is competitive with the current SOTA supervised method.

1 Introduction

Disfluency is a characteristic of spontaneous speech which is different from written texts. Detecting and removing the non-fluent word sequences in spoken language transcripts can improve the transcripts' quality and provide clean inputs for the downstream NLP tasks, such as parsing, machine translation, and summarization (Tree, 1995; Wang et al., 2020b). As shown in Figure 1, a standard annotation of the disfluency structure indicates the RM (*reparandum*, words that the speaker intends to discard), IM (*interregnum*, filled pauses, discourse cue words, etc.), RP (*repair*, the associated repair) (Shriberg, 1994).

Most previous works (Zayats et al., 2016; Wu et al., 2015; Lou and Johnson, 2017; Jamshid Lou et al., 2018; Zayats and Ostendorf, 2019) on disfluency detection heavily relies on human-annotated corpora, which is difficult and expensive to obtain in practice. Some researchers try to alleviate this issue with, for instance, self-supervised learning (Wang et al., 2020a) and semi-supervised learning techniques (Wang et al., 2018), but they still need a certain amount of human-annotated corpora to perform high performance. Wang et al. (2020b) completely removes the need for human-annotated data. They first perform disfluency detection with the self-training method in an unsupervised manner, which shows promising performance. They leverage the pseudo data constructed by rules to train a weak disfluency detection model as a teacher. Then, the student model is fine-tuned in an unsupervised manner, only using the pseudo-labels generated by the teacher. Finally, the student model achieves promising performance.

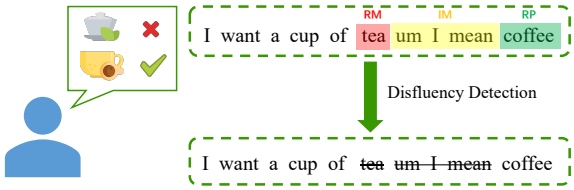

Figure 1: A sentence with disfluencies annotated. RM=Reparandum, IM=Interregnum, RP=Repair. The preceding RM is corrected by the following RP.

Traditional self-training methods utilize the probability estimations to select high-confidence pseudo-labels for re-training, which encounters two problems. (1) *Selection Bias.* The strategy that selects samples with the high confidence pseudo-labeled samples tends to neglect the hard and potentially informative samples. It results in the self-training procedure converging to a suboptimal solution. (2) *Error Accumulation.* The strategy that selects pseudo-labels with probability estimations, limited by the ability of the model, will inevitably introduce a large number of noisy labels and the noise will be accumulated with the increase of iteration times. The above problems become the bottleneck for self-training, which prevents model performance from growing as the number of iter-

*Email corresponding.

Proceedings of the 29th International Conference on Computational Linguistics, pages 7209–7218
October 12–17, 2022.

ations increases. To alleviate the error accumulation problem, Wang et al. (2020b) introduces a grammar checker into the selection process to enhance the ability to select high-quality pseudo-labels, which can be seen as introducing external knowledge. Concretely, Wang et al. (2020b) employed a grammar checker to discard samples that may be noise based on their grammatical correctness. However, their method has no ability to recall potential informative samples, which leaves the selection bias problem unmitigated. In addition, due to the limited performance of the grammar checker, the error accumulation problem still exists to a certain extent.

In this paper, we propose an adaptive self-training method that utilizes a re-weighting strategy to address both the problem of selection bias and error accumulation. Concretely, we assign specific weight to every pseudo-labeled sample based on its quality. We measure the quality of each sample by its grammatical feature and label confidence (see §2 for details). Compared with the method that selects samples with high confidence, our re-weighted method has two main advantages: First, instead of only selecting the pseudo-labeled samples with high confidence, our method makes all samples participate in the training process, which can alleviate the problem of selection bias to some extent. Second, our method can reduce the negative effect of noise data by lowering the weights of low-quality pseudo-labels and emphasizing high-quality pseudo-labels during the self-training process.

Besides, to help improve the performance of self-training, we also propose a more powerful contrastive grammar checker. The grammar checker proposed by Wang et al. (2020b) takes a single sentence as input and then judges whether the sentence is grammatical or not. In our practice, we found that the way of judging grammatical correctness by a single sentence alone faces great difficulties when using the ASR results as input, e.g. there are incomplete sentences with missing beginnings or endings in the ASR results, which may lead to the single-sentence grammar checker to misjudge. Therefore, it is hard to train a high-performance single-sentence grammar checker model. To solve this problem, we employ a contrastive mechanism to make it easier to judge whether the sentence is grammatical or not. We take both the sentence before and after removing disfluency elements as the input of our grammar checker. Therefore, our grammar checker can compare the two sentences to help judge whether the sentence is grammatical or not, which makes our grammar checker powerful.

We evaluate the proposed method on the commonly used English Switchboard dataset, and our method improves 2.3 points over the previous best result in unsupervised settings. Moreover, our method achieves competitive performance compared to the state-of-the-art supervised method which utilizes 60k labeled sentences. In addition, the experimental result on three other datasets shows that our method can consistently achieve competitive performance compared to the supervised systems.

The contributions of our work can be summarized as follows:

- To enhance the performance of unsupervised disfluency detection, we propose an adaptive self-training method utilizing a re-weighting strategy that can alleviate the problems of selection bias and error accumulation in the previous self-training-based method for disfluency detection.
- We propose a more powerful contrastive grammar checker which can better evaluate the quality of the pseudo-labeled data for disfluency detection.
- Experimental results on four datasets demonstrate our proposed method surpasses the performance of the existing SOTA method in unsupervised settings.

We will release our code and model[1].

2 Method

2.1 Procedure Overview

Figure 2 shows an overview of our adaptive unsupervised self-training framework for disfluency detection. Algorithm 1 is presented to help understand our method. Our method only takes unlabeled sentences as inputs, including news data and ASR outputs. We first use self-supervised learning to train the contrastive grammar checker on the large-scale constructed pseudo data (see §2.3 for details), which is used to judge whether a sentence is grammatical or not. Then, we use self-supervised learning to train a weak disfluency detection model with large-scale constructed pseudo data (see §2.2 for details), which is used as the teacher model.

[1] https://github.com/wyxstriker/ReweightingDisfluency

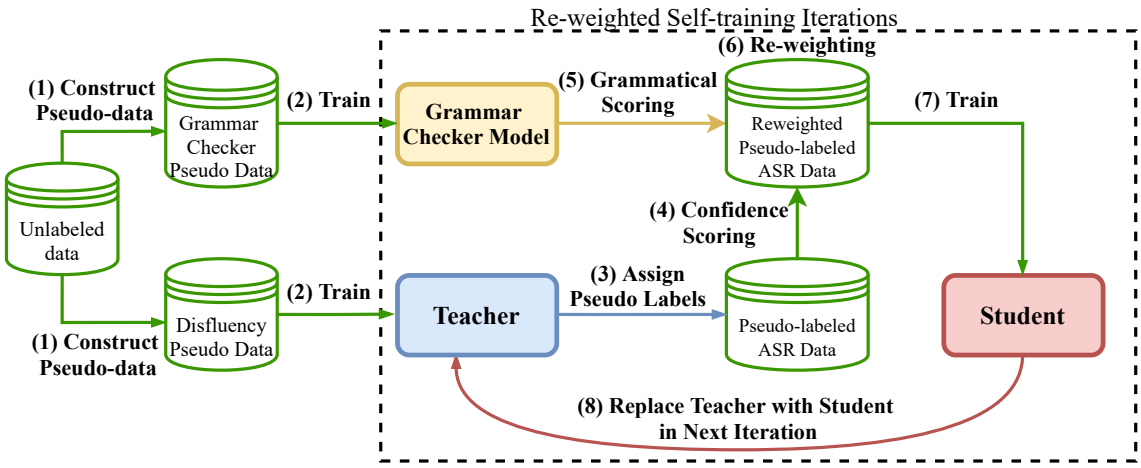

Figure 2: Illustration of our proposed method. Step 5 to Step 8 are looped until the student model performance no longer grows.

Next, we use the teacher model to assign pseudo-labels on unlabeled ASR outputs. We assign specific weight to each pseudo-labeled sentence from two different perspectives: 1) grammatical correctness, we emphasize the sentences that are more grammatical after removing the words with disfluency labels, and 2) pseudo-label confidence, we emphasize the high-confidence pseudo-labeled sentences. To calculate the final weight of each sentence (see §7 for details), we add grammatical correctness score and pseudo-label confidence together. We then train a student model on all pseudo-labeled sentences, which are assigned with different weights calculated in the way mentioned above. Then, the teacher model is replaced with the student model. The process will be looped until the performance of the student model stops growing.

2.2 Teacher Model

We formulate the disfluency detection task as a token-level classification task. The training goal is to detect the disfluency words by associating labels with them. For each token x_i in a sentence, the model assigns a label y_i which is "fluent" or "disfluent". To obtain the teacher model, we directly fine-tune the ELECTRA model (the discriminator) (Clark et al., 2020) by minimizing the cross-entropy loss:

$$loss_{teacher} = \frac{1}{N}\sum_{i=1}^{N} \mathrm{CE}(y_i, f_t(x_i, \theta_t)), \quad (1)$$

where $(x_i, y_i)_{i=1}^{N}$ denotes the input sentences and labels, CE denotes the cross-entropy loss function, f_t denotes our teacher model parameterized by θ_t.

Our teacher model trained on the pseudo data constructed by the self-supervised method. We follow Wang et al. (2020b) to construct the pseudo data. The unlabeled data to construct pseudo data comes from fluent news data. To simulate the sentence containing disfluencies, two types of random perturbations are introduced into the fluent sentences. One is repetition, we randomly select several continuous words in the fluent sentence to be repeated. The other is inserting, we randomly pick words from external vocabulary to insert into fluent sentences.

2.3 Grammar Checker Model

Traditional self-training methods utilize the probability estimations to select high-confidence pseudo-labels for re-training. These methods suffer from the problems of selection bias and error accumulation. To enhance the ability to select high-quality pseudo-labels in self-training, Wang et al. (2020b) proposes a grammar checker, which can be seen as external knowledge. However, we found that the grammar checker proposed in Wang et al. (2020b) may misjudge when dealing with the ASR outputs since there are incomplete sentences with missing beginnings or endings in the ASR output. To address this problem, we allow the grammar checker to compare the sentence before and after removing the disfluency elements, which can help judge whether a sentence is grammatical or not. Concretely, given an unlabeled sentence S, the disfluency detection model can assign a pseudo label for each word. By deleting the tokens with the "disfluent" label, we obtain the processed sentence $\bar{S}$. We concatenate S and $\bar{S}$ to be the input of our con-

Algorithm 1 : The learning algorithm of our unsupervised model for disfluency detection

Require: Pseudo data for disfluency detection $(x_i, y_i)_{i=1}^{N}$, pseudo data for grammar checker model $(\hat{x}_i, \hat{y}_i)_{i=1}^{M}$, and unlabeled ASR outputs $\{\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_K\}$.

1: Learn grammar checker model θ_g using $(\hat{x}_i, \hat{y}_i)_{i=1}^{M}$ via Eq. 2
2: Learn teacher model θ_t using $(x_i, y_i)_{i=1}^{N}$ via Eq. 1
3: Randomly sample data $\{\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_L\}$ from unlabeled ASR outputs $\{\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_K\}$ Use teacher model to generate pseudo labels $\tilde{y}_i$ and token-level weight s_{ij}^{lc} for sampled data via Eq. 5
4: Use grammar checker model to generate grammatical correctness score s_i^{gc}.
5: Learn a student model θ_s on sampled data $(\tilde{x}_i, \tilde{y}_i)_{i=K}^{N}$ with weight s_{ij}^{lc} and s_i^{gc} via Eq. 7
6: Iterative training until the performance stops growing: Use the student as a teacher and go back to step 3

trastive grammar checker. Our contrastive grammar checker can judge whether $\bar{S}$ is grammatical or not from two perspectives: 1) makes judgment from $\bar{S}$ itself, and 2) makes judgment by comparing S and $\bar{S}$.

We directly fine-tune the ELECTRA model (the discriminator) (Clark et al., 2020) on our pseudo data which minimizes the cross-entropy loss on:

$$loss_{checker} = \frac{1}{M}\sum_{i=1}^{M} \mathrm{CE}(\hat{y}_i, f_g(\hat{x}_i, \theta_g)), \quad (2)$$

where $(\hat{x}_i, \hat{y}_i)_{i=1}^{M}$ denotes input sentences and labels, CE denotes the cross-entropy loss function, f_g denotes our grammar checker model parameterized by θ_g.

Our contrastive grammar checker is trained with pseudo data constructed by the self-supervised learning method. Inspired by Wang et al. (2020b), we use three types of perturbations to simulate grammatical wrong sentences. The first two types of perturbations are repetition and inserting as described previously. The third type of perturbation is deletion. We randomly delete several words from the fluent sentence.

2.4 Re-weighting Mechanism

In this paper, we use two types of re-weighting mechanisms including grammatical correctness re-weighting and label confidence re-weighting. Fig 3 shows our re-weighting mechanism.

2.4.1 Grammatical Correctness Re-weighting

Once a sentence is labeled correctly by the disfluency detection model, the rest after removing the words with disfluency labels is fluent and grammatical (Wang et al., 2020b). Based on this fact, we employ a re-weight strategy to emphasize the sentences that are more grammatical after removing the words with disfluency labels.

With the grammatical correctness re-weighting mechanism, the training objects of the student model are formulated as follows:

$$loss_{gc} = \frac{1}{L}\sum_{i=1}^{L} s_i^{gc} \times \mathrm{CE}(\tilde{y}_i, f_s(\tilde{x}_i, \theta_s)), \quad (3)$$

where $(\tilde{x}_i, \tilde{y}_i)_{i=1}^{L}$ are sentences with pseudo-labels, s_i^{gc} denotes grammatical correctness score which is the logits output of the grammar checker model.

2.4.2 Label Confidence Re-weighting

We employ the label confidence obtained from our teacher model to re-weight the tokens in each sentence. Observing that pseudo-labels with higher confidence are more likely to be correctly labeled, we reinforce the role of high-confidence labels in the training process. With the label confidence re-weighting mechanism, the training object is formulated as follows:

$$loss_{lc} = \frac{1}{LN}\sum_{i=1}^{L}\sum_{j=1}^{N} s_{ij}^{lc} \times \mathrm{CE}(\tilde{y}_{ij}, f_s(\tilde{x}_{ij}, \theta_s)), \quad (4)$$

where $(\tilde{x}_i, \tilde{y}_i)_{i=1}^{L}$ are sentences with pseudo-labels, CE is the cross-entropy loss function, N is the length of the token sequence, s_{ij}^{lc} denotes the teacher model's processed logit output of the token $\tilde{x}_{ij}$.

In the early stages of the iterative self-training process, the performance of the student model is relatively weak. Therefore the confidence output of the student model is not reliable enough. We introduce the temperature mechanism to weaken the effect of confidence when it is unreliable in the early self-training stages.

The calculation of the s_{ij}^{lc} with the temperature mechanism can be formulated as follows:

$$s_{ij}^{lc} = N\frac{exp(\frac{g_{ij}}{T})}{\sum_{k=1}^{N} exp(\frac{g_{ik}}{T})}, \quad (5)$$

where s_{ij}^{lc} denotes the weight of jth token of sample $\tilde{x}_i$, T is the temperature, g_{ij} is the teacher model's confidence in $token_j$ of $\tilde{x}_i$, and N is the length of the token sequence.

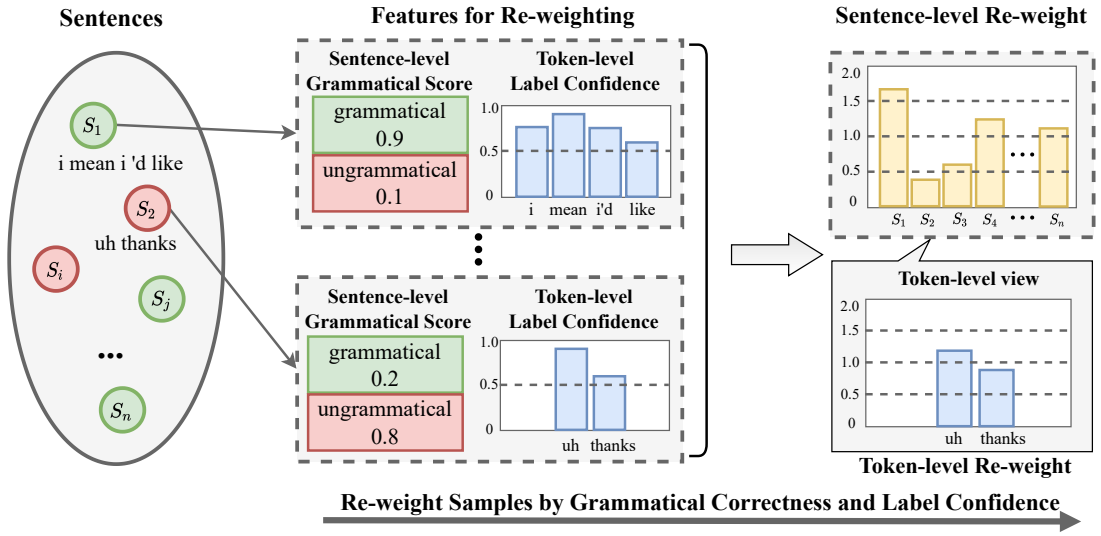

Figure 3: Illustration of proposed re-weighting mechanism. At the sentence-level, sentences with higher grammatical scores are given higher weights after removing words with the label "disfluent". At the token-level, labels with higher confidence are given higher weights.

The temperature T can be formulated as follows:

$$T = -\gamma * log(\frac{n_{sample}}{2 * n_{total}}), \quad (6)$$

where n_{sample} is the amount of data used in this iteration of self-training, n_{total} is the amount of all unlabeled data. γ is a hyper-parameter used to adjust the rate of temperature drop.

2.5 Student Teacher Iterative Training

In Section 2.4, we introduce two different re-weighting mechanisms. Finally, we combine two re-weighting mechanisms, as figure 3 shows, to train the student model, the training objects is formulated as follows:

$$loss_{student} = \frac{1}{LN} \sum_{i=1}^{L} \sum_{j=1}^{N} (s_i^{gc} + s_{ij}^{lc}) \times \tilde{\text{CE}}, \quad (7)$$

where $(\tilde{x}_i, \tilde{y}_i)_{i=1}^{L}$ represents pseudo-labeled sentences, $\tilde{\text{CE}}$ represents $\text{CE}(\tilde{y}_{ij}, f_s(\tilde{x}_{ij}, \theta_s))$, and N is the length of the token sequence, s_i^{gc} and s_{ij}^{lc} are calculated according to the Section 2.4.

We fine-tuned student models with pseudo-labeled data on the teacher model trained by pseudo data. The student will be a new teacher in the next iteration.

3 Experiment

3.1 Settings

Dataset English Switchboard (Godfrey et al., 1992) is a large multispeaker corpus of conversational speech and text, which is the largest standard corpus for disfluency detection. Following the experiment settings in (Wang et al., 2020b), we process the Switchboard corpus and split it into the train (60k sentences), dev (4k sentences), and test set (4k sentences). Notice that the train set is not used in our method. The news data are from WMT2017 monolingual language model training data (News Discussions. Version 2).[2] We conduct our main experiment on the English Switchboard.

To demonstrate the robustness of our method, we also test our model on three out-of-domain datasets (Zayats et al., 2014; Zayats and Ostendorf, 2018), including CallHome, SCOTUS, and FCIC. Notice that the train set of the three out-of-domain datasets is not used.

- ***CallHome***: phone conversations between family members and close friends. It consists of a training set of 46k words and a test set of 30k words.
- ***SCOTUS***: transcribed Supreme Court oral arguments between justices and advocates. It consists of a test set of 43k words and has no training set.
- ***FCIC***: two transcribed hearings from Financial Crisis Inquiry Commission. It consists of a test set of 54k words and has no training set.

Metric Following Wang et al. (2020b), we use token-based precision (P), recall (R), and F1 as the

[2] http://www.statmt.org/wmt17/translation-task.html

Method	F1
Supervised methods	
UBT (Wu et al., 2015)	85.1
Bi-LSTM (Zayats et al., 2016)	85.9
NCM (Lou and Johnson, 2017)	86.8
Transition-based (Wang et al., 2017)	87.5
Self-supervised (Wang et al., 2020a)	90.2
Self-training (Jamshid and Mark, 2020)	90.6
EGBC (Bach and Huang, 2019)	91.8
PG (Yang et al., 2020)	**92.3**
BERT fine-tuning	90.5
ELECTRA fine-tuning	91.2
Teacher fine-tuning	92.0
Unsupervised methods	
Unsupervised teacher	72.3
SSST (Wang et al., 2020b)	88.0
Our Method	**90.3**

Table 1: Comparison with previous state-of-the-art methods on the Switchboard test set. For robustness, we run our proposed supervised baselines and our proposed method 5 times and report the average metric.

evaluation metrics.

3.2 Training Details

For all experiments, we use the English ELECTRA-base discriminator model as an encoder, which has 110M hidden units, 12 heads, and 12 hidden layers. For the self-supervised teacher model, the max sequence length of teacher model input is set to 128. We fine-tune the teacher model using the AdamW optimizer for 30 epochs with a batch size of 256 and a learning rate of 1e-4. Since the grammar checker model's input consists of a pair of sentences, we set its max sequence length to be 256. We fine-tune the grammar checker model using AdamW optimizer for 30 epochs with a batch size of 256 and a learning rate of 1e-4.

When training the student model in the self-training procedure, we use the AdamW optimizer for 15 epochs with a batch size of 128 and a learning rate of 2e-5.

3.3 Performance on English Switchboard

Table 1 shows the comparison of the performance among different models on the Switchboard test set. In addition to the previous methods, we have also built the following baseline models for comparison.

- BERT fine-tuning: We directly fine-tune the BERT-Base model on the Switchboard train set.
- ELECTRA fine-tuning: We directly fine-tune ELECTRA-Base (the discriminator) model on the Switchboard train set.
- Unsupervised teacher: This model is the first teacher model trained by pseudo data.
- Teacher fine-tuning: We fine-tune the teacher model which is trained by pseudo data on the Switchboard train set.

Method	CallHome	SCOTUS	FCIC
Supervised methods			
ELECTRA fine-tuning	62.2	81.6	63.1
Teacher fine-tuning	63.4	**81.9**	63.8
Pattern-match	**65.2**	79.9	**66.1**
Unsupervised methods			
Unsupervised teacher	48.0	69.2	47.6
SSST	60.2	80.3	63.3
Our unsupervised	**61.6**	**80.7**	**63.6**

Table 2: F1 scores on cross-domain disfluency detection. For robustness, we run our proposed supervised baselines and our proposed method 5 times and report the average metric and standard deviation metric.

Our unsupervised model achieves 2.3 points improvements over the past state-of-the-art unsupervised method (Wang et al., 2020b) which is also based on self-training. Moreover, our unsupervised model achieves competitive performance compared to the current SOTA supervised (Yang et al., 2020) method, which trains BERT with 20M sentences of pseudo data and 173k sentences of human-annotated data.

3.4 Performance on Cross-domain Data

In this section, we experiment on three out-of-domain disfluency datasets to prove the robustness of our proposed approach.

We build four baseline systems for comparison, including Unsupervised teacher, ELECTRA fine-tuning, Teacher fine-tuning, SSST (Wang et al., 2020b), and Pattern-match (Zayats and Ostendorf, 2018). Unsupervised teacher, ELECTRA fine-tuning, and Teacher fine-tuning are the same as described in Section 3.3. Pattern-match (Zayats and Ostendorf, 2018) used a pattern match neural network architecture trained on the Switchboard train set. It achieves state-of-the-art performance in cross-domain scenarios.

Following the experiment setting in (Wang et al., 2020b), we test each model's performance on the three out-of-domain test sets without any retraining on the out-of-domain train set.

Table 2 shows the comparison of the performance among different models on the three out-

Method	P	R	F1
ST+SGC	90.2	89.1	89.6
ST+SGC+Re-weight	**90.5**	91.7	91.1
ST+CGC	89.5	91.0	90.3
ST+CGC+Re-weight	90.3	**92.9**	**91.6**

Table 3: Ablation results of our method on Switchboard dev set. "ST" denotes self-training. "SGC" denotes the single sentence grammar checker. "CGC" denotes the contrastive grammar checker. "Re-weight" denotes grammatical correctness re-weighting and label confidence re-weighting mechanism.

Method	P	R	F1
ST+CGC	89.5	91.0	90.3
ST+CGC+GCR	**91.2**	91.0	91.1
ST+CGC+LCR	90.2	92.1	91.1
ST+CGC+GCR+LCR	90.3	**92.9**	**91.6**

Table 4: Ablation results of our method on Switchboard dev set. "ST" denotes self-training. "CGC" denotes contrastive grammar checker. "GCR" denotes grammatical correctness re-weighting mechanism."LCR" denotes label confidence re-weighting mechanism.

T hyperparameter	P	R	F1
w/o Temperature	90.7	92.0	91.4
w/ Temperature γ=2	**91.5**	90.9	91.2
w/ Temperature γ=1	91.1	91.7	91.4
w/ Temperature γ=0.5	90.3	**92.9**	**91.6**

Table 5: Performance of our method on Switchboard dev set with different temperature.

of-domain test sets. It shows that our unsupervised methods can achieve good performance in the three cross-domain datasets without any retraining.

4 Ablation Studies

In this section, we analyze the effect of several components of our method.

4.1 Effect of re-weighting mechanism

To explore the role of the re-weighting mechanism, we conduct a comparative experiment for the re-weighting mechanism. We take the approach proposed by Wang et al. (2020b) as our baseline, which is a self-training-based method with a single sentence grammar checker (referred to as "ST+SGC"). We separately add the re-weighting mechanism to the baseline system (referred to as "ST+SGC+Re-weight"). Table 3 shows that the method with a re-weighting mechanism outperforms the baseline by 1.5 points, demonstrating the benefit of the re-weighting mechanism.

Our re-weighting mechanism consists of grammatical correctness re-weighting and the label confidence re-weighting. To explore the role of each part of the re-weighting mechanism, we conduct a comparative experiment. Table 4 shows that both the grammatical correctness re-weighting and the label confidence re-weighting bring improvement.

4.2 Effect of improved grammar checker

In our approach, we propose a contrastive grammar checker. To validate the effectiveness of our grammar checker, we conduct a comparative experiment. "ST+SGC" denotes a self-training approach with a single sentence grammar checker which is proposed by Wang et al. (2020b). "ST+CGC" denotes a self-training approach with a contrastive grammar checker which is proposed by us. The only difference between them is that they use different grammar checkers. Table 3 shows that the method with our grammar checker outperforms the method with the previous grammar checker by 0.7 points.

To compare the performance of two grammar checkers directly, we test them on the grammar check task dev dataset, which is constructed from the Switchboard dev dataset. Concretely, we take the sentences from the Switchboard without disfluency components as positive examples, while the sentences with disfluency components as negative examples. As shown in Figure 4, the "CGC"proposed by us archieves a higher AUC than the "SGC" proposed by Wang et al. (2020b). It is worth noting that we don't let the "CGC" see the sentence after removing disfluency components. We repeat the raw sentence twice as the input of "CGC".

These demonstrate that our contrastive grammar checker can better evaluate the quality of pseudo-labeled data for disfluency detection.

4.3 Effect of temperature mechanism performance

We proposed a temperature mechanism to weaken the confidence re-weighting mechanism effect in the early self-training stage when the confidence is not reliable enough. To reveal the temperature mechanism effect on model performance, we conduct comparative experiments using different γ which control the rate of temperature drop. The higher γ we set, the smaller the confidence re-weighting effect in the early self-training stage. The experiment results in table 5 show that reducing the confidence re-weighting effect in the early

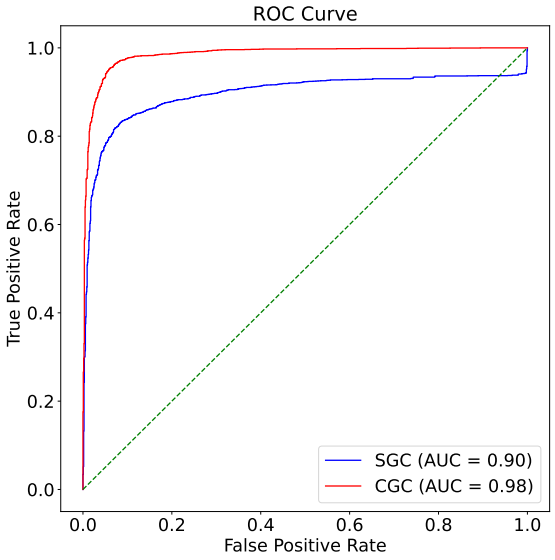

Figure 4: The ROC curve of grammar checkers. "SGC" denotes single sentence grammar checker. "CGC" denotes contrastive grammar checker.

self-training stage appropriately can improve the model performance. With these results, we chose the temperature value as 0.5.

5 Related Work

5.1 Disfluency Detection

Most of the previous work on disfluency detection focus on supervised learning methods. There are three main categories of methods to solve the disfluency detection problem including sequence tagging, noisy-channel, and parsing-based approaches. Sequence tagging methods use various models as classifiers to classify each word in the sentence to be fluent or disfluent, including conditional random fields (Georgila, 2009; Ostendorf and Hahn, 2013; Zayats et al., 2014), Max-Margin Markov Networks (M^3N) (Qian and Liu, 2013), Semi-Markov CRF (Ferguson et al., 2015), recurrent neural networks (Hough and Schlangen, 2015; Zayats et al., 2016; Wang et al., 2016) and transformer-based model (Wang et al., 2020a). Noisy-channel (Charniak and Johnson, 2001) methods use the similarity between reparandum and repair to detect disfluency. Parsing-based approaches (Rasooli and Tetreault, 2013; Wu et al., 2015) deal with disfluency detection by parsing the sentence.

Some previous work focuses on tackling the training data bottleneck, including self-supervised methods (Wang et al., 2020a), self-training methods (Jamshid and Mark, 2020) and active learning methods (Wang et al., 2021). Some researchers propose a method using unsupervised self-training to perform unsupervised disfluency detection which does need not any human-annotated data (Wang et al., 2020b).

Rocholl et al. (2021) uses the distillation method to make the disfluency detection model to be small, fast and on-device while maintaining competitive performance.

5.2 Self-training

The self-training method first leverages human-annotated data to train a teacher model. Then the teacher model is used to assign pseudo-label to unlabeled data. Finally, the labeled data and the pseudo-labeled data are merged to train the student model jointly Scudder (2006). Self-training has shown promising performance for a variety of tasks including leveraging noisy data (Veit et al., 2017), semantic segmentation (Babakhin et al., 2019) and text classification (Li et al., 2019). Xie et al. (2020) presents Noisy Student Training, which adds noise to the self-training process and changes the size of the student model during self-training iteration. Inspired by Xie et al. (2020), Wang et al. (2020b) combines self-supervised and self-training to build an unsupervised disfluency detection system for disfluency detection.

5.3 Re-weighting mechanism

Re-weighting approaches (Ren et al., 2018; Shu et al., 2019; Mei et al., 2020; Yaqing Wang et al., 2020) are proposed to give different loss weights on each sample to emphasize the important samples and discount the noisy samples. Re-weighting approaches often train a teacher model on a small human-annotated validation set and then use the teacher model to re-weight training samples. In our method, we do not need a teacher model trained by human-annotated data. We re-weight each training sample by the model trained by the self-supervised method.

6 Conclusion

In this work, we propose an adaptive unsupervised method to deal with the task of disfluency detection in an unsupervised manner. We introduce a re-weighting mechanism into the self-training to alleviate the problems of selection bias and error accumulation. Our experiments show that our re-weighting mechanism can alleviate the problems of selection bias and error accumulation efficiently,

and improve the performance of self-training for disfluency detection.

Acknowledgement

This work was supported by the National Key RD Program of China via grant 2020AAA0106501 and the National Natural Science Foundation of China (NSFC) via grant 61976072 and 62176078.

References

Yauhen Babakhin, Artsiom Sanakoyeu, and Hirotoshi Kitamura. 2019. Semi-supervised segmentation of salt bodies in seismic images using an ensemble of convolutional neural networks. In *Proc. of GCPR*.

Nguyen Bach and Fei Huang. 2019. Noisy bilstm-based models for disfluency detection. *Proc. Interspeech 2019*.

Eugene Charniak and Mark Johnson. 2001. Edit detection and parsing for transcribed speech. In *Proc. of ACL*.

Kevin Clark, Minh-Thang Luong, V. Quoc Le, and D. Christopher Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *ICLR*.

James Ferguson, Greg Durrett, and Dan Klein. 2015. Disfluency detection with a semi-markov model and prosodic features. In *Proc. of ACL*.

Kallirroi Georgila. 2009. Using integer linear programming for detecting speech disfluencies. In *Proc. of ACL*.

John J Godfrey, Edward C Holliman, and Jane McDaniel. 1992. Switchboard: Telephone speech corpus for research and development. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, volume 1, pages 517–520. IEEE Computer Society.

Julian Hough and David Schlangen. 2015. Recurrent neural networks for incremental disfluency detection. In *Proc. of INTERSPEECH*.

Paria Lou Jamshid and Johnson Mark. 2020. Improving disfluency detection by self-training a self-attentive model. *ACL*.

Paria Jamshid Lou, Peter Anderson, and Mark Johnson. 2018. Disfluency detection using auto-correlational neural networks. In *Proc. of EMNLP*.

Xinzhe Li, Qianru Sun, Yaoyao Liu, Qin Zhou, Shibao Zheng, Tat-Seng Chua, and Bernt Schiele. 2019. Learning to self-train for semi-supervised few-shot classification. In *Proc. of NeurIPS*.

Paria Jamshid Lou and Mark Johnson. 2017. Disfluency detection using a noisy channel model and a deep neural language model. *Proc. of ACL*.

Ke Mei, Chuang Zhu, Jiaqi Zou, and Shanghang Zhang. 2020. Instance adaptive self-training for unsupervised domain adaptation. In *European conference on computer vision*, pages 415–430. Springer.

Mari Ostendorf and Sangyun Hahn. 2013. A sequential repetition model for improved disfluency detection. In *INTERSPEECH*.

Xian Qian and Yang Liu. 2013. Disfluency detection using multi-step stacked learning. In *HLT-NAACL*.

Mohammad Sadegh Rasooli and Joel R Tetreault. 2013. Joint parsing and disfluency detection in linear time. In *EMNLP*.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. 2018. Learning to reweight examples for robust deep learning. In *Proc. of ICML*.

Johann C. Rocholl, Vicky Zayats, Daniel D. Walker, Noah B. Murad, Aaron Schneider, and Daniel J. Liebling. 2021. Disfluency detection with unlabeled data and small bert models. In *Proc. Interspeech*.

H. Scudder. 2006. Probability of error of some adaptive pattern-recognition machines. *IEEE Trans. Inf. Theor.*

Elizabeth Ellen Shriberg. 1994. *Preliminaries to a theory of speech disfluencies*. Ph.D. thesis.

Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. 2019. Meta-weight-net: Learning an explicit mapping for sample weighting. In *Proc. of NeurIPS*.

Jean E Fox Tree. 1995. The effects of false starts and repetitions on the processing of subsequent words in spontaneous speech. *Journal of memory and language*.

Andreas Veit, Neil Alldrin, Gal Chechik, Ivan Krasin, Abhinav Gupta, and Serge Belongie. 2017. Learning from noisy large-scale datasets with minimal supervision. In *Proc. of CVPR*.

Feng Wang, Wei Chen, Zhen Yang, Qianqian Dong, Shuang Xu, and Bo Xu. 2018. Semi-supervised disfluency detection. In *Proc. of COLING*.

Shaolei Wang, Wangxiang Che, Qi Liu, Pengda Qin, Ting Liu, and William Yang Wang. 2020a. Multi-task self-supervised learning for disfluency detection. In *Proc. of AAAI*.

Shaolei Wang, Wanxiang Che, and Ting Liu. 2016. A neural attention model for disfluency detection. In *Proc. of COLING*.

Shaolei Wang, Wanxiang Che, Yue Zhang, Meishan Zhang, and Ting Liu. 2017. Transition-based disfluency detection using lstms. In *Proc. of EMNLP*.

Shaolei Wang, Zhongyuan Wang, Wanxiang Che, and Ting Liu. 2020b. Combining self-training and self-supervised learning for unsupervised disfluency detection. In *Proc. of EMNLP*.

Shaolei Wang, Zhongyuan Wang, Wanxiang Che, Sendong Zhao, and Ting Liu. 2021. Combining self-supervised learning and active learning for disfluency detection. *Proc. of TALLIP*.

Shuangzhi Wu, Dongdong Zhang, Ming Zhou, and Tiejun Zhao. 2015. Efficient disfluency detection with transition-based parsing. In *Proc. of ACL*.

Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. 2020. Self-training with noisy student improves imagenet classification. In *Proc. of CVPR*.

Jingfeng Yang, Diyi Yang, and Zhaoran Ma. 2020. Planning and generating natural and diverse disfluent texts as augmentation for disfluency detection. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1450–1460, Online. Association for Computational Linguistics.

Yaqing Yaqing Wang, Subhabrata Mukherjee, Haoda Chu, Yuancheng Tu, Ming Wu, Jing Gao, and Ahmed Hassan Awadallah. 2020. Adaptive self-training for few-shot neural sequence labeling. *arXiv preprint arXiv:2010.03680*.

Vicky Zayats and Mari Ostendorf. 2018. Robust cross-domain disfluency detection with pattern match networks. *arXiv: Computation and Language*.

Vicky Zayats and Mari Ostendorf. 2019. Giving attention to the unexpected: Using prosody innovations in disfluency detection. *Proc. of NAACL*.

Vicky Zayats, Mari Ostendorf, and Hannaneh Hajishirzi. 2016. Disfluency Detection Using a Bidirectional LSTM. In *Proc. Interspeech 2016*, pages 2523–2527.

Victoria Zayats, Mari Ostendorf, and Hannaneh Hajishirzi. 2014. Multi-domain disfluency and repair detection. In *Proc. of INTERSPEECH*.