Modeling Selective Feature Attention for Lightweight Text Matching

Jianxiang Zang, **Hui Liu***
School of Statistics and Information,
Shanghai University of International Business and Economics
{21349110, liuh}@suibe.edu.cn,

Abstract

Representation-based Siamese networks have risen to popularity in lightweight text matching due to their low deployment and inference costs. While word-level attention mechanisms have been implemented within Siamese networks to improve performance, we propose **F**eature **A**ttention (**FA**), a novel downstream block designed to enrich the modeling of dependencies among embedding features. Employing "squeeze-and-excitation" techniques, the FA block dynamically adjusts the emphasis on individual features, enabling the network to concentrate more on features that significantly contribute to the final classification. Building upon FA, we introduce a dynamic "selection" mechanism called **S**elective **F**eature **A**ttention (**SFA**), which leverages a stacked BiGRU Inception structure. The SFA block facilitates multi-scale semantic extraction by traversing different stacked BiGRU layers, encouraging the network to selectively concentrate on semantic information and embedding features across varying levels of abstraction. Both the FA and SFA blocks offer a seamless integration capability with various Siamese networks, showcasing a plug-and-play characteristic. Experimental evaluations conducted across diverse text matching baselines and benchmarks underscore the indispensability of modeling feature attention and the superiority of the "selection" mechanism.

1 Introduction

The goal of the text matching task is to assess the semantic relevance between pairs of sentences and to determine their relationship. More specifically, it involves creating a classifier ξ that calculates the conditional probability $P(\text{label}|\boldsymbol{s}^a, \boldsymbol{s}^b)$, thereby predicting the relationship between the sentence pair $\boldsymbol{s}^a$ and $\boldsymbol{s}^b$. Here, label $\in \Omega$ represents different levels of sentence pair relationships, which can be {relevant, irrelevant} or {entailed, neutral, contradicted}. Representation-based Siamese networks [Wang *et al.*, 2017; Chen *et al.*, 2017; Yang *et al.*, 2019; Zang and Liu, 2023a] use dual encoders to compute text embeddings offline and aggregate them downstream for prediction. They offer the benefits of having low parameter sizes and reduced inference latency, making them extensively applicable in industrial contexts, including search engines and recommendation systems [Huang *et al.*, 2013; Khattab and Zaharia, 2020]. To enhance the post-interaction of text pairs, researchers have introduced various downstream attentions in Siamese networks [Chen *et al.*, 2017; Yang *et al.*, 2019; Cao *et al.*, 2020; Liu *et al.*, 2021]. Notably, these attention strategies solely capture ***word-level*** dependencies, neglecting the modeling of intricate relationships among ***embedding features***. Each feature in text embeddings captures certain semantic or syntactic properties of the vocabulary, though these properties are usually not directly interpretable. For example, particular dimensions in the embedding vector might be related to parts of speech, contextual information (the relationship of a word to surrounding words), or other linguistic attributes.

The Word-level Interaction Attention shown in Figure 1(a) is the most commonly used downstream attention in Siamese matching networks [Chen *et al.*, 2017; Tay *et al.*, 2018; Yang *et al.*, 2019], and is a mapping $(\boldsymbol{a}, \boldsymbol{b}) \rightarrow (\boldsymbol{x}, \boldsymbol{y})$, where $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^{L\times D}$. Here, $\boldsymbol{a}, \boldsymbol{b}$ represent the text embeddings of the text pair $\boldsymbol{s}^a, \boldsymbol{s}^b$. $\boldsymbol{x}, \boldsymbol{y}$ represent the embeddings of the text pair containing rich word level interaction information. L denotes the length of the text sequence, and D represents the dimension of the embedding features. To enhance the sensitivity of the Siamese network to the embedding features then construct a more robust downstream attention, as illustrated in Figure 1(b), we advocate further building a single-branch symmetric Feature Attention (FA) based on the Word-level Interaction Attention. The FA block is a mapping that does not change the tensor size: $\boldsymbol{x} \rightarrow \boldsymbol{u}$ or $\boldsymbol{y} \rightarrow \boldsymbol{v}$, where $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^{L\times D}$. It is noteworthy that, despite the FA blocks in both branches sharing the *same* form, we advocate *against* sharing their parameters.

The FA block incorporates a "squeeze-and-excitation" approach, which concentrates on the most influential embedding features, enhancing their significance in the final classification. Moreover, inspired by neuroscience, where the size of receptive fields in visual cortical neurons is modulated by external stimuli, we integrate a dynamic "selection" mechanism into Feature Attention based on the stacked BiGRU Inception structure. This results in the creation of Selective

*Corresponding author

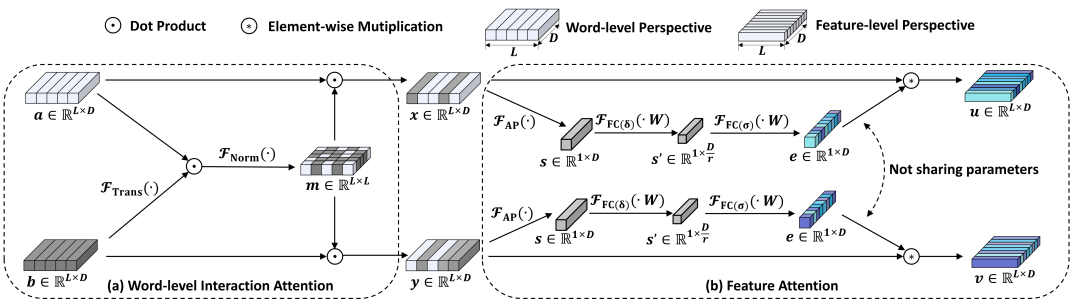

Figure 1: Our more robust downstream attention, composed of (a) Word-level Interaction Attention and (b) Feature Attention.

Feature Attention (SFA). The SFA block stimulates the network to dynamically adapt its focus on semantic information and embedding features across various levels of abstraction. Simultaneously, the "selection" within SFA effectively addresses the challenge of consistent gradient flow arising from the diverse scale semantic extraction in multi-branch Inception, achieving more efficient gradient flow management.

The FA and SFA blocks preserve tensor shape in their mappings, capable of seamless integration with virtually any Siamese network, offering a plug-and-play characteristic. In the experimental evaluation, we combine the FA and SFA blocks with six of the most commonly used baseline Siamese networks from 2020 to 2023, assessing their performance across various text matching benchmarks. Extensive experiments demonstrate that the integration of SFA with all networks significantly improves inference accuracy across all text matching benchmarks. Our primary contributions are highlighted in the following: (1)Based on our survey, we are the first to model dependencies at the embedding feature level for text matching. (2)We present the Feature Attention block and enhance it with a "selection" mechanism based on the stacked BiGRU Inception structure, resulting in the Selective Feature Attention. Extensive experiments confirm the superior performance of the "selection" in SFA block. (3)FA and SFA blocks are offering a plug-and-play characteristic, allowing them to be integrated with almost any Siamese network. [1]

2 Feature Attention

2.1 Squeeze-and-Excitation Network

Channel-level attention mechanisms have demonstrated exceptional performance in image classification [Hu *et al.*, 2018b; Hu *et al.*, 2018a; Woo *et al.*, 2018; Bello *et al.*, 2019] and segmentation [Hou *et al.*, 2020; Huang *et al.*, 2019; Fu *et al.*, 2019]. The Squeeze-and-Excitation Network (SE-Net) [Hu *et al.*, 2018b] pioneered channel attention by effectively constructing interdependencies among channels through the compression of each feature map. CBAM [Woo *et al.*, 2018] further refined this concept by introducing spatial information encoding via convolutions with large-size kernels. Subsequent studies such as GENet [Hu *et al.*, 2018a], GALA [Linsley *et al.*, 2019], TA [Misra *et al.*, 2021] expanded on this idea by adopting various spatial attention mechanisms or designing advanced attention blocks. While related work [Zang and Liu, 2023b] models feature dependencies through higher-dimensional semantic spaces, we devised a squeeze-and-excitation style Feature Attention to model dependencies among semantic features.

2.2 FA Block

As illustrated in Figure 1(b), FA block constitutes a 2D to 2D single-branch mapping that computes $\boldsymbol{x} \rightarrow \boldsymbol{u}$ or $\boldsymbol{y} \rightarrow \boldsymbol{v}$. For the sake of simplicity, our discussion will focus solely on the computational mapping of the $\boldsymbol{x}$ branch.

For the input $\boldsymbol{x} \in \mathbb{R}^{L \times D}$, to capture feature-level dependencies, we first execute a "squeeze" step using average pooling ($\mathcal{F}_{\text{AP}}(\cdot)$) to compress global information into a feature descriptor $\boldsymbol{s}$. Formally, $\boldsymbol{s} \in \mathbb{R}^{1 \times D}$ is generated by averaging $\boldsymbol{x}$ along the spatial dimension L, where the d^{th} element of $\boldsymbol{s}$ is computed as formulated in Equation 1. Throughout this paper, $\mathcal{F}(\cdot)$ represents the mapping that does not involve trainable weights, while $\mathcal{F}(\cdot, \boldsymbol{W})$ represents the mapping that involves trainable weights $\boldsymbol{W}$. Symbols with the subscript $\cdot_l$ denote spatial (word-level) descriptors, while those with the subscript $\cdot_d$ represent feature descriptors.

$$\boldsymbol{s}_d = \frac{1}{L} \sum_{l=1}^{L} \boldsymbol{x}_l \quad , \quad d \in [1, ..., D] \tag{1}$$

In the subsequent "excitation" step, aimed at enhancing the model's sensitivity to features, we filter out the embedding features that contribute more significantly to the final classification. For the aggregated information $\boldsymbol{s}$ from the previous steps, the "excitation" step is tasked with constructing a nonlinear, non-mutually exclusive gating mechanism. To ensure the excitation of multiple features, we have designed two fully connected layers for nonlinear mapping ($\mathcal{F}_{\text{FC}(\delta)}(\cdot, \boldsymbol{W})$ and $\mathcal{F}_{\text{FC}(\sigma)}(\cdot, \boldsymbol{W})$), namely a dimension-reducing layer with a Tanh function followed by a dimension-increasing layer with a Sigmoid function. Here, δ, σ represent the Tanh and Sigmoid function respectively, and $\boldsymbol{s}' \in \mathbb{R}^{1 \times \frac{D}{r}}$ is a transitional vector. The decay factor r introduces a bottleneck in the network to control parameter redundancy.

$$\boldsymbol{e} = \sigma(\delta(\boldsymbol{s}\boldsymbol{W}_{\text{FC}_1})\boldsymbol{W}_{\text{FC}_2}) \tag{2}$$

The vector $\boldsymbol{e} \in \mathbb{R}^{1 \times D}$ signifies the features that have been activated. Ultimately, this vector is merged with $\boldsymbol{x}$ through element-wise multiplication, enabling $\boldsymbol{x}$ to further develop into an embedding feature representation $\boldsymbol{u} =$

[1] Codes available:https://github.com/hggzjx/SFA

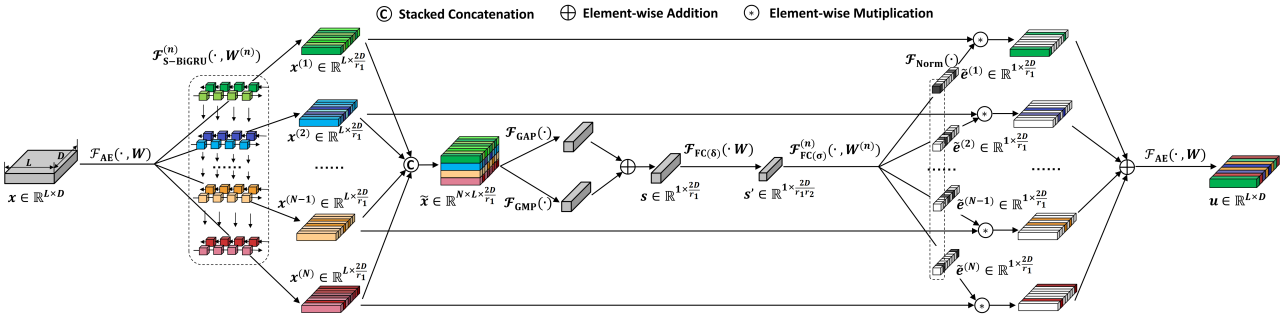

Figure 2: Selective Feature Attention

$[\boldsymbol{u}_1, \boldsymbol{u}_2, ..., \boldsymbol{u}_D] \in \mathbb{R}^{L\times D}$ that is more finely attuned to the final classification.

$$\boldsymbol{u}_d = \boldsymbol{e}_d * \boldsymbol{x}_d \quad , \quad d \in [1, ..., D] \tag{3}$$

3 Selective Feature Attention

3.1 Inception Structure

In the visual cortex, neurons' ability to gather multi-scale spatial information within the same processing stage stems from the varying receptive field sizes in the same region [Hubel and Wiesel, 1962]. The Inception structure [Szegedy *et al.*, 2017] leverages this characteristic, achieving superior performance in computer vision by directly concatenating features extracted from multiple scales. However, this linear aggregation may be insufficient to model the neurons' robust adaptability. Furthermore, the uniform treatment of semantic extraction at different scales gradient flow management during training, affecting training stability.

Related research has shown that stimuli also influence neuronal responses [Nelson and Frost, 1978]. The size of these receptive fields is not fixed but correlates with the stimulus contrast: lower contrast corresponds to a larger effective receptive field [Sceniak *et al.*, 1999]. Selective Kernel Networks (SK-Net) [Li *et al.*, 2019] were the first to model this phenomenon in the field of computer vision, achieving significant success in image classification and semantic segmentation. Motivated by this theory, we introduce the Selective Feature Attention, which encourages the network to dynamically adapt its focus on semantic information and embedding features across different levels of abstraction.

3.2 SFA Block

The SFA block comprises three phases: "split-and-fusion", "squeeze-and-excitation", and "selection", as illustrated in Figure 2. Initially, considering the potential complexity introduced by a multi-branch Inception structure, we advocate the incorporation of bottleneck structure at both ends of the SFA block for feature dimension scaling. Specifically, we employ a one-dimensional convolution kernel of size 1 as an auto encoder $\mathcal{F}_{\text{AE}}(\cdot, \boldsymbol{W})$ to map the size of the input $\boldsymbol{x}$ to $\mathbb{R}^{L\times \frac{D}{r_1}}$, where r_1 acts as a dimension reduction factor controlling the feature dimensionality of embeddings.

In the "split-and-fusion" phase, for $\boldsymbol{x} \in \mathbb{R}^{L\times \frac{D}{r_1}}$, we introduce an N-layer stacked BiGRU ($\mathcal{F}_{\text{S-BiGRU}}(\cdot, \boldsymbol{W}^{(n)})$) to capture the semantic representation at each layer, effectively "splitting" the original embedding into vectors $\{\boldsymbol{x}^{(n)}\}_{n=1}^{N}$, with $\boldsymbol{x}^{(n)} \in \mathbb{R}^{L\times \frac{2D}{r_1}}$, as formulated in Equation 4. Here, $h_l^{(n)}$ represents the hidden state at position l in the n^{th} layer of either the forward or backward GRU and $<;>$ denotes concatenation along the feature dimension.

$$\begin{aligned} \boldsymbol{x}_l^{(n)} =& <\overrightarrow{\text{GRU}}^{(n)}(\overrightarrow{h}_{l-1}^{(n)}, \boldsymbol{x}_l^{(n-1)}, \boldsymbol{W}_{\overrightarrow{\text{GRU}}}^{(n)}); \\ & \overleftarrow{\text{GRU}}^{(n)}(\overleftarrow{h}_{l+1}^{(n)}, \boldsymbol{x}_l^{(n-1)}, \boldsymbol{W}_{\overleftarrow{\text{GRU}}}^{(n)}) > \\ & , n \in [1, ..., N], l \in [1, ..., L] \end{aligned} \tag{4}$$

The shallower layers of BiGRU excel at capturing short-range dependencies between words, such as understanding the combination of words in compound words or phrases. On the other hand, deeper layers of BiGRU are capable of processing and capturing long-range word dependencies. This includes discerning a sentence's theme, which may hinge on words at the beginning and end of the sentence or require consideration of the entire sentence's content for accurate interpretation. The core idea behind "fusion" is to use a gating mechanism to enable information carrying different levels of semantic abstraction from multiple branches to flow towards the neurons of the next layer. To comprehensively and holistically preserve the semantic information of each branch, we employ stacked concatenation to amalgamate the results of all branches, as formulated in Equation 5. Here $[;]$ denotes the stacked concatenation.

$$\widetilde{\boldsymbol{x}}_l = [\boldsymbol{x}_l^{(1)}; \boldsymbol{x}_l^{(2)}; ...; \boldsymbol{x}_l^{(n)}], \quad n \in [1, ..., N], l \in [1, ..., L] \tag{5}$$

In the subsequent "squeeze-and-excitation" phase, for $\widetilde{\boldsymbol{x}} \in \mathbb{R}^{N\times L\times \frac{2D}{r_1}}$, we recommend the combined use of global average pooling ($\mathcal{F}_{\text{GAP}}(\cdot)$) and global max pooling ($\mathcal{F}_{\text{GMP}}(\cdot)$) to compress information simultaneously at both the BiGRU layer and word levels. As formulated in Equation 6, we obtain the sum $\boldsymbol{s}$ from the global average pooling and max pooling results, and then apply fully connected layers for activation.

$$\boldsymbol{s}_d = \frac{1}{N\times L}\sum_{n=1}^{N}\sum_{l=1}^{L}\widetilde{\boldsymbol{x}}_l + \max_{n=1}^{N}\max_{l=1}^{L}(\widetilde{\boldsymbol{x}}_l), \quad d \in [1, ..., \frac{2D}{r_1}] \tag{6}$$

It is noteworthy that the essence of SFA block is to capture and excite features of text embeddings at different levels

of abstraction, while adaptively adjust their relative importance. To accomplish this, we employ a single dimension-reducing fully connected layer ($\mathcal{F}_{\text{FC}(\delta)}(\cdot, \boldsymbol{W})$) alongside a series of dimension-increasing fully connected layers, with the count matching the number of branches ($\mathcal{F}_{\text{FC}(\sigma)}(\cdot, \boldsymbol{W}^{(n)})$) for excitation. This process results in the excited vectors $\{\boldsymbol{e}^{(n)}\}_{n=1}^{N}, \boldsymbol{e}^{(n)} \in \mathbb{R}^{1\times\frac{2D}{r_1}}$, as formulated in Equation 7. Similarly, δ,σ represent the Tanh, Sigmoid function, respectively. The decay factor r_2 creates a bottleneck in the network to avoid parameter redundancy.

$$\boldsymbol{e}^{(n)} = \sigma(\delta(\boldsymbol{s}\boldsymbol{W}_{\text{FC}_1})\boldsymbol{W}_{\text{FC}_2}^{(n)})) \quad , \quad n \in [1, ..., N] \tag{7}$$

In the most critical "selection" phase, we apply vector-level softmax normalization ($\mathcal{F}_{\text{Norm}}(\cdot)$) to the excitation vectors $\{\boldsymbol{e}^{(n)}\}_{n=1}^{N}$. The normalized results $\{\widetilde{\boldsymbol{e}}^{(n)}\}_{n=1}^{N}$ serve as adaptive weights for the different branches, and are used for element-wise multiplication with each branch's representation $\{\boldsymbol{x}^{(n)}\}_{n=1}^{N}$. The summation of these products yields the weighted sum representation of each branch $\boldsymbol{u} = [\boldsymbol{u}_1, \boldsymbol{u}_2 ..., \boldsymbol{u}_{\frac{2D}{r_1}}] \in \mathbb{R}^{L\times\frac{2D}{r_1}}$.

$$\boldsymbol{u}_d = \sum_{n=1}^{N} \widetilde{\boldsymbol{e}}_d^{(n)} * \boldsymbol{x}_d^{(n)} = \frac{\sum_{n=1}^{N} \exp(\boldsymbol{e}_d^{(n)}) * \boldsymbol{x}_d^{(n)}}{\sum_{n=1}^{N} \exp(\boldsymbol{e}_d^{(n)})}, d \in [1, ..., \frac{2D}{r_1}] \tag{8}$$

Finally, to align the input and output dimensions of the SFA block, we also employ the same auto encoder ($\mathcal{F}_{\text{AE}}(\cdot)$) to transform the shape of $\boldsymbol{u}$ to $\mathbb{R}^{L\times D}$. To ensure the expressive power of the embedding feature, the feature dimension reduction caused by the bottleneck layer in the entire paper must satisfy the Equation 9 [Lai *et al.*, 2016].

$$\frac{D}{r}, \frac{2D}{r_1}, \frac{2D}{r_1 r_2} > 8.33 \log L \tag{9}$$

3.3 Efficient Gradient Management

The traditional Inception structure aggregates multi-scale information linearly. However, this uniform updating of weights across different Inception layers impedes the differentiated flow of gradients, thereby impacting training stability. The distinct mapping branches $\mathcal{F}_{\text{S-BiGRU}}^{(n)}(\cdot, \boldsymbol{W}^{(n)}) : \boldsymbol{x} \rightarrow \boldsymbol{x}^{(n)}$ represent semantic extraction at various scales from the text embeddings, acknowledging that semantic extraction at different scales contributes differently to the final classification. In this section, we analyze the backpropagation within the SFA block during training, focusing on how the "selection" mechanism impacts training stability in the context of the gradient flow $\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{x}}$.

w/o selection Firstly, an SFA block without "selection" implies that there is only a single feature weight $\boldsymbol{e}$, thus we have $\boldsymbol{u} = \boldsymbol{e} * \widetilde{\boldsymbol{x}}$. Consequently, the gradient flow chain is jointly determined by the gradient propagations of both $\boldsymbol{e}$ and $\widetilde{\boldsymbol{x}}$, specifically $\frac{\partial \boldsymbol{u}}{\partial \widetilde{\boldsymbol{x}}} = \boldsymbol{e}$ and $\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{e}} = \widetilde{\boldsymbol{x}}$. The gradient flow in the direction of $\boldsymbol{e}$ is formulated in Equation 10.

$$\frac{\partial \boldsymbol{u}}{\partial \widetilde{\boldsymbol{x}}} = \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{e}} * \frac{\partial \boldsymbol{e}}{\partial \boldsymbol{s}'} * \frac{\partial \boldsymbol{s}'}{\partial \boldsymbol{s}} * (\frac{\partial \mathcal{F}_{\text{GMP}}(\widetilde{\boldsymbol{x}})}{\partial \widetilde{\boldsymbol{x}}} + \frac{\partial \mathcal{F}_{\text{GAP}}(\widetilde{\boldsymbol{x}})}{\partial \widetilde{\boldsymbol{x}}}) \tag{10}$$

Combining the backward propagation of gradients from $\boldsymbol{x} \rightarrow \widetilde{\boldsymbol{x}}$, the computation of the overall gradient flow is formulated in Equation 11. In this context, the transformation from $\boldsymbol{x}^{(n)} \rightarrow \widetilde{\boldsymbol{x}}$ is a stacked concatenation, hence it follows that $\frac{\partial \widetilde{\boldsymbol{x}}}{\partial \boldsymbol{x}^{(n)}} = 1$.

$$\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{x}} = 2 * \frac{\partial \boldsymbol{u}}{\partial \widetilde{\boldsymbol{x}}} * \sum_{n=1}^{N} \frac{\partial \widetilde{\boldsymbol{x}}}{\partial \boldsymbol{x}^{(n)}} * \frac{\partial \boldsymbol{x}^{(n)}}{\partial \boldsymbol{x}} = 2 * \frac{\partial \boldsymbol{u}}{\partial \widetilde{\boldsymbol{x}}} * \sum_{n=1}^{N} \frac{\partial \boldsymbol{x}^{(n)}}{\partial \boldsymbol{x}} \tag{11}$$

It can be observed that each semantic extraction mapping $\mathcal{F}_{\text{S-BiGRU}}^{(n)} : \boldsymbol{x} \rightarrow \boldsymbol{x}^{(n)}$ represents a gradient flow $\frac{\partial \boldsymbol{x}^{(n)}}{\partial \boldsymbol{x}}$, which impacts the overall gradient flow $\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{x}}$ in a uniform proportion of $2 * \frac{\partial \boldsymbol{u}}{\partial \widetilde{\boldsymbol{x}}}$. This implies that the feature weights of each Bi-GRU layer are updated to the same degree. However, this uniformity does not align with the necessity for differentiated gradient flow management across various semantic scales, resulting in unstable training.

w/ selection The "selection" mechanism of the SFA block introduces an adaptive weight for each branch, leading to the entire gradient flow chain being jointly determined by the gradient propagations of both $\boldsymbol{x}^{(n)}$ and $\widetilde{\boldsymbol{e}}^{(n)}$. This is expressed as $\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{x}^{(n)}} = \boldsymbol{e}^{(n)}$ and $\frac{\partial \boldsymbol{u}}{\partial \widetilde{\boldsymbol{e}}^{(n)}} = \boldsymbol{x}^{(n)}$. The gradient flow in the direction of $\widetilde{\boldsymbol{e}}^{(n)}$ is formulated in Equation 12. Since the process involves a straightforward addition, it results in $\frac{\partial \boldsymbol{u}}{\partial \widetilde{\boldsymbol{e}}^{(n)}} = 1$, as formulated in Equation 12.

$$\begin{aligned} \frac{\partial \boldsymbol{u}}{\partial \widetilde{\boldsymbol{x}}} = &\sum_{n=1}^{N} (\frac{\partial \boldsymbol{u}}{\partial \widetilde{\boldsymbol{e}}^{(n)}} * \frac{\partial \widetilde{\boldsymbol{e}}^{(n)}}{\partial \boldsymbol{e}^{(n)}} * \frac{\partial \boldsymbol{e}^{(n)}}{\partial \boldsymbol{s}'}) \\ &* \frac{\partial \boldsymbol{s}'}{\partial \boldsymbol{s}} * (\frac{\partial \mathcal{F}_{\text{GMP}}(\widetilde{\boldsymbol{x}})}{\partial \widetilde{\boldsymbol{x}}} + \frac{\partial \mathcal{F}_{\text{GAP}}(\widetilde{\boldsymbol{x}})}{\partial \widetilde{\boldsymbol{x}}}) \end{aligned} \tag{12}$$

After backpropagating the gradients in Equation 12, a complete gradient chain is formulated in Equation 13. It is evident that, upon the introduction of a selection mechanism, the coefficient preceding each $\frac{\partial \boldsymbol{x}^{(n)}}{\partial \boldsymbol{x}}$ becomes a linear function of $\widetilde{\boldsymbol{e}}^{(n)}$. This equation reflects how $\frac{\partial \boldsymbol{x}^{(n)}}{\partial \boldsymbol{x}}$ directly and, through the adaptive weights $\widetilde{\boldsymbol{e}}^{(n)}$, indirectly influences the total gradient flow. In contrast to the scenario without the "selection", where $\frac{\partial \boldsymbol{x}^{(n)}}{\partial \boldsymbol{x}}$ affects the total gradient flow uniformly, the adaptive weights $\widetilde{\boldsymbol{e}}^{(n)}$ manage the gradient flow across different branches in an adaptive and differentiated manner. This approach leads to a more robust training process, as reflected in Section 4.2.

$$\begin{aligned} \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{x}} &= \sum_{n=1}^{N} (\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{x}^{(n)}} * \frac{\partial \boldsymbol{x}^{(n)}}{\partial \boldsymbol{x}}) + \frac{\partial \boldsymbol{u}}{\partial \widetilde{\boldsymbol{x}}} * \sum_{n=1}^{N} (\frac{\partial \widetilde{\boldsymbol{x}}}{\partial \boldsymbol{x}^{(n)}} * \frac{\partial \boldsymbol{x}^{(n)}}{\partial \boldsymbol{x}}) \\ &= \sum_{n=1}^{N} (\widetilde{\boldsymbol{e}}^{(n)} + \frac{\partial \boldsymbol{u}}{\partial \widetilde{\boldsymbol{x}}}) * \frac{\partial \boldsymbol{x}^{(n)}}{\partial \boldsymbol{x}} \end{aligned} \tag{13}$$

4 Experimental Results & Analysis

4.1 Main Results

We selected the six most commonly used lightweight baselines in text matching tasks from 2020 to 2023. All these baselines reported parameter sizes in the articles with values less than ten million. The selected baselines are: BiMPM [Wang *et al.*, 2017], ESIM [Chen *et al.*, 2017], CAFE [Tay *et al.*, 2018], RE2 [Yang *et al.*, 2019], DIIN [Gong *et al.*, 2018], and DRCN [Kim *et al.*,

Network	r	N	QQP	MRPC	BoolQ	SNLI	MNLI(m/mm)	QNLI	Scitail	Avg.	$P_{avg.}$(M)	$IL_{avg.}$(ms)
BiMPM	-	-	85.54	70.38	69.75	87.22	72.34/72.02	79.24	79.45	76.99	1.83	0.403
+FA	2	1	85.79	70.61†	70.18†	87.19	72.74†/72.41†	80.19†	80.13†	77.41†	2.01	0.481
+SFA	(3.75, 4)	2	**86.13†**	**71.33†**	**70.86†**	**88.14†**	**73.19†/72.97†**	**80.94†**	**80.98†**	**78.17†**	2.06	0.708
ESIM	-	-	87.92	73.48	71.71	88.04	74.27/74.19	80.84	79.23	78.71	4.46	0.672
+FA	1	1	88.38†	73.59	72.32†	88.59†	74.28/74.11	81.16†	80.17†	79.08†	4.82	0.718
+SFA	(3,5)	3	**90.32†**	**75.88†**	**73.94†**	**90.02†**	**76.31†/76.19†**	**82.92†**	**81.62†**	**80.90†**	4.88	1.248
CAFE	-	-	88.01	73.18	71.23	87.68	75.02/74.45	81.65	80.54	78.97	4.75	0.672
+FA	1	1	89.04†	73.25	71.34	87.98†	75.34†/74.47	81.71	81.75	79.36	5.11	0.708
+SFA	(3,5)	3	**90.27†**	**74.54†**	**73.21†**	**89.72†**	**77.03†/76.33†**	**82.84†**	**82.82†**	**80.85†**	5.17	1.386
RE2	-	-	88.78	73.17	72.11	88.29	75.98/75.52	80.36	82.45	79.58	4.85	0.742
+FA	1	1	90.14†	73.51†	72.97†	88.87†	76.11/75.71†	80.53†	82.38	80.03†	5.21	0.805
+SFA	(3,5)	3	**90.97†**	**75.29†**	**74.28†**	**90.52†**	**77.61†/77.41†**	**81.97†**	**84.52†**	**81.57†**	5.27	1.478
DIIN	-	-	88.26	73.03	71.45	88.08	76.56/76.49	81.67	82.34	79.74	4.42	0.653
+FA	1	1	88.71†	73.42†	71.78†	88.56†	76.43/76.37	82.05†	82.87†	80.02†	4.78	0.689
+SFA	(3,5)	3	**90.34†**	**75.04†**	**73.73†**	**89.33†**	**78.03†/77.82†**	**82.68†**	**84.51†**	**81.44†**	4.84	1.289
DRCN	-	-	88.81	72.45	71.67	89.84	78.07/77.85	81.03	82.98	80.34	6.68	1.436
+FA	1	1	90.18	72.56	71.98†	90.25†	78.38†/78.17†	81.21	83.07	80.73†	7.40	1.608
+SFA	(3,5)	3	**90.53†**	**74.34†**	**73.57†**	**90.96†**	**79.25†/78.95†**	**82.38†**	**84.97†**	**81.87†**	7.51	2.803

Table 1: The evaluation accuracy (%) of introducing FA and SFA on 6 lightweight text matching baselines across 7 text matching benchmarks. The hyperparameters r and N represent the dimension reduction factors and the number of branches in the Inception network, respectively. $P_{avg.}$(M) denotes the model's average parameters (million), and $IL_{avg.}$(ms) indicates the sentence-level inference latency. The bolded parts represent the best values in each group, and †signifies a significant improvement over the baseline (t-test, $p < 0.05$).

2019]. In the experiments, we introduce FA block and SFA block to these baseline networks and evaluated their performance on following benchmarks: QQP [Iyer *et al.*, 2017], MRPC [Dolan and Brockett, 2005], BoolQ [Clark *et al.*, 2019], SNLI[Bowman *et al.*, 2015], MNLI [Williams *et al.*, 2018](matched&mismatched), QNLI [Wang *et al.*, 2018], and Scitail [Khot *et al.*, 2018].

Table 1 reports the evaluation accuracies of six lightweight text matching baselines, as well as their performances following the integration of FA and SFA blocks. Particularly, since DRCN is composed of multiple identical modules in series, we incorporated two FA and SFA blocks in each branch to align with the number of autoencoders present in DRCN. To minimize the impact of increased parameters on performance, we carefully controlled the values of r and N to maintain a consistent increment in parameters caused by the FA and SFA blocks. By regulating these hyperparameters, the additional parameters introduced by FA and SFA amounted to approximately 5%-10% of the network's original parameters. The sentence-level inference latency caused by the SFA block is higher than that of the FA block, due to the increased model complexity resulting from the stacked BiGRU. The incorporation of the FA block lead to overall improvements across all baselines, while the introduction of the SFA block significantly boost inference accuracy. The SFA block consistently demonstrate the best performance across all baselines compared to the FA block. It provides the most substantial average performance boost for the ESIM model (from 78.71% to 80.90%). With the integration of the SFA block, DRCN achieved the highest accuracy (81.87%) among all baselines, albeit at the cost of having the highest network parameters and inference latency.

Table 2 reports a comprehensive comparison of six SFA-enhanced lightweight baselines against several high-parameter networks, (Residual Stacked [Nie and Bansal, 2017], LSTM-Max [Conneau *et al.*, 2017], LM-Transformer [Vaswani *et al.*, 2017], BERT-base/large [Devlin *et al.*, 2019], AlBERT [Lan *et al.*, 2019], and RoBERTa [Liu *et al.*, 2019]). The comparison is based on evaluation accuracies, parameter sizes, and inference latencies across the QQP and SNLI benchmarks. In the QQP benchmark, it is observed that after integrating SFA, networks such as ESIM, CAFE, RE2, DIIN, and DRCN not only retain a substantial advantage in parameter volume and inference latency but also surpass the performance of pre-trained models like AlBERT (89.31%) and BERT (90.06%). Particularly noteworthy is RE2-SFA, which, with only 1.5% and 4.2% of the parameters and 5.4%, 12.3% of the inference latency, surpasses the accuracy of BERT-large (90.45%) and RoBERTa-base (90.73%) at 90.97%. In the SNLI benchmark, our networks with SFA outperform Residual Stacked, LSTM-Max, LM-Transformer, and AlBERT in terms of accuracy. Notably, DRCN, with just 6.9% and 2.2% of the parameter size and 36.2%, 10.3% of the inference latency, surpasses both BERT-base (90.16%) and BERT-large (90.82%). Despite having only 6.0% of the parameters and 23.2% of the inference latency, DRCN

Network	QQP	SNLI	$P_{avg.}$(M)	$IL_{avg.}$(ms)
Residual Stacked*	-	86.0	29	-
LSTM-Max*	-	84.5	40	-
LM-Transformer*	-	89.9	85	-
AlBERT	89.31	87.30	12	7.27
BERT-base	90.06	90.16	109	7.74
BERT-large	90.45	90.82	335	27.26
RoBERTa-base	**90.73**	**91.13**	125	12.05
BiPMP-SFA	86.13	88.14	2.06	0.71
ESIM-SFA	90.32	90.02	4.88	1.25
CAFE-SFA	90.27	89.72	5.17	1.39
RE2-SFA	**90.97**	90.52	5.27	1.48
DIIN-SFA	90.34	89.33	4.84	1.29
DCRN-SFA	90.53	**90.96**	7.51	2.80

Table 2: Evaluation results of the network with the SFA block integrated, compared with other large-scale networks on QQP and SNLI. * signifies that the result directly adopts the accuracy as reported in [Kim *et al.*, 2019].The bolded parts represent the two best results for each evaluation benchmark.

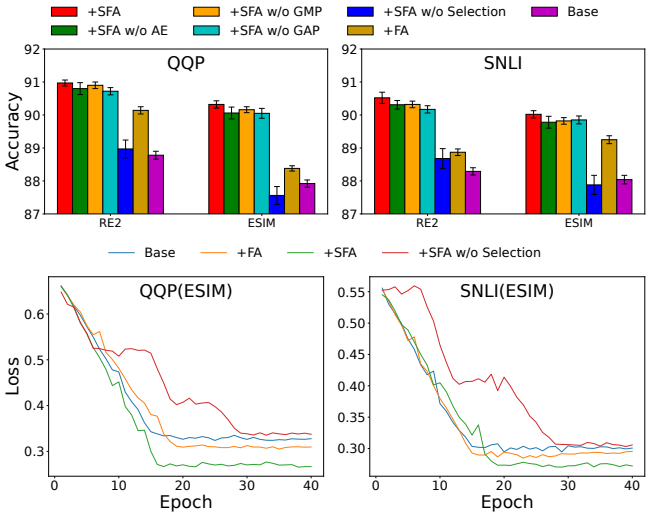

Figure 3: Ablation study on the components of SFA block on QQP and SNLI datasets, using RE2 and ESIM as baselines.

closely approaches the accuracy of RoBERTa-base (91.13%).

4.2 Ablation Study

To investigate the key factors contributing to the superiority of the SFA block over the FA block, we conducted detailed ablation studies on the various components of the SFA block. As indicated in Table 1, ESIM and RE2 demonstrated the most significant average performance improvement following the integration of the SFA block. Thus, for a clearer representation of the results, we chose these two networks, along with QQP and SNLI, as the baselines and benchmarks for our ablation experiments.

Figure 3 illustrates the changes in prediction accuracy when components such as auto encoder, global max pooling, global average pooling, and the "selection" within the SFA block are individually removed. It is evident that the exclusion of each of these components resulted in a decrease in performance. Notably, the exclusion of the 'selection' step resulted in a substantial drop in performance, dipping below the outcomes achieved with the FA block. This impact was pronounced enough to cause ESIM's inference accuracy on QQP and SNLI to fall below that of their original baseline networks. Furthermore, the error bars indicate increased training instability after the removal of the "selection".

Figure 3 illustrates the loss-epochs curves (dev.) for the ESIM model trained on two datasets. It is evident that the introduction of the FA and SFA blocks not only maintains or even improves training convergence speed but also enhances overall convergence performance. On the other hand, introducing a SFA block without the "selection" leads to numerous inefficient training processes. The loss exhibits little variation, significantly slowing down the convergence compared to previous models. As explained in Section 3.3, the omission of the "selection" yields uniform gradient updates across different Inception branches during feature extraction, leading to reduced training efficiency. This obstructed backpropagation process also hinders effective convergence during training, resulting in outcomes that may even be inferior to those of the base network.

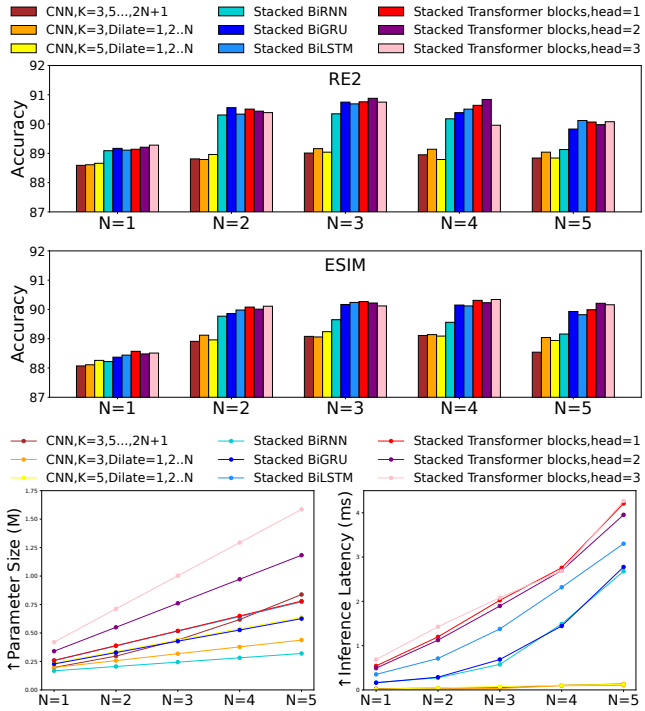

Figure 4: The average increase in evaluation accuracy (%) of SFA blocks on QQP and SNLI with different Inception networks (using RE2 and ESIM as baselines), along with the associated parameters and average inference latency growth.

4.3 Inception Networks

Section 4.2 corroborates that the "selection" is the pivotal element in the SFA block, which is built upon the foundation of multi-branch Inception networks. In this section, we discuss the effects of semantic multi-scale mapping using various Inception networks. These networks can be categorized into three types based on their fundamental architecture: CNN, RNN, and Transformer. For the CNN series, we investigate three forms: (1) CNNs with varying one-dimensional kernel sizes ($K = 2N + 1$) across different branches, (2) CNNs with a fixed kernel size of 3 but varying dilation factors (Dilation $= N$) across branches, and (3) CNNs with a fixed kernel size of 5 and varying dilation factors (Dilation $= N$). For the RNN series, we explore stacked BiRNN, stacked BiGRU, stacked BiLSTM. For the Transformer series, we examine three forms of stacked Transformer blocks with 1, 2, 3 attention heads, respectively. Specifically, we continue to use RE2 and ESIM as baselines and introduce SFA blocks with different Inception networks on top of them. Figure 4 illustrates the average accuracies of the networks on the QQP and SNLI datasets. A significant increase in accuracy is observed when the number of branches changes from 1 to 2 in each Inception structure, further emphasizing the criticality of the "selection" based on multi-branch Inception. Compared to the RNN and Transformer series, SFA blocks built on the

CNN series do not exhibit superior performance, possibly because RNN and Transformer architectures are suited for capturing sequential characteristics. When the branch amount reaches 3, the RNN and Transformer series generally achieve the highest inference accuracy.

Figure 4 illustrates the increase in parameters and inference latency introduced by different Inception based SFA blocks, under the constraint of identical bottleneck factors (r_1, r_2). It is observed that at $N = 3$, stacked BiLSTM and stacked Transformer blocks incur relatively high parameters and inference delay, contradicting the principle of lightweight text matching. In comparison, stacked BiRNN and stacked BiGRU strike a balance between performance and computational costs. SFA blocks based on stacked BiGRU exhibit superior and more stable accuracy compared to stacked BiRNN. This is the reason why we opted for a stacked BiGRU as the Inception network within the SFA block.

4.4 Attention Analysis

The interaction of sentence embeddings directly influences the performance of matching networks. To visually illustrate the impact mechanism of FA and SFA blocks on the network, We selected two sets of sentence pairs from the MRPC, both of which have an 'irrelevant' relationship between them: "*Robin Saunders, head of the bank's London-based principal finance unit, is also expected to quit.*" & "*Robin Saunders, head of the principal finance unit, has made clear she has funding to buy parts of the business.*" and "*In the second quarter, Anadarko now expects volume of 46 million BOE, down from 48 million BOE.*" & "*Production for the second quarter was cut to 46 million barrels from 48 million barrels.*".

We encoded the two sentence pairs using three types of trained ESIM (base, +FA, +SFA) respectively, and visualized the word-level dot product matrices of the base network $\boldsymbol{x}$ and $\boldsymbol{y}$, as well as those of the network $\boldsymbol{u}$ and $\boldsymbol{v}$ with FA and SFA integrated. For visualization purposes, we performed feature-level average pooling, as illustrated in Figure 5. It can be observed that the base network only activates attention between synonyms, such as "*head of*," "*principal finance unit*", "*second quarter*", "*46 million*" and "*48 million*". When the FA block is introduced, this situation remains unchanged, as focusing only on synonyms does not make the network aware of their lack of relevance. In contrast, the introduction of the SFA block prompts the network to activate additional segments, such as "*is also expected to quit*" & "*has funding to buy parts of the business*" and "*now expects*" & "*was cut to*". These segments are essential for the network to determine the "irrelevant" relationship between sentences. This is because the FA block only adjusts the weights among individual features of each word, not affecting the values post-average pooling of all features (the weight of the sum of word-level features), and thus cannot directly influence word-level interactions. On the other hand, the SFA block aggregates embedding features across different scales with weighting. The mappings at these different scales are nonlinear, which leads to changes in the weight of the sum of word-level features. This directly activates words that were previously not focused on by the network, enabling the extraction of semantic information at a finer granularity and capturing the semantic focus.

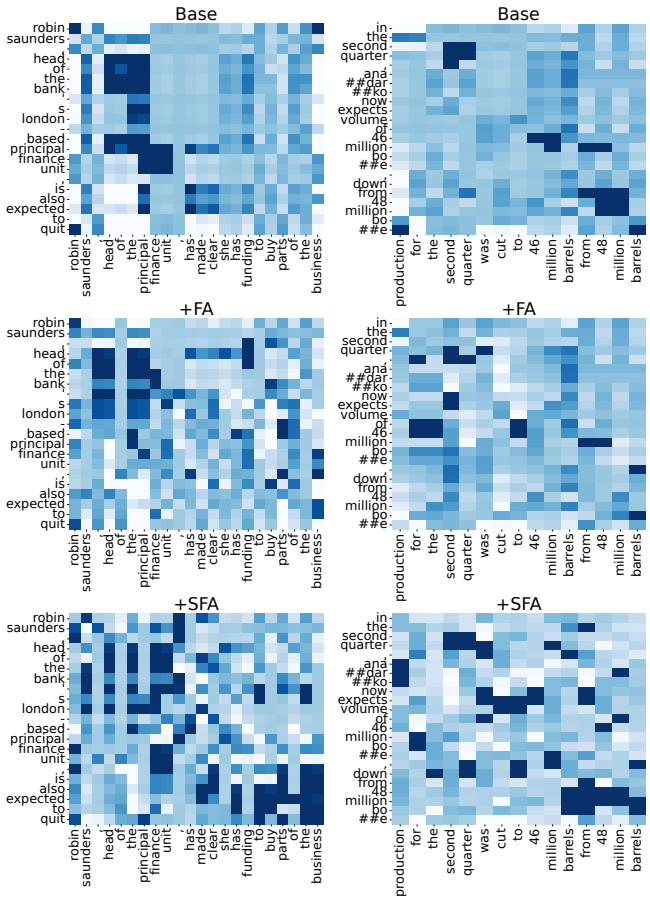

Figure 5: The heatmap of the dot product matrix for sentence pair embeddings, where deeper colors indicate higher levels of activated word-level attention.

5 Conclusion

In this paper, we introduce innovative attention modelling at the embedding feature level within lightweight text matching networks, presenting the FA and SFA blocks. The structures of the FA block and SFA block are concise, plug-and-play, and offer vast opportunities for expansion. In terms of structural design, beyond being a selection mechanism, the feature attention structure can support various forms, enabling finer-grained feature modeling. In terms of task format, feature attention is merely an activation of feature dependencies, unaffected by task formats. This indicates its applicability to various other semantic embedding-based tasks in NLP, such as text classification, entity recognition. We hope to encourage more researchers to explore diverse forms of feature-level attention across a broader range of NLP tasks, fostering a community ecosystem for feature attention in NLP.

References

[Bello *et al.*, 2019] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V Le. Attention augmented convolutional networks. In *Proceedings of the*

IEEE/CVF international conference on computer vision, pages 3286–3295, 2019.

[Bowman *et al.*, 2015] Samuel Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, 2015.

[Cao *et al.*, 2020] Qingqing Cao, Harsh Trivedi, Aruna Balasubramanian, and Niranjan Balasubramanian. Deformer: Decomposing pre-trained transformers for faster question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4497, 2020.

[Chen *et al.*, 2017] Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. Enhanced lstm for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1657–1668, 2017.

[Clark *et al.*, 2019] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of NAACL-HLT*, pages 2924–2936, 2019.

[Conneau *et al.*, 2017] A Conneau, D Kiela, H Schwenk, L Barrault, and A Bordes. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680. Association for Computational Linguistics, 2017.

[Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

[Dolan and Brockett, 2005] Bill Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Third International Workshop on Paraphrasing (IWP2005)*. Asia Federation of Natural Language Processing, January 2005.

[Fu *et al.*, 2019] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3146–3154, 2019.

[Gong *et al.*, 2018] Yichen Gong, Heng Luo, and Jian Zhang. Natural language inference over interaction space. In *International Conference on Learning Representations*, 2018.

[Hou *et al.*, 2020] Qibin Hou, Li Zhang, Ming-Ming Cheng, and Jiashi Feng. Strip pooling: Rethinking spatial pooling for scene parsing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4003–4012, 2020.

[Hu *et al.*, 2018a] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Andrea Vedaldi. Gather-excite: Exploiting feature context in convolutional neural networks. *Advances in neural information processing systems*, 31, 2018.

[Hu *et al.*, 2018b] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7132–7141, 2018.

[Huang *et al.*, 2013] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*, pages 2333–2338, 2013.

[Huang *et al.*, 2019] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 603–612, 2019.

[Hubel and Wiesel, 1962] David H Hubel and Torsten N Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1):106, 1962.

[Iyer *et al.*, 2017] Shankar Iyer, Nikhil Dandekar, Kornél Csernai, et al. First quora dataset release: Question pairs. data. quora. com. 2017.

[Khattab and Zaharia, 2020] Omar Khattab and Matei Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48, 2020.

[Khot *et al.*, 2018] Tushar Khot, Ashish Sabharwal, and Peter Clark. Scitail: a textual entailment dataset from science question answering. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, pages 5189–5197, 2018.

[Kim *et al.*, 2019] Seonhoon Kim, Inho Kang, and Nojun Kwak. Semantic sentence matching with densely-connected recurrent and co-attentive information. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6586–6593, 2019.

[Lai *et al.*, 2016] Siwei Lai, Kang Liu, Shizhu He, and Jun Zhao. How to generate a good word embedding. *IEEE Intelligent Systems*, 31(6):5–14, 2016.

[Lan *et al.*, 2019] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: Alite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

[Li *et al.*, 2019] Xiang Li, Wenhai Wang, Xiaolin Hu, and Jian Yang. Selective kernel networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 510–519, 2019.

[Linsley *et al.*, 2019] Drew Linsley, Dan Shiebler, Sven Eberhardt, and Thomas Serre. Learning what and where to attend. In *International Conference on Learning Representations*, 2019.

[Liu *et al.*, 2019] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[Liu *et al.*, 2021] Hanxiao Liu, Zihang Dai, David So, and Quoc V Le. Pay attention to mlps. *Advances in Neural Information Processing Systems*, 34:9204–9215, 2021.

[Misra *et al.*, 2021] Diganta Misra, Trikay Nalamada, Ajay Uppili Arasanipalai, and Qibin Hou. Rotate to attend: Convolutional triplet attention module. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 3139–3148, 2021.

[Nelson and Frost, 1978] JI Nelson and BJ Frost. Orientation-selective inhibition from beyond the classic visual receptive field. *Brain research*, 139(2):359–365, 1978.

[Nie and Bansal, 2017] Yixin Nie and Mohit Bansal. Shortcut-stacked sentence encoders for multi-domain inference. *RepEval 2017*, page 41, 2017.

[Sceniak *et al.*, 1999] Michael P Sceniak, Dario L Ringach, Michael J Hawken, and Robert Shapley. Contrast's effect on spatial summation by macaque v1 neurons. *Nature neuroscience*, 2(8):733–739, 1999.

[Szegedy *et al.*, 2017] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

[Tay *et al.*, 2018] Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. Compare, compress and propagate: Enhancing neural architectures with alignment factorization for natural language inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1565–1575, 2018.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

[Wang *et al.*, 2017] Zhiguo Wang, Wael Hamza, and Radu Florian. Bilateral multi-perspective matching for natural language sentences. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 4144–4150, 2017.

[Wang *et al.*, 2018] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Association for Computational Linguistics, 2018.

[Williams *et al.*, 2018] Adina Williams, Nikita Nangia, and Samuel R Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2018*, pages 1112–1122. Association for Computational Linguistics (ACL), 2018.

[Woo *et al.*, 2018] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018.

[Yang *et al.*, 2019] Runqi Yang, Jianhai Zhang, Xing Gao, Feng Ji, and Haiqing Chen. Simple and effective text matching with richer alignment features. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4699–4709, 2019.

[Zang and Liu, 2023a] Jianxiang Zang and Hui Liu. How to extract and interact? nested siamese text matching with interaction and extraction. In *International Conference on Artificial Neural Networks*, pages 523–535. Springer, 2023.

[Zang and Liu, 2023b] Jianxiang Zang and Hui Liu. Improving text semantic similarity modeling through a 3d siamese network. In *ECAI 2023*, pages 2970–2977. IOS Press, 2023.