BaSFormer: A Balanced Sparsity Regularized Attention Network for Transformer

SUBMISSION DATE / POSTED DATE

15-05-2023 / 19-05-2023

CITATION

Jiang, Shuoran; Chen, Qingcai; Xiang, Yang; Pan, Youcheng (2023): BaSFormer: A Balanced Sparsity Regularized Attention Network for Transformer. TechRxiv. Preprint.
https://doi.org/10.36227/techrxiv.22824908.v1

DOI

10.36227/techrxiv.22824908.v1

BaSFormer: A Balanced Sparsity Regularized Attention Network for Transformer

Shuoran Jiang, Qingcai Chen, Yang Xiang, Youcheng Pan, Xiangping Wu

Abstract—Attention networks often make decisions relying solely on a few pieces of tokens, even if those reliances are not truly indicative of the underlying meaning or intention of the full context. That can lead to over-fitting in Transformers and hinder their ability to generalize. Attention regularization and sparsity-based methods have been used to overcome this issue. However, these methods cannot guarantee that all tokens have sufficient receptive fields for global information inference. Thus, the impact of individual biases cannot be effectively reduced. As a result, the generalization of these approaches improved slightly from the training data to new data. To address these limitations, we proposed a balanced sparsity (BaS) regularized attention network on top of the Transformers, called BaSFormer. BaS regularization introduces the K-regular graph constraint on self-attention connections, which replaces SoftMax with SparseMax in the attention transformation. In BaS-regularized self-attentions, SparseMax assigns zero attention scores to low-scoring connections, highlighting influential and meaningful contexts. The K-regular graph constraint ensures that all tokens have an equal-sized receptive field to aggregate information, which facilitates the involvement of global tokens in the feature update of each layer and reduces the impact of individual biases. As no continuous loss can be used as the K-regular graph regularization, we proposed an exponential extremum loss with augmented Lagrangian. Experimental results show that BaSFormer improves debiasing effectiveness compared to the newest large language models, such as the chatGPT, GPT-4 and LLaMA. In addition, BaSFormer achieves new state-of-the-art results in text generation tasks. Interestingly, this paper also evaluates that BaSFormer can learn hierarchically linguistic dependencies in gradient attributions, which improves interpretability and adversarial robustness. Our implementation is anonymously available at Google Drive.

Index Terms—over-fitting, Transformers, attention regularization, receptive field, balanced sparsity, generalization

1 Introduction

Transformer is one of the breakthroughs in natural language processing (NLP). It has emerged as the dominant architecture for pre-trained large language models (LLMs) [1], such as the Generative Pre-trained Transformer (GPT) models [2], T5 [3] and LLaMA [4]. In Transformer-based language models, the self-attention network is a critical component [5]–[7] due to its powerful capability to model the temporal sequence of tokens. It contributes to impressive results in NLP [8] [9], computer vision [10], multimodal information processing [11], and other areas. The self-attention network aims to learn the alignment between every pair of tokens in a sequence, and update the token features from the aligned positions [9]. However, Transformers often over-fit on a few pieces of the sequence to make decisions instead of considering the full context. This is due to the lack of regularization on the attention connections [12]. For example, a model may learn to associate certain words or phrases with a particular class or polarity, even if those associations do not truly indicate the underlying meaning or intention of the text [13]. Such individual biases can lead to biased or inaccurate predictions on unseen data, which is also one of the most critical challenges for the generalization of Transformers [14].

Numerous sparse attention methods have been proposed to overcome the above issue [15]. The local-band sparsity is one of the early research, which considers a small window around the keys in attention networks. As the local segments usually have grammatical relationships, the local-band sparsity can alleviate the over-fitting problem by focusing more on local associations [16]. However, the localness limits the ability to model long-term and non-consecutive dependencies. Subsequently, non-localized sparsity mechanisms were proposed, including DropAttention [17], Cluster-Former [18] and L_1 sparsity-regularized attention (L1-attention) [19], [20]. DropAttention randomly sets attention weights to zero, interpreted as dropping a set of neurons along different dimensions. DropAttention encourages the model to utilize the full context of the input sequences rather than relying solely on a small piece of features [17]. The Cluster-Former marries sliding-window and hashing-based methods to achieve effective local and long-range dependency encoding via two encoding layers [18]. The first type (Sliding-Window Layer) focuses on extracting local information within a sliding window. The second type (Cluster-Former Layer) encodes global information beyond the initial chunked sequences [18]. The L_1 sparsity-regularized attention introduced a L_1 sparse prior, which minimizes contributions of the irrelevant connections in the feature learning process [19]. These methods encourage the model to make decisions relying on the full context of the input sequences rather than a few pieces of input.

- *Qingcai Chen and Yang Xiang are Corresponding authors.*
- *Shuoran Jiang, Qingcai Chen and Xiangping Wu are with the College of Computer Science and Technology, Harbin Institute of Technology, China, 518055.*
  E-mail: shuoran.chiang@gmail.com, qingcai.chen@hit.edu.cn, wxpleduole@gmail.com
- *Yang Xiang and Youcheng Pan are with Peng Cheng Laboratory, China, 518055.*
  E-mail: xiangy@pcl.ac.cn, panyoucheng4@gmail.com

Manuscript received April 19, 2005; revised August 26, 2015.

Another line of research includes regularization and pruning methods, which aim to alleviate the over-fitting problem from the view of head-level performance. Attention head pruning tries to prune the redundant heads of multi-head attention (MHA). Based on the actual phenomena in BERT, it was proposed that multiple heads in the same layer exhibit similar behaviors in attention distributions [21]. However, attention pruning could affect the model's capacity to exploit the plentiful language dependencies, and impact the test accuracy [22]. The diversity regularization among multiple attention heads is an effective resolution [23], enlarging the distributions of different heads in the same layer and encouraging all attention connections to be considered. Disagreement regularization [24] implements diversity regularization by maximizing the cosine distances between the input sub-spaces and output representations, and dispersing the positions attended by multiple heads with element-wise multiplication of attention matrices. Experimental results demonstrated that a small Transformer network with disagreement regularization achieved comparable performance with a big one without it. Meanwhile, the training speed was nearly twice faster. Constrained attention networks (CAN) [25] introduce an orthogonal sparsity, in which the attention weights concentrate on different parts of the sentence with less overlap. Collaborative multi-head attention [26] enables all heads in the same layer to learn shared key/query projections, which helps heads to extract meaningful shared query/key features.

However, while non-localized sparsity in attention connections effectively mitigates over-fitting on a few pieces in the sequence, it cannot guarantee that every token has a sufficiently sizeable receptive field to learn representations. Attention regularization forces all heads to learn the diversified attention distributions, enlarging the averaged receptive fields over all tokens [27]. However, it still cannot prevent individual bias on a few pieces. According to these analyses, a good attention regularization strategy should reconcile the receptive fields over all tokens and the sparsity. This paper proposed a balanced sparsity (BaS) regularization for the self-attention network, which encourages attention networks to learn sufficient and large enough receptive fields for every token in a single head. In the BaS regularized attention, SparseMax replaced SoftMax transformation to align the attention scores. To implement the BaS regularization in Transformer networks (BaSFormer), we defined a continuous loss function via exponential extremum with augmented Lagrangian. Finally, the experimental results on text classification and neural machine translation (NMT) tasks proved that BaSFormer has better debiasing ability and generalizability than the newest language models. Furthermore, our experimental results also showed that BaSFormer learned more grammatical connections in both attention scores and gradient attributions.

The main contributions are listed below.

- In order to improve the debiasing ability and generalizability in the Transformer-based language model, we proposed the balanced sparsity (BaS) regularized attention networks. The BaS regularized attentions can learn the non-local and balanced sparsity in a single attention head. Moreover, the accumulated receptive fields can also cover the full context, effectively prompting the robustness of dataset biases.
- We enforced the BaS regularized attention network into the Transformer framework (BaSFormer), and defined BaS regularization as a continuous loss function.
- Experimental results show that BaSFormer effectively improved the debiasing ability and generalizability. Moreover, more exciting results show that BaSFormer demonstrated better interpretability, adversarial robustness and fairness for social biases.

2 Background

2.1 Fully-connected Self-Attention Network

As one of the most pivotal components in Transformers, the self-attention network captures semantic dependencies from the input sequence with a parallel mode. The typical self-attention architecture uses the scaled dot-product attention to learn the intra-interactions within a sequence [6], [17]. Given the input sequence $\mathbf{X} \in \mathbb{R}^{L\times d}$ with length L and feature dimension d, the self-attention projects it into query $\mathbf{Q}$, key $\mathbf{K}$ and value $\mathbf{V}$. And the scale-product attention computes the output representation. This process is described as the following equation,

$$\begin{aligned} \mathbf{Q},\mathbf{K},\mathbf{V} &= \mathbf{X}\mathbf{W}_q, \mathbf{X}\mathbf{W}_k, \mathbf{X}\mathbf{W}_v \\ \mathrm{Attn}\left(\mathbf{X}\right) &= \underbrace{\mathrm{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}}\right)}_{\mathbf{A}}\mathbf{V} \end{aligned} \tag{1}$$

where $\mathbf{W}_q \in \mathbb{R}^{d\times d_k}$, $\mathbf{W}_k \in \mathbb{R}^{d_k\times d}$, $\mathbf{W}_v \in \mathbb{R}^{d\times d_v}$ represent the transformation weights for query, key and value respectively, and $\mathbf{A} \in \mathbb{R}^{L\times L}$ represents the attention matrix.

Every token pair $\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle$, $1 \le i,j \le L$, in the fully-connected self-attention is interdependent. The feature updater on each token $\boldsymbol{x}_i$ depends on all tokens in the same sequence $p\left(\widehat{\boldsymbol{x}}_i | \boldsymbol{x}_1, \cdots, \boldsymbol{x}_L\right)$. This feature updater on fully-connected dependencies easily captures the spurious correlations in the training data, and it is the reason why the attention gradient attributed to adversarial patterns can fool Transformers-based PLMs easily [28].

2.2 SparseMax Transformation

The two core components of the attention mechanism are the alignment model and the transformation function [6].

$$\underbrace{\mathbf{A} = \mathrm{Soft\,max}\underbrace{\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}}\right)}_{\text{Alignment}}}_{\text{Transformation}} \tag{2}$$

The former $\boldsymbol{Q}\boldsymbol{K}^T/\sqrt{d}$ is used to compute attention matrix, and the later Softmax() is performed row-wise and transforms the attention matrix into probabilities. Usually, the well-known Softmax transformation [29] returns positive values and dense output probabilities [30]. The dense alignments assign non-zero weights to all positions in the input sequence in attention networks [31], which cannot distinguish the meaningful semantic connections.

Algorithm 1: Sparsemax Evaluation

Input: $\boldsymbol{z}$
Sort $\boldsymbol{z}$ as $z_{(1)} \geq \cdots \geq z_{(K)}$
Find $k(\boldsymbol{z}) \max\{k \in [K] | 1 + kz_{(k)} > \sum_{j \leq k} z_{(j)}\}$
Define $\tau(\boldsymbol{z}) = \frac{(\sum_{j \leq k(\boldsymbol{z})} \boldsymbol{z}_{(j)}) - 1}{k(\boldsymbol{z})}$
Output: $\boldsymbol{p}$ s.t. $p_i = [z_i - \tau(\boldsymbol{z})]_+$.

Peters B et al. [30] and Gong H et al. [31] proposed the adaptive sparsity on multi-heads attention architectures with the SparseMax transformation, which tends to yield zero for the low-scoring in the vector,

$$\text{SparseMax}\,(\boldsymbol{z}) \underset{\boldsymbol{p} \in \Delta^{K-1}}{\arg\min} \|\boldsymbol{p} - \boldsymbol{z}\|^2 \tag{3}$$

where $\Delta^{K-1}\{\boldsymbol{p} \in \mathbb{R}^K | \mathbf{1}^T \boldsymbol{p} = 1, \boldsymbol{p} \geq \mathbf{0}\}$ represents a $(K-1)$-dimensional simplex, $\boldsymbol{z}$ is the input vector and $\boldsymbol{p}$ is the output vector. The SparseMax projects the input vector $\boldsymbol{z}$ onto the probability simplex, where the simplex boundary is to be hit and the output vector SparseMax $(\boldsymbol{z})$ becomes sparse. A closed-form solution [32] for this simplex is described in **Algorithm 1**. In it, the $\tau(\boldsymbol{z})$ is a threshold function. All coordinates above this threshold will be shifted by this amount and the others will be truncated to zero [32].

The experimental results evaluated that the SparseMax transformation can yield exactly zero probabilities for irrelevant paired positions, and it improves generalizability and interpretability [33].

3 Methodology

This section described how to implement the balanced sparsity (BaS) regularization on attention networks. First, the concept of the BaS regularization was described in detail, and a continuous loss function on it was defined via the exponential extremum with augmented Lagrangian (EXPEAR). Next, the Softmax transformation was replaced with SparseMax in the BaS-regularized attention networks, and they were enforced into the Transformer (BaSFormer). Apart from the proposed BaS regularized attention as shown in Fig. 1, the model schema of BaSFormer is similar with standard Transformer proposed in [6].

3.1 BaS-Regularized Attention Network

In algebraic graph theory, if the in-degree and out-degree of any node in the graph are equal, this graph is called a balanced graph [34]. In this study, we used a special case - K-regular graph in graph theory to define the balanced regularization on attention connections.

Definition 1. *A regular directed graph must satisfy the restrictive condition that the in-degree and out-degree of each vertex are equal to each other. A regular graph with vertices of degree K is called a K-regular graph. [35]*

As the balanced regularization has an restrictive constraint on both the in-degree and out-degree, which means that all tokens in the attention network have the sane size of receptive fields. Moreover, as shown in Figure 1, for fear of the over-fitting on a few pieces and insufficient dependency

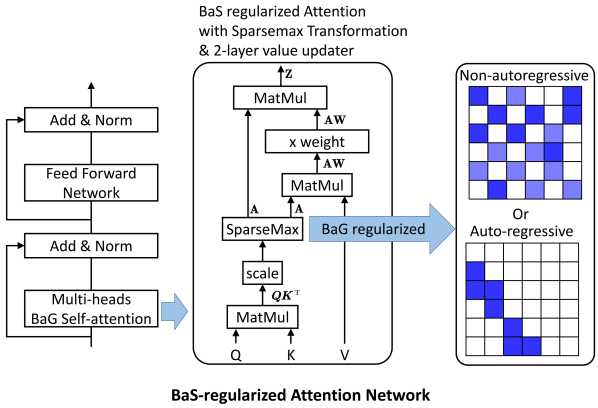


Fig. 1. The schema about the balanced sparsity in the proposed BaS regularized attention networks.

connections, the receptive fields of all tokens must be low-overlapped. The balanced regularization couples with the SparseMax transformation (BaSFormer) is naturally suitable for defining the non-local and balanced sparsity in attention networks, which ensures the information extracted across all layers of the Transformer rely on the full context.

The BaS regularization is achieved when the following restrictions can be satisfied.

$$\begin{aligned} &\|\mathcal{E}(\boldsymbol{A}) - (N - k + ke^{1/k})\| \leqslant \boldsymbol{\varepsilon} \\ &\text{where} \quad \mathcal{E}(\boldsymbol{A}) = \sum_{i=1}^{L} e^{A_i}, \quad \boldsymbol{\varepsilon} \in \mathbb{R}^L, \quad \boldsymbol{\varepsilon} \geqslant 0 \\ &\qquad \boldsymbol{A} = \mathbf{A}_{i,\cdot} \quad \text{or} \quad \boldsymbol{A} = \mathbf{A}_{\cdot,j} \\ &\qquad \sum_{i=1}^{N} A_{i,\cdot} = 1, \quad \text{and} \quad i, j = 1, \cdots, L \end{aligned} \tag{4}$$

where k is the the number of degree in the BaS regularized graph, L represents the sequence length, and the slack variable $\boldsymbol{\varepsilon} \in \mathbb{R}^L$ is used to tolerate the self-loops when the self-attention $\mathbf{A} \in \mathbb{R}^{L \times L}$ is defined in the non-autoregressive encoding networks.

The constraint Eq.(4) can be defined as a Lagrange dual problem as follows,

$$\begin{aligned} &\min : \sum_{i=1}^{N} \varepsilon_i \\ &s.t. \quad \varepsilon_i - \|\boldsymbol{\varepsilon}(\boldsymbol{A}_i) - (N - 2 + 2e^p)\|^2 \geqslant 0 \\ &\qquad \varepsilon_i \geqslant 0 \end{aligned} \tag{5}$$

This Lagrangian duality on k-regular graph can be defined as a continuous optimization $h(\mathbf{A})$ via **exp**onential **e**xtremum with **a**ugmented Lag**r**angian (**EXPEAR**) as follows,

$$\begin{aligned} h(\mathbf{A}) = &\sum_i |\varepsilon_i^{'} + \varepsilon_i^{''} - 2\varepsilon_{max}| \\ &+ \sum_i |2\delta_1 - \varepsilon(\mathbf{A}_{i,\cdot}) - \varepsilon(\mathbf{A}_{\cdot,i}) + 2\varepsilon_i^{'}| \\ &+ \sum_i |\varepsilon(\mathbf{A}_{i,\cdot}) + \varepsilon(\mathbf{A}_{\cdot,i}) - 2\delta_2 + 2\varepsilon_i^{''}| \\ &\to 0 \end{aligned} \tag{6}$$

The augmented Lagrangian method is used to enforce the BaS regularization into Transformer with the task objective together:

$$\begin{aligned} \min_{\mathbf{A}\in\mathbb{R}^{L\times L}} & \ell\left(\mathbf{X}\right)+\frac{\rho}{2}\left|h\left(\mathbf{A}\right)\right|^{2} \\ \text{s.t.} \quad & h\left(\mathbf{A}\right)\leq 0 \end{aligned} \tag{7}$$

where $\ell\left(\mathbf{X}\right)$ is the loss function in downstream tasks, and the final optimization is defined with Lagrange multiplier α and ρ.

$$\mathcal{L}\left(\boldsymbol{\theta}\right)=\ell\left(\mathbf{X}\right)+\frac{\rho}{2}\left|h\left(\mathbf{A}\right)\right|^{2}+\alpha h\left(\mathbf{A}\right) \tag{8}$$

where $\boldsymbol{\theta}$ is all trainable parameters in the Transformer.

3.2 BaSFormer Framework

Similar to the standard Transformer framework, all encoder and decoder blocks in BaSFormer are composed of the multi-heads of BaS regularized self-attention networks, cross-attention networks, normalization (Norm) layers and position-wise feed-forward (FFN) layer. There is also another difference that our self-attention network adapts the SparseMax transformation with a bilayer value updater. The attention networks in the encoding blocks have self-loops and the auto-regressive decoders have not.

3.2.1 BaS Regularized Self-Attention

Given the input sequence $\mathbf{X}\in\mathbb{R}^{L_x\times d}$, the feature updater in the BaS-regularized self-attention is defined as follows,

$$\begin{aligned} \mathbf{Q},\mathbf{K},\mathbf{V} &= \mathbf{X}\mathbf{W}_q,\mathbf{X}\mathbf{W}_k,\mathbf{X}\mathbf{W}_v^{(0)} \\ \mathbf{A} &= \text{SparseMax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}}\right) \\ \mathbf{Z} &= \mathbf{A}\sigma\left(\mathbf{A}\mathbf{V}\right)\mathbf{W}_v^{(1)} \end{aligned} \tag{9}$$

where $\mathbf{W}_q\in\mathbb{R}^{d\times d_k}$, $\mathbf{W}_k\in\mathbb{R}^{d_k\times d}$, $\mathbf{W}_v^{(0)}\in\mathbb{R}^{d\times d_v}$ and $\mathbf{W}_v^{(1)}\in\mathbb{R}^{d_v\times d_v}$ represent the learnable transforming weights in query, key value and the second value updater layer respectively, and $\sigma()$ represents the non-linear activation.

As shown in Fig. 1, we replaced the single-layer value updater [6] with a bilayer value updater. As the natural language sentences have both sequential and hierarchical structures to understand them [36], [37], the value updater in attention networks needs multiple layers to constrcut the complex connections. The bilayer value updater transfers more information throughout attention connections in single-head attention.

3.2.2 Encoder Layer

BaSFormer has the similar encoder layer as the one defined in standard Transformer, which is named as $\text{EncoderLayer}\left(\mathbf{X}\right)=\mathbf{Z}_{enc}$ and defined as follows,

$$\begin{aligned} \mathbf{H}_{enc} &= \text{LayerNorm}\left(\text{MHWA}\left(\text{PE}\left(\mathbf{X}\right)+\mathbf{X}\right)+\mathbf{X}\right) \\ \mathbf{Z}_{enc} &= \text{LayerNorm}\left(\text{FFN}\left(\mathbf{H}_{enc}\right)+\mathbf{H}_{enc}\right) \end{aligned} \tag{10}$$

where MHWA() represents the multi-heads of BaS-regularized self-attention networks as defined in Eq. (9), PE() represents the absolute position encodings. For each position index t in the sequence $\mathbf{X}$, the encoding is a vector $\boldsymbol{p}_t=\text{PE}\left(t\right)\in\mathbb{R}^d$.

$$\text{PE}\left(t\right)_i=\begin{cases}\sin\left(w_i t\right) & \text{if i is even,}\\ \cos\left(w_i t\right) & \text{if i is odd,}\end{cases} \tag{11}$$

where t is the position index and i is the dimension.

The FFN() is the position-wise feed-forward layers, which operates on each position independently. This function consists of two linear transformations with a ReLU activation in between.

$$\text{FFN}\left(\mathbf{X}\right)=\text{ReLU}\left(\mathbf{X}\mathbf{W}_1+\mathbf{b}_1\right)\mathbf{W}_2+\mathbf{b}_2 \tag{12}$$

where $\mathbf{W}_1,\mathbf{W}_2\in\mathbb{R}^{d\times d}$ and $\mathbf{b}_1,\mathbf{d}_2\in\mathbb{R}^{L_x\times d}$.

In addition, the LayerNorm() represents a normalization operation on the residual connection between attention output and input, and FFN output and input respectively.

3.2.3 Decoder Layer

BaSFormer uses the auto-regressive generation in the decoder. Giving the target sequence $\mathbf{Y}\in\mathbb{R}^{L_y\times d}$, the basic decoding block is named as $\text{DecoderLayer}\left(\mathbf{Y}\right)=\mathbf{Z}_{dec}$ defined as follows,

$$\begin{aligned} \mathbf{H}_{dec} &= \text{LayerNorm}\left(\text{MMHWA}\left(\text{PE}\left(\mathbf{Y}\right)+\mathbf{Y}\right)+\mathbf{Y}\right) \\ \mathbf{H}_{cro} &= \text{LayerNorm}\left(\text{MHCA}\left(\mathbf{Z}_{enc},\mathbf{H}_{dec}\right)+\mathbf{H}_{dec}\right) \\ \mathbf{Z}_{dec} &= \text{LayerNorm}\left(\text{FFN}\left(\mathbf{H}_{cro}\right)+\mathbf{Z}_{cro}\right) \end{aligned} \tag{13}$$

where $\mathbf{H}_{dec}\in\mathbb{R}^{L_y\times d}$, $\mathbf{H}_{cro}\in\mathbb{R}^{L_y\times d}$ and $\mathbf{Z}_{dec}\in\mathbb{R}^{L\times d}$. The MMHWA() is the masked multi-heads of BaS-regularized self-attention network, in which the upper triangular of attention matrix is set as zero. And the MHCA() is the multi-head cross-attention network defined as follows,

$$\begin{aligned} \mathbf{Q}^{cro},\mathbf{K}^{cro},\mathbf{V}^{cro} &= \mathbf{H}_{dec}\mathbf{W}_q^{cro},\mathbf{Z}_{enc}\mathbf{W}_k^{cro},\mathbf{H}_{dec}\mathbf{W}_v^{cro} \\ \mathbf{A}^{cro} &= \text{Softmax}\left(\frac{\mathbf{Q}^{cro}\mathbf{K}^{croT}}{\sqrt{d}}\right) \\ \mathbf{Z}_{cro} &= \mathbf{A}^{cro}\mathbf{V}^{cro} \end{aligned} \tag{14}$$

where $\mathbf{W}_q^{cro},\mathbf{W}_k^{cro},\mathbf{W}_v^{cro}\in\mathbb{R}^{d\times d}$, $\mathbf{A}^{cro}\in\mathbb{R}^{L_x\times L_y}$ and $\mathbf{Z}_{cro}\in\mathbb{R}^{L_y\times d}$.

3.2.4 Prediction Layer

Word-Level Classification BaSFormer performs word-level predictions $\mathbf{Y}\in\mathbb{R}^{L_x\times c}$ on input sequence $\mathbf{X}\in\mathbb{R}^{L_x\times d}$ just from $\mathbf{Z}_{enc}$ output from encoding block, where c represents the class number. Its scheme is shown in the Fig. (a).

$$\widehat{\mathbf{Y}}=f_\sigma\left(\mathbf{Z}_{enc}\mathbf{W}_\sigma+\mathbf{b}_\sigma\right) \tag{15}$$

where the $f_\sigma()$ is a non-linear function, $\mathbf{Z}_{enc}\in\mathbb{R}^{L_x\times d}$ $\widehat{\mathbf{Y}}\in\mathbb{R}^{L_X\times c}$, $\mathbf{W}_\sigma\in\mathbb{R}^{d\times c}$, $\mathbf{b}_\sigma\in\mathbb{R}^c$, and L_x is the sequence length and c is the class number.

Sentence Classification: BaSFormer sends the representation on the first token `[CLS]` to the classifier to perform sentence-level classification.

$$\widehat{\boldsymbol{Y}}=f_\sigma\left(\boldsymbol{z}_{[CLS]}\mathbf{W}_\sigma+b_\sigma\right) \tag{16}$$

where $\widehat{\boldsymbol{y}} \in \mathbb{R}^c$, $\boldsymbol{z}_{[CLS]} \in \mathbb{R}^d$, $\mathbf{W}_\sigma \in \mathbb{R}^{d\times c}$, c represents the class number.

Sequence-to-Sequence Generation: BaSFormer uses the auto-regressive decoding model for the sequence-to-sequence generation,

$$\widehat{\mathbf{Y}} = \text{Softmax}\left(\mathbf{Z}_{dec}\mathbf{W}_\sigma + \mathbf{b}_\sigma\right) \tag{17}$$

where $\mathbf{Z}_{dec} \in \mathbb{R}^{L_y\times d}$, $\mathbf{W}_\sigma \in \mathbb{R}^{d\times|\mathcal{V}|}$, $\mathbf{b}_\sigma \in \mathbb{R}^{L_y\times|\mathcal{V}|}$, in which $\mathcal{V}$ represents the word vocabulary in the training corpus and $|\mathcal{V}|$ is the vocabulary size.

3.2.5 Loss Function

The final loss is defined with the BaS regularization together as follows,

$$\begin{aligned}\ell\left(\mathbf{Y}\right) =& -\sum_{l=1}^{L_y} y_l \log\left(\widehat{y}_l\right)\\ \mathcal{L}\left(\boldsymbol{\theta}\right) =& \ell\left(\mathbf{Y}\right) + \frac{\rho}{2}\left(\sum_{n=1}^{N}\left|h\left(\mathbf{A}_{\text{enc}}^{(n)}\right)\right|^2 + \sum_{n=1}^{N}\left|h\left(\mathbf{A}_{\text{dec}}^{(n)}\right)\right|^2\right)\\ &+ \alpha\left(\sum_{n=1}^{N} h\left(\mathbf{A}_{\text{enc}}^{(n)}\right) + \sum_{n=1}^{N} h\left(\mathbf{A}_{\text{dec}}^{(n)}\right)\right)\end{aligned} \tag{18}$$

where $\ell\left(\mathbf{Y}\right)$ is the loss function on downstream tasks, N represents the layer numbers in encoder and decoder, *enc* and *dec* represent the encoding and decoding blocks respectively.

4 Experiment

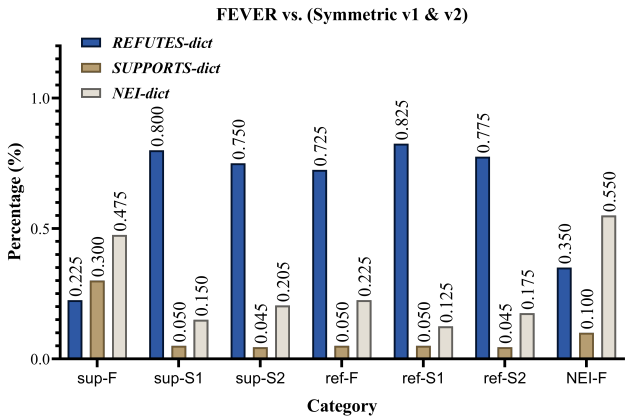
4.1 Exmerimental Setting

We empirically evaluated BaSFormer from three aspects: (i) the effectiveness of generalizability, (ii) the interpretability, (iii) the adversarial robustness and (iv) fairness for social biases.

4.1.1 Setting for Generalizability Evaluation

Datasets:

- The Fact Extraction and VERification (FEVER) dataset [38] verifies whether a claim *supports* or *refutes* or *is not enough information for* the given evidence. In the training set of FEVER, there is a dataset bias that a few bi-grams in claims highly co-occur with labels. To evaluate the robustness for this kind of bias, Schuster et al. [39] introduced out-of-distribution (OOD) evaluations - FEVER-Symmetric V1 and v2 with 717 and 712 examples, respectively. The statistics about the co-occurrence between exclusive bi-grams with labels in FEVER, and are shown in Fig. 2.
- The paraphrase identification (PI) task identifies alternative linguistic expressions of the same meaning at different textual levels (document, paragraph, sentence, word, or combination). QQP [40] dataset is a popular dataset to train PI models, and it contains 795,267 pairs of questions annotated as "paraphrase" or "non-paraphrase". However, QQP has a bias in training data that the high lexical overlaps between paired sentences mainly occur in the "paraphrase" examples. PAWS (Paraphrase Adversaries from Word Scrambling) dataset [41] balances the dataset bias on "paraphrase" and "non-paraphrase" examples. The statistics about the lexical overlaps in different classes in QQP and PAWS are shown in Fig. 3.



REFUTES-dict: {did not, yet to, does not, refused to, failed to, only ever, incapable being, to be, unable to, not have}

SUPPORTS-dict: {united states, least one, at least, person who, stars actor, won award, american actor, starred movie, from united, from america}

NEI-dict: {worked in, s name, award winning, wyatt earp, finished college, and it, will ferrell, can be, and he, tim rice}

Fig. 2. The statistics about the biased correlations from category to specific bi-grams. We abbreviated the "supports" in FEVER, $\mathcal{S}_1$ and $\mathcal{S}_2$ to "sup-F", "sup-S1" and "sup-S2" respectively, and "refutes" in three datasets to "ref-F", "ref-S1" and "ref-S2", and "not enough information (NEI)" in FEVER to "NEI-F".

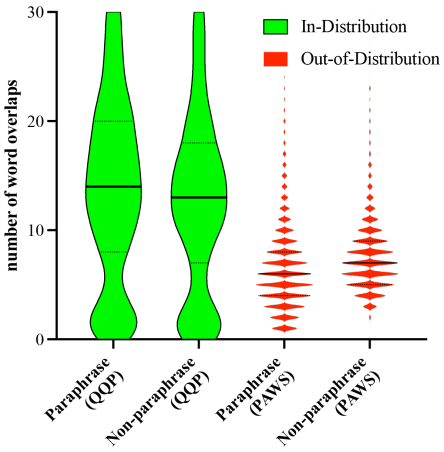


Fig. 3. The statistics about the lexical overlaps in QQP with its OOD dataset - QQP, where the vertical axis presents the number of word overlaps between sentence pair in an example, and the horizontal axis presents the different labels.

Compared Baselines

- **Transformer variants with Sparse Attentions: Longformer** [42] combines local windowed attention with task-motivated global attention. **BigBird** [43] uses a sparse attention mechanism applied token by token,

which is different from BERT as the attention mechanism is applied to the entire input just once.

- **Debiasing Language Models:** The experiments compared various debiasing models to evaluate their effectiveness on dataset biases. **BERT-base**+$\mathcal{F}_{\texttt{BoW}}$ and **BERT-base**+$\mathcal{F}_{\texttt{BiLSTM}}$ [44] fine-tuned BERT-base on the forgettable examples in BoW and BiLSTM models, which could improve the OOD generalization. **ReWeighting** [45] used the bias-only model to capture the inference strategy on the training data and trained a second model in an ensemble, including the pre-trained bias-only model, which learned an alternative strategy to improve the OOD generalization. **Reg-conf** [46] uses the confidence regularization method to improve the OOD generalization. **CrossAug** [47] uses the generated claim-evidence pairs to debias the fact verification models. **Product-of-Experts (PoE)** [45] first trains a naive model to make predictions exclusively based on dataset bias, and then trains a robust model as part to compose an ensemble with the naive one together. **DRiFt** [48] formalizes the concept of dataset bias from the perspective of distribution shift and presents a simple **D**ebiasing algorithm based on **R**es**i**dual **Fi**t**t**ing (DRiFt). **Inverse-Reweight (InvR)** [49] reweights the sampling probabilities by the inverse of the confidences induced by the bias-only model, which aims to balance the in-distribution (ID) and OOD samples. **Learned-Mixin (LMin)** [45] is a variant of PoE. **MoCaD** [50] extends the traditional two-stage ensemble-based debiasing framework to a three-stage one including the bias Modeling, model Calibrating, and Debiasing stages. **Self-debiasing** [51] jointly identifies biased examples and features in an end-to-end manner, which does not require an extra training stage, or manual bias feature engineering. **DePro** [52] is an end-to-end method that can eliminate spurious correlations in a fine-grained. **BERT-Whitening** [53], [54] uses a simple and effective post-processing technique (whitening) to tackle the anisotropic problem of sentence embeddings [55]. **AdaTest** [56] is a process to uses large-scale language models in partnership with human feedback to automatically to write unit tests highlighting bugs in a target model automatically. **Prompting GPT-3** [57] systematically studied the reliability of GPT-3 from key facets of generalizability, fairness, calibration, and factuality with knowledge updating. **Sentence-BERT (SBERT)** [58] is a bi-encoder approach, which encodes sentences separately and generates high-quality embeddings for each. **Debiasing Masks** [59] focus on removing bias from an existing model instead of the now-common approach of training again from scratch.
- **Pre-trained large language models (LLMs):** As the pre-trained Transformers on large-scale corpus can learn more invariant representations, experiments compared the proposed model with (i) the fine-tuning PLMs: **BERT** [60], **RoBERTa** [61], **ALBERT** [62], **GPT2** [63], LLaMA [4] and (ii) the OpenAI API tool [1] for **prompting GPT-3** [57], **prompting ChatGPT** [64] and **prompting GPT-4** [65] test, on a wide range of text prompts and scenarios.

4.1.2 Setting for Machine Translation

Datasets. The WMT 2014 English-to-German (En-De) [66] with $4.5M$ parallel pairs and WMT 2017 English-to-Romanian (En-Ro) [67] with $610K$ parallel pairs were chose in the machine translation task.

Compared Baselines

- **Auto-regressive (AT) Machine Translation models:** standard **Transformer** proposed in original paper [6]. **Levenshtein Transformer (LevT)** [68] is a partially auto-regressive model that combines the "insertion" and "deletion" operations to devise more flexible and amenable sequence generation.
- **Non-autoregressive (NAT) Machine Translation models: Glancing Transformer (GLAT)** [69] uses the single-pass parallel generation model to generate high-quality translation with $8\times \sim 15\times$ speedup. **Mask-Predict** [70] uses the language masking mechanism in training that predicts any subset of the target words conditioned on both the input text and a partially masked target translation. **JM-NAT** [71] trains Transformer by masking inputs with an n-gram loss function, which alleviates the problem of translating duplicate words. The imitation learning framework for non-autoregressive machine translation (**imit-NAT**) [72] introduces a knowledgeable auto-regressive machine translation (AT) demonstrator to supervise each decoding state of the NAT model. **FlowSeq** [73] designs several layers of generative flow tailored for modeling the conditional density of sequential latent variables. **NAT-DCRF** [74] incorporates an efficient approximation of positional contexts into the non-autoregressive models, which improves the decoding consistency and reduces the inference cost. **Imputer** [75] makes conditional independence assumptions within a generation step to achieve parallel generation and models conditional dependencies across generation steps.
- **Transformers with Attention Regularization:** Constrained Sparsemax transformation in attention network (**CSparseMax**) [76] replaces the traditional Softmax by SparseMax transformation and sets the upper bound for the amount of attention a word received. **Disagreement regularization** [24] maximizes cosine distances between the input sub-spaces and output representations. Cross-model Back-translated Distillation (**CBD**) [77] induces a novel component to the standard unsupervised machine translation framework, which aims to induce another level of data diversification. Cross-lingual language models (**XLMs**) [78] define an unsupervised generative pre-training method, which only relies on monolingual data to leverage parallel data with a new cross-lingual language model objective. Simultaneous Neural Machine Translation (**SNMT**) [79] is a

1. https://platform.openai.com/docs/introduction

generic framework to integrate linguistic and extra-linguistic information into simultaneous models. MAsked Sequence to Sequence pre-training (**MASS**) [80] adopts the encoder-decoder framework to reconstruct a sentence fragment given the remaining part of the sentence to predict this masked fragment. **Transformer_Rep** [81] uses the perturbation methods in the training process for Transformer.

- **Pre-trained Large Language Models (LLMs): M2M100** is a state-of-the-art multilingual neural machine translation model using an encoder-decoder architecture with a Transformer-based neural network [82]. **DeltaLM** is a pre-trained multilingual encoder-decoder model that regards the decoder as the task layer of off-the-shelf pre-trained encoders [83].

4.1.3 Setting for Interpretability Analysis

Analysis tool. The multi-head self-attention mechanism learns word-to-word dependencies within the input sequence and encodes contextual information across all layers. Hao et al. [28] proposed the self-attention attribution tree (AttAttr) tool, which effectively explains how each token interacts with each other across attention layers to reach the final prediction. AttAttr first identifies the most important attention connections in each layer. This step finds that attention weights only sometimes correlate well with their contributions to the model prediction. Next, AttAttr introduces a heuristic algorithm to construct self-attention attribution trees based on the backward gradients, which discovers the information flow inside Transformer. In addition, a quantitative analysis is applied to justify how much the edges of an attribution tree contribute to the final prediction.

Datasets. This experiment was conducted on the natural language inference (NLI) task with the MNLI [84] dataset and text classification task with SST-2 [85] dataset. MNLI is a benchmark containing 431,992 sentence pairs annotated with textual entailment information (neutral, entailment, or contradiction), and SST-2 is a standard dataset for predicting sentiment from longer movie reviews.

4.1.4 Setting for Adversarial Robustness

Adversarial Attack Construction: AttAttr [28] can also be used to construct the adversarial triggers to attack language models. These adversarial triggers are extracted from the interaction patterns contributing most to the model decision. During the attack, these adversarial triggers are inserted into the test input at the same relative position and segment as in the original sentence.

Datasets. The adversarial attack was also conducted on the MNLI and SST-2 datasets.

4.2 Effectiveness of Generalization Performance

4.2.1 *Text Classification*

This experiment evaluated the effectiveness of the proposed BaSFormer on debiasing and generalization. All comparison models were trained on ID datasets - FEVER and QQP separately, and tested on evaluations from in-distribution (ID) and out-of-distribution (OOD) datasets.

TABLE 1
The test accuracies on FEVER, QQP and their OOD evaluations.

Models	Fact Verification			PI	
	ID	OOD		ID	OOD
	FEVER	$\mathcal{S}$-v1	$\mathcal{S}$-v2	QQP	PAWS
PLM-based debiasing models					
BERT-base [44]	86.1	57.7	64.7	90.9	36.1
BERT $+\mathcal{F}_{\text{BoW}}$ [44]	87.1	61.0	67.0	89.0	48.8
BERT $+\mathcal{F}_{\text{BiLSTM}}$ [44]	86.5	61.7	66.6	88.0	47.6
ReWeighting [45]	85.5	61.7	66.5	83.5	69.2
Reg-conf [46]	86.4	60.5	66.2	88.3	55.4
CrossAug [47]	85.3	61.7	66.5	-	-
PoE [45]	87.1	65.9	69.1	89.2	55.2
DRiFt [45]	87.4	65.7	69.0	87.8	65.2
InvR [49]	84.3	60.8	65.2	-	-
LMin [45]	84.7	59.8	65.3	-	-
MoCaD [50]	87.1	65.9	69.1	-	-
Self-debiasing [46]	87.9	66.1	-	-	-
DePro [52]	84.5	65.2	69.2	-	-
AdaTest [56]	-	-	-	**91.9**	53.8
SBERT [58]	-	-	-	90.7	68.9
Debiasing Masks [59]	85.0	63.4	-	89.6	44.3
BERT-whitening [53]	85.9	65.9	68.3	87.3	58.3
Prompting GPT-3 [57]	-	61.5	64.3	83.5	73.7
LLaMa-7B [57]	**94.1**	51.3	49.2	90.5	46.9
Prompting GPT-3.5	-	**69.7**	**70.6**	-	61.4
Prompting GPT-4	-	60.5	60.7	-	69.4
Transformers trained from scratch					
Longformer [42]	82.1	63.4	65.5	81.6	68.8
BigBird [43]	77.4	43.4	42.1	81.9	66.1
Transformer [44]	87.9	47.8	47.4	83.0	40.6
BaSFormer	92.6	66.2	**69.4**	87.8	**78.1**
w/o BaS $h(\mathbf{A})$	85.2	52.9	51.3	78.5	62.3
w/ single-layer	87.9	47.8	47.4	81.7	47.7
w/o BaS & bilayer	87.5	45.7	42.3	82.5	40.6

Table 1 shows the experimental results of the ID test on FEVER, QQP and corresponding OOD test on Symmetric-V1, Symmetric-V2 and PAWS, respectively. The word embeddings, layer number and head number in BaSFormer were set to the same size as the BERT-base, and the model was trained for 200 epochs with $12,800$ warm-up steps. All compared models were trained on FEVER and QQP and did not undergo any fine-tuning for OOD evaluations - Symmetric-V1, Symmetric-V2 and PAWS. In addition, the dropout was not used for the BaS regularized self-attention networks, and the word embeddings were initialized from the 300-d GloVe [86]. The prompting ChatGPT and GPT-4 used the templates: " Are the following evidence sentence supports, refutes or is not enough information for the claim sentence? Claim: sentence1, Evidence: sentence2. Answer me with "entailment", "neutral" or "contradiction." for the FEVER and Symmetric V1, V2 datasets. And " Are the following two questions equivalent or not? Answer me with "equivalent" or "not equivalent. Sentence1. Sentence2." for the QQP and PAWS datasets.

In Table 1, the fine-tuned LLaMA-7B and AdaTest achieved the best ID results on FEVER and QQP with 94.1% and 91.9%, respectively. Although the proposed BaSFormer did not achieve top-1 ID test accuracy, it ranked second on FEVER and performed comparably with most LLMs. BaSFormer outperformed all other baselines on the OOD test sets for OOD generalization - PAWS, with test

accuracies of 78.1%. Specifically, although BaSFormer did not outperform the Prompting ChatGPT on Symmetric-V1, and Symmetric-V2, it achieved the comparable result of 69.4% with prompting ChatGPT of 70.6% on Symmetric-V1, and surpassed the prompting GPT4 by more than 5.5%, 8.5% and 8.5% on three OOD test sets respectively. These results demonstrated that the performance of OpenAI GPTs deteriorates with increasing pre-training data and training epochs.

In this experiment, BaSFormer was set with the same model architectures and parameter size as BERT-base, and its results beat the BERT-base by more than 6.5%, 8.5%, 4.7%, 42.0% on FEVER, Symmetric-V1, Symmetric-V2 and PAWS. Furthermore, compared with the state-of-the-art (SOTA) debiasing models, BaSFormer beat the old top-1 models - Self-debiasing, DePro, Prompting GPT-3 by at least 0.1%, 0.2% and 4.4% on Symmetric-V1, Symmetric-V2 and PAWS, respectively. Notably, even though BaSFormer did not improve the ID test accuracy on QQP compared with other baselines, the improvement for OOD generalization on PAWS was significant, with a rise of 4.4%.

Table 1 also conducted a series of ablation studies to evaluate the effectiveness of the main technical contributions in the proposed model: BaS regularization $h(\mathbf{A})$, bilayer value updater and their combination. Compared to the undivided BaSFormer, the model without (w/o) BaS, with (w/) a single-layer value updater and w/o both BaS and (&) bilayer settings showed significant decreases by at least 5.1% in ID test accuracy and 13.3% 37.5% in OOD test accuracy. These ablation studies prove that all proposed ideas in this study can significantly improve the robustness of dataset bias in training and generalization.

4.2.2 *Machine Translation*

TABLE 2
The sacreBLEU scores on WMT14 and WMT16.

Models	WMT14		WMT16	
	EN→DE	DE→EN	EN→RO	RO→EN
Transformer [6]	27.30	-	-	-
GLAT [69]	27.48	31.27	33.70	34.05
LevT [68]	27.27	-	-	33.26
Mask-Predict [70]	27.03	30.53	33.08	33.31
JM-NAT [71]	27.31	31.02	-	-
imit-NAT [72]	22.44	25.67	28.61	28.90
Flowseq [73]	23.72	28.39	29.73	30.72
NAT-DCRF [74]	23.44	27.22	-	-
Imputer [75]	25.80	28.40	32.30	31.70
CSparseMax [76]	29.85	-	-	29.77
Disagreement [24]	28.51	-	-	-
CBD [77]	30.10	36.30	36.30	33.80
XLM [78]	27.20	34.30	34.60	32.70
SNMT [79]	29.50	33.90	33.70	32.50
MASS [80]	27.10	35.20	35.10	33.40
Transformer_Rep [81]	32.35	-	-	-
M2M100-418M [87]	33.90	35.60	27.90	34.10
DeltaLM-base	35.41	33.58	**39.82**	31.42
BaSFormer	**36.74**	**37.88**	36.96	**35.77**

Our experiment also evaluated the sequence-to-sequence generation effectiveness for the Encoder-Decoder framework of BaSFormer. The experiment was conducted on the WMT2014 En-De and WMT2016 En-Ro machine translation tasks. The WMT2014 English-German (EN-DE) preprocessed dataset in [6] and WMT2016 English-Romanian (EN-RO) preprocessed in ([88]) were chosen as the benchmarks to compare the proposed BaSFormer with other state-of-the-art baselines. Both datasets were tokenized and segmented into subword units with the BPE encodings [89], and the word embedding was initialized randomly and trained with the model together.

Table 2 shows all results on these tasks, in which the compared baselines include the newest auto-regressive and non-autoregressive machine translation models and the Transformer variants with attention regularization. These results show that BaSFormer outperforms the previous baselines on WMT14 DE-EN, EN-DE and WMT16 EN-RO tasks with 36.74, 37.88 and 35.77 sacreBLEU scores, respectively. The previous state-of-the-art (SOTA) baselines on these four translation directions are Transformer_Rep, CBD, CBD and GLAT, respectively. Compared with these SOTA results, BaSFormer raised the BLEU score from 32.35 to 36.74 on WMT14 EN-DE, 36.30 to 37.88 on WMT14 DE-EN and 34.05 to 35.77 on WMT16 RO-EN respectively. Specifically, compared with the newest Transformation variants enforced with attention regularization (CSparseMax, disagreement regularization, CBD), BaSFormer also achieved better results with the sacreBLEU scores improvements by 6.64, 1.58, 0.46 and 1.97 at least. Noticeably, even though the fine-tuned DeltaLM-base achieved the top-1 sacreBLEU score on WMT16 RO-EN, its performance on WMT16 EN-RO with 31.42 was far below the BaSFormer's result of 35.77.

All experimental results in text classification and machine translation proved that the proposed BaSFormer improves the generalization of both the encoder-only and encoder-decoder frameworks.

4.3 Interpretability Analysis

The second experiment was conducted on NLI and text classification tasks, which aims to analyze the interpretability of the BaS regularized attentions. The datasets used for the two tasks are MNLI and SST-2, respectively, and the BaSFormer was set as the same size as BERT-base. Both BaSFormer and Transformer were trained from scratch with 200 epochs with a learning rate $r = 2e - 5$.

4.3.1 *Hierarchical Information Flows*

To evaluate whether BaS regularized attention could learn more interpretable knowledge than the unconstrained ones. This experiment used the attention attribution (AttAttr) tool to interpret the gradient attributions inside BaSFormer and compared LLMs, respectively. Fig. 4 shows the self-attention attribution trees across all Transformer layers on MNLI and SST-2 examples.

On the MNLI example 1, the gradient attribution tree across BaSFormer layers constructs dependencies within local segments, and the information interactions hierarchically aggregate into tokens of *"know"*, *"what"* and *"how"* scattered over paired sentences. The *"[CLS]"* token collects information flows over these three words to make the prediction *"Entailment"*. Compared with the fine-tuned BERT, GPT2, GPT-neo, T5 and LLaMA, BaSFormer captures more grammatical and consecutive dependencies in the self-attention attribution trees. For example, the dependencies

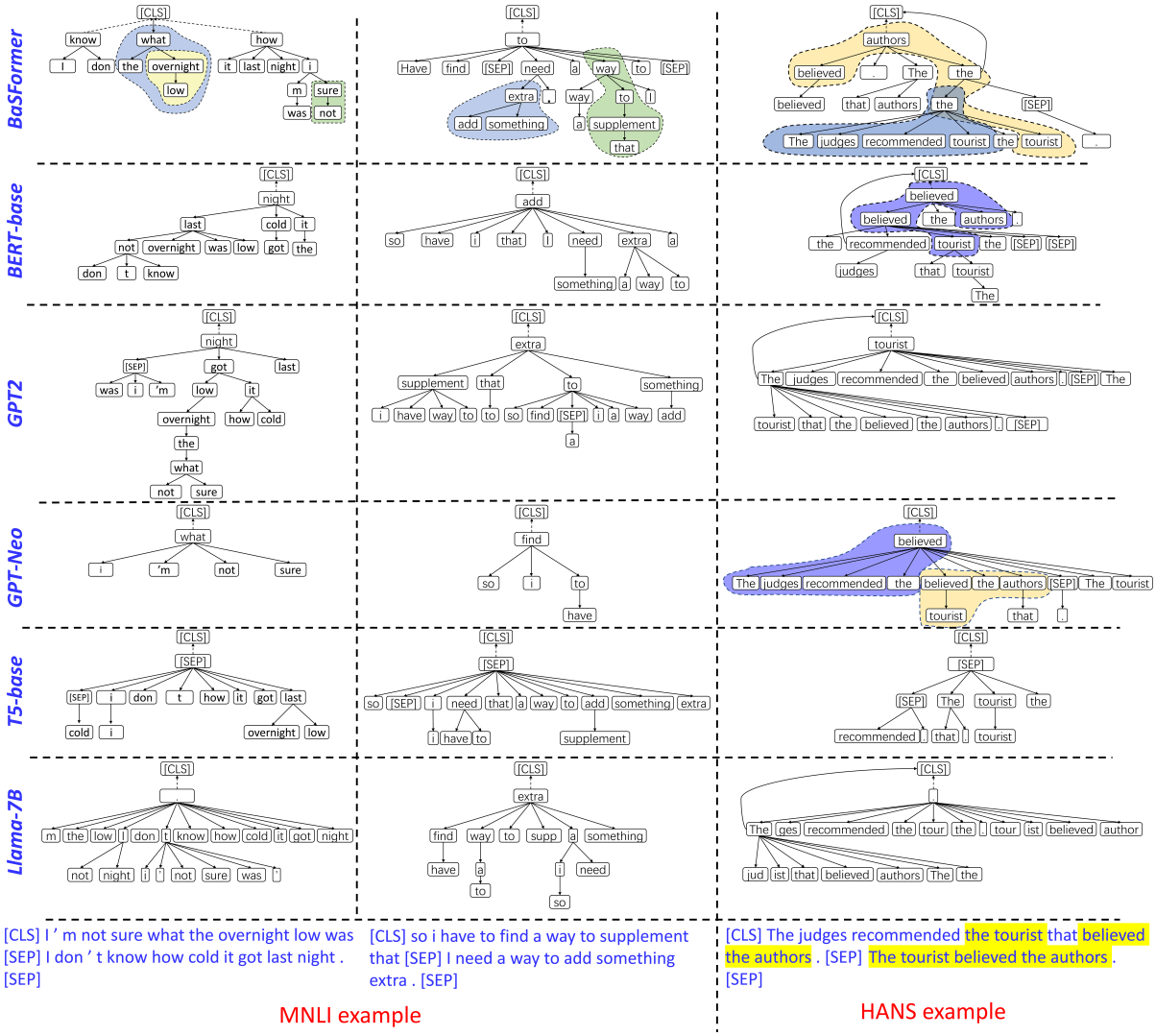


Fig. 4. The case study of attention attribution trees on MNLI with it OOD challenging evaluations - HANS, in which the words with yellow background among paired sentences are overlapped. Where the subtrees labeled with colored background are local and grammatical dependencies.

of { *overnight low* }, { *what the overnight low* } and { *not sure*}. On the MNLI example 2, BaSFormer also captured the grammatical dependencies { *a way to supplement that*} and { *add something extra* } that do not appear in BERT, GPT2, GPT-Neo, T5 and LLaMA. Moreover, the fine-tuned BERT, GPT2, GPT-Neo, T5 and LLaMA learned attention attribution trees have many confused and meaningless dependencies across paired sentences, and the final prediction mainly depends on the single token like *night* and *add* on two MNLI examples, which is the reason for the over-fitting on dataset bias. The superiority of BaSFormer in learning interpretable and meaningful associations contributes to its outperformance on OOD generalization. The HANS example in Fig. 4 shows that BaSFormer learned most grammatical associations. By comparison, aside from BERT and GPT-Neo, which can learn few meaningful connections, other compared LLMs make decisions solely on biased dependencies.

As shown in Fig. 5, the experimental comparison on the SST-2 examples is more significant, where BaSFormer captures more hierarchical grammatical dependencies to construct the phrases and multiple concepts.

4.3.2 *Non-local Receptive Field*

To better demonstrate the effectiveness of non-local dependency modeling in BaSFormer, Fig. 6 plotted the averaged interaction distances across all BaSFormer and BERT layers. These results were counted from $1,000$ and 500 evaluation examples from MNLI and SST-2 respectively. As shown in Fig.6, for the paired sentences in MNLI, both BaSFormer and fine-tuned BERT have averaged interaction distances over 20. Specifically, in the 1st$\sim$3rd, 6th$\sim$12th layers, BaSFormer has broader receptive fields than BERT's corresponding layers. For the single sentence input in SST-2, BaSFormer extracts longer-term dependencies, especially in the 1st and 7th$\sim$11th layers.

Notably, BaSFormer and fine-tuned BERT performed the similar trends on the interaction distances across all layers. Moreover, in the 4th-5th layers on MNLI and the 2nd-6th layer on SST-2, the receptive fields of BaSFormer restricted more local dependencies than BERT. In the deeper layers of two datasets, BaSFormer extracts more long-term dependencies and models richer knowledge than the fine-tuned BERT.

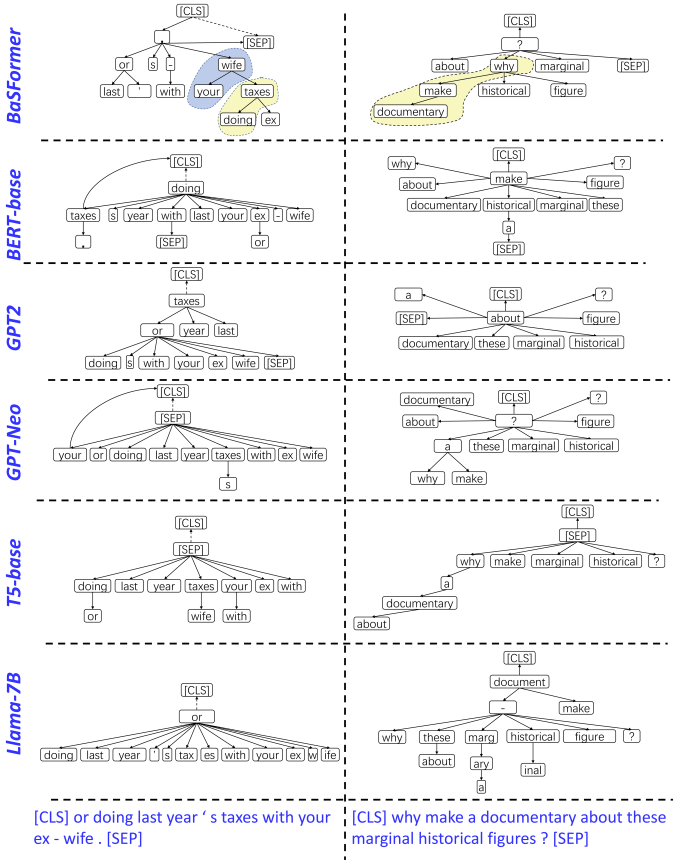


Fig. 5. The case study of attention attribution trees on single sentence classification (SSC) task on SST-2 dataset, where the subtrees labeled with colored background are local and grammatical dependencies.

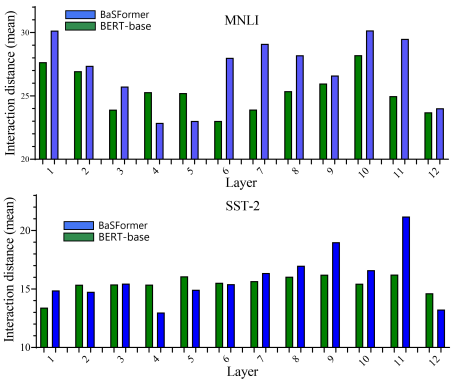


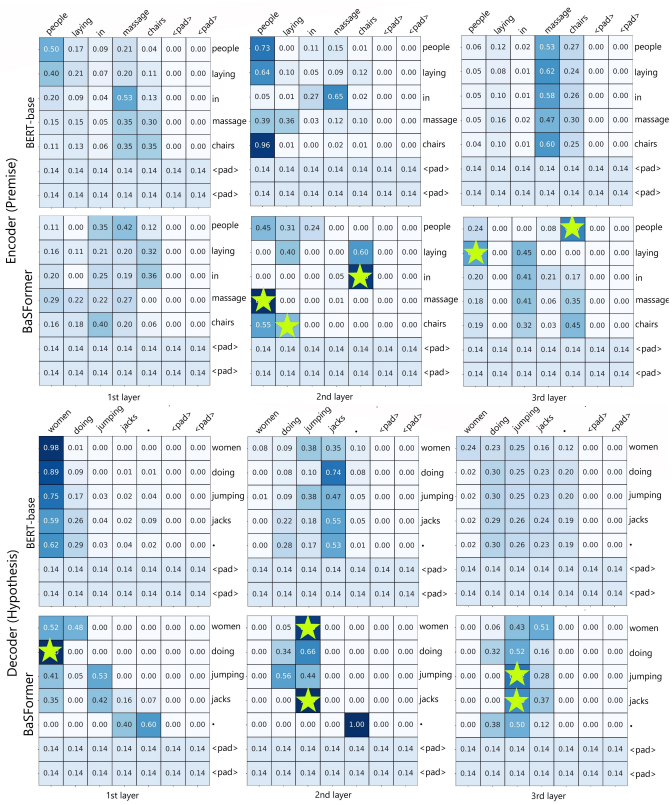
Fig. 6. The means of interaction distances extracted by the ATTATTR built tree in every layer from Transformer and BaSFormer.

4.3.3 *Grammatically Meaningful Attention Connections*

Apart from the attention attribution analysis, this paper also analyzed the attention heatmaps to evaluate whether BaSFormer can model grammatically meaningful dependencies in attention connections. This case study was applied to two MNLI examples, as shown in Fig. 7.

In this experiment, the BaSFormer and Transformer were set as an encoder-decoder framework, where the encoder and decoder learn the representations from "premise" and "hypothesis" sentences, respectively.

Fig. 7 visualized the heatmaps of self-attentions in BaSFormer trained from scratch and fine-tuned BERT-base, and these results plotted the 1st heads in the top three layers. These experimental results show that the BaSFormer learned more long-term dependencies than the BERT, and these dependencies must conform to more grammatical rules than the BERT. Concretely, on the premise - *"People laying in massage chairs."*, BaSFormer gave significant attention scores on connections {*people, massage*}, {*in, chairs*} and {*massage, chairs*}. These frequently-used expressions did not appear in the BERT-base. Moreover, on the hypothesis - *"Women doing jumping jacks."*, BaSFormer also learned the phrases {*women, doing*}, {*jumping, jacks*} with significant attention scores, and the fine-tuned BERT had not.



Premise: People laying in massage chairs.
Hypothesis: Women doing jumping jacks.
Label: Neutral

Fig. 7. The heatmaps of self-attention connections on two MNLI examples from BaSFormer and BERT-base. The exhibited attention scores are selected from the 1st heads in the top three layers, in which the connections labeled with yellow stars have the significant scores with grammatically meaningful dependencies.

The above analysis of "hierarchical information flows", "non-local receptive field" and the "grammatically meaningful attention connections" demonstrate that the BaS regularized attentions can learn sparser connections with more ground-truth knowledge than the BERT. It is also the reason for the outperformance of BaSFormer compared with other SOTA baselines for debiasing effectiveness and generalization.

4.4 Robustness of Attacks

The third experiment conducted adversarial attacks on BaSFormer and BERT-base to evaluate the adversarial robust-

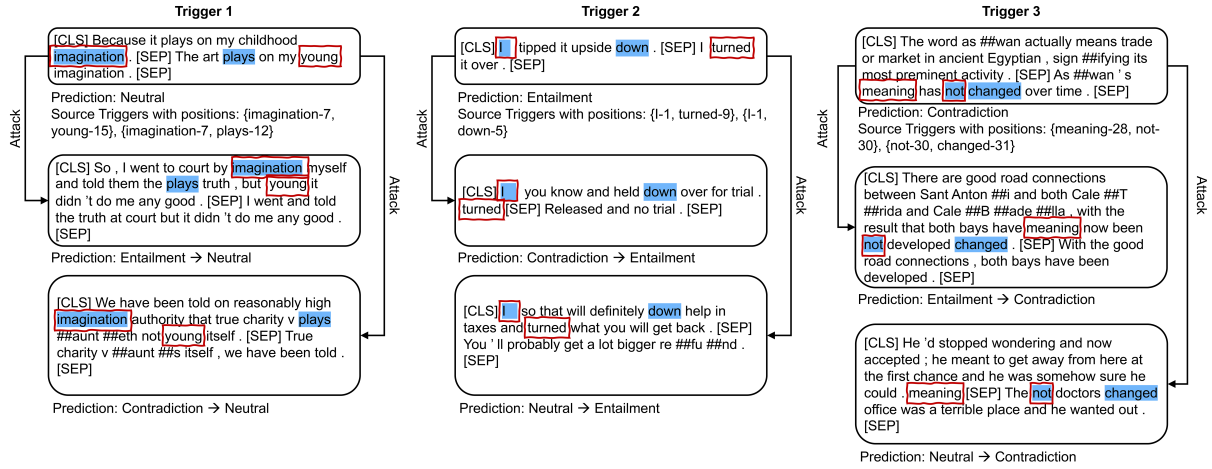


Fig. 8. The top-3 triggers (i.e., highlighted and underlined word patterns) extracted from AttAttr tool on the MNLI, where the adversarial triggers were inserted into test examples at the same relative position and segment as in the original sentence.

ness. This experiment was conducted on MNLI and the adversarial examples were constructed by the adversarial triggers found in AttAttr [28]. Specifically, each adversarial trigger includes two paired pairs extracted from the interaction patterns contributing most to the model decision. During the attack, these adversarial triggers are inserted into test sentences at the same relative position and segment as in the original sentences. This experiment reports the adversarial results on top-1 triggers for three labels, and the details about the source triggers and adversarial patterns are shown in Fig. 8.

TABLE 3
Attack results of the top-1 triggers on dev-matched test data in MNLI.

Label	**No trigger**	**Trigger1**			
	BERT-base	BERT-base		BaSFormer	
	baseline	Acc.	Δ	Acc.	Δ
Contradiction	84.94	63.49	-21.45	65.89	-19.05
Entailment	82.87	29.26	-53.61	39.67	-43.20
Neutral	82.00	77.84	-4.16	82.80	0.80
Avg.	83.27	56.86	-26.41	62.79	-20.48
Label	**No trigger**	**Trigger2**			
	Transformer	Transformer		BaSFormer	
	baseline	Acc.	Δ	Acc.	Δ
Contradiction	84.94	68.19	-16.75	72.30	-12.64
Entailment	82.87	60.22	-22.65	63.00	-19.87
Neutral	82.00	66.95	-15.05	79.86	-2.14
Avg.	83.27	65.12	-18.15	71.72	-11.55
Label	**No trigger**	**Trigger3**			
	Transformer	Transformer		BaSFormer	
	baseline	Acc.	Δ	Acc.	Δ
Contradiction	84.94	68.00	-16.94	73.14	-11.80
Entailment	82.87	33.20	-49.67	43.69	-39.18
Neutral	82.00	72.91	-9.09	78.77	-1.23
Avg.	83.27	58.04	-25.23	65.20	-18.07

Table 3 reported the attacking results on top-1 adversarial triggers for three labels in MNLI. Even though three triggers decreased the test accuracy in both fine-tuned BERT and BaSFormer, the proposed BaSFormer was more robust. Specifically, BaSFormer outperforms the fine-tuned BERT on three adversarial triggers by at least 2.4% for "contradiction" label. This advantage is further expanded on the other two labels with averaged increases of 7.89% and 7.91%, respectively.

5 Conclusion

This study proposed a balanced sparsity (BaS) regularization on attention networks to define the non-local and balanced sparsity with SparseMax transformation. In order to enforce the BaS regularization into Transformer (BaSFormer), this paper defined a continuous loss function via EXPonential extremum with Augmented lagRangian (EXPEAR). Our experimental results demonstrated that the BaS regularization approach significantly improves debiasing and generalization effectiveness, even when compared with state-of-the-art LLMs such as Prompting chatGPT, GPT-4 and LLaMA. Moreover, interpretability and robustness analyses show that the BaS regularized attentions construct hierarchically linguistic dependencies that are closer to how humans understand language. These findings suggest that BaS regularization has the potential to enhance the interpretability, robustness, and generalization of Transformer networks in various NLP tasks.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant No. 62006061, 61872113, 62106115), Stable Support Program for Higher Education Institutions of Shenzhen (No.GXWD20201230155427003-20200824155011001), Strategic Emerging Industry Development Special Funds of Shenzhen (No. XMHT20190108009), and Fundamental Research Fund of Shenzhen (No. JCYJ20190806112210067).

References

[1] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, *et al.*, "A survey of large language models," *arXiv preprint arXiv:2303.18223*, 2023.

[2] M. R. Chavez, T. S. Butler, P. Rekawek, H. Heo, and W. L. Kinzler, "Chat generative pre-trained transformer: Why we should embrace this technology," *American Journal of Obstetrics and Gynecology*, 2023.

[3] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *The Journal of Machine Learning Research*, 2020.

[4] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[5] G. Novakovsky, N. Dexter, M. W. Libbrecht, W. W. Wasserman, and S. Mostafavi, "Obtaining genetics insights from deep learning via explainable artificial intelligence," *Nature Reviews Genetics*, 2022.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Proc. of NeurIPS*, 2017.

[7] L. Yunxiang, L. Zihan, Z. Kai, D. Ruilong, and Z. You, "Chatdoctor: A medical chat model fine-tuned on llama model using medical domain knowledge," *arXiv preprint arXiv:2303.14070*, 2023.

[8] T. Lin, Y. Wang, X. Liu, and X. Qiu, "A survey of transformers," *AI Open*, 2022.

[9] H. Wang, X. Shen, M. Tu, Y. Zhuang, and Z. Liu, "Improved transformer with multi-head dense collaboration," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2022.

[10] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu, *et al.*, "A survey on visual transformer," *arXiv preprint arXiv:2012.12556*, 2020.

[11] S. Yao and X. Wan, "Multimodal transformer for multimodal machine translation," in *Proc. of ACL*, 2020.

[12] G. Wang, R. Ying, J. Huang, and J. Leskovec, "Improving graph attention networks with large margin-based constraints," *arXiv preprint arXiv:1910.11945*, 2019.

[13] M. McKinnon and C. O'Connell, "Perceptions of stereotypes applied to women who publicly communicate their stem work," *Humanities and social sciences communications*, 2020.

[14] Z. Wang and A. Culotta, "Identifying spurious correlations for robust text classification," in *Proc. of EMNLP Findings*, 2020.

[15] T. Liu, C. Wang, C. Chen, M. Gao, and A. Zhou, "Understanding long programming languages with structure-aware sparse attention," in *Proc. of SIGIR*, 2022.

[16] B. Yang, Z. Tu, D. F. Wong, F. Meng, L. S. Chao, and T. Zhang, "Modeling localness for self-attention networks," in *Proc. of EMNLP*, 2018.

[17] L. Zehui, P. Liu, L. Huang, J. Chen, X. Qiu, and X. Huang, "Dropattention: A regularization method for fully-connected self-attention networks," *arXiv preprint arXiv:1907.11065*, 2019.

[18] S. Wang, L. Zhou, Z. Gan, Y.-C. Chen, Y. Fang, S. Sun, Y. Cheng, and J. Liu, "Cluster-former: Clustering-based sparse transformer for question answering," in *Proc. of ACL Findings*, 2021.

[19] C. Jiao, C. Chen, S. Gou, X. Wang, B. Yang, X. Chen, and L. Jiao, "L sparsity-regularized attention multiple-instance network for hyperspectral target detection," *IEEE Transactions on Cybernetics*, 2021.

[20] P. Manakul and M. Gales, "Long-span summarization via local attention and content selection," in *Proc. of ACL*, 2021.

[21] 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.

[22] Z. Li, S. Ghodrati, A. Yazdanbakhsh, H. Esmaeilzadeh, and M. Kang, "Accelerating attention through gradient-based learned runtime pruning," in *Proc. of ISCA*, 2022.

[23] X. Qin, Z. Dou, Y. Zhu, and J.-R. Wen, "Gdesa: Greedy diversity encoder with self-attention for search results diversification," *ACM Transactions on Information Systems*, 2023.

[24] J. Li, Z. Tu, B. Yang, M. R. Lyu, and T. Zhang, "Multi-head attention with disagreement regularization," in *Proc. of EMNLP*, 2018.

[25] M. Hu, S. Zhao, L. Zhang, K. Cai, Z. Su, R. Cheng, and X. Shen, "Can: Constrained attention networks for multi-aspect sentiment analysis," in *Proc. of EMNLP*, 2019.

[26] J.-B. Cordonnier, A. Loukas, and M. Jaggi, "Multi-head attention: Collaborate instead of concatenate," *arXiv preprint arXiv:2006.16362*, 2020.

[27] A. Mohamed, H.-y. Lee, L. Borgholt, J. D. Havtorn, J. Edin, C. Igel, K. Kirchhoff, S.-W. Li, K. Livescu, L. Maaløe, *et al.*, "Self-supervised speech representation learning: A review," *IEEE Journal of Selected Topics in Signal Processing*, 2022.

[28] Y. Hao, L. Dong, F. Wei, and K. Xu, "Self-attention attribution: Interpreting information interactions inside transformer," in *Proc. of AAAI*, 2021.

[29] J. S. Bridle, "Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition," in *Neurocomputing: Algorithms, architectures and applications*, 1990.

[30] B. Peters, V. Niculae, and A. F. Martins, "Sparse sequence-to-sequence models," in *Proc. of ACL*, 2019.

[31] H. Gong, X. Li, and D. Genzel, "Adaptive sparse transformer for multilingual translation," *arXiv preprint arXiv:2104.07358*, 2021.

[32] A. Martins and R. Astudillo, "From softmax to sparsemax: A sparse model of attention and multi-label classification," in *Proc. of ICML*, 2016.

[33] V. Niculae and M. Blondel, "A regularized framework for sparse and structured neural attention," *Proc. of NeurIPS*, 2017.

[34] C. Godsil and G. F. Royle, *Algebraic graph theory*. Springer Science & Business Media, 2001.

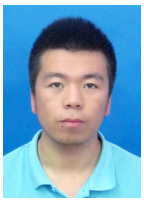
[35] Y. Li, W. Wang, and Z. Yang, "The connected vertex cover problem in k-regular graphs," *Journal of Combinatorial Optimization*, 2019.

[36] B. Hu, Z. Lu, H. Li, and Q. Chen, "Convolutional neural network architectures for matching natural language sentences," *Proc. of NeurIPS*, 2014.

[37] H. Wang and J. Leskovec, "Combining graph convolutional neural networks and label propagation," *ACM Transactions on Information Systems (TOIS)*, 2021.

[38] J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal, "Fever: A large-scale dataset for fact extraction and verification," in *Proc. of NAACL*, 2018.
[39] T. Schuster, D. Shah, Y. J. S. Yeo, D. R. F. Ortiz, E. Santus, and R. Barzilay, "Towards debiasing fact verification models," in *Proc. of EMNLP*, 2019.
[40] S. Iyer, N. Dandekar, K. Csernai, *et al.*, "First quora dataset release: Question pairs. data. quora. com," 2017.
[41] Y. Zhang, J. Baldridge, and L. He, "Paws: Paraphrase adversaries from word scrambling," in *Proc. of NAACL*, 2019.
[42] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The long-document transformer," *arXiv preprint arXiv:2004.05150*, 2020.
[43] M. Zaheer, G. Guruganesh, K. A. Dubey, J. Ainslie, C. Alberti, S. Ontanon, P. Pham, A. Ravula, Q. Wang, L. Yang, *et al.*, "Big bird: Transformers for longer sequences," *Proc. of NeurIPS*, 2020.
[44] Y. Yaghoobzadeh, S. Mehri, R. T. des Combes, T. J. Hazen, and A. Sordoni, "Increasing robustness to spurious correlations using forgettable examples," in *Proc. of EACL*, 2021.
[45] C. Clark, M. Yatskar, and L. Zettlemoyer, "Don't take the easy way out: Ensemble based methods for avoiding known dataset biases," in *Proc. of EMNLP*, 2019.
[46] P. A. Utama, N. S. Moosavi, and I. Gurevych, "Mind the trade-off: Debiasing nlu models without degrading the in-distribution performance," in *Proc. of ACL*, 2020.
[47] M. Lee, S. Won, J. Kim, H. Lee, C. Park, and K. Jung, "Crossaug: A contrastive data augmentation method for debiasing fact verification models," in *Proc. of CIKM*, 2021.
[48] H. He, S. Zha, and H. Wang, "Unlearn dataset bias in natural language inference by fitting the residual," in *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, 2019.
[49] G. Zhang, B. Bai, J. Liang, K. Bai, S. Chang, M. Yu, C. Zhu, and T. Zhao, "Selection bias explorations and debias methods for natural language sentence matching datasets," in *Proc. of ACL*, 2019.
[50] R. Xiong, Y. Chen, L. Pang, X. Cheng, Z.-M. Ma, and Y. Lan, "Uncertainty calibration for ensemble-based debiasing methods," *Proc. of NeurIPS*, 2021.
[51] A. Ghaddar, P. Langlais, M. Rezagholizadeh, and A. Rashid, "End-to-end self-debiasing framework for robust nlu training," in *Proc. of ACL Findings*, 2021.
[52] S. Dou, R. Zheng, T. Wu, S. Gao, J. Shan, Q. Zhang, Y. Wu, and X.-J. Huang, "Decorrelate irrelevant, purify relevant: Overcome textual spurious correlations from a feature perspective," in *Proc. of COLING*, 2022.
[53] J. Su, J. Cao, W. Liu, and Y. Ou, "Whitening sentence representations for better semantics and faster retrieval," *arXiv preprint arXiv:2103.15316*, 2021.
[54] S. Gao, S. Dou, Q. Zhang, and X. Huang, "Kernel-whitening: Overcome dataset bias with isotropic sentence embedding," *arXiv preprint arXiv:2210.07547*, 2022.
[55] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Proc. of EMNLP*, 2019.
[56] M. T. Ribeiro and S. Lundberg, "Adaptive testing and debugging of nlp models," in *Proc. of ACL*, 2022.
[57] C. Si, Z. Gan, Z. Yang, S. Wang, J. Wang, J. Boyd-Graber, and L. Wang, "Prompting gpt-3 to be reliable," *arXiv preprint arXiv:2210.09150*, 2022.
[58] Q. Peng, D. Weir, J. Weeds, and Y. Chai, "Predicate-argument based bi-encoder for paraphrase identification," in *Proc. of ACL*, 2022.
[59] J. M. Meissner, S. Sugawara, and A. Aizawa, "Debiasing masks: A new framework for shortcut mitigation in nlu," *arXiv preprint arXiv:2210.16079*, 2022.
[60] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proc. of NAACL*, 2019.
[61] 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," *arXiv preprint arXiv:1907.11692*, 2019.
[62] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "Albert: A lite bert for self-supervised learning of language representations," *arXiv preprint arXiv:1909.11942*, 2019.
[63] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019.
[64] C. Leiter, R. Zhang, Y. Chen, J. Belouadi, D. Larionov, V. Fresen, and S. Eger, "Chatgpt: A meta-analysis after 2.5 months," *arXiv preprint arXiv:2302.13795*, 2023.
[65] B. Peng, C. Li, P. He, M. Galley, and J. Gao, "Instruction tuning with gpt-4," *arXiv preprint arXiv:2304.03277*, 2023.
[66] O. Bojar, C. Buck, C. Federmann, B. Haddow, P. Koehn, J. Leveling, C. Monz, P. Pecina, M. Post, H. Saint-Amand, *et al.*, "Findings of the 2014 workshop on statistical machine translation," in *Proceedings of the ninth workshop on statistical machine translation*, 2014.
[67] O. Bojar, R. Chatterjee, C. Federmann, Y. Graham, B. Haddow, S. Huang, M. Huck, P. Koehn, Q. Liu, V. Logacheva, *et al.*, "Findings of the 2017 conference on machine translation (wmt17)," in *Proceedings of the Second Conference on Machine Translation*, 2017.
[68] J. Gu, C. Wang, and J. Zhao, "Levenshtein transformer," *Proc. of NeurIPS*, 2019.
[69] L. Qian, H. Zhou, Y. Bao, M. Wang, L. Qiu, W. Zhang, Y. Yu, and L. Li, "Glancing transformer for non-autoregressive neural machine translation," in *Proc. of ACL*, 2021.
[70] M. Ghazvininejad, O. Levy, Y. Liu, and L. Zettlemoyer, "Mask-predict: Parallel decoding of conditional masked language models," in *Proc. of EMNLP*, 2019.
[71] J. Guo, L. Xu, and E. Chen, "Jointly masked sequence-to-sequence model for non-autoregressive neural machine translation," in *Proc. of ACL*, 2020.
[72] B. Wei, M. Wang, H. Zhou, J. Lin, and X. Sun, "Imitation learning for non-autoregressive neural machine translation," in *Proc. of ACL*, 2019.

[73] X. Ma, C. Zhou, X. Li, G. Neubig, and E. Hovy, "Flowseq: Non-autoregressive conditional sequence generation with generative flow," in *Proc. of EMNLP*, 2019.
[74] Z. Sun, Z. Li, H. Wang, D. He, Z. Lin, and Z. Deng, "Fast structured decoding for sequence models," *Proc. of NeurIPS*, 2019.
[75] C. Saharia, W. Chan, S. Saxena, and M. Norouzi, "Non-autoregressive machine translation with latent alignments," in *Proc. of EMNLP*, 2020.
[76] C. Malaviya, P. Ferreira, and A. F. Martins, "Sparse and constrained attention for neural machine translation," in *Proc. of ACL*, 2018.
[77] X.-P. Nguyen, S. Joty, T.-T. Nguyen, K. Wu, and A. T. Aw, "Cross-model back-translated distillation for unsupervised machine translation," in *Proc. of ICML*, 2021.
[78] A. Conneau and G. Lample, "Cross-lingual language model pretraining," *Proc. of NeurIPS*, 2019.
[79] S. R. Indurthi, M. A. Zaidi, B. Lee, N. K. Lakumarapu, and S. Kim, "Language model augmented monotonic attention for simultaneous translation," in *Proc. of NAACL*, 2022.
[80] K. Song, X. Tan, T. Qin, J. Lu, and T.-Y. Liu, "Mass: Masked sequence to sequence pre-training for language generation," in *Proc. of ICML*, 2019.
[81] S. Takase and S. Kiyono, "Rethinking perturbations in encoder-decoders for fast training," in *Proc. of NAACL*, 2021.
[82] A. Fan, S. Bhosale, H. Schwenk, Z. Ma, A. El-Kishky, S. Goyal, M. Baines, O. Celebi, G. Wenzek, V. Chaudhary, *et al.*, "Beyond english-centric multilingual machine translation," *The Journal of Machine Learning Research*, 2021.
[83] S. Ma, L. Dong, S. Huang, D. Zhang, A. Muzio, S. Singhal, H. H. Awadalla, X. Song, and F. Wei, "Deltalm: Encoder-decoder pre-training for language generation and translation by augmenting pretrained multilingual encoders," *arXiv preprint arXiv:2106.13736*, 2021.
[84] A. Williams, N. Nangia, and S. Bowman, "A broad-coverage challenge corpus for sentence understanding through inference," in *Proc. of NAACL*, 2018.
[85] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proc. of EMNLP*, 2013.
[86] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proc. of EMNLP*, 2014.
[87] S. Ma, L. Dong, S. Huang, D. Zhang, A. Muzio, S. Singhal, H. H. Awadalla, X. Song, and F. Wei, "Deltalm: Encoder-decoder pre-training for language generation and translation by augmenting pretrained multilingual encoders," *ArXiv*, 2021.
[88] J. Lee, E. Mansimov, and K. Cho, "Deterministic non-autoregressive neural sequence modeling by iterative refinement," in *Proc. of EMNLP*, 2018.
[89] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proc. of ACL*, 2016.



Shuoran Jiang is currently pursuing a Ph.D. degree with the School of computer science and technology, Harbin Institute of Technology, Shenzhen, China. His research interests include machine learning, natural language inference, and causal inference.

Qingcai Chen received the B.S., M.S., and Ph.D. degrees from the Harbin Institute of Technology, Harbin, China, in 1996, 1998, and 2003, respectively. He is currently a Professor and the Executive Deputy Director of the Intelligent Computing Research Center, School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen. He is also a PI of the Peng Cheng Laboratory, Shenzhen.

Yang Xiang (Member, IEEE), Ph.D., is an assistant professor at Peng Cheng Laboratory, Shenzhen, China. His expertise is in natural language processing, medical informatics, and clinical artificial intelligence.

Youcheng Pan , Ph.D., is currently a postdoctoral researcher at Peng Cheng Laboratory, Shenzhen, China. His research interests include natural language processing, text generation, and machine translation.

Xiangping Wu received the M.S. and Ph.D. degrees in computer science from Harbin Institute of Technology (Shenzhen), China, in 2015 and 2021. Her research interests mainly include computer vision, multimodal machine learning, and pattern recognition.