Semi-Supervised Disfluency Detection

Feng Wang^{1,2}, Wei Chen^{1*}, Zhen Yang^{1,2} , Qianqian Dong^{1,2}, Shuang Xu^{1}, and Bo Xu^{1}
^{1}Institute of Automation, Chinese Academy of Sciences
^{2}University of Chinese Academy of Sciences, Beijing, China
{feng.wang, wei.chen.media,yangzhen2014
dongqianqian2016 ,shuang.xu, xubo}@ia.ac.cn

Abstract

While the disfluency detection has achieved notable success in the past years, it still severely suffers from the data scarcity. To tackle this problem, we propose a novel semi-supervised approach which can utilize large amounts of unlabelled data. In this work, a light-weight neural net is proposed to extract the hidden features based solely on self-attention without any Recurrent Neural Network (RNN) or Convolutional Neural Network (CNN). In addition, we use the unlabelled corpus to enhance the performance. Besides, the Generative Adversarial Network (GAN) training is applied to enforce the similar distribution between the labelled and unlabelled data. The experimental results show that our approach achieves significant improvements over strong baselines.

1 Introduction

A characteristic of spontaneous speech is different from written text, since it's usually accompanied by disfluencies. Identifying and removing these non-fluent factors would help to improve the spontaneous speech quality. It often plays a significant role in understanding the semantics of these sentences and it's vital for the downstream NLP tasks, such as question answering, machine translation, and information extraction.

$$\text{I want to flight [} \underbrace{\text{to Boston}}_{\text{RM}} + \{ \underbrace{\text{um}}_{\text{IM}} \} + \underbrace{\text{to Denver}}_{\text{RP}} \text{]}$$

Figure 1: Example of disfluency annotation style in Switchboard corpus.

Figure 1 shows a standard annotation of disfluency structure (Shriberg, 1994), which includes three types of annotations: RM (*reparandum*, words that are discarded, or corrected by the following words), RP (*repair*, the correct words), and IM (*interregnum*, such as filled pauses, discourse cue words, etc.) The interruption point (+) marks the end of the reparandum and an optional interregnum. Ignoring the interregnum, disfluencies can be further categorized into three types: repetition (when RP is same as RM), repair, and restart (when RP is empty). Table 1 gives a few examples. However, it is usually difficult to identify the reparandum non-fluent factors. The main challenge behind is that this type of structure is flexible, variable in length, able to occur anywhere in a sentence, and in many cases will be nested. Detecting the disfluency can work in arbitrary form.

Recently, a number of approaches based on deep neural networks have been proposed to address the problem of disfluency detection under the framework of sequence labeling or sequence to sequence(Hough and Schlangen, 2015; Zayats et al., 2016; Wang et al., 2016; Wang et al., 2017). For most of the approaches mentioned above, the researchers switch between the RNN and CNN. But the RNN/CNN shows less flexibility than the self-attention layer which has achieved great success in neural machine translation (Shen et al., 2015; Wu et al., 2016; Gehring et al., 2017) and language understanding

[*] Wei Chen is the corresponding author of this paper

Proceedings of the 27th International Conference on Computational Linguistics, pages 3529–3538
Santa Fe, New Mexico, USA, August 20-26, 2018.

Type	Annotation
repair	[I just + I] enjoy work
repair	[we want + well in our area we want] to
repetition	[it's + {um} + it's] almost like
repetition	[the + {um} + the] decision was
restart	[by +]it was attached to
restart	[we would like +] let's go to the

Table 1: Different types of disfluencies.

(Shen et al., 2017). In the framework of sequence to sequence mapping, the model of NMT has achieved great success. Therefore, it is natural for us to investigate how to apply the NMT system to the problem of disfluency detection. In this paper, we take the disfluency detection as the translation task and build our model following the architecture of newly emerged NMT system, i.e., Transformer(Vaswani et al., 2017). The Transformer has achieved state-of-the-art results on both WMT2014 English-German and English-French translation tasks. Inspired by the success of self-attention, we propose a RNN/CNN-free network for disfluency detection through multi-task learning, which is built only on the self-attention layers. In the proposed model, we apply the self-attention layers to extract the hidden features of the input sentence. And two softmax layers, namely the label softmax and word softmax, are applied to perform classifications. The label softmax is used to calculate the probability of the disfluency label, and the word softmax is applied to compute the probability of the word.

In addition, we leverage unlabelled data to improve the performance of the disfluency detection. Due to lack of tagging information, using the unlabelled corpora is very promising, since the unlabelled corpora is usually easy to be collected. To utilize the unlabelled data, we extend the traditional encoder-decoder model by leveraging two independent encoders but with some layers shared. Specifically, we share the weights of the last few layers of two encoders that are responsible for extracting high-level representations of input sentences. In the proposed model, one encoder is utilized to encode the unlabelled data and the other is applied for the labelled data. For the unlabelled data, the corresponding encoder and decoder perform an Auto-Encoder (AE), where the encoder generates the latent representations from the perturbed input sentences and the decoder reconstructs the sentences from the latent representations. We utilize the GAN to constrain the latent representations from the labelled and unlabelled corpus to subject to a similar distribution, whereby the encoders try to fool a discriminator which is simultaneously trained to distinguish whether the latent representation is from labelled or unlabelled data.

In summary, we mainly make the following contributions:

- We propose a novel semi-supervised approach for the problem of disfluency detection, which can utilize large amounts of unlabelled data.

- We firstly introduce the self-attention mechanism into the disfluency detection through multi-task learning . Without relying on the RNN/CNN, the proposed model has more flexibility in sequence length and allows for more parallelization, which makes great significance.

- We conduct extensive experiments to test the proposed model. Experimental results show that the proposed approach consistently achieves great success.

2 Related Work

One of the most common approaches to disfluency detection is taking the task as a sequence labeling problem, where each sentential word is assigned with a label. Many approaches had achieved good performance by using the Condition-based Random Field (CRF) as a classifier (Ostendorf and Hahn, 2013; Liu et al., 2006; Cho et al., 2013; Zayats et al., 2014). (Qian and Liu, 2013) proposed a detection model based on maximum interval Markov random field (M^3N), which outperformed CRF by using an f-score matching objective function. (Ferguson et al., 2015) proposed the Semi-Markov CRF model for disfluency detection and achieved the best performance of linear statistical sequence labeling methods by integrating prosodic features. Sequence labeling methods mentioned above are not powerful enough to model long-range dependencies of complicated disfluencies. RNN has been widely used to disfluency detection (Zayats et al., 2016; Hough and Schlangen, 2015), since it can capture dependencies at any length in theory. However, these RNN methods still suffer from the defect of unable to model the linguistic structural integrity, since it does not model the transition between tags which is important in recognizing the repair phrases of multi-words. Recently, sequence-to-sequence methods based on deep neural networks are applied to disfluency detection. (Wang et al., 2016) proposed an attention-based detection model and achieved good performance. It can capture a global representation of the input sentence by RNN when encoding and model tag transition when decoding.

Another branch of researches adopt transition-based parsing model, which jointly perform dependency parsing and disfluency detection (Rasooli and Tetreault, 2013; Honnibal and Johnson, 2014; Wu et al., 2015). These syntax-based models are capable of capturing long-range dependencies by modeling the repair phrases on a syntax tree. However, their high requirements for training data reduce their practicality, since it is expensive to obtain training data containing both syntax trees and disfluency annotations. Furthermore, the performance of syntax parsing also restricts the performance of disfluency detection. (Wang et al., 2017) proposed a transition-based model for disfluency detection, which learned representation of both chunks and global contexts without using any syntax information. It achieved the state-of-the-art f-score on the commonly used English Switchboard test set.

Lack of large scale of labelled data is always a bottleneck for improving the performance of disfluency detection (Wang et al., 2016; Wang et al., 2017). Recently, the unsupervised and semi-supervised NMT have arose many interests in the research area. To improve the translation performance of NMT, they use the auto-encoder and back-translation to train the model(Saha et al., 2016; Cheng et al.,). Following the same idea, we firstly propose the semi-supervised training for disfluency detection.

3 The Approach

In this section, we will give more details about the design of our proposed semi-supervised disfluency detection model.

3.1 The Input and Output

Since we consider the problem of disfluency detection as a translation task, translating the non-fluent sequence into the fluent sequence, we prepare our input and output as the training examples of translation. In traditional translation, the training example consists of one source-side sequence and one target-side sequence. However, for the proposed model, the source-side input is the non-fluent sequence, the target side includes two output sequences, one is the fluent sequence, which is the real needed output of disfluency detection task, and the other is the label sequence, which is an auxiliary information sequence. Considering a sentence from the training data, the non-fluent input of source side is:

”I want to flight to Boston um to Denver”

The target-side fluent sequence is represented as:

”I want to flight E E E to Denver ”

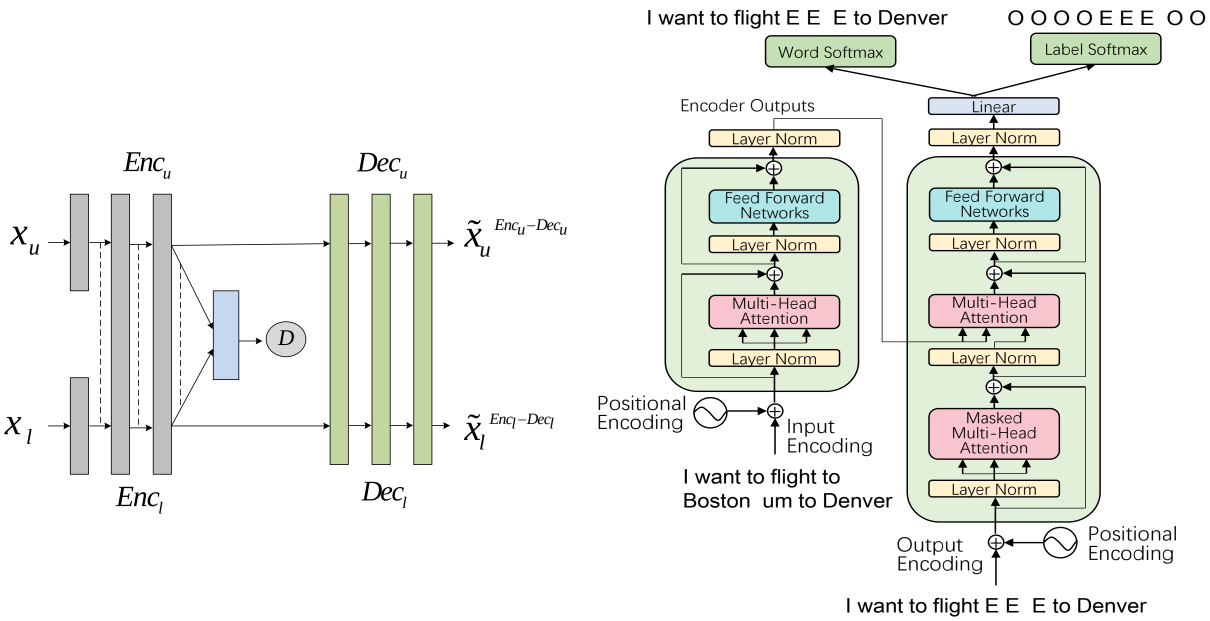
And the target-side label sequence can be represented as:

”O O O O E E E O O”

Where ’O’ indicates that the word at this time step is fluent, which should be copied from the source side to the fluent sequence, and ’E’ means the word at this time step is non-fluent, i.e., RM and IM words, which should be discarded. The length of the label sequence is the same as the input sequence. The motivation of the design of two output sequences will be elaborated in section 3.2.

3.2 The Model Architecture

Our proposed semi-supervised model is extended on traditional encoder-decoder framework. As illustrated in Figure 2(a), the model architecture is composed of four subnetworks: including two partially



(a) Weight sharing model with GAN

(b) Multi-task Transformer model

Figure 2: The framework of the proposed model. (a) is the whole architecture of our model, which contains two independent encoders with some weight sharing and the fully-shared decoder. (b) is the specific architecture of the proposed model which extends the Transformer into multi-task learning setting.

shared encoders Enc_l and Enc_u, one totally shared decoder Dec_l (Dec_u is the same as Dec_l), and a discriminator D, where subscript 'l' and 'u' represent labelled and unlabelled data, respectively. Each encoder consists of three identical layers, where each layer consists of a multi-head self-attention and a simple position-wise fully connected feed-forward network, which follows the powerful self-attention based model Transformer (Vaswani et al., 2017). Specifically, the encoder is composed of a stack of three identical layers. Symmetrically, the decoder is also a stack of three identical layers, and each layer also follows the Transformer. The key difference is the last output layer of the decoder, where we use two softmax layers to predict label sequence and word sequence separately (see Figure 2(b)). The decoding is constrained by the label prediction and word prediction at the same time, which is critical important for improving the disfluency detection performance.

To utilize unlabelled data x_u for training, the output of Enc_u should have similar distribution with the output of Enc_l. Therefore, we apply the weight sharing constraint to associate the two encoders by sharing the weights of the last few layers of two encoders that are responsible for extracting high-level representations of input sequences. Afterwards, a discriminator, implemented as a multilayer perceptron, is used to further constrain the latent representations from the labelled and unlabelled data to have similar distribution, whereby the encoder tries to fool the discriminator which is simultaneously trained to distinguish whether the latent representation is from the labelled or unlabelled data. As a matter of fact, the encoders and the discriminator constitute a GAN tactfully, where the encoders act as the cheater and the discriminator plays the role of police. The weight sharing and GAN training are pivotal strategies to make the semi-supervised training manner feasible.

To give more detailed interpretation of the weight sharing model, we piece together several roles by combining some subnetworks of the model, which are summarized in table 2. If only consider using labelled data x_l for training, it is enough to compose a complete model only with the encoder Enc_l and the decoder Dec_l, which will be back to a self-attention based Transformer model with multi-task learning. As for the unlabelled data, the encoder Enc_u generates the latent representation of the perturbed input sentence (such as adding random noises). And the corresponding decoder Dec_u reconstructs the normal input sentence from the latent representation. Actually, Enc_u and Dec_u constitute an AE. The key information of these roles will be described in the following subsections successively.

Networks	Roles
$\{Enc_l, Dec_l\}$	Transformer for labelled data
$\{Enc_u, Dec_u\}$	AE for unlabelled data
$\{Enc_l, Enc_u\}$	Weight sharing of labelled and unlabelled data
$\{Enc_l, Enc_u, D\}$	GAN for discriminating data reality

Table 2: Interpretation of the roles for the subnetworks in the weight sharing model.

3.2.1 Multi-task Learning and Constrained Decoding

Our improvement on the Transformer model focuses on the last output layer of the decoder. Traditional output layer of Transformer is only one softmax layer. However, either the word sequence or the label sequence is incapable of handling the problem of disfluency detection independently. For the word sequence, it guides the model to treat each fluent word and disfluency word equally. However, in disfluency detection task, the model just needs to distinguish the non-fluent words from the fluent words in sentence. Too careful classification will detriment the classification performance. For the label sequence, it guides the model to output the fluent label or the non-fluent label. Nevertheless, it may cause overfitting during training since the fluent label 'O' and non-fluent label 'E' dominate in the training example. In other words, the word sequence needs a simple classification as the label sequence does, and the label sequence can benefit from the semantic information of the word sequence. Therefore, we fuse the disfluency detection sequence and the label sequence through multi-task learning to improve the performance.

Multi-task Learning

Given N such non-fluent source word sequences, target word sequences and label sequences. Our goal is to maximize the probability of observing the target word sequence y_n and minimize the label classification error rate of the label softmax, with regard to the model parameter θ. Different from the traditional end-to-end model in disfluency detection which only maximizes the probability of observing the target label sequence, the word error rate of the word softmax is also leveraged in the proposed model. From the multi-goal learning perspective, two different goals are used to dictate the parameters of the proposed model to converge to a better region. Formally, the objective function used in the joint punctuation model can be represented as:

$$\operatorname*{argmax}_{\theta} \frac{1}{N} \sum_{n=1}^{N} (log_{p_\theta}(y_n|x_n) + \gamma * log_{p'_\theta}(z_n|x_n)) \tag{1}$$

Apart from the cross entropy loss used in the label neural network model, the error rate of the word neural network is also considered to train the proposed model.

$$J = J_w + \gamma * J_l \tag{2}$$

here J_w refers to the loss of the word, J_l denotes the loss of the label and γ is the hyper-parameter. We use the γ to balance the loss between the word softmax and label softmax.

Constrained Decoding

In the test decoding, we design a constrained decoding algorithm to adapt the proposed self-attention based network. In this algorithm, we need to judge the class of the label softmax to determine the decode sequence of the model. Specifically, if the label softmax outputs 'O', we copy the word from the source side and feed it into the model for the next time step. And if the output of the label softmax is 'E', we feed the generated 'E' into the model for the decoding in the next time step. The decoding process repeats until the "eos", i.e., the end of sentence token, is emerged and all the source words are copied. The constrained decoding algorithm could effectively avoid the inconsistency between the source word and the target word in a transduction model.

3.2.2 Denoising Auto-Encoding

In order to learn some knowledge from the unlabelled corpus to make up the scarcity of labelled data, an AE is utilized to train the unlabelled data via reconstructing their inputs. Specifically, the encoder is responsible for composing the high-level latent representation of the input and the decoder endeavors to decompose this representation into its corresponding input sequence. However, the AE usually fails to capture any internal structure of the involved sentences under unconstrained condition. Instead it just learns to copy every word one by one. To solve this problem, we employ the similar strategy of denoising AE (Vincent et al., 2008) and randomly add some noises to the input sentences (Hill et al., 2016; Artetxe et al., 2017).

To this end, for each sentence, we add two types of noises separately, one is repeating, the other is inserting. In repeating operation, we randomly choose one to three positions in sentence. And for each position, one to five words starting from selected position are repeated randomly, which is abide by the constraint that the repeat fragment does not overlap with the next selected position. In inserting operation, we still randomly choose one to three positions in sentence to add noises. Different from repeating operation which simply repeats words in the sentence, we insert words randomly selected from a top frequent dictionary, which is extracted from the corpus by keeping the top 10,000 most frequent 1-grams to 5-grams respectively beforehand. Thus, inserting operation can introduce more complicated noises. In this way, the system needs to learn some useful structure of the involved languages to be able to recover the correct word order.

3.2.3 Weight Sharing

To decrease even eliminate the distribution differences between the labelled and unlabelled corpora in some ways, a weight sharing constraint is applied to the two encoders according to the shared-latent space assumption. Different from (Cheng et al., 2016; Saha et al., 2016) which adopt the fully shared encoder, we share only partial weights for the encoders, considering that 1) The independent weights are devoted to encode the hidden features about the internal characteristics of each corpus, such as the terminology, style, and sentence structure; 2) The shared weights are concentrated on projecting those hidden features into the shared-latent space. Therefore, we share the weights of the last few layers of encoders Enc_l and Enc_u, which are committed to capture the high-level representations of the input sentences. A totally shared decoder is applied to decode those high-level representations that are vital for reconstructing the fluent sentences, since both for the labelled and unlabelled input sentences, they share the same decoding space and target.

3.2.4 Discriminator

For the weight sharing constraint based on the shared-latent space assumption, we would expect that the corresponding sentences in the labelled or unlabelled corpus will have the similar latent representations. However, the weight sharing constraint alone does not necessarily guarantee the same latent representations. To further enforce the shared-latent space, we train a discriminative neural network to classify whether the encoding representation is from the auto-encoded unlabelled or the labelled sentences.

The discriminator is implemented as a multi-layer perception with two hidden layers of size 256. The discriminator performs as a binary classifier and predicts whether the input is the labelled or unlabelled data.Given the encoding vector Enc(x), i.e., the output of the encoders either from the Enc_u or Enc_l, we build the perception layer as:

$$x_c = \rho(BN(W \times Enc(x) + b)) \tag{3}$$

where W and b is the learning parameters. ρ is a non-linear activation function which is implemented as ReLU. At last, we pass x_c into a softmax layer to generate the probability $p(f|Enc(x))$ as:

$$p(f|Enc(x)) = softmax(V * x_c) \tag{4}$$

where V is the transformation matrix and f is the data class, satisfying $f \in \{l, u\}$, where l means labelled data and u means unlabelled data. The discriminator is trained to predict the label probability by minimizing the following cross-entropy loss:

$$L_D(\theta_D) = -\mathbb{E}_{x \in x_u}[\log p(f = u|Enc_u(x))] - \mathbb{E}_{x \in x_l}[\log p(f = l|Enc_l(x))] \tag{5}$$

here θ_D represents the discriminator parameters. The encoders are trained to fool the discriminator:

$$L_{Enc_u}(\theta_{Enc_u}) = -\mathbb{E}_{x \in x_u}[\log p(f = l|Enc_u(x))] \tag{6}$$

$$L_{Enc_l}(\theta_{Enc_l}) = -\mathbb{E}_{x \in x_l}[\log p(f = u|Enc_l(x))] \tag{7}$$

where θ_{Enc_u} and θ_{Enc_l} are the parameters of the two encoders.

4 Experiments

4.1 Datasets

To evaluate the effectiveness of utilizing unlabelled corpus on disfluency detection task, we firstly conduct an experiment solely on a labelled corpus named Switchboard. Then we expand the training data by introducing large-scale unlabelled corpus.

Switchboard is the largest available corpus for disfluency detection task, which is portion of the English Penn Treebank. Two annotation layers are provided in Switchboard corpus: one for syntactic bracketing (MRG files), the other for disfluencies (DPS files). Our proposed method only needs the DPS files. The disfluency annotation style of DPS files has been described in introduction. In our experiments, the word labelled with RM and IM would be tagged with 'E' and others would be tagged with 'O'. Some previous work has been done on this corpus. To keep the direct comparison possible, we follow the experiment settings in (Johnson and Charniak, 2004), taking directory 2 and directory 3 in subcorpus of PARSED/MRG/SWBD as training set, and splitting directory 4 into test and development sets. Table 3 shows the detailed scale of each dataset, where "total" means the total number of sentences and "non-fluent" means the number of non-fluent sentences among total.

Unlabelled corpus is randomly extracted from the source side of WMT2014 English-German corpus, consisting of English news. Noises, which make the unlabelled fluent sentence non-fluent, are added to the unlabelled datasets (named as WMT2014 English corpus) according to the description in subsection of denoising auto-encoding. The scale of training, development and test sets of WMT2014 English corpus is shown in Table 4.

Dataset	Total	Non-fluent
Training set	261,882	87,701
Dev set	2,000	1,455
Test set	2,000	1,425

Table 3: Scale of Switchboard corpus.

Dataset	Total	Non-fluent
Training set	1,000,000	1,000,000
Dev set	2,000	1,500
Test set	2,000	1,500

Table 4: Scale of WMT2014 English corpus.

4.2 Training Details

BPE Byte Pair Encoding (Sennrich et al., 2015) is a simple data compression technique, which is highly efficient to reduce the OOV quantity by iteratively replacing the most frequent pair of bytes in a sequence with a single unused byte. In our experiments, we set source number operations to 20,000 and the vocabulary size to 30,000.

Hyper-parameter The configuration of the hyper-parameter is critical to the performance of the proposed self-attention based model, since it directly affects the generalization and regression of the model. To balance the whole information from the word sequence and the label sequence, we introduce γ parameter. The hyper-parameter γ is designed to balance the loss from the label sequence and the word sequence. In order to highlight the word sequence loss, we set the γ value to 0.15 to weaken the influence of label sequence .

Metric Following previous works (Wang et al., 2016), token-based precision (P), recall (R), and F-score (F1) are used as the evaluation metrics.

	Switchboard			WMT2014		
Sharing layer	P	R	F1	P	R	F1
0	89.2	87.9	88.5	68.5	59.5	63.7
1	84.2	78.9	81.4	94.5	96.4	95.4
2	92.1	90.2	91.1	93.2	95.9	94.5
3	90.7	87.8	89.2	95.3	96.9	96.1

Table 5: Performance on Switchboard and unlabelled WMT2014 test set with different sharing layers.

4.3 Number of Weight-Sharing Layers

To investigate how the number of weight-sharing layers affects the translation performance, we vary the number of weight-sharing layers of the two encoders from 0 to 3. Table 5 shows the experimental results. It is observed that the number of weight-sharing layers has a significant impact on the performance of disfluency detection. When we set the number of weight-sharing layers to 0, resulting in two independent encoders, we get pool F-score on both test sets, especially on the WMT2014 test set. This is mainly because in testing phase, the unlabelled data will be encoded by encoder Enc_l and decoded by the totally shared decoder, while Enc_l could not encode the unlabelled data well under this condition since the encoder Enc_l could not learn any knowledge from the unlabeled data in training phase. Furthermore, the decoder needs to balance the two different kinds of input data, which leads to the small decline of performance on Switchboard corpus as well. From another perspective, it also indicates that without the weight sharing constraint, the GAN alone is insufficient to enforce the labelled data and unlabelled data to learn the similar latent representations, which results in a negative role. This confirms our intuition that the shared layers are crucial to project the labelled and unlabelled latent representations into a shared-latent space. At the opposite extreme, when all three layers of the two encoders are shared, resulting in a totally shared encoder, it is equal to feeding labelled and unlabelled data into a multi-task Transformer directly. In spite of a larger amount of training corpus, the F-score is better than the one of two independent encoders but still 1.8% lower than the one of two weight-sharing layers on Switchboard corpus. This verifies our conjecture that unlabelled data can enhance the generalization ability of model, but the shared encoder may also detrimental to the performance since it will weaken the unique characteristic of different corpora. Choosing proper number of weight-sharing layers makes much difference. We achieve best F-score of 91.1% on the commonly used Switchboard corpus test set when the two encoders share the last two layers. Under this condition, model can learn both the general characteristic and the internal characteristics such as the terminology and sentence structure of each corpus well.

Method	P	R	F1
Multi-task Transformer	91.5	87.1	89.2
Weight sharing	92.1	90.2	91.1
Transition-based (Wang et al., 2017)	91.1	84.1	87.5
Attention-based (Wang et al., 2016)	91.6	82.3	86.7
Bi-LSTM (Zayats et al., 2016)	91.6	80.3	85.9
semi-CRF (Ferguson et al., 2015)	90.0	81.2	85.4
UBT (Wu et al., 2015)	90.3	80.5	85.1
M^3N (Qian and Liu, 2013)	-	-	84.1

Table 6: Comparison with previous approaches on the test set of English Switchboard.

4.4 Results and Analysis

We compare our self-attention based models with six previous top performing systems. Table 6 shows the comparable results on English Switchboard test set. Our first model, named *multi-task Transformer*, only utilizes the Switchboard corpus for training as other baselines. It is designed to assess the performance of self-attention based model directly. As it shown in Table 6, our multi-task Transformer achieves 89.2% F-score, outperforming the Transition-based method (Wang et al., 2017), which is the state-of-the-art, by 1.7 point improvements. Compared with the semi-CRF method (Ferguson et al., 2015), which gains the best performance of linear statistical sequence labeling methods, our multi-task Transformer achieves 3.8 point improvements without leveraging any prosodic features. The performance gab between our multi-task Transformer and UBT (Wu et al., 2015), the best syntax-based method for disfluency detection, is even larger. The F-score of multi-task Transformer increases by 5.1% than UBT. Furthermore, our second model focuses on how to take the advantage of unlabelled data, since labelled data is usually scarce. A weight sharing model (named *weight sharing* in Table 6) is introduced to tackle the training problem of mixed data. It improves the best F-score of our self-attention based models from 89.2% to 91.1%, which indicates that our novel proposed weight sharing model is capable of learning useful language knowledge from unlabelled noisy dataset automatically. As a result, our weight sharing semi-supervised model achieves 3.6% improvements than the previous state-of-the-art.

5 Conclusion

In this paper, we propose a novel semi-supervised model based on weight sharing mechanism and self-attention based multi-task learning scheme, which views the disfluency detection as a translation task. In the multi-task self-attention based network, the word sequence information and labelling information are incorporated into the model at the same time, and a constrained decoding method in testing phase is applied. Experimental results show that the proposed model achieves significant improvements than the strong baseline models. We are among the first endeavors to use the translation model to handle the disfluency detection task, which can be applied to any other sequence labelling task easily. In addition, we utilize the unlabelled corpus to enhance the performance by introducing the weight sharing strategy and the generative adversarial training to enforce the similar distribution between the labelled and unlabelled data. Experimental results show that the semi-supervised model can further improve the performance significantly.

Acknowledgements

The research work is supported by the National Key Research and Development Program of China under Grant No. 2017YFB1002102 and the NSFC project 61702514.

References

[Artetxe et al.2017] Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2017. Unsupervised neural machine translation. *arXiv preprint arXiv:1710.11041*.

[Cheng et al.] Yong Cheng, Qian Yang, Yang Liu, Maosong Sun, Wei Xu, Yong Cheng, Qian Yang, Yang Liu, Maosong Sun, and Wei Xu. Joint training for pivot-based neural machine translation. In *Twenty-Sixth International Joint Conference on Artificial Intelligence*.

[Cheng et al.2016] Yong Cheng, Yang Liu, Qian Yang, Maosong Sun, and Wei Xu. 2016. Neural machine translation with pivot languages. *arXiv preprint arXiv:1611.04928*.

[Cho et al.2013] Eunah Cho, Thanh-Le Ha, and Alex Waibel. 2013. Crf-based disfluency detection using semantic features for german to english spoken language translation. *IWSLT, Heidelberg, Germany*.

[Ferguson et al.2015] James Ferguson, Greg Durrett, and Klein Dan. 2015. Disfluency detection with a semi-markov model and prosodic features. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 257–262.

[Gehring et al.2017] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. *arXiv preprint arXiv:1705.03122*.

[Hill et al.2016] Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning distributed representations of sentences from unlabelled data. *arXiv preprint arXiv:1602.03483*.

[Honnibal and Johnson2014] Matthew Honnibal and Mark Johnson. 2014. Joint incremental disfluency detection and dependency parsing. *Transactions of the Association of Computational Linguistics*, 2(1):131–142.

[Hough and Schlangen2015] Julian Hough and David Schlangen. 2015. Recurrent neural networks for incremental disfluency detection. *Interspeech 2015*.

[Johnson and Charniak2004] Mark Johnson and Eugene Charniak. 2004. A tag-based noisy channel model of speech repairs. In *Meeting of the Association for Computational Linguistics, 21-26 July, 2004, Barcelona, Spain*, pages 33–39.

[Liu et al.2006] Yang Liu, Elizabeth Shriberg, et al. 2006. Enriching speech recognition with automatic detection of sentence boundaries and disfluencies. *IEEE Transactions on audio, speech, and language processing*.

[Ostendorf and Hahn2013] Mari Ostendorf and Sangyun Hahn. 2013. A sequential repetition model for improved disfluency detection. In *INTERSPEECH*, pages 2624–2628.

[Qian and Liu2013] Xian Qian and Yang Liu. 2013. Disfluency detection using multi-step stacked learning. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 820–825.

[Rasooli and Tetreault2013] Mohammad Sadegh Rasooli and Joel Tetreault. 2013. Joint parsing and disfluency detection in linear time. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 124–129.

[Saha et al.2016] Amrita Saha, Mitesh M Khapra, Sarath Chandar, Janarthanan Rajendran, and Kyunghyun Cho. 2016. A correlational encoder decoder architecture for pivot based sequence generation. *arXiv preprint arXiv:1606.04754*.

[Sennrich et al.2015] Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

[Shen et al.2015] Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2015. Minimum risk training for neural machine translation. *arXiv preprint arXiv:1512.02433*.

[Shen et al.2017] Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Shirui Pan, and Chengqi Zhang. 2017. Disan: Directional self-attention network for rnn/cnn-free language understanding. *arXiv preprint arXiv:1709.04696*.

[Shriberg1994] Elizabeth Ellen Shriberg. 1994. *Preliminaries to a theory of speech disfluencies*. Ph.D. thesis, Citeseer.

[Vaswani et al.2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010.

[Vincent et al.2008] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM.

[Wang et al.2016] Shaolei Wang, Wanxiang Che, and Ting Liu. 2016. A neural attention model for disfluency detection. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 278–287.

[Wang et al.2017] Shaolei Wang, Wanxiang Che, Yue Zhang, Meishan Zhang, and Ting Liu. 2017. Transition-based disfluency detection using lstms. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2785–2794.

[Wu et al.2015] Shuangzhi Wu, Dongdong Zhang, Ming Zhou, and Tiejun Zhao. 2015. Efficient disfluency detection with transition-based parsing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 495–503.

[Wu et al.2016] Yonghui Wu, Mike Schuster, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

[Zayats et al.2014] Victoria Zayats, Mari Ostendorf, and Hannaneh Hajishirzi. 2014. Multi-domain disfluency and repair detection. In *Fifteenth Annual Conference of the International Speech Communication Association*.

[Zayats et al.2016] Vicky Zayats, Mari Ostendorf, and Hannaneh Hajishirzi. 2016. Disfluency detection using a bidirectional lstm. *arXiv preprint arXiv:1604.03209*.