

# Context-Aware Interaction Network for Question Matching

Zhe Hu<sup>1</sup>, Zuohui Fu<sup>2</sup>, Yu Yin<sup>3</sup>, and Gerard de Melo<sup>4</sup>

<sup>1</sup>Baidu Inc., China, <sup>2</sup>Rutgers University, USA

<sup>3</sup>Northeastern University, USA, <sup>4</sup>HPI/University of Potsdam, Germany

<sup>1</sup>huzhe01@baidu.com, <sup>2</sup>zuohui.fu@rutgers.edu

<sup>3</sup>yin.yul@northeastern.edu, <sup>4</sup>gdm@demelo.org

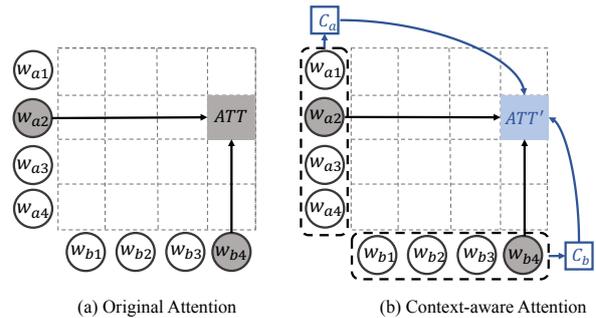
## Abstract

Impressive milestones have been achieved in text matching by adopting a cross-attention mechanism to capture pertinent semantic connections between two sentence representations. However, regular cross-attention focuses on word-level links between the two input sequences, neglecting the importance of contextual information. We propose a context-aware interaction network (COIN) to properly align two sequences and infer their semantic relationship. Specifically, each interaction block includes (1) a context-aware cross-attention mechanism to effectively integrate contextual information when aligning two sequences, and (2) a gate fusion layer to flexibly interpolate aligned representations. We apply multiple stacked interaction blocks to produce alignments at different levels and gradually refine the attention results. Experiments on two question matching datasets and detailed analyses demonstrate the effectiveness of our model.

## 1 Introduction

Semantic text matching is among the most fundamental tasks in natural language processing. Given two sentences, the goal is to predict their semantic relationship. In this work, we focus in particular on question matching (QM) benchmarks.

Recently, the availability of large-scale annotated datasets has led to a proliferation of deep neural architectures for text matching (Williams et al., 2018; Chen et al., 2017; Wang et al., 2017). Most existing neural models fall into two categories, namely the sentence encoding and the sentence interaction approaches (Lan and Xu, 2018). The former encodes sentences as fixed-length vector representations, which are then consulted to make the final prediction. The latter considers interactions between two sequences to identify their semantic connections, which tends to yield better results.



**Figure 1:** The original attention mechanism (left) and the proposed context-aware attention (right).  $w_*$  represents the two sequences (more generally, they can be regarded as query and key).  $C_*$  denotes the contextual features.

Attention mechanisms are widely adopted for the sentence-interaction approaches, relying on a word-by-word attention matrix to obtain alignment information between two sequences. This has proven fruitful in modeling sentence pair relationships (Parikh et al., 2016; Rocktäschel et al., 2015; Wang and Jiang, 2016). Nonetheless, when computing the cross-sentence attention, existing models mostly focus on word-level local matching and fail to fully account for the overall semantics: *each value of the attention matrix is based on just two individual tokens from the sequences without full consideration of the context*. As shown in Figure 1, in the original attention mechanism, each token individually attends to the other tokens without accounting for important contextual information. However, accurate matching may require a deeper understanding of the two sentences along with pertinent linguistic patterns and constructions (Storks et al., 2019). Yang et al. (2019a) show that contextualizing the self-attention network may improve the original representations, but they do not consider the scenario of sentence pairs with cross-attention.

In this work, we aim to generalize the notion of cross-sentence attention by enabling it to incorporate rich contextual signals. We propose a

**C**ontext-aware **I**nteraction **N**etwork (**COIN**) with a novel *context-aware attention layer*. This layer enables the model to consult contextual information while computing the cross-attention matrix to measure the word relevance, yielding better contextualized alignments for semantic reasoning. We leverage the self-alignment on each sequence to produce contexts that represent salient features for each token. The subsequent gate fusion layer is designed to enable the model to selectively integrate the aligned representations and control to what extent the new information is to be passed to the following layers, which is similar to a skip connection in mitigating the additional model complexity coming from the deeper structure. Finally, an aggregation layer and a multi-head pooling layer are adopted to infer high-level semantic representations for the sequences and predict the result based on the refined representations.

To validate the effectiveness of our method, we conduct extensive experiments on the Quora and LCQMC datasets, along with further analyses of model components and a case study visualizing the alignment. The results show that by incorporating rich context into cross-attention, our model outperforms state-of-the-art methods without the huge number of model parameters and pre-training on extrinsic data of BERT models.

## 2 Method

Question matching can be viewed as a classification task that seeks a label  $y \in \mathcal{Y} = \{\text{DUPLICATE}, \text{NON-DUPLICATE}\}$  for a given sentence pair  $(S_a, S_b)$ . Figure 2 illustrates our novel sentence interaction approach for this task. In the following, we describe the individual ingredients of this approach.

### 2.1 Input Representation Layer

The input representation layer converts each sentence into matrix representations with an embedding and encoding layer. We invoke word embeddings without additional lexical features and adopt a multi-layer convolutional encoder on top of the embedding layer. In addition, we concatenate the contextual representations with the original embeddings to produce better alignments in the following interaction blocks. This serves a similar purpose as skip connections to represent words at different levels (Wang et al., 2018).

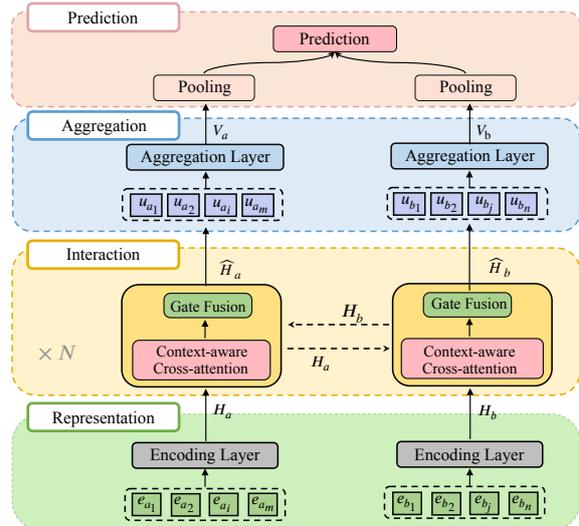


Figure 2: Overview of our model structure.

### 2.2 Context-aware Interaction Block

Our proposed interaction block consists of a context-aware cross-attention and a gate fusion layer. Several such interaction blocks are stacked to obtain refined alignments.

#### 2.2.1 Cross-Attention Layer

We first review the original cross-sentence attention before introducing our context-aware form of attention. Assume the two inputs of the current layer are  $\mathbf{H}_a = (\mathbf{h}_{a_1}, \dots, \mathbf{h}_{a_m})$  and  $\mathbf{H}_b = (\mathbf{h}_{b_1}, \dots, \mathbf{h}_{b_n})$ , where  $m$  and  $n$  are the corresponding sequence lengths. The word-by-word attention matrix is first calculated as follows:

$$\mathbf{E}_{ij} = \text{Att}(\mathbf{h}_{a_i}, \mathbf{h}_{b_j}) = F_1(\mathbf{h}_{a_i})^T F_1(\mathbf{h}_{b_j}), \quad (1)$$

where  $F_1$  is a feed-forward neural network. Then the similarity matrix  $\mathbf{E}$  is used to compute aligned representations of each sequence as a weighted summation with regard to the other sentence:

$$\mathbf{a}_i = \text{softmax}(\mathbf{E}_{i:}), \quad \mathbf{b}_j = \text{softmax}(\mathbf{E}_{:j}) \quad (2)$$

$$\mathbf{h}'_{b_j} = \sum_{k=1}^m \mathbf{b}_{kj} \mathbf{h}_{a_k}, \quad \mathbf{h}'_{a_i} = \sum_{k=1}^n \mathbf{a}_{ik} \mathbf{h}_{b_k} \quad (3)$$

**Limitation.** It is evident in Eq. 1 that each value of the attention matrix is governed by the parameters of the feed-forward layer with respect to only the individual token pairs, so the layer does not take advantage of valuable contextual signals.

#### 2.2.2 Context-Aware Cross-Attention Layer

We propose a novel context-aware cross-attention layer by incorporating contextual representations into the cross-attention. The goal is to enable the

model to identify salient contextual features for each token, and consider these features when computing the cross-attention matrix  $\mathbf{E}$ .

Given  $\mathbf{C}_a = (\mathbf{c}_{a_1}, \dots, \mathbf{c}_{a_m})$ ,  $\mathbf{C}_b = (\mathbf{c}_{b_1}, \dots, \mathbf{c}_{b_n})$  as contextual representations for the two sentences, we modify the attention mechanism from Eq. 1 to be able to draw on these as additional inputs when computing the word-by-word attention matrix:

$$\begin{aligned} \mathbf{E}_{ij}^c &= \text{Att}_{\text{context}}(\mathbf{h}_{a_i}, \mathbf{h}_{b_j}, \mathbf{c}_{a_i}, \mathbf{c}_{b_j}) \\ &= F_1(\mathbf{h}_{a_i} + \mathbf{c}_{a_i})^T F_1(\mathbf{h}_{b_j} + \mathbf{c}_{b_j}) \end{aligned} \quad (4)$$

By incorporating the contextual vectors, the model is able to take advantage of the full context and enable better alignments.

**Contextual Representations.** In order to compute such representations of the contexts, given each sequence, we adopt a self-alignment layer to aggregate pertinent contextual information. Each contextual vector is computed by attending to the input hidden states and conducting a weighted summation. Formally, for the input  $\mathbf{H} = (\mathbf{h}_1, \dots, \mathbf{h}_n)$ :

$$\mathbf{A} = \text{ReLU}(\mathbf{W}_c \mathbf{H})^T \text{ReLU}(\mathbf{W}_c \mathbf{H}) \quad (5)$$

$$\mathbf{C} = \text{softmax}(\mathbf{A}) \mathbf{H} \quad (6)$$

Here,  $\mathbf{W}_c$  is a trainable parameter.

Leveraging the self-alignment to produce contextual signals also mirrors human behavior in the sense that when matching two sentences, people tend to first process each sentence paying attention to the important contents, and then compare the two sentences and connect relevant elements (words or phrases) with contextual features to identify their relationship, rather than just comparing individual words.

### 2.2.3 Gate Fusion Layer

Subsequently, a gate fusion layer compares the original sequences against the aligned representations and blends them together as new sequence representations. Specifically, we first compare the original representation ( $\mathbf{H}_a$ ) with the aligned one ( $\mathbf{H}'_a$ ) from three perspectives, and then combine them with a non-linear transformation:

$$\tilde{\mathbf{h}}_{a_i}^1 = G_1([\mathbf{h}_{a_i}; \mathbf{h}'_{a_i}]) \quad (7)$$

$$\tilde{\mathbf{h}}_{a_i}^2 = G_2([\mathbf{h}_{a_i}; \mathbf{h}_{a_i} - \mathbf{h}'_{a_i}]) \quad (8)$$

$$\tilde{\mathbf{h}}_{a_i}^3 = G_3([\mathbf{h}_{a_i}; \mathbf{h}_{a_i} \odot \mathbf{h}'_{a_i}]) \quad (9)$$

$$\tilde{\mathbf{h}}_{a_i} = \text{ReLU}(\mathbf{W}_f [\tilde{\mathbf{h}}_{a_i}^1; \tilde{\mathbf{h}}_{a_i}^2; \tilde{\mathbf{h}}_{a_i}^3] + \mathbf{b}_f) \quad (10)$$

Then a gated connection is applied to enable the model to selectively integrate the aligned features:

$$\mathbf{f}_i = \sigma(\mathbf{W}_1 \mathbf{h}_{a_i} + \mathbf{W}_2 \tilde{\mathbf{h}}_{a_i} + \mathbf{b}_g) \quad (11)$$

$$\hat{\mathbf{h}}_{a_i} = \mathbf{f}_i \odot \mathbf{h}_{a_i} + (\mathbf{1} - \mathbf{f}_i) \odot \tilde{\mathbf{h}}_{a_i} \quad (12)$$

Here  $\sigma$  is a Sigmoid nonlinear transformation, while  $\mathbf{W}_*$  and  $\mathbf{b}_g$  are trainable parameters. The same operation is conducted on sentence  $S_b$ , thereby yielding the outputs  $\hat{\mathbf{H}}_a$  and  $\hat{\mathbf{H}}_b$ . With these operations, the model can flexibly interpolate the aligned information by controlling the gate, especially when multiple interactions are applied.

## 2.3 Aggregation Layer

To obtain high-level semantic representations for each sentence, we apply another convolutional neural network on top of the interaction blocks to obtain the aggregated sentence representations  $\mathbf{V}_a$ ,  $\mathbf{V}_b$ , serving as the inputs for the prediction layer.

## 2.4 Pooling and Prediction Layer

We compute a weighted summation of the hidden states to get sentence vectors. To allow the model to represent each sequence in different representation subspaces, we adopt multi-head pooling following Liu and Lapata (2019). For each head  $z$ , we first transform the sequence into attention scores  $\mathbf{S}^z$  and values  $\tilde{\mathbf{V}}^z$ :

$$\mathbf{S}^z = \text{softmax}(\mathbf{W}_a^z \mathbf{V}_a) \quad (13)$$

$$\tilde{\mathbf{V}}^z = \mathbf{W}_v^z \mathbf{V}_a \quad (14)$$

where  $\mathbf{W}_a^z \in \mathbb{R}^{1 \times d}$  and  $\mathbf{W}_v^z \in \mathbb{R}^{d_h \times d}$  are trainable parameters, with  $d_h = d/n_h$  as the dimensionality of each head and  $n_h$  as the number of heads. The pooling vector of head  $z$  is computed as

$$\hat{\mathbf{V}}^z = \sum_{i=1}^n \mathbf{s}_i^z \tilde{\mathbf{v}}_i^z, \quad (15)$$

where  $\mathbf{s}_i^z$  and  $\tilde{\mathbf{v}}_i^z$  denote the calculated attention scores and values. The pooling vectors of all heads are concatenated to form the final vector representations of each sequence  $\mathbf{V}'_a$  and  $\mathbf{V}'_b$ . We combine  $\mathbf{V}'_a$  and  $\mathbf{V}'_b$  to produce the overall representation by concatenating the different operations:

$$\mathbf{V} = [\mathbf{V}'_a; \mathbf{V}'_b; \mathbf{V}'_a - \mathbf{V}'_b; \mathbf{V}'_a \odot \mathbf{V}'_b] \quad (16)$$

Finally, the prediction layer takes the representation  $\mathbf{V}$  and passes it to a fully-connected network component to predict the ultimate target scores.

Model	Acc (%)	F1 (%)
Lattice-CNN	82.1	82.4
ESIM (Chen et al., 2017)	82.0	84.0
BiMPM (Wang et al., 2017)	83.3	84.9
GMN (Chen et al., 2020)	84.6	86.0
COIN (Ours)	<b>85.6</b>	<b>86.5</b>
BERT (Devlin et al., 2019)	85.7	86.8
SBERT (Reimers and Gurevych, 2019)	85.4	86.6
COIN (ensemble)	<b>86.2</b>	<b>87.0</b>

Table 1: Experimental results on LCQMC.

### 3 Experiments

#### 3.1 Experimental Setup

**Datasets.** We conduct experiments on two datasets: 1) The Quora Questions Pairs corpus (Quora) contains over 400k English question pairs selected from Quora.com, for which we use the same data split as Wang et al. (2017). 2) LCQMC (Liu et al., 2018) is a large-scale open-domain Chinese question matching corpus constructed from Baidu Knows. We follow the data splits in the original papers, and apply a hard cut-off of the sentence length on both datasets by cropping or padding. The length is set to 32 for Quora and 50 for LCQMC.

**Training Details and Parameters.** For Quora, we use 300 dimensional GloVe embeddings (Pennington et al., 2014). For LCQMC, following Li et al. (2019), we avoid word segmentation and instead use a randomly initialized character embedding matrix. The kernel size is 3 for convolutional layers with padding. We tune the dimensionality of the feed-forward layers from 150 to 300. The batch size is tuned from 32 to 128. Adam optimization is used with an initial learning rate of 0.001 and exponential decay. We use ReLU (Glorot et al., 2011) as the activation function in all feed-forward networks. To prevent over-fitting, dropout with a retention probability of 0.8 is applied. We apply 3 context-aware interaction blocks for Quora and 2 interaction blocks for LCQMC. For BERT (Devlin et al., 2019), we choose the BERT-base version (12 layers, 768 hidden dimensions and 12 attention heads). Further training details are given in the appendix.

#### 3.2 Experimental Results

We compare our model against recent prior work, including state-of-the-art neural models and BERT based methods. ESIM (Chen et al., 2017) and BiMPM (Wang et al., 2017) are two strong sentence-interaction baselines. GMN (Chen et al., 2020) is a neural graph matching network with

Model	Acc. (%)	Params
BiMPM (Wang et al., 2017)	88.2	1.6M
DIIN (Gong et al., 2017)	89.0	4.4M
CAFE (Tay et al., 2018)	88.7	4.7M
OSOA-DFN (Liu et al., 2019)	89.0	10.0M
RE2 (Yang et al., 2019b)	89.2	2.8M
ESAN (Hu et al., 2020)	89.3	3.9M
Enhanced-RCNN (Peng et al., 2020)	89.3	7.7M
COIN (ours)	<b>89.4</b>	6.5M
BERT (Devlin et al., 2019)	90.1	109.5M
SBERT (Reimers and Gurevych, 2019)	90.6	109.5M
COIN (ensemble)	<b>90.7</b>	32.5M

Table 2: Experimental results on Quora dataset.

multi-granular input information. DIIN (Gong et al., 2017) extracts semantic features from the interaction space. OSOA-DFN (Liu et al., 2019) uses multiple original semantics-oriented attention, and RE2 (Yang et al., 2019b) adopts richer features for alignment processes to improve the performance. ESAN (Hu et al., 2020) is a sentence-interaction model with gated feature augmentation. For pre-trained methods, we consider BERT (Devlin et al., 2019) and SBERT (Reimers and Gurevych, 2019). We also include ensemble results of our method where we consider the majority vote of the results given by 5 runs of the same model under different random parameter initialization.

Results on LCQMC are listed in Table 1. Our single model achieves better accuracy and F1-score than all non-pretrained baselines, and the results of COIN are fairly comparable to BERT despite *not* being pretrained on any extrinsic data. In fact, our ensemble model (5 runs) even outperforms BERT.

The results on Quora are given in Table 2. Our approach outperforms the non-pretrained baselines with 89.4% test accuracy, and our ensemble model again achieves better results than BERT and SBERT with fewer parameters (32.5M vs. 109.5M). This confirms our model’s ability to be applied in real-world scenarios that require less computational complexity and a smaller model footprint.

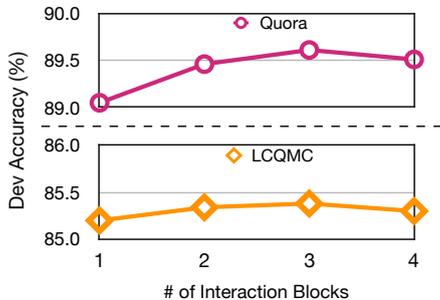
Overall, the above results on two question matching datasets reflect our model’s effectiveness at capturing semantic interactions between the sentences and properly inferring their relationship. In-depth analyses of the model’s efficiency are given in the appendix.

#### 3.3 Model Analysis

**Effect of Model Components.** In Table 3, we study the contribution of different model components. Without context in cross-attention, the accuracy decreases by 0.5 and 0.6 percentage points,

Model	Quora	LCQMC
original	89.6	85.4
w/o context	89.1	84.8
simple fusion	88.8	85.2
w/o aggregat.	89.2	84.9
simple pool	89.4	85.2

**Table 3:** Ablation study on Quora and LCQMC dev set.

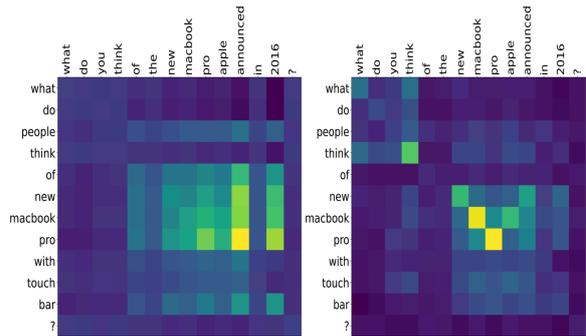


**Figure 3:** Ablation study [Left] and effect of number of interaction blocks [Right] on Quora and LCQMC dev sets.

respectively. This confirms that, by incorporating the context, our model can better capture sentence relationships in the alignments. We then replace the gate fusion with a simplified fusion layer, where we feed the concatenation of the two representations to a feed-forward network, observing a performance drop on both datasets. This shows the effectiveness of our context-aware interaction blocks. We then remove the aggregation layer, finding that the accuracy decreases to 89.2% and 84.9%. This confirms that the aggregation layer is useful to produce high-level representations for the final prediction. In the last ablation, we replace the multi-head pooling by max-pooling to produce the sentence vector, and the results decrease on both datasets.

**Effect of Interaction Block Depth.** Figure 3 plots the accuracy with varying numbers of interaction blocks. Evidently, a smaller number of interaction blocks may not suffice to fully capture the sentence relationships, and adding further such blocks may improve the model’s ability to reason across the sequences and boost the performance. However, increasing the depth of interactions more than necessary harms the performance. Additionally, there is a trade-off between performance and efficiency since adding more interaction blocks increases the number of parameters. For computational cost reasons, we use at most three interactions blocks in our experiments.

**Case Study.** We analyze the context-aware interaction results by visualizing the attention to show how the model learns aligned features at different levels of interaction in Figure 4. We consider a sample from Quora with the target label DUPLICATE.



**Figure 4:** Visualization of alignment in the first and third interactions. Lighter colors indicate higher values.

The left image shows the contextualized cross-attention in the first interaction block. Aided by the context, the model learns to correctly align the salient phrase “*new macbook pro*” across the inputs. The attention results in the third interaction block are visualized in the right image. As we can observe, the model refines the alignment results with a sharper distribution on the salient phrases than in the first interaction block, and the structured phrase “*what do you think of*” is also connected. The model thus predicts the relationship between the two sentences correctly. This corroborates our model’s ability to gradually refine and adjust the attention scores in higher layers.

## 4 Conclusion

In this work, we propose a context-aware interaction network for question matching. We improve the cross-attention by incorporating contextual cues, and further leverage a gate fusion layer to flexibly integrate the aligned features. Experiments on two datasets validate the effectiveness of our architecture and show that accounting for the context enhances the original cross-attention.

## Acknowledgements

We thank the anonymous reviewers for their constructive suggestions.

## References

- Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283.
- Lu Chen, Yanbin Zhao, Boer Lyu, Lesheng Jin, Zhi Chen, Su Zhu, and Kai Yu. 2020. [Neural graph](#)

- matching networks for Chinese short text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6152–6158, Online. Association for Computational Linguistics.
- Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. **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, Vancouver, Canada. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. **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, Minneapolis, Minnesota. Association for Computational Linguistics.
- Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. **Deep sparse rectifier neural networks**. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 315–323, Fort Lauderdale, FL, USA. PMLR.
- Yichen Gong, Heng Luo, and Jian Zhang. 2017. Natural language inference over interaction space. *arXiv preprint arXiv:1709.04348*.
- Zhe Hu, Zuohui Fu, Cheng Peng, and Weiwei Wang. 2020. **Enhanced sentence alignment network for efficient short text matching**. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 34–40, Online. Association for Computational Linguistics.
- Wuwei Lan and Wei Xu. 2018. **Neural network models for paraphrase identification, semantic textual similarity, natural language inference, and question answering**. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3890–3902, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Xiaoya Li, Yuxian Meng, Xiaofei Sun, Qinghong Han, Arianna Yuan, and Jiwei Li. 2019. **Is word segmentation necessary for deep learning of Chinese representations?** In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3242–3252, Florence, Italy. Association for Computational Linguistics.
- Mingtong Liu, Yujie Zhang, Jinan Xu, and Yufeng Chen. 2019. **Original semantics-oriented attention and deep fusion network for sentence matching**. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2652–2661, Hong Kong, China. Association for Computational Linguistics.
- Xin Liu, Qingcai Chen, Chong Deng, Huajun Zeng, Jing Chen, Dongfang Li, and Buzhou Tang. 2018. **LCQMC: a large-scale Chinese question matching corpus**. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1952–1962, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Yang Liu and Mirella Lapata. 2019. **Hierarchical transformers for multi-document summarization**. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5070–5081, Florence, Italy. Association for Computational Linguistics.
- Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. **A decomposable attention model for natural language inference**. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255, Austin, Texas. Association for Computational Linguistics.
- Shuang Peng, Hengbin Cui, Niantao Xie, Sujian Li, Jiaxing Zhang, and Xiaolong Li. 2020. **Enhanced-rnn: An efficient method for learning sentence similarity**. In *Proceedings of The Web Conference 2020*, pages 2500–2506.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. **Glove: Global vectors for word representation**. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.
- Nils Reimers and Iryna Gurevych. 2019. **Sentence-BERT: Sentence embeddings using Siamese BERT-networks**. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.
- Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiský, and Phil Blunsom. 2015. Reasoning about entailment with neural attention. *arXiv preprint arXiv:1509.06664*.
- Shane Storks, Qiaozi Gao, and Joyce Y Chai. 2019. Recent advances in natural language inference: A survey of benchmarks, resources, and approaches. *arXiv preprint arXiv:1904.01172*.
- Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. 2018. **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, Brussels, Belgium. Association for Computational Linguistics.

Shuohang Wang and Jing Jiang. 2016. [Learning natural language inference with LSTM](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1442–1451, San Diego, California. Association for Computational Linguistics.

Wei Wang, Ming Yan, and Chen Wu. 2018. [Multi-granularity hierarchical attention fusion networks for reading comprehension and question answering](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1705–1714, Melbourne, Australia. Association for Computational Linguistics.

Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral multi-perspective matching for natural language sentences. *arXiv preprint arXiv:1702.03814*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. [A broad-coverage challenge corpus for sentence understanding through inference](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Baosong Yang, Jian Li, Derek F Wong, Lidia S Chao, Xing Wang, and Zhaopeng Tu. 2019a. Context-aware self-attention networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 387–394.

Runqi Yang, Jianhai Zhang, Xing Gao, Feng Ji, and Haiqing Chen. 2019b. [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, Florence, Italy. Association for Computational Linguistics.

## A Experiment Details

**Data Statistics.** Statistics of the datasets are given in Table 4. For LCQMC, we follow the same data split as in the original work (Liu et al., 2018), and for Quora we use the same split as Wang et al. (2017).

Dataset	Train	Dev	Test	# Classes
QUORA	384K	10K	10K	2
LCQMC	239K	9K	13K	2

**Table 4:** Statistics on the datasets for experiments.

**Preprocessing.** We apply a hard cut-off of the sentence length on both datasets by cropping or

padding. Recent work has shown that character-based models typically outperform word-based models over Chinese NLP tasks (Li et al., 2019), so we apply character-based modeling for LCQMC. For Quora, we set the length as 32, and for LCQMC we set the length as 50. We mask the padding tokens during the experiments.

**Embedding Details.** For Quora, we use 300-dimensional GloVe CommonCrawl 840B word embeddings (Pennington et al., 2014) and fix the weights during training. For LCQMC, following Li et al. (2019), we avoid word segmentation and instead use a randomly initialized character embedding matrix. We set the dimensionality of character embeddings to 200, and train the weights. For sentence preprocessing, we tokenize and lowercase all words. For efficiency and more generalizable results, we do not incorporate any additional lexical features in our experiments.

**Training Details.** The kernel size is 3 for convolutional layers with padding. We apply 2 layers of convolutional encoder and 1 layer of convolutional aggregation in all experiments. We tune the dimensionality of the feed-forward layers from 150 to 300, and the number of interaction blocks from 2 to 4. The batch size is tuned from 32 to 128. Adam optimization is used with an initial learning rate of 0.001 and exponential decay. We apply ReLU (Glorot et al., 2011) as the activation function in all feed-forward networks. To prevent over-fitting, dropout with a retention probability of 0.8 is applied.

Cross-entropy serves as the loss function during training. Adam optimization is used with an initial learning rate of 0.001, and  $\beta_1$  is set as 0.9 and  $\beta_2$  as 0.999 during training. Exponential decay is also applied. Moreover, we add L2 regularization, and set the threshold for gradient clipping as 5. We apply 3 context-aware interaction blocks for Quora, and 2 interaction blocks for LCQMC. We implement our model using TensorFlow (Abadi et al., 2016) and train the models on NVIDIA Tesla V100 GPUs and NVIDIA Tesla P4 GPUs. For BERT (Devlin et al., 2019), we choose the BERT-base version (12 layers, 768 hidden dimensions and 12 attention heads), and fine-tune the model using the official implementation<sup>1</sup>. The Chinese pre-trained BERT is adopted from <https://huggingface.co/bert-base-chinese>. For SBERT (Reimers and Gurevych,

<sup>1</sup><https://github.com/google-research/bert>

2019), we utilize the original implementation<sup>2</sup>, and add a softmax classifier on top of the output of the two Transformer networks as in the original paper.

## B Model Efficiency

Models	parameter size	time (s/batch)
COIN	6.5M	0.12 $\pm$ 0.03
BERT	109.5M	1.19 $\pm$ 0.06

**Table 5:** Parameter size and inference time for COIN and BERT on Quora dataset.

Pretrained language models such as BERT (Devlin et al., 2019) have drawn much attention for their substantial gains across a range of different natural language processing tasks. However, BERT is fairly demanding in terms of the computational requirements. For additional analysis, we compare our model efficiency with BERT-base on Quora. We set the sentence lengths as 32 (64 for BERT after concatenating the two sequences). Both models need to make predictions for a batch of 8 sentence pairs on a MacBook Pro with Intel Core i7 CPUs. For BERT, we add a linear layer on top of the [CLS] token for classification, as in the original paper. We report the average and the standard deviation of processing 1,000 batches.

As shown in Table 5, COIN contains far fewer parameters than BERT and is much faster in terms of the CPU inference speed. Additionally, our single model produces comparable results to BERT on both Quora and LCQMC. This shows that our proposed method is effective at tackling text matching tasks with substantially fewer parameters and high computational efficiency.

---

<sup>2</sup><https://github.com/UKPLab/sentence-transformers>