SELF-ATTENTION FOR INCOMPLETE UTTERANCE REWRITING

Yong Zhang, Zhitao Li, Jianzong Wang, Ning Cheng, Jing Xiao*

Ping An Technology (Shenzhen) Co., Ltd.

ABSTRACT

Incomplete utterance rewriting (IUR) has recently become an essential task in NLP, aiming to complement the incomplete utterance with sufficient context information for comprehension. In this paper, we propose a novel method by directly extracting the coreference and omission relationship from the self-attention weight matrix of the transformer instead of word embeddings and edit the original text accordingly to generate the complete utterance. Benefiting from the rich information in the self-attention weight matrix, our method achieved competitive results on public IUR datasets.

***Index Terms*—** Incomplete Utterance Rewriting, Self-Attention Weight Matrix, Text Edit

1. INTRODUCTION

The incomplete utterance rewriting (IUR) has attracted dramatic attention in recent years due to its potential commercial value in conversation tasks. The main goal of IUR is to tackle the coreference and complement the ellipsis in the incomplete utterance and make the semantic information complete for understanding without referring to the context utterance. For the example of the multi-turn dialogue utterances (u_1, u_2, u_3) in Table 1, u_3 is the incomplete utterance that omits the subject "Shenzhen" and "this" actually refers to the "raining heavily recently" given the context utterances u_1 and u_2.

Table 1: The example of incomplete utterance rewriting

Turns	Utterance (Translation)
u_1	深圳的天气怎么样 (How is the weather in Shenzhen)
u_2	最近一直下暴雨 (It keeps raining heavily recently)
u_3	为什么这样 (why is this)
u_3^*	深圳为什么最近一直下暴雨 (Why is Shenzhen keeps raining heavily recently)

Notes: u_1 and u_2 denote the context utterances in the dialogue and u_3 is the incomplete utterance with u_3^* indicates the referenced complete utterance.

Given most omitted and coreference words come from contexts utterances, current methods mainly apply the seq2seq methods with copy mechanism[1][2] or pointer network[3] to deal with IUR. Su et al.[4] proposes a hyper-parameter λ to distinguish the attention of context and incomplete utterance based on transformer-based seq2seq model and pointer network. Pan et al.[5] apply a "pick and combine" (PAC) method, which first picks omitted words in the context utterances and utilizes the pointer generative network to take omitted words as extra features to produce the output. CSRL[6] exploits additional information from semantic role labeling (SRL) to enhance BERT representation for rewriting utterances, which requires more processes. Although they achieved promising results, they still unavoidably suffer from exposure bias and low autoregressive generation speed.

To improve the speed, SARG[7] fuses the sequence labeling and non-autoregressive generation first to identify required operations for incomplete utterance and insert words from context utterances to the incomplete utterance accordingly. RAST[8] formulates IUR task as a span prediction task of deletion and insertion with reinforcement learning to improve fluency. RUN[9] formulates the IUR task as semantic segmentation based on the feature map constructed by the similarity function on the word embeddings and achieves better performance with faster speed.

Above mentioned methods depend heavily on encoders' output which could be the information bottleneck whereas rich semantics dependency information hidden in the attention weight matrix was overlooked. In this work, we propose to shed more light on the signal hidden in the self-attention weight matrix and leverage a segmentation CNN from computer vision to extract more information for the IUR task. The self-attention weight matrix can naturally capture the coreference and omission relationships between the context utterances and the incomplete utterance. Without outputting the word embedding, we directly apply a segmentation CNN to map the learned token2token relationship in the self-attention weight matrix to edit operations in parallel. The final complete utterance can be produced by editing the incomplete utterance and context utterances based on the generated edit type tags. Our contributions are summarized below:

1. We explore the self-attention weight for the token relationship representation and apply it to the IUR.
2. We propose a straightforward and efficient model structure based on the self-attention weight matrix with low resource cost.

*Corresponding author: Jianzong Wang, jzwang@188.com. This paper is supported by the Key Research and Development Program of Guangdong Province under grant No. 2021B0101400003 and the National Key Research and Development Program of China under grant No. 2018YFB0204403.

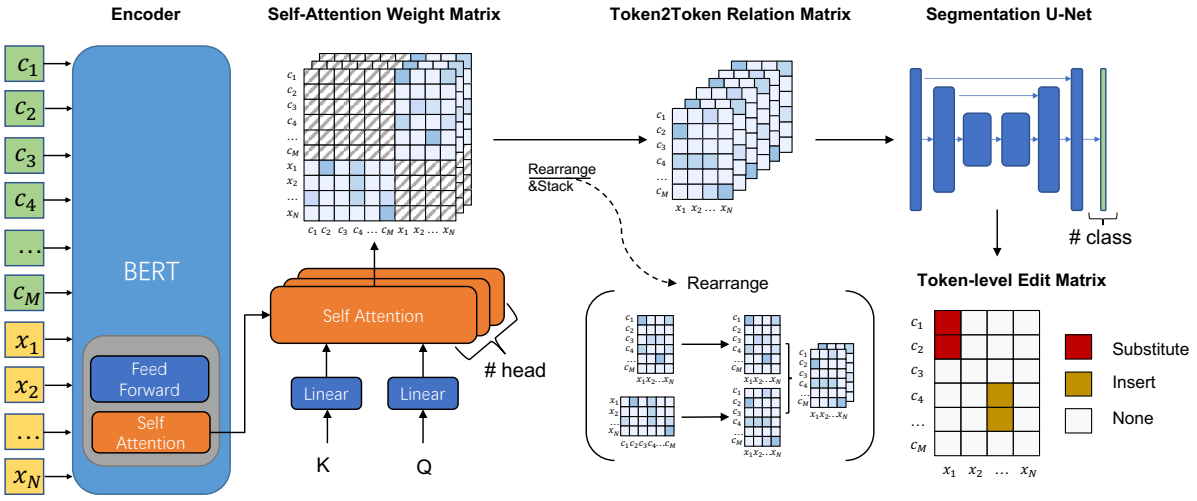

Fig. 1: The architecture of our proposed model

3. Experimental results demonstrate that our proposed method performs better than current baselines on the RESTORATION[5] and REWRITE[4] benchmark.

2. PROPOSED METHOD

In this section, we will introduce our method in detail. As shown in Figure 1, we propose a straightforward model structure with BERT[10] as the encoder to produce the token2token relation matrix and U-Net[11] as the classifier. We named our model as Rewritten Attention U-Net (RAU).

Formally, given multi-turn dialogue utterances $(u_1, u_2, \ldots, u_t)$, we concatenate all context utterances $(u_1, u_2, \ldots, u_{t-1})$ into an M-length word sequence $c = (c_1, c_2, \ldots, c_M)$ and using a special mask $[SEP]$ to separate different utterance. Besides, the last utterance in the dialogue: the incomplete utterance u_t is denoted as an N-length word sequence $x = (x_1, x_2, \ldots, x_N)$.

2.1. Token2Token Relation Mapping

Encoder We use a pre-trained language model BERT[10] as the encoder to learn the context information. The concatenation of context utterance sequence c and incomplete utterance sequence x will first be passed to the corresponding tokenizer to generate tokens and further processed by the BERT to get the contextual information among utterances. Since the model does not require the hidden state of the word for representation, the last layer's feed-forward layer is abandoned in the structure.

Token2Token Relation Matrix On top of the context-aware information learned by BERT, we propose to apply BERT's self-attention weight matrix as the representation of the classifier to learn edit operations. With pre-trained knowledge, the self-attention weight of each layer can further learn the token to token positional, syntax, and semantic relationship. And different heads of the layer pays attention to diverse perspective.

The calculation of self-attention weight[12] relies on the query dimensionality d_q and the key dimensionality d_k. For each token, dot products are performed by the query with all keys among the tokens in the input and divided each by $\sqrt{d_k}$ to smooth the value. Finally, a softmax function is applied to obtain the attention weights distribution. And the attention weight can be calculated simultaneously by packing queries and keys together into matrix Q and matrix K as:

$$\text{Attention Weight } (Q, K) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \quad (1)$$

Multi-head attention allows the model to learn the information from different aspects with different sets of query weight matrixes W_i^Q and key weight matrixes W_i^K. $Head_i \in \mathbb{R}^{(M+N)\times(M+N)}$ is self-attention weight matrix with i indicates the corresponding head.

$$\text{Head}_i = \text{Attention Weight}\left(QW_i^Q, KW_i^K\right) \quad (2)$$

Since the self-attention weight matrix includes the self-dependency of each token, the model has to select desired attention of the token in context utterances with the token in the incomplete utterance. As shown in the Token2Token Relation Matrix of Figure 1, for each head's self-attention weight matrix, the top right and the bottom-left part corresponding to the token relationship between the context utterance and the incomplete utterance are selected. And rearrange is required for the bottom left part to maintain the same shape and the order of the token. Finally, for each attention head, it can acquire a token2token relation weight matrix $\text{Head}_i^* \in \mathbb{R}^{M\times N\times 2}$:

$$\text{Head}_i^* = \text{Slice}_1\ \text{Head}_i \oplus \text{Rearrange}\ (\ \text{Slice}_2\ \text{Head}_i) \quad (3)$$

Where $Slice_1$ and $Slice_2$ respectively corresponds to the mentioned two selection operations and $\oplus$ indicates the concatenation.

Visualization As mentioned before, self-attention with different heads can help recognize the position, syntax, and semantic information. We statistically analyze the last layer's self-attention weight matrix to complement the proposed method. As shown in Figure 2, it can be observed that most

of the heads of the last layer pay more attention to semantic information (Coreference and Omission). Also, different heads will learn some syntax relationships and other information. Take Figure 3's one head's self-attention weight matrix visualization[4] as an example, this head has aligned the coreference subject "My daughter" in c with the pronoun "She" in the x, representing the semantic ability. Besides, it also highlights the omission of "eat napkins" in the target insertion position. We argue it is due to the head's position detection ability to identify the position of the current token in the correct word order cooperated with semantic knowledge.

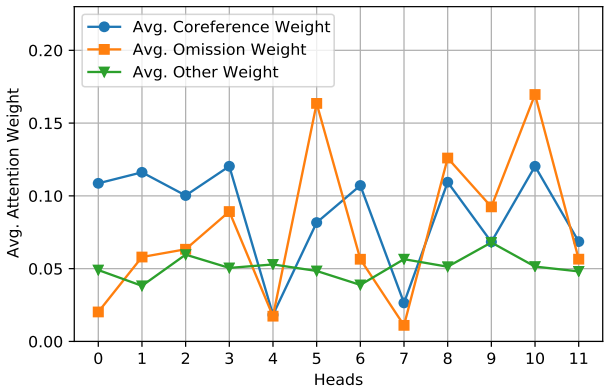

Fig. 2: Self-attention weight statistic of BERT last layer's 12 heads on the REWRITE dev set. Coreference and Omission: mentioned IUR token2token relationship. Other: other relationship. Avg. Weight: the average weight of all heads's attention for the same token2token relation type in all relation matrix cells.

Another advantage is that utilizing the self-attention weight matrix could simplify the model architecture omitting the feedforward structure of the last layer and contribute to the speed of the training and prediction.

Segmentation Regarding the token2token relation matrix as a multi-channel image, we apply the U-Net[11] to integrate low-level features to high-level features and as the classifier to map the token2token relationship to the corresponding edit type. U-Net is proposed for image segmentation in the area of CV, and it is originally used for pixels' interactively parallel classification, which is naturally suitable in our case. The down-sampling blocks of U-Net can enlarge the receptive field of token-to-token relevance embedding Head $_i^*(c_m, x_n)$ and fuse the global information learned from the encoder. And the up-sampling blocks help distribute the fused information to each cell. The output of U-Net is the same width and height as the input matrix with channel amount aligned with the edit operation amount (Substitute, Insert, and None). Each cell of the channel matrix corresponds to the score of the edit type.

$$F = \text{U-Net}\left(\oplus_i^I(\text{ Head }_i^*)\right) \quad (4)$$

$$Edit\left(c_m, x_n\right) = \text{ ArgMax } F\left(c_m, x_n\right) \quad (5)$$

where I is the amount of heads and $\oplus_i^I(\text{ Head }_i^*)$ denotes concatenating all $Head_i^*$. The $F \in \mathbb{R}^{M \times N \times C}$ is the output of the U-Net with C class channels. The class of each cell $Edit\left(c_m, x_n\right)$ is the ArgMax of $F\left(c_m, x_n\right) \in \mathbb{R}^{1 \times 1 \times C}$.

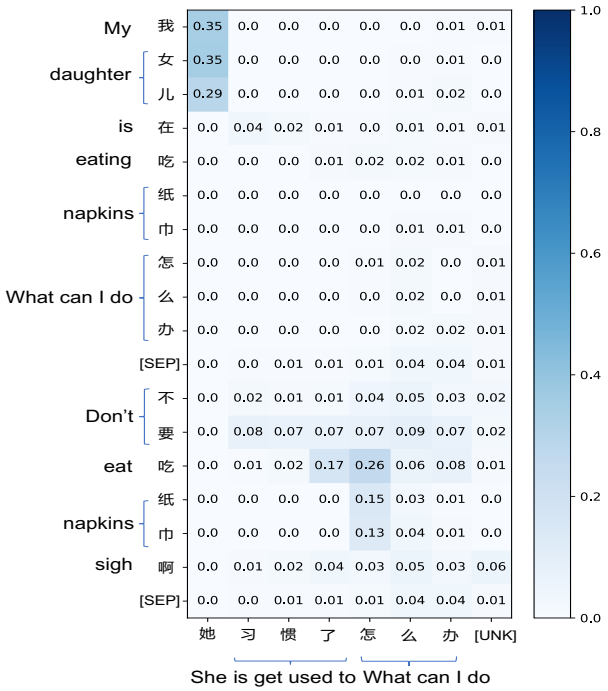

Fig. 3: Example of BERT last layer's one head's self-attention weight matrix

2.2. Incomplete Utterance Edit

After obtaining the token-level editing matrix $Edit \in \mathbb{R}^{M \times N}$ with each entry of the matrix represents the token2token editing type between c and x, we can use a simple editing algorithm to generate the complete utterance. The example is shown in Figure 3, the coreference relationship $\rightarrow$ Substitute operation: "My daughter" will substitute the "She" in x, and the omission relationship $\rightarrow$ Insert before operation: "eat napkins" will be inserted before the "What can I do". Nothing is done for None operation of the other relationship.

3. EXPERIMENTS

3.1. Setup

Datasets We conduct our experiments on RESTORATION-200K [5] and REWRITE[4] which are split as $0.8/0.1/0.1$ and $0.9/0.1/-$ for training/development/testing according to the previous methods. The dataset consists of multi-turn dialogue sentences as input and "correctly" rewritten sentences as label.

Comparing methods We compare the performance of our method with the following methods as described in INTRODUCTION: the transformer based pointer generator (T-Ptr-Gen)[2], T-Ptr-λ[4], PAC[5], CSRL[6], SARG[7], RAST[8], and RUN (BERT)[9]. For benchmark details, please refer to the corresponding paper.

Evaluation We follow the previous work's usage to apply BLEU[13], ROUGE[14], EM and restoration score[5] as the automatic evaluation metrics to compare our proposed method with others.

Table 2: The results of all compared models trained and evaluated on the RESTORATION.

Model	$\mathcal{P}_1$	$\mathcal{R}_1$	$\mathcal{F}_1$	$\mathcal{P}_2$	$\mathcal{R}_2$	$\mathcal{F}_2$	$\mathcal{P}_3$	$\mathcal{R}_3$	$\mathcal{F}_3$	$\mathbf{B}_1$	$\mathbf{B}_2$	$\mathbf{R}_1$	$\mathbf{R}_2$
T-Ptr-λ[4]	–	–	51.0	–	–	40.4	–	–	33.3	90.3	87.4	90.1	83.0
PAC[5]	70.5	58.1	63.7	55.4	45.1	49.7	45.2	36.6	40.4	89.9	86.3	91.6	82.8
CSRL[6]	–	–	–	–	–	–	–	–	–	90.6	89.7	91.1	85.0
SARG[7]	–	–	62.4	–	–	52.5	–	–	46.3	92.2	89.6	92.1	**86.0**
RAST[8]	–	–	–	–	–	–	–	–	–	90.4	89.6	91.2	84.3
RUN (BERT)[9]	73.2	64.6	68.6	59.5	53.0	56.0	50.7	45.1	47.7	92.3	89.6	92.4	85.1
RAU (Ours)	**75.0**	**65.5**	**69.9**	**61.2**	**54.3**	**57.5**	**52.5**	**47.0**	**49.6**	**92.4***	**89.6**	**92.8***	**86.0***

Notes: P_n, R_n, and F_n denote precision, recall, and F-score of n-grams restored word in rewritten utterance based on incomplete and complete utterances. The detail is described in restoration score[5]. B_n indicates n-gram BLEU score and R_n represents n-gram ROUGE score. The - indicates result is not reported in the paper. And the * indicates the result is statistically significant against all the baselines with the p-value < 0.05. The marks are the same for Table 3.

Model setting We utilize bert-base-chinese from HuggingFace's community[15] as our pre-trained BERT and it is fine-tuned as part of the training. The number of layers is 12 with 12 attention heads. Only the last layer's self-attention weight is used since it achieves the best result in our experiment. Adam[16] is utilized to optimize the model with a learning rate of 1e-5. Weighted cross-entropy is applied to address the imbalanced class distribution of mentioned three edit operations.

Table 3: The results of all compared models trained and evaluated on the REWRITE.

Model	**EM**	$\mathbf{B}_2$	$\mathbf{B}_4$	$\mathbf{R}_2$	$\mathbf{R}_L$
T-Ptr-Gen[2]	53.1	84.4	77.6	85.0	89.1
T-Ptr-λ[4]	52.6	85.6	78.1	85.0	89.0
T-Ptr-λ (BERT)[4]	57.5	86.5	79.9	86.9	90.5
RUN (BERT)[9]	66.4	91.4	86.2	90.4	93.5
RAU (Ours)	**68.4***	**91.6***	**86.6***	**90.6***	**93.9***

Notes: EM indicates the exact match score and R_L is ROUGE score based on the longest common subsequence (LCS).

3.2. Main Result

The result of Restoration and Rewrite are shown in Table 2 and Table 3. For Restoration, our method performs better than the previous best model RUN (BERT) in all n-grams F-score, that P_1, P_2 and P_3 averagely raise 2.1 points and different n-grams recall achieves comparable performance. The result indicates our method can help correctly recognize more target words with the help of sufficient information of attention weight. In addition, our model outperforms the previous model on all the BLEU and ROUGE. Although the improvement is slight, it also supports our model is robust since the BLEU and ROUGE scores of all previous models are close even restoration scores are different and our model has the highest restoration score.

For Rewrite, our method also performs better on all scores, significantly improves 2 points on the most challenging EM score, which requires an exact match of rewritten utterance with the referenced complete utterance.

3.3. Ablation Study

We conduct a series of ablation studies to evaluate the effectiveness of attention weight learned by different layers and heads of BERT. The results are depicted in Table 4.

As expected, the higher the layer, the better high-level information can be learned by the head attention. All evaluation metric scores drop consistently with lowering the layer. Given the phenomenon observed by Jawahar et al.[17] that the lower layer tends to learn the surface feature, the middle and the high layer prefer the syntax feature and semantic feature, we also try to aggregate different layer's attention into the token2token matrix. All combination's experiment result indicates last layer's information is far sufficient for the current task. We also observe that learned different level information distributes in various heads of different layers, and some heads may be lazy, which is similar to the previous observation[18]. We try to prune the heads with the first six kept. The result shows that BERT can transfer the learned information to desired heads with finetuning setting.

Table 4: Ablation results on the RESTORATION test set.

Model	$\mathcal{F}_1$	$\mathcal{F}_2$	$\mathcal{F}_3$	$\mathbf{B}_2$	$\mathbf{R}_2$
RAU $L12$	69.9	57.5	49.6	**89.6**	**86.0**
RAU $L6$	69.9	55.3	46.3	86.2	84.1
RAU $L1$	58.8	44.3	35.2	83.8	80.9
RAU $L6, 12$	**70.8**	57.5	49.1	87.2	84.9
RAU $L1, 6, 12$	70.7	**58.0**	**50.0**	87.0	85.0
RAU L *all*	70.2	57.0	48.5	87.9	85.1
RAU $L12\ H1-6$	70.2	57.1	48.7	89.2	85.6

Notes: L and H denote the layer and head of BERT with the next digit indicates the index from 1-12; "L all" means all layers are included.

4. CONCLUSIONS

In this paper, we discovered the potential usage of the overlooked self-attention weight matrix from the transformer and proposed a straightforward and effective model for the IUR task. Our model has achieved state-of-the-art performance on public IUR datasets. Deeper research on the incorporation of self-attention weight matrix for other NLP tasks and linguistics studies can be conducted in the future.

5. ACKNOWLEDGEMENTS

This paper is supported by the Key Research and Development Program of Guangdong Province under grant No. 2021B0101400003 and the National Key Research and Development Program of China under grant No. 2018YFB0204403. Corresponding author is Jianzong Wang from Ping An Technology (Shenzhen) Co., Ltd (jzwang@188.com).

6. REFERENCES

[1] J. Gu, Z. Lu, H. Li, and V. O. Li, "Incorporating copying mechanism in sequence-to-sequence learning," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1631–1640.

[2] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, Jul. 2017, pp. 1073–1083.

[3] O. Vinyals, M. Fortunato, and N. Jaitly, "Pointer networks," in *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 2*, 2015, pp. 2692–2700.

[4] H. Su, X. Shen, R. Zhang, F. Sun, P. Hu, C. Niu, and J. Zhou, "Improving multi-turn dialogue modelling with utterance rewriter," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 22–31.

[5] Z. Pan, K. Bai, Y. Wang, L. Zhou, and X. Liu, "Improving open-domain dialogue systems via multi-turn incomplete utterance restoration," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 1824–1833.

[6] K. Xu, H. Tan, L. Song, H. Wu, H. Zhang, L. Song, and D. Yu, "Semantic Role Labeling Guided Multi-turn Dialogue ReWriter," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Nov. 2020, pp. 6632–6639.

[7] M. Huang, F. Li, W. Zou, and W. Zhang, "Sarg: A novel semi autoregressive generator for multi-turn incomplete utterance restoration," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 14, 2021, pp. 13 055–13 063.

[8] J. Hao, L. Song, L. Wang, K. Xu, Z. Tu, and D. Yu, "Robust dialogue utterance rewriting as sequence tagging," *arXiv preprint arXiv:2012.14535*, 2020.

[9] Q. Liu, B. Chen, J.-G. Lou, B. Zhou, and D. Zhang, "Incomplete utterance rewriting as semantic segmentation," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Nov. 2020, pp. 2846–2857.

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 4171–4186.

[11] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Cham: Springer International Publishing, 2015, pp. 234–241.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 6000–6010.

[13] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.

[14] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, Jul. 2004, pp. 74–81.

[15] T. Wolf, J. Chaumond, L. Debut, V. Sanh, C. Delangue, A. Moi, P. Cistac, M. Funtowicz, J. Davison, S. Shleifer *et al.*, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2020, pp. 38–45.

[16] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.*

[17] G. Jawahar, B. Sagot, and D. Seddah, "What does BERT learn about the structure of language?" in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 3651–3657.

[18] K. Clark, U. Khandelwal, O. Levy, and C. D. Manning, "What does bert look at? an analysis of bert's attention," in *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, 2019, pp. 276–286.