RSMLP: A light Sampled MLP Structure for Incomplete Utterance Rewrite

Lunjun Liu^{a,b}, *Weilai Jiang^{a,b}, Yaonan Wang^{a,b}

^aCollege of Electrical and Information Engineering, Hunan University, Changsha, P.R.China

^bGreater Bay Area Institute for Innovation, Hunan University, Guangzhou, P.R.China

{barryyyliu, jiangweilai, yaonan}@hnu.edu.cn

Abstract—The Incomplete Utterance Rewriting (IUR) task has garnered significant attention in recent years. Its goal is to reconstruct conversational utterances to better align with the current context, thereby enhancing comprehension. In this paper, we introduce a novel and versatile lightweight method, Rewritten-Sampled MLP (RSMLP). By employing an MLP-based architecture with a carefully designed down-sampling strategy, RSMLP effectively extracts latent semantic information between utterances and makes appropriate edits to restore incomplete utterances. Due to its simple yet efficient structure, our method achieves competitive performance on public IUR datasets and in real-world applications.

Index Terms—Incomplete Utterance Rewriting, Fully MLP Structure, Down-Sampling, Text Edit.

I. INTRODUCTION

In recent years, conversation-based tasks have gained increasing attention, such as dialogue response generation [1] [2] and dialogue understanding [3] [4]. The advent of Large Language Models (LLMs) has shifted the focus from single-turn to multi-turn dialogues. In multi-turn dialogue scenarios, users tend to use incomplete utterances, which often omit or reference entities or concepts from previous dialogue context, a phenomenon known as ellipsis and anaphora. Studies have shown that over 70% of utterances exhibit these phenomena [5], which significantly impacts the accuracy of semantic understanding in dialogue systems.

To address this issue, recent research has introduced the Incomplete Utterance Rewriting (IUR) task [6] [7] [8]. The goal of the IUR task is to rewrite incomplete utterances into new sentences with the same semantics, where the new sentences can be understood without referring to the context. As shown in Table I, (u_1,u_2,u_3) form a multi-turn dialogue, where u_3 is an incomplete utterance that omits "Shenzhen" and uses "this" to refer to "wet". The revised u_3^* is a complete sentence that can be understood independently. By explicitly rewriting the omitted information into the latest utterance, downstream dialogue models only need to process the final

This paper is supported by the National Natural Science Foundation of China under Grant 62473138, the Project of Natural Science Foundation Youth Enhancement Program of Guangdong Province under Grant 2024A1515030184, the Project of Guangzhou City Zengcheng District Key Research and Development under Grant 2024ZCKJ01, and the General Project of Natural Science Foundation of Hunan Province under Grant 2022JJ30162. *Corresponding author is Weilai Jiang from Hunan University (Jiangweilai@hnu.edu.cn)

 $\label{thm:table interpolation} TABLE\ I$ The example of incomplete utterance rewriting .

Turns	Utterance (Translation)
u_1	深圳的气候怎么样
	(How is the climate in Shenzhen)
u_2	十分潮湿
	(It is quite wet)
u_3	为什么会这样
	(Why is this)
u_3^*	深圳的气候为什么会十分潮湿
0	(Why is the climate in Shenzhen so wet)

utterance. This significantly alleviates the model's burden during long-term reasoning.

Although significant progress has been made in previous work, balancing the quality of sentence rewriting with autoregressive generation speed remains a challenge for the IUR task. To improve speed, RUN [9] framed the IUR task as a semantic segmentation problem based on feature mappings constructed from word embeddings. RAU [10] extracted coreference and ellipsis relationships from the self-attention weight matrices of transformers. Both approaches utilize U-Net, a complex model that significantly impacts rewriting efficiency. To enhance quality, SRL [11] trained a semantic role labeling model to emphasize the core meaning of each input dialogue, preventing the rewriter from violating key content. They manually annotated SRL information for over 27,000 dialogue turns, a time-consuming and costly process. PAC [7] constructs a "pick-and-combine" model to extract omitted tokens from the context in order to restore incomplete sentences. RAST [12] formulated the IUR task as a span prediction problem of deletion and insertion, using reinforcement learning to improve fluency, which heavily depends on the encoder's output.

In this paper, we explore the use of a light neural network architecture for sentence rewriting, specifically utilizing a simple MLP architecture based on a down-sampling strategy. Our approach involves using down-sampling and MLP to sequentially refine local and global semantic information. Subsequently, we perform similarity calculations on the output semantic matrices to obtain similarity feature maps, which are then used to construct a token-level edit matrix. Finally, we edit incomplete utterances based on the predicted edit type tokens to generate the rewritten sentences. Our contributions are summarized as follows:

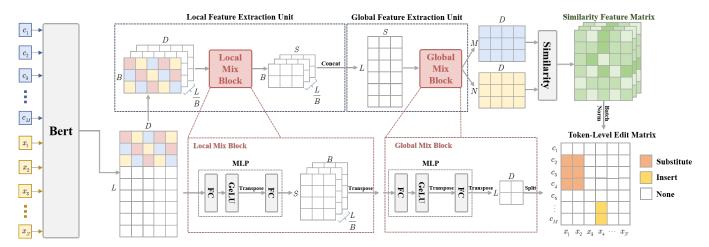


Fig. 1. The architecture of RSMLP.

- We investigate capturing local information in the IUR task using a down-sampling approach.
- We propose a model structure, RSMLP, composed solely of MLPs, which is both simple and efficient.
- Experimental results demonstrate that our method achieves a balance between rewriting quality and inference speed, and performs well in real-world scenarios.

II. RELATED WORKS III. METHOD

In this section, we will provide a detailed explanation of our method.

A. Problem Definition

We formally define the utterances that need to be rewritten. For multi-turn dialogue utterances (u_1,u_2,\ldots,u_t) , we concatenate all the context utterances (u_1,u_2,\ldots,u_{t-1}) into a word sequence of length M, denoted as $c=(c_1,c_2,\ldots,c_M)$, and use a special [SEP] token to separate utterances from different contexts. The final utterance u_t in the dialogue is defined as the incomplete utterance, represented as a word sequence of length N, $x=(x_1,x_2,\ldots,x_N)$.

B. Architecture

As shown in Figure 1, we propose a simple model architecture composed of four components: an Encoder, a Local Feature Extraction Unit, a Global Feature Extraction Unit, and a Similarity Feature Matrix. Since our core concept is an MLP based on a sampling mechanism, we named the model Rewritten-Sampled MLP (RSMLP).

Encoder We use BERT [13] as our Encoder to extract contextual information between utterances. First, we concatenate the context utterance c with the incomplete utterance x, forming a joint token sequence of length L=M+N. This sequence is then fed into BERT to produce a word embedding matrix $A \in \mathbb{R}^{L \times D}$.

Local Feature Extraction Unit To capture both local and global information, we divide the core of our framework

into two components: the Local Feature Extraction Unit and the Global Feature Extraction Unit. The design of the Local Feature Extraction Unit is primarily inspired by the concept of down-sampling. Since our sentence rewriting task involves editing and replacing parts of the original sentence—similar to classification tasks—down-sampling effectively balances the model's ability to recognize different categories, thereby enhancing its rewriting capabilities. To address the issue of potential information loss caused by naïve down-sampling methods, we introduce continuous sampling, which helps the model capture local information more accurately.

Our sampling method converts each multi-turn dialogue sequence of length L into multiple non-overlapping subsequences of length B, resulting in $\frac{L}{B}$ subsequences. The word matrix can then be decomposed as:

$$A = \{A_1, A_2, \dots, A_{\frac{L}{R}}\}$$
 (1)

where each $A_i \in \mathbb{R}^{B \times D}$, $i \in [1, \frac{L}{B}]$ is a submatrix, and $A \in \mathbb{R}^{B \times \frac{L}{B} \times D}$. This segmentation divides the sequence into continuous short fragments, allowing the model to focus more on local semantic information. Additionally, this design improves training and inference efficiency, as the model only processes a small portion of the sequence at a time.

Next, we feed the subsequences into the Local Mix Block, which is the basic building block designed for local information exchange, primarily consisting of MLPs. The Local Mix Block includes two fully connected layers and a non-linear activation function, all applied to 2D matrices. The purpose of the Local Mix Block is to facilitate information exchange along two different dimensions, producing another feature matrix. The resulting matrix can be viewed as the feature extraction of the input matrix. The information exchange along the embedding dimension is referred to as input projection:

$$Z = W_2(\sigma(W_1 A + B_1))^T + B_2 \tag{2}$$

where $Z=\{Z_1,Z_2,\ldots,Z_{\frac{L}{B}}\}\in\mathbb{R}^{B\times\frac{L}{B}\times S}$ is the output matrix, and W_1,B_1,W_2,B_2 represent the weights and biases

of the first and second fully connected layers in the MLP, respectively. The function $\sigma(\cdot)$ denotes the non-linear activation function, with GeLU [14] being used in our method. It should be noted that S is smaller than D, creating a bottleneck structure.

Global Feature Extraction Unit — After refining the local information, we designed the Global Feature Extraction Unit to extract global information. We concatenate the matrices $Z_{i\in [1,\frac{L}{B}]}$ to obtain $Z\in \mathbb{R}^{L\times S}$ Next, we feed Z into the Global Mix Block, which facilitates the exchange of global information across two dimensions:

$$Z^* = W_4(\sigma(W_3Z + B_3))^T + B_4 \tag{3}$$

Where $Z^* \in \mathbb{R}^{L \times D}$ is the output matrix, and W_3, B_3, W_4, B_4 represent the weights and biases of the first and second fully connected layers in the MLP, respectively. The function $\sigma(\cdot)$ denotes the non-linear activation function. Additionally, we apply padding and replication mechanisms [15] [16] to ensure consistency in sequence length.

Similarity Feature Matrix To further capture the correlation between words, we used several similarity functions to encode the relationships. Specifically, for each word embedding \mathbf{E}_{x_N} in the incomplete utterance and each word embedding \mathbf{E}_{c_M} in the context utterance, we modeled their relationship using three similarity matrices: dot product similarity, cosine similarity, and bilinear similarity. These matrices together form the Similarity Feature Matrix $\mathbf{S}(c_M, x_N)$, as shown below:

$$\mathbf{S}(c_M, x_N) = [\mathbf{E}_{x_N} \cdot \mathbf{E}_{c_M}; cos(\mathbf{E}_{x_N}, \mathbf{E}_{c_M}); \tag{4}$$

$$bilinear(\mathbf{E}_{x_N}, \mathbf{E}_{c_M})]$$
 (5)

These similarity feature matrices model the correlation between words from different perspectives.

Finally, we apply BatchNorm [17] to reduce the dimensionality of matrix **S**. At this stage, each feature vector is mapped to one of three label types: *Substitute*, *Insert*, or *None*, resulting in the generation of a Token-level Edit Matrix.

C. Incomplete Utterance Edit

Since the existing dataset only contains rewritten sentences, we need a process to automatically derive the Token-level Edit Matrix and use these examples for training. We first identify the Longest Common Subsequence (LCS) between the incomplete and rewritten utterances. Then, we enumerate the alignment among the incomplete utterance, rewritten utterance, and the LCS. For words in the rewritten sentence that are not in the LCS, they are labeled as [ADD]. Conversely, for words in the incomplete sentence but not in the LCS, they are labeled as [DEL]. Consecutive words with the same label are merged into a span. By comparing spans, any [DEL] span in the rewritten sentence that corresponds to an [ADD] span in the same context is marked as *Substitute*. Otherwise, the added span is considered an *Insert*, while words that do not undergo any changes are labeled as *None*.

IV. EXPERIMENTS

A. Experimental Setup

Datasets We conducted experiments on IUR benchmarks from three different domains and languages: Restoration - 200k [7], REWRITE [5], and CANARD [18]. The datasets were split into training, evaluation, and testing sets with the following proportions: 80%/10%/10% for Restoration - 200k, 90%/10%/- for REWRITE, and 80%/10%/10% for CANARD. These datasets consist of multi-turn dialogue contexts, incomplete sentences to be rewritten, and examples of correct rewrites.

Baselines We compared the performance of RSMLP with the following methods: transformer-based pointer generator (T-Ptr-Gen) [19], Seq2Seq model L-Gen [20], the hybrid pointer generator (L-Ptr-Gen) [19], L-Ptr- λ /T-Ptr- λ [5], PAC [7], CSRL [21], SARG [22], RAST [12] and RUN (BERT) [9]. For details on the benchmarks, please refer to the respective papers.

Evaluation Following previous practices, we used BLEU [23], ROUGE [24], Exact Match (EM), and Restoration Score [7] as automatic evaluation metrics to compare our proposed method with other approaches.

Model Setting We used bert-base-chinese from the HuggingFace community [25] as our pre-trained BERT model and fine-tuned it as part of the training process. The model has 12 layers and 12 attention heads. We optimized the model using Adam [26] with a learning rate of 1e-5 and computed the loss using weighted cross-entropy.

B. Main Results

Tables II and III present the experimental results on the Restoration-200K and Rewrite datasets, respectively. For the Restoration dataset, our proposed RSMLP model outperforms the previously best-performing model, RUN (BERT), on nearly all metrics. In particular, the metrics \mathcal{P}_1 , \mathcal{P}_2 , and \mathcal{P}_3 show an average improvement of 3.5 points, while \mathcal{F}_1 , \mathcal{F}_2 , and \mathcal{F}_3 also exhibit significant gains. This demonstrates the effectiveness of our architecture, as RSMLP successfully captures both local and global information, thereby enhancing the rewriting capability. Furthermore, although the differences are small, RSMLP also surpasses previous models in terms of BLEU and ROUGE scores, supporting the robustness of our model.

For the Rewrite dataset, RSMLP similarly achieves better performance across nearly all metrics. Notably, our method improves the EM score by 1.5 points. This indicates that the rewritten sentences generated by our model perfectly match the reference sentences, showcasing its deep understanding of contextual semantics.

C. Inference Speed

Table IV presents the inference speed results for a single sentence. All models were run on a single NVIDIA 3070 Laptop, implemented using PyTorch. We can observe that compared to the state-of-the-art (SOTA) methods, our RSMLP model achieves the fastest inference speed. Specifically, it

TABLE II
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	${f R}_2$
$\overline{\text{T-Ptr-}\lambda}$	-	-	51.0	-	-	40.4	-	-	33.3	90.3	87.4	90.1	83.0
L-Gen	65.5	40.8	50.3	52.2	32.6	40.1	43.6	27.0	33.4	84.9	81.7	88.8	80.3
L-Ptr-Gen	66.6	40.4	50.3	54.0	33.1	41.1	45.9	28.1	34.9	84.7	81.7	89.0	80.9
PAC	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
SARG	-	-	62.4	-	-	52.5	-	-	46.3	92.2	89.6	92.1	86.0
RAST	-	-	-	-	-	-	-	-	-	90.4	89.6	91.2	84.3
RUN (BERT)	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
RSMLP (Ours)	76.4	64.0	69.6	62.9	53.1	57.3	54.5	46.1	49.7	93.3	90.2	92.5	86.1

*Note: All results are taken from the original papers. Dashes: results are not reported in the responding literature.

TABLE III
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
L-Gen	47.3	81.2	73.6	80.9	86.3
L-Ptr-Gen	50.5	82.9	75.4	83.8	87.8
L-Ptr- λ	42.3	82.9	73.8	81.1	84.1
T-Ptr- λ	52.6	85.6	78.1	85.0	89.0
T-Ptr-Gen	53.1	84.4	77.6	85.0	89.1
RUN (BERT)	66.4	91.4	86.2	90.4	93.5
RSMLP (Ours)	67.9	91.5	86.5	90.7	93.4

*Note: **EM** indicates the exact match score and \mathbf{R}_L is ROUGE score based on the LCS.

TABLE IV
THE INFERENCE SPEED COMPARISON BETWEEN REMOD AND BASELINES
ON CANARD.

Model	Speedup
T-Ptr-Gen (n_Beam=1)	1×
T-Gen (n_Beam=1)	$2\times$
L-Gen (n_Beam=1)	$4\times$
L-Ptr-Gen (n_Beam=1)	$4\times$
SARG (n_Beam=1)	$18 \times$
RUN (BERT)	$18 \times$
RSMLP (Ours)	21 ×

*Note: n_Beam refers to the beam size used in beam search, which is not applicable to the RUN and ReMod models.

is 21 times faster than T-Ptr-Gen (n_Beam=1). Furthermore, compared to the second-fastest model, RUN (BERT), our approach also demonstrates superior speed. This highlights that our lightweight MLP architecture significantly enhances sentence inference speed while maintaining high rewriting quality.

D. Ablation Study

To validate the effectiveness of the Local Feature Extraction Unit (LU) and Global Feature Extraction Unit (GU), we conducted a comprehensive ablation study, as presented in Table V.

As expected, the absence of both feature extraction units resulted in a decrease across all metrics, demonstrating that our framework significantly enhances model performance. Moreover, using only one of the units also yielded suboptimal results due to the lack of understanding of certain aspects of

TABLE V
THE ABLATION STUDY ON REWRITE DATASET.

Model	\mathbf{EM}	\mathcal{F}_2	\mathcal{P}_2	\mathbf{B}_2	\mathbf{R}_2
RSMLP	67.9	82.6	86.5	91.5	90.7
w/o GU	66.7	81.5	85.2	90.8	90.5
w/o LU	66.4	81.1	85.3	90.6	90.2
w/o both	65.0	80.1	82.3	90.3	89.6

*Note: 'w/o' stands for 'without'.

TABLE VI THE REAL-WORLD EXPERIMENT

Mode	I ROM	Inference Speed	l Accuracy
RSML	P 368MB	70ms	96%

*Note: 'ROM' stands for 'Read-Only Memory', and 'MB' stands for 'MegaByte'.

the information. This further confirms that RSMLP achieves outstanding performance only when both local and global semantic information are simultaneously utilized.

E. Real-World Experiment

We integrated RSMLP into the vehicle-based Retrieval Augmented Generation (RAG) pipeline for real-world experimentation. This system is designed to quickly address issues users encounter while operating their vehicles. In this real-world scenario, referential and elliptical phenomena in multiturn dialogues lead to difficulties in retrieving the correct documents on the RAG recall side. Our experiments aim to address this issue. As shown in Table VI, after integrating RSMLP, the RAG model achieved a recall accuracy of 96% in multi-turn scenarios, with an inference speed of only 70 milliseconds per sentence. Moreover, our model's ROM usage is only slightly larger than that of BERT, making it highly suitable for edge devices with limited computational power and memory, and demonstrating excellent applicability in such environments.

V. Conclusions

In this paper, we propose a simple and efficient model for the IUR task, which utilizes an MLP architecture based on a down-sampling strategy. Our model achieves advanced performance and inference speed on public IUR datasets. Future work will involve exploring the extension of this framework to other conversational domains.

REFERENCES

- [1] S. Zhang, "Personalizing dialogue agents: I have a dog, do you have pets too," arXiv preprint arXiv:1801.07243, 2018.
- [2] H. Zhou, C. Zheng, K. Huang, M. Huang, and X. Zhu, "Kdconv: A chinese multi-domain dialogue dataset towards multi-turn knowledgedriven conversation," arXiv preprint arXiv:2004.04100, 2020.
- [3] K. Xu, H. Wu, L. Song, H. Zhang, L. Song, and D. Yu, "Conversational semantic role labeling," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 2465–2475, 2021.
- [4] D. Yu, K. Sun, C. Cardie, and D. Yu, "Dialogue-based relation extraction," arXiv preprint arXiv:2004.08056, 2020.
- [5] H. Su, X. Shen, R. Zhang, F. Sun, P. Hu, C. Niu, and J. Zhou, "Improving multi-turn dialogue modelling with utterance rewriter," arXiv preprint arXiv:1906.07004, 2019.
- [6] V. Kumar and S. Joshi, "Non-sentential question resolution using sequence to sequence learning," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 2016, pp. 2022–2031.
- [7] 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.
- [8] K. Zhou, K. Zhang, Y. Wu, S. Liu, and J. Yu, "Unsupervised context rewriting for open domain conversation," arXiv preprint arXiv:1910.08282, 2019.
- [9] Q. Liu, B. Chen, J.-G. Lou, B. Zhou, and D. Zhang, "Incomplete utterance rewriting as semantic segmentation," arXiv preprint arXiv:2009.13166, 2020.
- [10] Y. Zhang, Z. Li, J. Wang, N. Cheng, and J. Xiao, "Self-attention for incomplete utterance rewriting," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 8047–8051.
- [11] K. Xu, H. Tan, L. Song, H. Wu, H. Zhang, L. Song, and D. Yu, "Semantic role labeling guided multi-turn dialogue rewriter," arXiv preprint arXiv:2010.01417, 2020.
- [12] J. Hao, L. Song, L. Wang, K. Xu, Z. Tu, and D. Yu, "Rast: Domain-robust dialogue rewriting as sequence tagging," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021, pp. 4913–4924.
- [13] J. D. M.-W. C. Kenton and L. K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings* of naacl-HLT, vol. 1, 2019, p. 2.
- [14] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," arXiv preprint arXiv:1606.08415, 2016.
- [15] X. Zeng, D. Zeng, S. He, K. Liu, and J. Zhao, "Extracting relational facts by an end-to-end neural model with copy mechanism," in *Proceedings* of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2018, pp. 506–514.
- [16] J. Gu, Z. Lu, H. Li, and V. O. Li, "Incorporating copying mechanism in sequence-to-sequence learning," arXiv preprint arXiv:1603.06393, 2016.
- [17] S. Ioffe, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," arXiv preprint arXiv:1502.03167, 2015.
- [18] A. Elgohary, D. Peskov, and J. Boyd-Graber, "Can you unpack that? learning to rewrite questions-in-context," Can You Unpack That? Learning to Rewrite Questions-in-Context, 2019.
- [19] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," *arXiv preprint arXiv:1704.04368*, 2017
- [20] D. Bahdanau, "Neural machine translation by jointly learning to align and translate," arXiv preprint arXiv:1409.0473, 2014.
- [21] K. Xu, H. Tan, L. Song, H. Wu, H. Zhang, L. Song, and D. Yu, "Semantic role labeling guided multi-turn dialogue rewriter," arXiv preprint arXiv:2010.01417, 2020.
- [22] 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.
- [23] 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.

- [24] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text summarization branches out*, 2004, pp. 74–81.
- [25] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz et al., "Transformers: Stateof-the-art natural language processing," in *Proceedings of the 2020* conference on empirical methods in natural language processing: system demonstrations, 2020, pp. 38–45.
- [26] P. K. Diederik, "Adam: A method for stochastic optimization," (No Title), 2014.