

Enhanced Sentence Alignment Network for Efficient Short Text Matching (Supplementary Material)

Zhe Hu¹ Zuohui Fu² Cheng Peng¹ Weiwei Wang¹

¹Baidu Inc., Beijing, China

²Rutgers University, NJ, USA

¹{huzhe01, pengcheng06}@baidu.com, elegate@qq.com

²zuohui.fu@rutgers.edu

A Additional Experiment Details

Data Statistics. The statistics of datasets are shown in Table 1. For SNLI and MultiNLI, we follow the same data split as original papers (Bowman et al., 2015; Williams et al., 2018), and for Quora we use the same split as Wang et al. (2017). Notably, the test set labels of MultiNLI are not provided, and we obtain the test accuracy from submission on Kaggle¹.

Dataset	Train	Dev	Test	# Classes
SNLI	549K	9.8K	9.8K	3
Quora	384K	10K	10K	2
MultiNLI-1	392k	9.8K	9.8K	3
MultiNLI-2	392k	9.8K	9.8K	3

Table 1: Statistics on the datasets for experiments. MultiNLI-1 represents in-domain setting, and MultiNLI-2 indicates out-domain setting.

Preprocessing. We use hard cutoff for sentence length on all three datasets with cropping or padding. For Quora and SNLI, we set length as 30, and for MultiNLI we set length as 48. We mask the padding tokens during experiments. We only tokenize the sentence during preprocessing.

Training Details. We implement our model using TensorFlow (Abadi et al., 2016) and train the experiments on NVIDIA Tesla V100 GPU. cuDNN implementation for BiLSTM network is used to improve speed. For all feed-forward layers, we apply ReLU (Glorot et al., 2011) as activation function, and Adam optimizer (Kingma and Ba, 2014) is used with β_1 to be 0.9 and β_2 to be 0.999 during training. We use cropping or padding to limit each token to have 16 characters in char embedding.

¹In-domain: <https://www.kaggle.com/c/multinli-matched-open-evaluation/leaderboard>;
out-domain: <https://www.kaggle.com/c/multinli-mismatched-open-evaluation/overview>

The threshold for gradient clipping is set to 5, and l_2 regularizer strength is set to $6e-5$. Each epoch takes around 4.4 minutes with a batch size of 128 on Quora. Cross-entropy is applied as loss function during training.

B Does Feature Augmentation Improve Alignments?

To better understand how our model uses augmented features to enhance the cross-sentence alignments, we also calculate attention results using the original intermediate representations and show the visualizations in Figure 1. The two figures in the upper row are attention results computed with original intermediate representations, and the lower row shows the attention results computed with enhanced representations². The sentence 1 is “police officer with riot shield stands in front of crowd” and the sentence 2 is “a police officer stands in front of a crowd?”

As we can see, in the first alignment, computing the cross-sentence attention with original intermediate representations would bring some noisy alignments (shown in upper left). However, the attention results with enhanced representations contain less noises and the key components such as “police officer” and “crowd” are correctly aligned between two sequences (shown in lower left). In the second alignment, similar as previous, the attention with original representations are noisier and the dark cluster covers more irrelevant parts (shown in upper right). With the augmentation of original semantic features, we can observe in the lower right figure the attention is properly conducted with better connections between two sequences.

Above all, the attention results with original intermediate representations contain more noises,

²Notably here we only calculate the additional attention results with the original intermediate representations, and do not use them as inputs for the following layers.

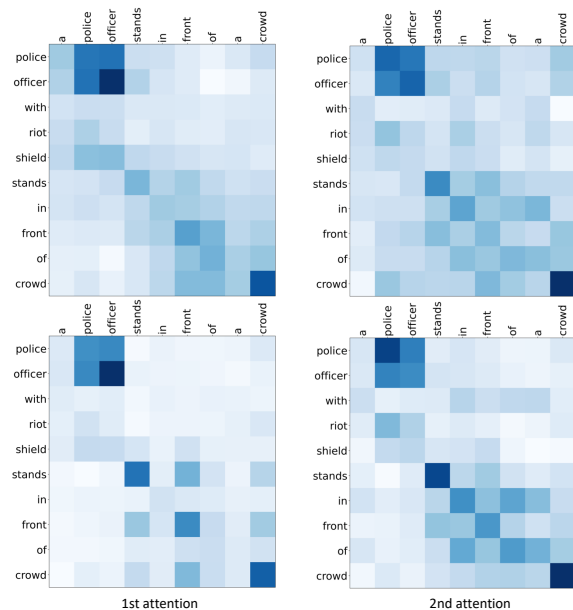


Figure 1: Visualization of attention results. The upper row are the attention computed using original intermediate representations, and the lower row are computed using enhanced sentence representations.

which would lead to incorrect alignments and unstable matching. With the augmentation of the original semantic features, the model is able to produce a proper alignments and thus better capture their semantic relationship.

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.
- Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. [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, Lisbon, Portugal. 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.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- 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.