Token-Level Supervised Contrastive Learning for Punctuation Restoration

Qiushi Huang[1,2], *Tom Ko*[1*], *H Lilian Tang*[2], *Xubo Liu*[2], *Bo Wu*[3]

[1]Department of Computer Science and Engineering
Southern University of Science and Technology, Shenzhen, China
[2]School of Computer Science and Electronic Engineering, University of Surrey, UK
[3]MIT-IBM Watson AI Lab, Cambridge, USA

qiushi.huang@surrey.ac.uk, tomkocse@gmail.com,{h.tang, xubo.liu}@surrey.ac.uk,
bo.wu@ibm.com

arXiv:2107.09099v3 [cs.CL] 3 Mar 2023

Abstract

Punctuation is critical in understanding natural language text. Currently, most automatic speech recognition (ASR) systems do not generate punctuation, which affects the performance of downstream tasks, such as intent detection and slot filling. This gives rise to the need for punctuation restoration. Recent work in punctuation restoration heavily utilizes pre-trained language models without considering data imbalance when predicting punctuation classes. In this work, we address this problem by proposing a token-level supervised contrastive learning method that aims at maximizing the distance of representation of different punctuation marks in the embedding space. The result shows that training with token-level supervised contrastive learning obtains up to 3.2% absolute F_1 improvement on the test set.[1]

Index Terms: punctuation restoration, supervised contrastive learning, imbalance data

1. Introduction

Punctuation symbols are often absent in the transcript generated by the automatic speech recognition (ASR) system. That often causes degradation in readability for both humans and machines [1]. Removing punctuation from transcriptions notably impacts the comprehension of text. For subsequent machine learning tasks, such as intent detection or slot filling, the performance suffers from missing punctuation since such models usually are trained on clean text with punctuation marks. Thus, it is essential to predict and insert punctuation for the speech transcripts.

Many methods have been proposed to tackle this problem. They can generally divide into three paradigms. The first approach considers the prosodic feature as an essential clue to punctuation insertion [2, 3]. It is natural to consider adding punctuation by the prosody of the speech since humans often adopt this approach to add punctuation in their minds. Nevertheless, the acoustic prosody feature is often noisy and error-prone, leading to a sub-optimal result. The second paradigm fuses prosodic features with lexical information. These methods incorporate acoustic and textual features by embedding different features [4, 5, 6]. However, the dataset with the acoustic and textual features is not always readily available. The third approach is to incorporate just the lexical information in the transcripts. Early attempts of lexical approach on restoring punctuation utilize n-gram language model that treats the prediction as hidden events [7, 8, 9]. Later, some proposed work demonstrates the methods based on recurrent neural network (RNN) on the punctuating task [10, 11, 1]. Unlike previous attempts, RNN can be better capture textual context within the segment than the n-gram methods. Recently, the transformer [12] based language model dominates language-related tasks, which has better performance and the capability to capture contextual information across longer distances than the RNN. Meanwhile, several transformer-based language models [13, 14, 15] that are pre-trained on gigabytes of corpora greatly enhance the performance of downstream tasks, such as Named Entity Recognition. A few approaches through pre-trained models for the punctuating task have been proposed with promising performance [16, 17, 18, 19, 20]. This work will explore this approach as our based model and work with textual data only. Therefore, the problem can be as well simplified to the textual sequential labeling task.

As most existing works [16, 17, 18, 19, 20] consider this problem a token-level classification task, it leads to a data imbalance problem. In the IWSLT2011[2] dataset, which is commonly used in the automatic punctuation restoration task, over 85% of tokens are NO PUNCTUATION, and only less than 15% of them are with punctuation labels. Moreover, the question mark accounts for only 0.54% of the total data, much less than that of other classes. The class imbalance problem will influence the performance of the punctuating task and should be addressed.

Contrastive learning [21, 22] can leverage the information from all classes rather than those from the corresponding class during training. By contrasting information from all labels, the problem of data imbalance can be alleviated. This paper incorporates supervised contrastive learning (SCL) to demonstrate the learning effectiveness from imbalanced data. Unlike the previous method, which tries to solve this problem [17] by adopting the weighted term to loss, contrastive learning contrasts the information from the same class against other classes utilizing all the information within a batch in the loss calculation. Meanwhile, clusters of the same classes in the embedding space are pulled together, making the classifier trivial to find the boundaries in the latent space. One linear layer that follows the transformer-based model can effectively classify the punctuation without complex structures like bidirectional LSTM [23, 24] or transformers [25].

The main contribution of this paper is to incorporate token-level supervised contrastive learning to address the data imbalance problem in punctuating restoration. Experiments conducted on the IWSLT dataset show the models trained with this approach gain up to 3.2% absolute overall F_1 score than the model trained with just the cross-entropy loss.

The rest of the paper is organized as follows: Section 2 reviews the contrastive learning theory. Section 3 describes our

* corresponding author

[1]The code is available at https://github.com/hqsiswiliam/punctuation-restoration-scl

[2]http://hltc.cs.ust.hk/iwslt/index.php/evaluation-campaign/ted-task.html

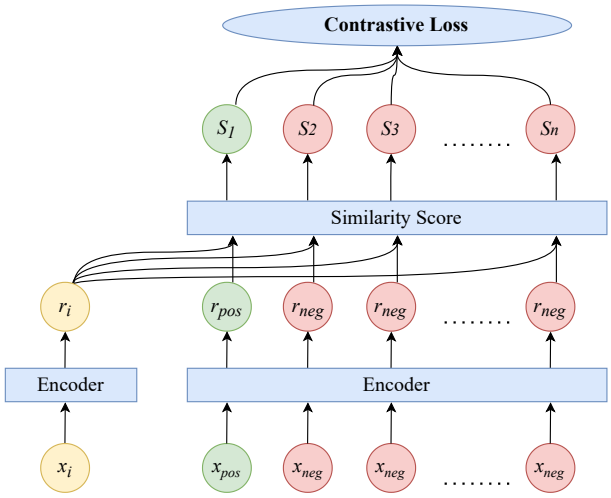

Figure 1: *Given an anchor sample x_i, a positive sample x_{pos}, and negative samples $\{x_{neg}\}$ in a batch, the encoder encodes each sample into latent space as the representation. The similarity scores $\{S_1, ..., S_n\}$ are calculated through the pairs of the anchor and each positive/negative sample. Afterward, a contrastive loss is calculated based on the similarity scores.*

proposed method, followed by detailed experiments in Section 4. Section 5 concludes and opens up points for future work.

2. Preliminaries

2.1. Contrastive learning

The concept of contrastive learning [21] builds on self-supervised learning (SSL). Self-supervised learning is a learning paradigm to capture the inherent patterns and context in data without human labeling. Therefore, self-supervised learning often constructs pretext tasks solely based on the unlabeled data and forces the network to train with these tasks to learn meaningful representation from data. Contrastive learning is to explore this approach through contrasting samples. As shown in Figure 1, given an anchor sample, a positive sample, and negative samples, similarities are calculated pairwise between the given anchor and the rest. A positive sample means it belongs to the same class as the anchor sample (e.g., augmented image of the anchor image or different time slices from audio). In contrast, negative samples mean those not in the same class as the anchor. Then, a contrastive loss is computed using the similarity scores. The formula of the self-supervised contrastive loss takes the following form.

$$\mathcal{L}_{CL} = -\sum_{i \in I} \log \frac{f(z_i, z_{j(i)})}{\sum_{k \in A(i)} f(z_k, z_i)} \quad (1)$$

Here, $f(z_i, z_{j(i)}) = \exp(z_i \cdot z_{j(i)}/\tau)$ calculates the similarity between z_i and $z_{j(i)}$. τ is the temperature, a scalar to stabilize the calculation. $z_i = \text{Encoder}(x_i)$ denotes the representation calculated by the encoder. i denotes the anchor sample; $j(i)$ denotes its positive sample; $A(i)$ is the set that contains the positive sample and the negative samples.

2.2. Supervised contrastive learning

In Supervised Contrastive Learning (SCL) [22], prior knowledge in the labeled data can be utilized. Since the anchor label is known, the label information can identify the positive and

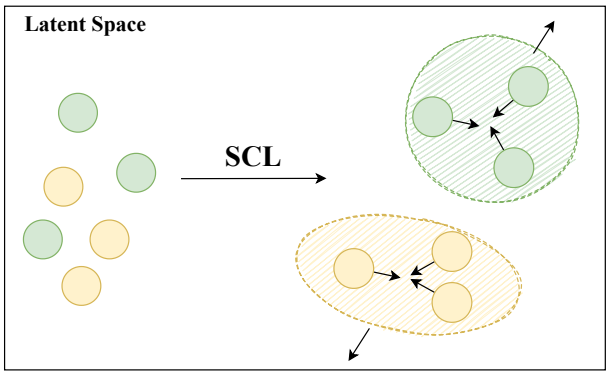

Figure 2: *The representations belonging to the same class are pulled together in the latent space while simultaneously pushing apart clusters of samples from different classes by Supervised Contrastive Learning (SCL).*

negative samples. Therefore, equation 1 can be generalized as follows.

$$\mathcal{L}_{SCL} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{f(z_i, z_p)}{\sum_{k \in A(i)} f(z_k, z_i)} \quad (2)$$

The $P(i)$ presented in the $\mathcal{L}_{SCL}$ means all positives correspond to anchor i, retrieved by the label information within a batch. The $|P(i)|$ means the number of items in this set.

By adopting supervised contrastive learning in training, representations belonging to the same class are pulled together in the latent space while simultaneously pushing apart clusters of samples from different classes, as shown in Figure 2. Supervised contrastive learning is closely related to the triplet loss [22], one of the widely used losses in face recognition [26], speaker identification tasks [27]. The triplet loss is a special case of supervised contrastive learning where only one negative is used.

3. Approach

There are two simultaneous objectives during the training. The first one is to train the representations of each token that have distinctive clusters for different labels. The second is to build the boundaries among these clusters to accomplish the classification task. To achieve this, we incorporate a supervised contrastive learning term with the standard cross-entropy loss.

3.1. Problem setting

Given an input sequence $X = \{x_1, x_2, ..., x_n\}$ where n denotes the length of the sequence, each $x_i \in X$ denotes the word in a document. The input X is encoded into representation vectors in latent space as $R = \text{Encoder}(X)$ where $R = \{r_1, r_2, ..., r_n\}$. Then, the ground truth is formalized as $Y = \{y_1, y_2, ..., y_n\}$ where length n in Y is equal to the length n in X. Meanwhile, the y_i is a predefined list of the four possible punctuation in the documents, which can be *O (No Punctuation), COMMA, PERIOD, QUESTION (Question Mark)*. Afterward, the predicted result from the model is formalized as $\hat{Y} = \{\hat{y_1}, \hat{y_2}, ..., \hat{y_n}\}$.

3.2. Token-level supervised contrastive learning

In contrast to the work in [28], which applies SCL to contrast sentences within a batch, we apply token-level SCL to contrast words within a batch. To leverage the label information into both

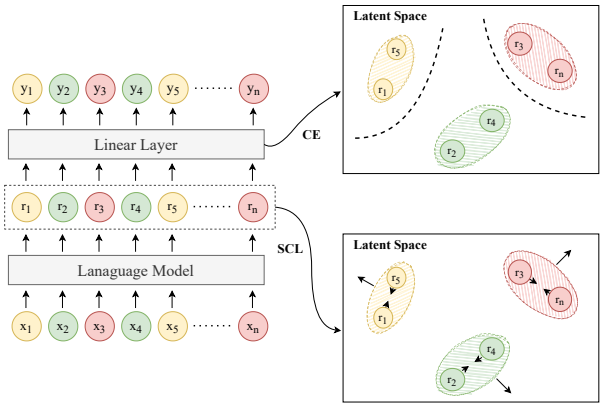

Figure 3: *The supervised contrastive learning (SCL) separates the representations of different classes. Meanwhile, cross-entropy (CE) draws the decision boundaries onto the clusters built by SCL.*

contrastive learning and supervised learning, we add supervised contrastive learning with cross-entropy concurrently, which can be given by the following formula.

$$\mathcal{L} = (1-\lambda)\mathcal{L}_{CE} + \lambda\mathcal{L}_{SCL} \tag{3}$$

$$\mathcal{L}_{CE} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\cdot\log\hat{y}_{i,c} \tag{4}$$

$$\mathcal{L}_{SCL} = \sum_{i\in I}\frac{-1}{|P(i)|}\sum_{p\in P(i)}\log\frac{\exp(\Phi(r_i)\cdot\Phi(r_p)/\tau)}{\sum_{k\in A(i)}\exp(\Phi(r_i)\cdot\Phi(r_k)/\tau)} \tag{5}$$

Here, we add a hyperparameter λ to control the balance between cross-entropy and supervised contrastive learning loss in equation 3.

By introducing equation 3, we can simultaneously calculate the clusters between different representations r and build the boundaries among different clusters in latent space. Therefore, the vectors for the same label in latent space are pulled together. Meanwhile, the cross-entropy takes charge of classifying the labels from the clusters sorted by supervised contrastive learning.

For the supervised contrastive loss, set I in $\mathcal{L}_{SCL}$ means the four types of labels: *O, COMMA, PERIOD, QUESTION*. Therefore, representations of all four labels are computed where the label contexts for punctuation and non-punctuation are fully incorporated within equation 5.

The Φ denotes the ℓ_2 normalization of R after we get the R from Encoder(X). Since we get ℓ_2 normalization on R. Therefore, the dot product between r_i and r_p (or r_i and r_k) means the cosine similarity between the representation pair since the normalized representations exclude the influence from their magnitudes. Meanwhile, adding ℓ_2 on r avoids the issue that calculates the exp on a large number, which might cause the infinity error in the empirical practice. The N is the number for the samples that belong to the labels that differ from i.

The τ is the temperature to control the smoothness of the whole calculation. Small τ is beneficial to the training process since it is sensitive to the change of similarity score, which causes fluctuation for a slight change between two r. However, this would be hard to train since the loss and gradient for small τ are numerically unstable, which results in under-fitting.

3.2.1. Supervised contrastive learning on language model

As shown in Figure 3, we adopt the above loss onto a pre-trained language model. The combined loss function calculates the gradients based on different aspects regarding the clusters of representations and the nature of the classification task.

We trained the language model on the punctuation restoration with equation 3. The hidden states from the last layer of the language model are used as the representations for the punctuating tasks since they should have captured the context information throughout the feed-forward pass from previous layers. Meanwhile, by fine-tuning the task, the hidden states' distribution from the last layer is closer to the distribution of ground truth than those from previous layers. Therefore, we conduct supervised contrastive learning on the hidden states from the language model's last layer, which is denoted as R. Clusters $r_i \in R$ from different labels are pushed away that create spaces for cross-entropy to build boundaries on the sparse latent space tractably.

4. Experiments

4.1. Dataset

IWSLT TED Talk dataset [29] is a commonly used dataset for punctuation restoration tasks. This dataset consists of $2.1M$ words in the training set, $295k$ words in the validation set from the IWSLT20212 machine translation track, and $12.6k$ words (reference transcripts) in the test set are from the IWSLT2011 ASR track. Each word is labeled by one of four classes: *O, COMMA, PERIOD, QUESTION*. The ratios of these four labels are 85.7%, 7.53%, 6.3%, 0.47% respectively. The detail of the data distribution can be found in Table 3. In the preprocessing phase, we convert all words in the dataset into lower case, which prevents the sentence segment information leakage from word capitalization.

4.2. Metrics

The prediction results are evaluated by precision (P), recall (R), and F_1 score (F_1). Since we only focus on punctuation restoration task performance, the correctly predicted no punctuation would be ignored. The metrics on *COMMA, PERIOD, QUESTION* are evaluated on the test set, and the overall metrics on those three punctuation marks are calculated as well.

4.3. Experiment setup

We examined our proposed methods onto two language models: RoBERTa [14] and BERT [13]. We set the max length of the input token to 256 with a batch size of 32. Adamw optimizer [33] is used in our experiment with a learning rate of 1×10^{-5}. The dropout rate in our experiment is 0.1. Figure 3 shows the architecture of our model. We place only one linear layer after the language model to exclude the potential performance gain by other deep network structures from the contrastive loss. Meanwhile, fine-tuning the language model on merely one extra layer also shows that supervised contrastive learning separates the representations from different classes by making spaces among different clusters, creating a tractable space for one linear layer to build boundaries. We trained our model for 20 epochs, and the validation set is used to select the best model. The evaluation results are performed on the test set. The training process jointly fine-tunes the parameters of the entire network.

Table 1: *Model comparisons on IWSLT2011 Ref dataset. P, R, F_1 denote the Precision, Recall, and F_1 Score on test dataset respectively*

Models	COMMA			PERIOD			QUESTION			OVERALL		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
T-BRNN-pre [30]	65.5	47.1	54.8	73.3	72.5	72.9	70.7	63.0	66.7	70.0	59.7	64.4
BLSTM-CRF [31]	58.9	59.1	59.0	68.9	72.1	70.5	71.8	60.6	65.7	66.5	63.9	65.1
Teacher-Ensemble [31]	66.2	59.9	62.9	75.1	73.7	74.4	72.3	63.8	67.8	71.2	65.8	68.4
DRNN-LWMA-pre [32]	62.9	60.8	61.9	77.3	73.7	75.5	69.6	69.6	69.6	69.9	67.2	68.6
Self-attention [4]	67.4	61.1	64.1	82.5	77.4	79.9	80.1	70.2	74.8	76.7	69.6	72.9
Multi-task Learning [16]	68.2	68.8	68.5	81.2	81.3	81.2	81.2	81.3	82.1	76.8	77.1	77.2
Bert-Punct-BASE [20]	72.1	72.4	72.3	82.6	83.5	83.1	77.4	**89.1**	82.8	77.4	81.7	79.4
Focal Loss [17]	74.4	77.1	75.7	**87.9**	88.2	88.1	74.2	88.5	80.7	78.8	84.6	81.6
Focal Loss*	68.8	71.7	70.2	82.5	82.8	82.7	78.5	86.5	82.1	76.4	80.4	78.3
Self-Ensemble (RoBERTa$_{\text{LARGE}}$) [18]	74.3	**76.9**	75.5	85.8	**91.6**	**88.6**	83.7	**89.1**	86.3	81.3	**85.9**	83.5
Self-Ensemble (RoBERTa$_{\text{BASE}}$) [18]	76.9	75.4	**76.2**	86.1	89.3	87.7	88.9	87.0	87.9	84.0	83.9	**83.9**
Token-Level SCL (Ours)	**78.4**	73.1	75.7	86.9	87.2	87.0	**89.1**	**89.1**	**89.1**	**84.8**	83.1	**83.9**

* This is the result of our implementation for focal loss

Table 2: *F_1 Scores on cross-entropy (CE) and our supervised contrastive learning (SCL)*

Loss	BERT$_{\text{BASE}}$	RoBERTa$_{BASE}$	RoBERTa$_{LARGE}$
CE	76.4	78.5	80.9
SCL	79.6	80.4	83.9

Table 3: *Label distributions of the IWSLT dataset*

	Train	Validation	Test
Empty	1,801,727	252,922	10,943
Comma	158,392	22,451	830
Period	132,393	18,910	807
Question	9,905	1,517	46

4.4. Result

The best result reported in Table 1 is based on RoBERTa$_{\text{LARGE}}$, which has $355M$ parameters, 16 attention heads, and 24 layers of transformer encoders. The λ and temperature used in the best result are 0.1 and 0.6. To stabilize the training, we divide the temperature by a base temperature of 0.07 to get a final temperature τ, which means $\tau = temperature/base\ temperature$.

Table 1 shows that our model has the best result on *QUESTION*, which is the smallest class. We attribute this to the alleviation of the data imbalance problem by our proposed method.

4.5. Comparison with other models

T-BRNN-pre, BLSTM-CRF, Teacher-Ensemble, DRNN-LWMA-pre, Bert-Punct BASE, Multi-task Learning, and Focal Loss are trained with only text data, while the *Self-attention* model utilized both lexical and prosody features. We use the lexical data to train our model. The models based on transformer architectures outperform those on RNNs. Overall, our model achieves the best result among all the baseline methods listed in Table 1.

4.6. Comparison with focal loss

Since we have a similar base structure with the focal loss model [17], we implemented the focal loss on our experiment setting as one of the baseline methods. The model and focal loss settings used in the experiment are the same as the proposed paper, BERT$_{\text{BASE}}$ model with $\gamma = 2$ for focal loss. Our result over BERT$_{\text{BASE}}$ model is shown in Table 2. By adopting token-level supervised contrastive learning, we still gain 1.3% definite improvement on the overall F_1.

4.7. Comparison with cross-entropy

To demonstrate the effectiveness of our method, we made comparisons between our method and cross-entropy over the same experiment settings in Table 2. Here we can see over 1.9% absolute F_1 improvement on all three language models. The BERT$_{\text{BASE}}$ benefits the most from our method with an improvement of up to 3.2% absolute F_1 score.

4.8. Comparison with self-ensemble

Our approach has comparable results to the self-ensemble method. The self-ensemble method uses RoBERTa as the base language model followed by the point-wise feed-forward network with the dimension of 1568. Meanwhile, this method applies the mechanism of the sliding window to get multiple predictions for a token. Our method only has one layer after the RoBERTa$_{\text{LARGE}}$ without any ensemble mechanism. On the RoBERTa$_{\text{LARGE}}$ model, we achieved 83.9, higher than the RoBERTa$_{\text{LARGE}}$ through the self-ensemble method, with fewer layers and direct output, showing more efficiency than the self-ensemble method.

5. Conclusion

This paper leveraged the token-level supervised contrastive learning to alleviate the data imbalance problem in punctuation restoration. The results on the IWSLT2011 dataset showed the ability of supervised contrastive learning with one linear layer after the transformer-based language model. Incorporating supervised contrastive learning into the task yielded absolute performance improvement on F_1 from 1.9% to 3.2%. Also, our method achieved comparable results with the ensemble models. In future work, we plan to conduct automatic data augmentation and other techniques onto supervised contrastive learning that enables better performance and robustness in both supervised and semi-supervised learning settings.

6. References

[1] N. Ueffing, M. Bisani, and P. Vozila, "Improved models for automatic punctuation prediction for spoken and written text," in *Proc. Interspeech*, 2013.

[2] H. Christensen, Y. Gotoh, and S. Renals, "Punctuation annotation using statistical prosody models." in *Proc Isca Workshop on Prosody in Speech Recognition and Understanding*, 2001.

[3] J. Kim and P. Woodland, "A combined punctuation generation and speech recognition system and its performance enhancement using prosody," *Speech Commun.*, vol. 41, pp. 563–577, 2003.

[4] J. Yi and J. Tao, "Self-attention based model for punctuation prediction using word and speech embeddings," in *ICASSP, 2019*, pp. 7270–7274.

[5] G. Szaszák and M. Ákos Tündik, "Leveraging a Character, Word and Prosody Triplet for an ASR Error Robust and Agglutination Friendly Punctuation Approach," in *Proc. Interspeech*, 2019, pp. 2988–2992.

[6] A. Nanchen and P. N. Garner, "Empirical evaluation and combination of punctuation prediction models applied to broadcast news," in *ICASSP, 2019*, pp. 7275–7279.

[7] D. Beeferman, A. Berger, and J. Lafferty, "Cyberpunc: a lightweight punctuation annotation system for speech," in *ICASSP, 1998*, pp. 689–692 vol.2.

[8] Y. Liu, E. Shriberg, A. Stolcke, D. Hillard, M. Ostendorf, and M. Harper, "Enriching speech recognition with automatic detection of sentence boundaries and disfluencies," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 5, p. 1526–1540, Sep. 2006.

[9] A. Gravano, M. Jansche, and M. Bacchiani, "Restoring punctuation and capitalization in transcribed speech," in *ICASSP, 2009*, USA, p. 4741–4744.

[10] W. Lu and H. T. Ng, "Better punctuation prediction with dynamic conditional random fields," in *EMNLP*, 2010, pp. 177–186.

[11] P. Żelasko, P. Szymanski, J. Mizgajski, A. Szymczak, Y. Carmiel, and N. Dehak, "Punctuation prediction model for conversational speech," in *Proc. Interspeech*, 2018.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017.

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL*, 2019, pp. 4171–4186.

[14] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach," 2020.

[15] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A lite BERT for self-supervised learning of language representations," in *International Conference on Learning Representations*, 2020.

[16] B. Lin and L. Wang, "Joint Prediction of Punctuation and Disfluency in Speech Transcripts," in *Proc. Interspeech*, 2020, pp. 716–720.

[17] J. Yi, J. Tao, Z. Tian, Y. Bai, and C. Fan, "Focal Loss for Punctuation Prediction," in *Proc. Interspeech*, 2020, pp. 721–725.

[18] M. Courtland, A. Faulkner, and G. McElvain, "Efficient automatic punctuation restoration using bidirectional transformers with robust inference," in *Proceedings of the 17th International Conference on Spoken Language Translation*, 2020, pp. 272–279.

[19] A. Nagy, B. Bial, and J. Ács, "Automatic punctuation restoration with BERT models," *arXiv preprint arXiv:2101.07343*, 2021.

[20] K. Makhija, T.-N. Ho, and E. Chng, "Transfer learning for punctuation prediction," in *Proceedings of APSIPA*, 2019, pp. 268–273.

[21] A. V. D. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[22] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 18 661–18 673.

[23] T. Alam, A. Khan, and F. Alam, "Punctuation restoration using transformer models for high-and low-resource languages," in *W-NUT 2020*, pp. 132–142.

[24] J. Yi, J. Tao, Y. Bai, Z. Tian, and C. Fan, "Adversarial transfer learning for punctuation restoration," 2020.

[25] Q. Chen, M. Chen, B. Li, and W. Wang, "Controllable time-delay transformer for real-time punctuation prediction and disfluency detection," in *ICASSP, 2020*, pp. 8069–8073.

[26] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering." in *CVPR*, 2015.

[27] H. Bredin, "Tristounet: triplet loss for speaker turn embedding," in *ICASSP, 2017*, pp. 5430–5434.

[28] B. Gunel, J. Du, A. Conneau, and V. Stoyanov, "Supervised contrastive learning for pre-trained language model fine-tuning," in *International Conference on Learning Representations*, 2021.

[29] X. Che, C. Wang, H. Yang, and C. Meinel, "Punctuation prediction for unsegmented transcript based on word vector," in *LREC*, 2016.

[30] O. Tilk and T. Alumäe, "Bidirectional recurrent neural network with attention mechanism for punctuation restoration," in *Proc. Interspeech*, 2016, pp. 3047–3051.

[31] J. Yi, J. Tao, Z. Wen, and Y. Li, "Distilling knowledge from an ensemble of models for punctuation prediction," in *Proc. Interspeech*, 2017, pp. 2779–2783.

[32] S. Kim, "Deep recurrent neural networks with layer-wise multi-head attentions for punctuation restoration," in *ICASSP, 2019*, pp. 7280–7284.

[33] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2019.