IMPROVING DISFLUENCY DETECTION WITH MULTI-SCALE SELF ATTENTION AND CONTRASTIVE LEARNING

Peiying Wang, Chaoqun Duan, Meng Chen[*]*, Xiaodong He*

JD AI, Beijing, China
{wangpeiying3, duanchaoqun1, chenmeng20, xiaodong.he}@jd.com

ABSTRACT

Disfluency detection aims to recognize disfluencies in sentences. Existing works usually adopt a sequence labeling model to tackle this task. They also attempt to integrate into models the feature that the disfluencies are similar to the correct phrase, the so-called "rough copy". However, they heavily rely on hand-craft features or word-to-word match patterns, which are insufficient to precisely capture such rough copy and cause under-tagging and over-tagging problems. To alleviate these problems, we propose a multi-scale self-attention mechanism (MSAT) and design contrastive learning (CL) loss for this task. Specifically, the MSAT leverages token representations to learn representations for different scales of phrases, and then compute similarity among them. The CL adopts the fluent version of the input to build the positive and negative samples and encourages the model to keep the fluent version consistent with the input in semantics. We conduct experiments on a public English dataset Switchboard, and an in-house Chinese dataset Waihu, which is derived from an online conversation bot. Results show that our method outperforms the baselines and achieves superior performance on both datasets.

Index Terms— Disfluency detection, multi-scale self-attention, contrastive learning

1. INTRODUCTION

Disfluency detection aims to remove the non-fluent word sequence from a sentence. As in previous works [1, 2, 3, 4], disfluency consists of three distinct parts: *interregnum*, *reparandum*, and *repair*. Specifically, the *interregnum* refers to filled pauses and discourse cue words, such as "uh", "I mean", etc. The *reparandum* means what the speaker wants to replace, and the *repair* is the content that the speaker intends to adopt to replace the reparandum. Since the interregnum is always in some specific formats, it is easier to detect [5, 6]. Compared with the former, the format of the reparandum is more flexible, and it is difficult to detect. Therefore, existing works mainly concentrate on detecting the reparandum part.

Utt$_1$:	I was we were so glad to meet her	
Out$_1$:	we were so glad to meet her	(Correct)
Utt$_2$:	I camp every month camp at least one weekend	
Out$_2$:	I ~~every~~ camp at least one weekend	(Under-tagging)
Utt$_3$:	I 'm sure within those people 's minds it 's justified	
Out$_3$:	I 'm sure within those it 's justified	(Over-tagging)

Table 1: Examples of Switchboard. Phrases with red color is the reparandum and the one with blue color is the repair.

The sequence labeling model is a vital branch of approaches to this task. In the early stage, semantic feature-based discriminative models, such as conditional random fields (CRF) [7] and the semi-Markov model, were leveraged to tackle this task [8, 9]. However, they neglected the feature that the reparandum is similar to the repair, so-called rough copy, which is a salient phenomenon of this task [4, 10, 11]. To integrate such features into models, two categories of methods were explored. The first category proposes to design hand-crafted features to recognize rough copies [2, 8]. However, the hand-crafted features have obvious limitations in scalability and generalization. To eliminate the tedious work of feature engineering, the second adopts CNN models [12] to acquire repeating patterns based on the word-to-word match patterns [4, 13, 14, 15].

Despite their success, this task still faces the problems of **under-tagging** and **over-tagging** which mean either missing out some disfluencies or recognizing some correct phrases as disfluencies. Table 1 depicts three examples produced by previous works. For Utt$_1$, since the phrase "I was" is similar to "we were" in words and word order, these methods accurately acquire the similarity between them based on the relations of "I" and "we", "was" and "were". However, when facing utterances like Utt$_2$, since "camp every month" and "camp at least one weekend" are different in words, the word-to-word relations cause a wrong prediction for their relation. And the model misses out the "every", namely under-tagging. Therefore, it is insufficient to just take word-level similarity into account. In addition, existing works only focus on detecting disfluencies, while lacking constraints to keep the output fluent version consistent with the input in semantics. We argue that it may cause over-tagging like Out$_3$ in which "people 's minds" is mistaken as a disfluency.

[*] Corresponding author.

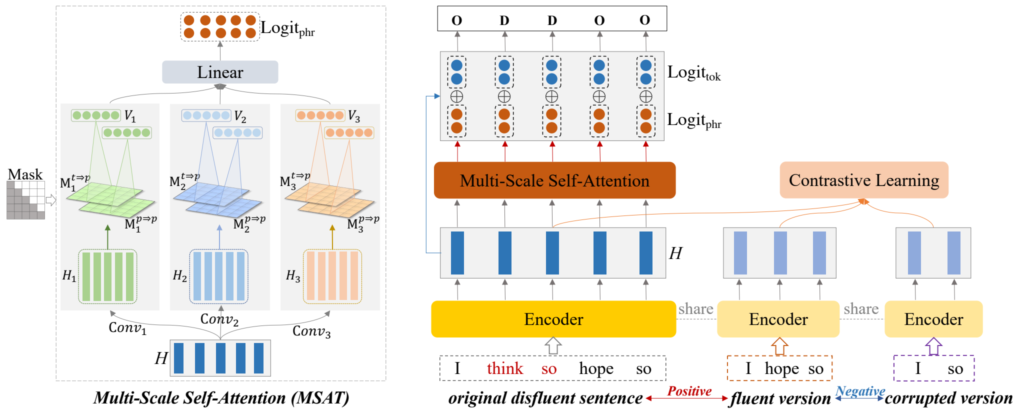


Fig. 1: The architecture of the model. The left part is multi-scale self-attention module and the right part is contrastive learning.

Based on the analysis above, we propose to improve disfluency detection with multi-scale self-attention (MSAT) and contrastive learning (CL). Specifically, we intend to capture more precisely phrase match patterns to mitigate the under-tagging problem. To this end, we first leverage the output of the pre-trained language model (PLM) [16, 17, 18] to obtain representations for different scales of phrases, and then utilize the proposed MSAT mechanism to explicitly compute similarity among them. To alleviate the over-tagging problem, we devise an auxiliary CL loss [19] to constrain the training. Specifically, we take the fluent version of the input as a positive sample and delete some words from it to build a negative sample. Based on these two samples, our model can ensure the input is close to the fluent version and away from the over-tagged one. To verify the effectiveness of our method, we conduct extensive experiments on Switchboard and Waihu. Compared with baselines, our method has made significant improvements and achieved superior performance on them.

2. METHODOLOGY

Following previous works [1, 5], we formulate the disfluency detection as a sequence labeling task and detect the target by assigning a label $T \in \{D, O\}$ to each word in the input, where D denotes the word is a part of the disfluency and O means not. Finally, we take words labeled as D as disfluencies of the input and the rest as its fluent version.

Figure 1 shows the overview architecture of the proposed method. Specifically, it consists of three parts: a contextual encoder, the proposed multi-scale self-attention mechanism, and a classifier. Inspired by the success of PLM [16, 17, 18] on numerous NLP tasks, we apply the BERT [16] as the contextual encoder without loss of generality. As for classifier, we adopt a MLP layer.

2.1. Multi-scale self-attention for disfluency detection

Inspired by the success of the multi-granularity self-attention in acquiring context information in neural machine translation [20, 21], we propose a novel multi-scale self-attention (MSAT) module to acquire relations among different phrases.

Multi-scale phrase representation. We first compute phrase representations based on the output of the contextual encoder $\mathbf{H} \in \mathbb{R}^{n \times d}$ by utilizing multiple CNN layers with different widths of filters, where n is the length of the input and d is the dimension of the hidden state. Taking phrases with length f as examples, their representations are computed as follows:

$$\mathbf{H}_f = \mathrm{Conv}_f(\mathbf{H}), \tag{1}$$

where $\mathrm{Conv}_f(\cdot)$ denotes the convolution operation with the kernel size f. To keep the sentence length unchanged, padding is adopted. We apply this operation multiple times to obtain representations for different scales of phrases.

Multi-scale self-attention. After obtaining phrase representations, we adopt them to calculate relations among different phrases. Specifically, we calculate two kinds of relations for each scale of phrases as follows:

$$\begin{aligned}\mathbf{M}_f^{t\Rightarrow p} &= \mathrm{softmax}\Big(\frac{(\mathbf{H}\mathbf{W}_q)(\mathbf{H}_f\mathbf{W}_k)^T}{\sqrt{d}}\Big),\\ \mathbf{M}_f^{p\Rightarrow p} &= \mathrm{softmax}\Big(\frac{(\mathbf{H}_f\mathbf{W}_q)(\mathbf{H}_f\mathbf{W}_k)^T}{\sqrt{d}}\Big),\end{aligned} \tag{2}$$

where $\mathbf{M}_f^{t\Rightarrow p} \in \mathbb{R}^{n \times n}$ represents relations among words and phrases with length f in the input, and $\mathbf{M}_f^{p\Rightarrow p} \in \mathbb{R}^{n \times n}$ contains the relation of each pair of phrases with length f. The softmax operation is conducted along the horizontal axis (the 2nd dimension).

Considering disfluencies usually appear before their corresponding correct phrase [10, 22], we just take the relations between a phrase and the phrases after it into account. Formally, we conduct a mask mechanism on $\mathbf{M}_f^{t\Rightarrow p}$ and $\mathbf{M}_f^{p\Rightarrow p}$ to drop their lower triangular part.

After obtaining phrase relations, we conduct max-pooling on $\mathbf{M}_f^{t\Rightarrow p}$ and $\mathbf{M}_f^{p\Rightarrow p}$ along the horizontal axis (the 2nd dimension) to obtain the relation score between each word or phrase and its most similar phrase as follows:

$$\begin{aligned}\mathbf{V}_f^{t\Rightarrow p} &= \mathrm{Max\_Pooling}(\mathbf{M}_f^{t\Rightarrow p}),\\ \mathbf{V}_f^{p\Rightarrow p} &= \mathrm{Max\_Pooling}(\mathbf{M}_f^{p\Rightarrow p}),\end{aligned} \tag{3}$$

where $\mathbf{V}_f^{t\Rightarrow p}$, $\mathbf{V}_f^{p\Rightarrow p} \in \mathbb{R}^n$. After that, we concatenate them together as $\mathbf{V}_f = [\mathbf{V}_f^{t\Rightarrow p} : \mathbf{V}_f^{p\Rightarrow p}]$, where $\mathbf{V}_f \in \mathbb{R}^{2\times n}$. Similarly, we conduct max-pooling on other scales of phrases and concatenate all of their outputs as $\mathbf{V}$. Subsequently, we feed it into a linear projection to fuse these features. Specifically, we take $\{D, O\}$ as the label set in this work. Therefore, each element of $\mathbf{V}$ is converted into a 2-D vector as follows:

$$\mathbf{Logit}_{\text{phr}} = \text{MLP}(\mathbf{V}), \tag{4}$$

where $\mathbf{Logit}_{\text{phr}} \in \mathbb{R}^{2\times n}$ and $\text{MLP}(\cdot)$ is a linear projection. Besides, to leverage the original token information, we also transform $\mathbf{H}$ into a 2-D vector $\mathbf{Logit}_{\text{tok}}$ by another linear projection. Then the sum of them is fed into the classifier and the cross-entropy loss $\mathcal{L}_{ce}$ is taken as the training objective.

2.2. Contrastive learning for semantic consistency

To alleviate the over-tagging problem, we introduce an auxiliary CL loss to constrain the training. Formally, given the input S, we take its fluent version as the positive sample S^+ since the fluent version contains all of the indispensable parts of the input. As to the negative sample, we design two self-supervised strategies by corrupting some words from the fluent version and denote it as S^-. For the first one, given that any words in the fluent sentence may be misrecognized, we randomly delete 15% of words in a fluent sentence and keep the rest part as the negative sample. We denote it as "random deletion" (RD). The second strategy constructs the negative sample by deleting top K[1] keywords, which are selected based on the TF-IDF scores [23] from the fluent sentence, and we denote it as "importance deletion" (ID).

After obtaining the positive and negative samples, we feed them into the encoder to learn their representations $\mathbf{H}^+$ and $\mathbf{H}^-$. Then, the CL loss is defined as follows:

$$\mathcal{L}_{cl} = -\sum_{\text{S}\in\mathcal{S}} \log \frac{e^{\text{sim}(\mathbf{H},\mathbf{H}^+)/\tau}}{e^{\text{sim}(\mathbf{H},\mathbf{H}^+)/\tau} + e^{\text{sim}(\mathbf{H},\mathbf{H}^-)/\tau}} \tag{5}$$

where $\mathcal{S}$ is all sentences in the dataset and τ is a temperature hyperparameter. $\text{sim}(\cdot)$ denotes a similarity function and we implement it with the cosine distance. Finally, we combine $\mathcal{L}_{ce}$ and $\mathcal{L}_{cl}$ as $\mathcal{L} = \mathcal{L}_{ce} + \lambda\mathcal{L}_{cl}$, where λ is a weight to balance the importance of $\mathcal{L}_{ce}$ and $\mathcal{L}_{cl}$.

Datasets	Train	Dev	Test
SWBD	63203	4100	4038
Waihu	20007	1962	1976

Table 2: Dataset statistics.

3. EXPERIMENTS

3.1. Experimental setup

Dataset. Experiments are conducted on two datasets, and their statistics are shown in Table 2: (1) **SWBD** [10] is the largest available public dataset of disfluency detection. Following [22], we split it into train, dev, and test sets. (2)**Waihu** is an in-house Chinese dataset for disfluency detection, which is collected from the logs of an online voice-enabled customer service bot. Specifically, we collected about 24k spoken sentences from personal statement transcriptions and invited three professional annotators to label them according to the guideline of SWBD.

Models	SWBD			Waihu		
	P	R	F1	P	R	F1
HFLSTM* [2]	91.80	80.60	85.90	-	-	-
ACNN [4]	89.50	80.00	84.50	71.21	71.20	71.61
Trans-based [24]	91.10	84.10	87.50	-	-	-
MTL [5]	93.40	87.30	90.20	76.56	73.12	74.80
LSTM	90.56	74.80	81.93	71.46	58.28	64.20
w/ MSAT	91.06	77.34	83.62	72.29	62.95	67.30
CNN	90.94	74.85	82.11	77.53	60.93	68.23
w/ MSAT	91.01	79.96	85.52	77.01	68.41	72.28
BERT	93.92	87.44	90.56	79.12	74.79	76.90
w/ MSAT	**94.83**	88.42	91.51	81.73	75.17	78.32
BERT+MSAT+CL	94.03	**89.30**	**91.61**	**82.91**	**75.39**	**78.97**

Table 3: Performance comparison on SWBD and Waihu (* denotes performance is evaluated on combination of interregnum and reparandum).

Baselines. We compare our method with the following baselines. Specifically, HFLSTM [2] proposes to integrate handcrafted features into a BiLSTM model. ACNN [4] designs an auto-correlation operator to augment the CNN model to capture rough copies. Trans-based [24] models the problem of disfluency detection by using a transition-based framework. MTL [5] proposes two self-supervised tasks for disfluency detection to tackle the training data bottleneck. LSTM and CNN adopt vanilla BiLSTM and CNN as encoders and only take sentences as input respectively. BERT [16] is directly fine-tuned with the training set.

Implementation Detail. We use the BERT-base as the contextual encoder. For phrase scales, we simultaneously adopt 1, 2, 3, and 4. When computing the CL loss, we take the average token embeddings as the sentence representation, and τ is set to 0.05. When training, we use the AdamW [25] as the optimizer, and the learning rate is 1e-5. The batch size is 24, and all models are trained for 3 epochs. In this work, we only analyze the performance of models on detecting the reparandum. Following previous works [2, 4, 10], precision, recall and f-score (P/R/F1) are adopted as evaluation metrics.

3.2. Main results and ablation study

Table 3 depicts the performance comparison of baselines and our method. Results show that our method achieves 91.61 and 78.97 on SWBD and Waihu, and outperforms all baselines on these two datasets, which proves the effectiveness of our method. To verify the universality of the MSAT, we also plug the model into the vanilla LSTM, CNN, and BERT, and then compare the performance of the variants and the original models. Results show that the variants are all superior to the original ones, demonstrating that phrase-to-phrase simi-

[1] Empirically, K $\in \{1, 2, 3, 4, 5\}$ is tuned and 3 is selected based on the results of the dev set.

Models	SWBD			Waihu		
	P	R	F1	P	R	F1
BERT	93.92	87.44	90.56	79.12	74.79	76.90
w/ 1-gram	93.72	**89.20**	91.40	81.70	74.48	77.92
w/ 2-gram	93.62	88.78	91.13	81.09	73.56	77.14
w/ 3-gram	93.93	88.74	91.26	81.21	73.44	77.13
w/ 4-gram	93.14	88.92	90.98	81.46	74.23	77.67
w/ MSAT	**94.83**	88.42	**91.51**	**81.73**	**75.17**	**78.32**

Table 4: Effects of different scales of phrases.

Neg. Sample	SWBD			Waihu		
	P	R	F1	P	R	F1
BERT	93.92	87.44	90.56	79.12	74.79	76.90
CL w/ RD	94.19	88.46	91.24	81.38	**75.49**	78.32
CL w/ ID	**94.52**	**88.67**	**91.48**	**82.81**	74.54	**78.46**

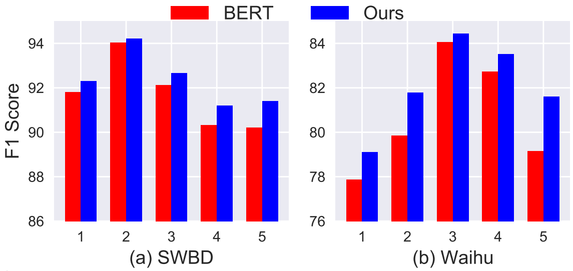
Table 5: Comparison of CL with different strategies of generating negative samples.

larity is essential for this task and the proposed MSAT can effectively acquire such information. Specially, the BERT w/ MSAT outperforms the BERT model by 0.95 on the SWBD and 1.42 on the Waihu in terms of the F1 score. It indicates that the gain of our method is not only from the PLM but also from the MSAT mechanism. Furthermore, the CNN w/ MSAT is superior to the ACNN, which explicitly proves the advantage of the MSAT on capturing the "rough copy". Finally, we equip the best setting BERT+MSAT with the CL loss. The result shows it makes a further improvement, which indicates that CL loss is also beneficial for this task.

3.3. Discussion

Effect of different scale of phrases. We range the scale of phrases from 1 to 4 and observe the performance change. As shown in Table 4, these variants all outperform the vanilla BERT. This result proves that it is essential to capture different scales of phrases for this task. Specifically, the variant equipped with 1-gram phrases achieves the greatest improvement. This is because that disfluencies with one word account for a large proportion in the dataset, e.g. 58.05% for SWBD and 32.99% for Waihu. On the other hand, many disfluencies are similar to correct phrases in words and word orders. Therefore, the word-to-word relations are sufficient to handle such simple cases. However, BERT w/ MSAT outperforms it. which demonstrates that multi-scale phrase match patterns are more beneficial for this task.

Performance on disfluencies with different length. Since multi-scale phrases are introduced into our method, we argue that it has an advantage in recognizing long disfluencies. To verify this assumption, we split the test set into 5 groups based on the length of disfluencies, and then compare our method with the BERT. As shown in Figure 2, our method outperforms BERT in terms of F1 on all groups. Specifically, starting from the group with length 3, with the increase of the length, the gain of our model becomes larger, which is completely consistent with our expectations.



(a) SWBD

(b) Waihu

Fig. 2: Trends of F1 with different length of disfluencies.

Effect of the CL. To verify if the CL can alleviate the over-tagging, we compare our model variants with CL and without CL on the subset of the test set in which all sentences are fluent. Intuitively, if a model detects any disfluency, it must be wrong. Results show that the model w/o CL can correctly predict 98.07% of such instances on SWBD and 90.55% on Waihu. While the model w/ CL achieves 98.59% on the SWBD and 92.77% on the Waihu. It indicates that the CL effectively prevents the model from over-tagging.

Furthermore, we compare different strategies of generating the negative samples. As shown in Table 5, both strategies improve the performance of the baseline. This result proves they are both beneficial for this task. Furthermore, the CL w/ ID is slightly superior to the CL w/ RD on both datasets. It indicates that random deletion may introduce noise, while the importance deletion tends to generate hard negative samples.

BERT:	I camp every month camp at least one weekend
Ours:	I camp every month camp at least one weekend
BERT:	I work in I 'm a on the professional administrative
Ours:	I work in I 'm a on the professional administrative

Table 6: Comparison of the BERT and our method on two examples of the test set of SWBD (Highlighted words mean the ground truth and underlined text denotes the prediction).

Case study. Table 6 illustrates two examples from the outputs of BERT and our method. Note that BERT misses out "every" for the first one and "I work" for the second one. Meanwhile, it mistakes "I 'm" as disfluencies. Compared with it, our method produces correct answers for them. These results prove that our method can effectively alleviate under-tagging and over-tagging problems.

4. CONCLUSIONS

In this work, we attempt to adopt a novel multi-scale self-attention mechanism (MSAT) and contrastive learning (CL) to enhance disfluency detection. Specifically, MSAT captures the similarity between different scales of phrases, which can effectively capture the "rough copy" in the input. Meanwhile, we also introduce CL based on semantic consistency to prevent our method from mistaking correct content as disfluencies. Our method significantly outperforms baselines both on SWBD and Waihu datasets. In the future, we will explore integrating different linguistic features to facilitate this task.

5. REFERENCES

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

[2] Vicky Zayats, Mari Ostendorf, and Hannaneh Hajishirzi, "Disfluency detection using a bidirectional LSTM," in *Proceedings of Interspeech*, 2016, pp. 2523–2527.

[3] Paria Jamshid Lou and Mark Johnson, "Disfluency detection using a noisy channel model and a deep neural language model," in *Proceedings of ACL*, 2017, pp. 547–553.

[4] Paria Jamshid Lou, Peter Anderson, and Mark Johnson, "Disfluency detection using auto-correlational neural networks," in *Proceedings of EMNLP*, 2018, pp. 4610–4619.

[5] Shaolei Wang, Wanxiang Che, Qi Liu, Pengda Qin, Ting Liu, and William Yang Wang, "Multi-task self-supervised learning for disfluency detection," in *Proceedings of AAAI*, 2020, pp. 9193–9200.

[6] Shaolei Wang, Zhongyuan Wang, Wanxiang Che, and Ting Liu, "Combining self-training and self-supervised learning for unsupervised disfluency detection," in *Proceedings of EMNLP*, 2020, pp. 1813–1822.

[7] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of ICML*, 2001, pp. 282–289.

[8] Eunah Cho, Thanh-Le Ha, and Alex Waibel, "Crf-based disfluency detection using semantic features for german to english spoken language translation," in *Proceedings of the 10th International Workshop on Spoken Language Translation: Papers*, 2013.

[9] James Ferguson, Greg Durrett, and Dan Klein, "Disfluency detection with a semi-Markov model and prosodic features," in *Proceedings of NAACL*, 2015.

[10] John J. Godfrey, Edward Holliman, and Jane McDaniel, "SWITCHBOARD: telephone speech corpus for research and development," in *Proceedings of ICASSP*, 1992, pp. 517–520.

[11] Victoria Zayats, Mari Ostendorf, and Hannaneh Hajishirzi, "Multi-domain disfluency and repair detection," in *Proceedings of Interspeech*, 2014, pp. 2907–2911.

[12] Keiron O'Shea and Ryan Nash, "An introduction to convolutional neural networks," *arXiv preprint arXiv:1511.08458*, 2015.

[13] Vicky Zayats and Mari Ostendorf, "Robust cross-domain disfluency detection with pattern match networks," *CoRR*, vol. abs/1811.07236, 2018.

[14] Xuezhe Ma and Eduard H. Hovy, "End-to-end sequence labeling via bi-directional lstm-cnns-crf," in *Proceedings of ACL*, 2016.

[15] Julian Hough and David Schlangen, "Joint, incremental disfluency detection and utterance segmentation from speech," in *Proceedings of EACL*, 2017, pp. 326–336.

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL*, 2019, pp. 4171–4186.

[17] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut, "ALBERT: A lite BERT for self-supervised learning of language representations," *CoRR*, vol. abs/1909.11942, 2019.

[18] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning, "ELECTRA: pre-training text encoders as discriminators rather than generators," in *Proceedings of ICLR*, 2020.

[19] Tianyu Gao, Xingcheng Yao, and Danqi Chen, "SimCSE: Simple contrastive learning of sentence embeddings," in *Proceedings of EMNLP*, 2021, pp. 6894–6910.

[20] Jie Hao, Xing Wang, Shuming Shi, Jinfeng Zhang, and Zhaopeng Tu, "Multi-granularity self-attention for neural machine translation," in *Proceedings of EMNLP*, 2019, pp. 887–897.

[21] Phi Xuan Nguyen and Shafiq R. Joty, "Phrase-based attentions," 2018, vol. abs/1810.03444.

[22] Eugene Charniak and Mark Johnson, "Edit detection and parsing for transcribed speech," in *Proceedings of NAACL*, 2001.

[23] Slobodan Beliga, "Keyword extraction: a review of methods and approaches," *University of Rijeka, Department of Informatics, Rijeka*, vol. 1, no. 9, 2014.

[24] Shaolei Wang, Wanxiang Che, Yue Zhang, Meishan Zhang, and Ting Liu, "Transition-based disfluency detection using lstms," in *Proceedings of EMNLP*, 2017, pp. 2785–2794.

[25] Ilya Loshchilov and Frank Hutter, "Decoupled weight decay regularization," in *Proceedings ICLR*, 2019.