Revision Notes: EAAI-22-3985R1

TCAMixer: A lightweight Mixer Based on a Novel Triple Concepts Attention Mechanism for NLP

Dear Editors and Reviewers:

Thanks very much for taking the time to review this manuscript. We appreciate all your comments and suggestions. We provide a detailed response to all of the previous reviews, as well as an updated manuscript highlighting in BLUE.

Thanks again.

1. RESPONSE TO REVIEWER #1

Comment 1: Regarding the grouped convolution, the line of work in modular neural networks similarly divides the network into modules that are computed on different devices. It is based on Top-k routing for reducing the computation cost. It would be better to also discuss the proposed method's relation with studies of modularity, such as [1][2].

[1] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, and et al. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In ICLR, 2017.

[2] Sara Sabour, Nicholas Frosst, Geoffrey E. Hinton. Dynamic Routing Between Capsules. NIPS, 2017.

Response: We appreciate the comments of the reviewers and carefully read the recommended papers [1-2]. The literature [1] uses sparse gate selection expert models to improve model capacity without increasing the amount of computation. In Literature [2], multiple vectors are obtained through capsules to represent more information about features, such as position, direction, size, etc. In order to obtain such a feature representation, the author uses the grouping idea in Capsules, and each group of convolutions obtains an information representation of the feature. From the perspective of research motivation, our research goal is to lighten the model, the literature [1] is how to better select expert models and how to include more information in feature representation in literature [2]. So we regret to indicate that these papers will not be compared with group convolution in the paper. However, we believe that the method proposed in the above paper has high research value, so we have cited the above literature in the Introduction at paragraph 1.

Modification: Large pre-trained models, such as Text Classification (Beltagy et al., 2020; Shazeer et al., 2017), Image Classification (Sabour et al., 2017), Image Fusion (Tang et al.,2022), against adversarial attacks (Shaukat et al., 2020), Recommendation System (Javed et al., 2021), and Paraphrase Identification. However, an inadequacy of these pre-trained models is obvious that both the application and deployment are facing huge challenges because of the high memory footprint and computing costs. In order to facilitate the deployment of artificial intelligence models at edges, how to build lightweight models is one of the main research hotspots at present and in the future.

Comment 2: I think a concise summarization and assessment of the state-of-the-art can be added in the Abstract. This statement would help the readers understand the research gap the paper is concerned with.

Response: Thanks for pointing out the problem. Based on your suggestion, we have revised the Abstracts and the modification is as follows:

Modification: Large scale model size and expensive computing costs cause the large pre-trained models facing challenges on application and deployment. Hence, this paper designs a novel Triple Concepts attention mechanism and a lightweight TCAMixer for edges to classify texts. To go further, the TCAMixer abstracts textual concepts in human way, which is thoughtless of other counterparts, such as pNLP-Mixer (a projection-based MLP-Mixer model for NLP) and HyperMixer (a token mixing MLP dynamically using hypernetwork). Experimental results on several pubic data sets show that the TCAMixer outperforming the pNLP-Mixer and HyperMixer by a great gap, e.g. 3% accuracy, with lower model size (0.177M). Additionally, the TCAMixer achieved over 85% to 98.7% performance of large pre-trained models, but only own 1/3000 to 1/2000 size of those models on most of test data sets.

Comment 3: Since the method described in this work aims to reduce the computational cost by leveraging its sparsity of parameters, it seems reasonable to me that more complicated tasks can be added to this work.

Response: We thank the reviewers for this suggestion, which shows that the reviewers fully recognize our work. We have replied to similar suggestions in the R1 version, but the answer is not very adequate. When processing more complex tasks, most SOTA models use pre-training techniques. In order to compare the proposed method with these models more fairly, we also have set out to improve our model using pre-training techniques. However, due to financial constraints, it is not possible to purchase more hardware devices, so it will take some time for the results of our pre-trained model on more complex tasks to be obtained. We will write another paper on TCAMixer models using pre-training techniques and improved techniques to compare with other SOTA models, please follow our related research or visit our github address: https://github.com/Liu-Xiaoyan97/

2. RESPONSE TO REVIEWER #2

Response: We are very grateful to the reviewers for their recognition of our work, and we will conduct more in-depth research in the future.

3. RESPONSE TO REVIEWER #3

Response: Thanks for the reviewers for their valuable comments to help us improve the quality of our manuscripts.

4. RESPONSE TO REVIEWER #4

Comment 1: Do not find the summary or gap from the related work? Just cite reference by reference?

Response: Thanks to the reviewer's comment, it was very helpful for improving our paper. We have added Table 1 at the end of the Related Work to summarize the advantages and disadvantages of each literature to improve readability.

Modification:

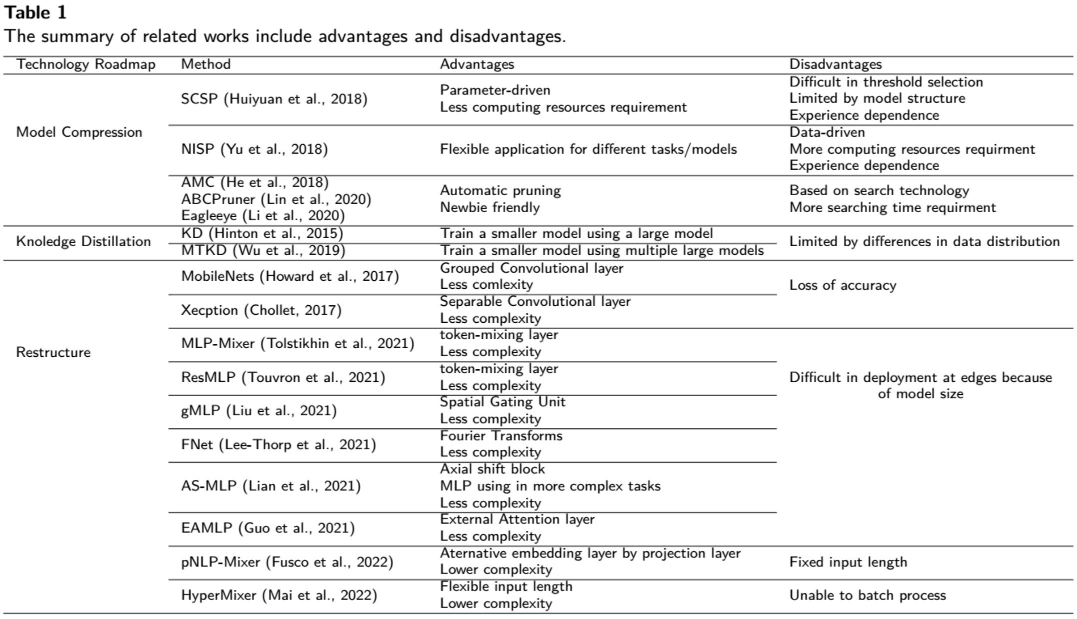

Table 1
The summary of related works include advantages and disadvantages.

Technology Roadmap	Method	Advantages	Disadvantages
Model Compression	SCSP (Huiyuan et al., 2018)	Parameter-driven Less computing resources requirement	Difficult in threshold selection Limited by model structure Experience dependence
	NISP (Yu et al., 2018)	Flexible application for different tasks/models	Data-driven More computing resources requirment Experience dependence
	AMC (He et al., 2018) ABCPruner (Lin et al., 2020) Eagleeye (Li et al., 2020)	Automatic pruning Newbie friendly	Based on search technology More searching time requirment
Knoledge Distillation	KD (Hinton et al., 2015)	Train a smaller model using a large model	Limited by differences in data distribution
	MTKD (Wu et al., 2019)	Train a smaller model using multiple large models	
Restructure	MobileNets (Howard et al., 2017)	Grouped Convolutional layer Less comlexity	Loss of accuracy
	Xecption (Chollet, 2017)	Separable Convolutional layer Less complexity	
	MLP-Mixer (Tolstikhin et al., 2021)	token-mixing layer Less complexity	Difficult in deployment at edges because of model size
	ResMLP (Touvron et al., 2021)	token-mixing layer Less complexity	
	gMLP (Liu et al., 2021)	Spatial Gating Unit Less complexity	
	FNet (Lee-Thorp et al., 2021)	Fourier Transforms Less complexity	
	AS-MLP (Lian et al., 2021)	Axial shift block MLP using in more complex tasks Less complexity	
	EAMLP (Guo et al., 2021)	External Attention layer Less complexity	
	pNLP-Mixer (Fusco et al., 2022)	Aternative embedding layer by projection layer Lower complexity	Fixed input length
	HyperMixer (Mai et al., 2022)	Flexible input length Lower complexity	Unable to batch process

Comment 2: Please complete the real revision work on the English revision.

Response: Thanks for advices. We proofread our manuscript for the spelling, coefficients, functional, notations and grammar issues in this revision, and update the figures accordingly

Thanks for all the valuable reviews which help us improve the quality of our manuscript and give us more insights into the research.

Best regards,

Xiaoyan Liu, Huanling Tang, Jie Zhao, Quansheng Dou, Mingyu Lu

2023.1.20