Multi-Level Matching Networks for Text Matching

Chunlin Xu, Zhiwei Lin, Shengli Wu, Hui Wang*
Faculty of Computing, Engineering and Built Environment, Ulster University
{xu-c,z.lin,s.wu1,h.wang}@ulster.ac.uk

ABSTRACT

Text matching aims to establish the matching relationship between two texts. It is an important operation in some information retrieval related tasks such as question duplicate detection, question answering, and dialog systems. Bidirectional long short term memory (BiLSTM) coupled with attention mechanism has achieved state-of-the-art performance in text matching. A major limitation of existing works is that only high level contextualized word representations are utilized to obtain word level matching results without considering other levels of word representations, thus resulting in incorrect matching decisions for cases where two words with different meanings are very close in high level contextualized word representation space. Therefore, instead of making decisions utilizing single level word representations, a multi-level matching network (MMN) is proposed in this paper for text matching, which utilizes multiple levels of word representations to obtain multiple word level matching results for final text level matching decision. Experimental results on two widely used benchmarks, SNLI and Scaitail, show that the proposed MMN achieves the state-of-the-art performance.

KEYWORDS

text matching, attention, multi-level matching network

ACM Reference Format:
Chunlin Xu, Zhiwei Lin, Shengli Wu, Hui Wang. 2019. Multi-Level Matching Networks for Text Matching . In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '19), July 21–25, 2019, Paris, France.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3331184.3331276

1 INTRODUCTION

Text matching is to compare two texts in order to establish their relationship – such as whether the two texts share the same meaning or not. It is an important operation in some information retrieval related tasks such as question duplicate detection [16], question answering [13], and dialog systems [15].

Deep learning has been widely applied to text matching in recent years [1, 2, 9] and the existing deep models for text matching can be categorized into two approaches. The first approach models two texts by encoding each text separately and then predicts their relationship based on the two extracted representations, taking no account of interaction between two texts [1, 3]. The second approach is based on the attention mechanism [2, 4, 14], in which words in two texts are matched firstly. Then these word level matching results are aggregated into a fixed-size vector for making final text level matching decision. Combining with bidirectional long short term memory (BiLSTM) [6], previous models based on attention mechanisms have achieved the state-of-the-art performances for text matching [2, 14].

However, the problem with the second approach lies in the fact that previous models only use the final representaions of words to obtain the word level matching results for text level matching decision without considering other levels of word representations. For example, the state-of-the-art model ESIM [2], firstly uses the low level pre-trained word embeddings [10] as inputs to a BiLSTM layer to generate high level contextualized word representations for representing words and their contextual information. Then an attention mechanism is employed to conduct word level matching solely based on high level contextualized word representations without considering low level representations, which can not capture sufficient information for modeling complex matching relations. For example, obviously, "I went to London yesterday" and "I went to Beijing yesterday" have different meanings and they should not be matched. However, because the contextual information of 'London' and 'Beijing' are very similar, the high level contextualized representations of these two words generated by BiLSTM layer will be very close in word representation space, which may not be sufficient to differentiate the two words, thus leading to incorrect matching decision. If the low-level word embeddings of these two words, which may be far from each other in embedding space, are also considered, the model would be aware of the difference between words 'London' and 'Beijing', which is helpful for making correct matching decision. Therefore, it is important to have multiple levels of matching to capture more matching information, hence yielding correct matching decision.

In order to address the above limitation, this paper presents a multi-level matching network (MMN) for text matching, which utilizes multiple levels of word representations to obtain multiple word level matching results for final text level matching decision. In each matching level, an attention mechanism is firstly used to learn the attention-aware representation of each word in two texts and to make word level matching at current level. Next, a fusion gate is used to combine the attention-aware representation with original representation of each word for word representation refinement. Then, a BiLSTM encoder is employed to generate new word representations which will be used as the inputs for next matching level. The above process is repeated for k times. Finally, the matching results of k matching levels are aggregated for final decision. The contributions of this paper are summarized as follows:

*Corresponding Author

SIGIR '19, July 21–25, 2019, Paris, France

ACM ISBN 978-1-4503-6172-9/19/07…$15.00
https://doi.org/10.1145/3331184.3331276

- A new multi-level matching network (MMN) is proposed for text matching. The model can capture more matching information by utilizing multiple levels of word representations.
- An attention aware representation fusion (AARF) layer is devised to refine word representaions in each matching level.
- The model is evaluated on two popular benchmarks, SNLI and Scaitail. Experimental results show that the model outperforms state-of-the-art baselines.

2 THE MMN MODEL

The overall framework of the proposed MMN model is shown in Fig. 1. It consists of five layers:

(1) the input layer for a pair of texts;
(2) the word embedding layer for representing each word in the two texts as a vector;
(3) the multi-level matching layer whose architecture is shown in Fig. 2;
(4) the aggregation layer, where matching results from all matching levels are aggregated into a fixed-size vector;
(5) the prediction layer.

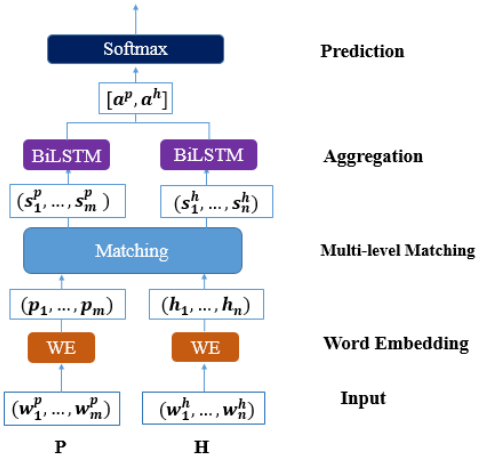

Figure 1: The framework of the proposed MMN.

2.1 Word Embedding Layer

Given a pair of texts $P = (w_1^p, \ldots, w_m^p)$ with m words and $H = (w_1^h, \ldots, w_n^h)$ with n words, the purpose of this layer is to convert each word in text P and text H into a d-dimensional vector denoted as $p_i \in \mathbb{R}^d$ and $h_j \in \mathbb{R}^d$. The d-dimensional column vector is composed of two parts: a word-level embedding and a character-level embedding. The word-level embedding is obtained from a pre-trained word embedding matrix Glove [10]. Then we feed each character within a word into a BiLSTM, and the last time-step output of the BiLSTM is used as the character-level embedding, which is the same as [14].

2.2 Multi-level Matching Layer

This layer obtains the matching results based on different levels of word representations. During the k-th matching level, given the representations of two texts P and H computed in the previous matching level: $(p_1^k, \ldots, p_m^k)$ and $(h_1^k, \ldots, h_n^k)$. A word-by-word matching layer is firstly employed to obtain the word level matching results at current level.

Word-by-Word Matching. To perform word level matching, co-attention matrix $A^k = (\alpha_{ij}^k)_{m \times n}$ between two texts are computed by the following equations:

$$\alpha_{ij}^k = {p_i^k}^T \cdot h_j^k \tag{1}$$

where α_{ij}^k indicates the relevance between the i-th word p_i^k of text P and j-th word h_j^k of text H, and $\alpha_{ji}^k = {\alpha_{ij}^k}^T$ otherwise. Next, for each word in one text, the relevant semantics in the other text is extracted and composed based on the co-attention matrices α_{ij}^k and α_{ji}^k by the following equations:

$$\overline{p}_i^k = \sum_{j=1}^{n} \frac{\exp\left(\alpha_{ij}^k\right)}{\sum_{r=1}^{n} \exp\left(\alpha_{ir}^k\right)} h_j^k \tag{2}$$

$$\overline{h}_j^k = \sum_{i=1}^{m} \frac{\exp\left(\alpha_{ji}^k\right)}{\sum_{r=1}^{m} \exp\left(\alpha_{rj}^k\right)} p_i^k \tag{3}$$

where $\overline{p}_i^k$ and $\overline{h}_j^k$ are the attention-aware representations of p_i^k and h_j^k, representing the contents in $\{h_j^k\}_{j=1}^n$ related to p_i^k and the contents in $\{p_i^k\}_{i=1}^m$ related to h_j^k, respectively. Then we use a vector matching function on each pair of $< p_i^k, \overline{p}_i^k >$ and $< h_j^k, \overline{h}_j^k >$ to obtain the word level matching results at current level between two texts.

$$t_{ki}^p = p_i^k \odot \overline{p}_i^k \tag{4}$$

$$t_{kj}^h = h_j^k \odot \overline{h}_j^k \tag{5}$$

where $\odot$ is the element-wise product operation, t_{ki}^p and t_{kj}^h are the matching results at k-th level of matching P against H and matching H against P, respectively.

Attention Aware Representaion Fusion (AAFR). This layer is utilized to refine the word representation vectors. In this paper, a fusion gate is used to incorporate the attention-aware representations into original representations of each word in two texts for word representation refinement.

$$F_p = sigmoid(\boldsymbol{W_{p1}} p_i^k + \boldsymbol{W_{p2}} \overline{p}_i^k + b_p) \tag{6}$$

$$F_h = sigmoid(\boldsymbol{W_{h1}} h_j^k + \boldsymbol{W_{h2}} \overline{h}_j^k + b_h) \tag{7}$$

$$\widetilde{p}_i^k = F_p \odot p_i^k + (1 - F_p) \odot \overline{p}_i^k \tag{8}$$

$$\widetilde{h}_j^k = F_h \odot h_j^k + (1 - F_h) \odot \overline{h}_j^k \tag{9}$$

where $\boldsymbol{W_{p1}}, \boldsymbol{W_{p2}}, \boldsymbol{W_{h1}}, \boldsymbol{W_{h2}} \in \mathbb{R}^{d_l \times d_l}$ and $b_p, b_h \in \mathbb{R}^{d_l}$ are the learable parameters of the fusion gate. $\odot$ is the element-wise product operation. $\widetilde{p}_i^k, \widetilde{h}_j^k \in \mathbb{R}^{d_l}$ are the refined word representation vectors after fusion.

BiLSTM Encoder. Next, a BiLSTM encoder layer is employed to encode the contexutal information into the above refined representation vectors to generate new word representations.

$$p_i^{k+1} = BiLSTM(\widetilde{p}_i^k, p_{i-1}^{k+1}, p_{i+1}^{k+1}) \tag{10}$$

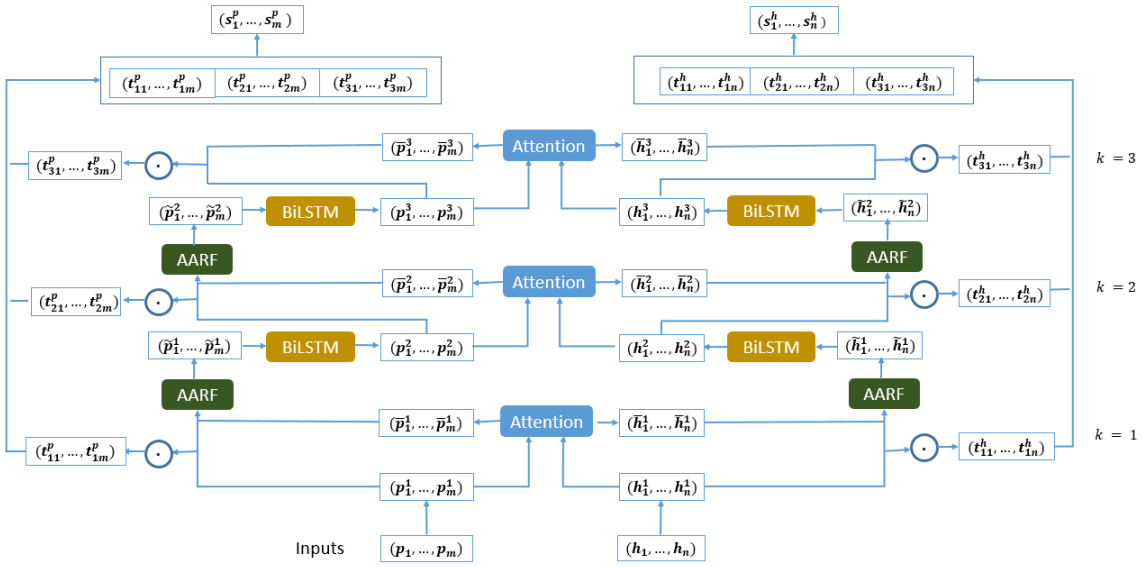

Figure 2: Architecture of multi-level matching layer (taking $k = 3$ as an example).

$$h_j^{k+1} = BiLSTM(\widetilde{h}_j^k, h_{j-1}^{k+1}, h_{j+1}^{k+1}) \tag{11}$$

where $p_i^{k+1}, h_j^{k+1} \in \mathbb{R}^{d_l}$ will be used as the inputs for next matching level.

Finally, The matching results from all matching levels are concatenated and used as the outputs of the multi-level matching layer.

$$s_i^p = [t_{1i}^p; \ldots; t_{ki}^p] \tag{12}$$

$$s_j^h = [t_{1j}^h; \ldots; t_{kj}^h] \tag{13}$$

where $s_i^p, s_j^h \in \mathbb{R}^{kd_l}$ are the concatenated matching results of matching P against H and matching H against P, respectively.

2.3 Aggregation and Prediction Layer

To aggregate all the matching results into a fixed-size vector, we pass the above concatenated matching results into another BiLSTM.

$$u_i^p = BiLSTM(s_i^p, u_{i-1}^p, u_{i+1}^p) \tag{14}$$

$$u_j^h = BiLSTM(s_i^h, u_{j-1}^h, u_{j+1}^h) \tag{15}$$

Then a mean pooling method is used to obtain the fixed-size vectors.

$$a^p = \frac{1}{m}\sum_{i=1}^{m} u_i^p \tag{16}$$

$$a^h = \frac{1}{n}\sum_{j=1}^{n} u_j^h \tag{17}$$

Finally, the above fixed-size vectors a^p and a^h are concatenated and then passed to a MLP classifier which includes a *tanh* activation and *softmax* output layer to obtain the final prediction.

3 EXPERIMENTS AND RESULTS

3.1 Dataset and Experimental Setup

We evaluate our model on two datasets: SNLI [1], and SciTail dataset [7]. SNLI contains over 570K human annotated sentence pairs, each labeled with one of the following relationships: *entailment, contradiction, neutral*. Scitail is constructed from science domain, which contains about 27K sentence pairs. Unlike the SNLI dataset, SciTail uses only two labels: *entailment, neutral*.

In this paper, word embeddings are initialized with the 300d GloVe word vectors [10]. The dimensions of the BiLSTM encoders are set as 400 in multi-level matching layer and 600 for aggregation layer. The number of aggregation BiLSTM layers is set as 2. The number of matching levels is tuned from [1, 4]. Batch sizes are 32 for Scitail dataset and 128 for SNLI dataset. The Adam optimizer [8] is used for training, and the initial learning rate is set as 0.001. To avoid overfitting, we apply dropout to all layers of the model and the dropout ratio is set as 0.2 .

3.2 Experimental Results

The accuracy metric is used to evaluate the performance of the proposed MMN and baseline models on datasets SNLI and Scitail. The performance of all baseline models come from respective papers.

SNLI. Table 1 shows the results of different models on the training and test sets of SNLI. DecompAtt [9] divides the text matching task into several sub-tasks using soft attention. BiMPM [14] performs matching at multi-perspective and two directions. ESIM [2] enhances the local inference procedure and achieves the state-of-the-art performance. DIIN [5] and CIN[4] are two advanced models based on CNN. From Table 1 we can see that the proposed MMN achieves an accuracy of 88.2% in the test sets, which outperforms all the baselines and achieves the state-of-the-art performance.

Table 1: Performances on the SNLI dataset

Model	Train	Test
DecompAtt [9]	89.5	86.3
BiMPM [14]	90.9	87.5
ESIM [2]	92.6	88.0
DIIN [5]	91.2	88.0
CIN[4]	93.2	88.0
MMN	89.3	**88.2**

Scitail. Table 2 shows results of the proposed MMN model and baselines on the SciTail dataset. DGEM is the decomposed graph entailment model proposed in [7]. HCRN [12] obtain the attention matrix using the complex-valued inner product (Hermitian products). CAFE [11] utilizes the word level matching results for augmentation of the base word representation instead of aggregating them for prediction. From Table 2 we can see our model MMN outperforms all baselines and achieves the state-of-the-art performance with an accuracy of 84.8%. Comparing with ESIM which achieves the state-of-the-art performance on SNLI dataset, the proposed MMN outperforms it by a large margin over 14%.

Table 2: Performances on the Scitail dataset

Model	Accuary
DecompAtt[9]	72.3
ESIM[2]	70.6
DGEM[7]	77.3
HCRN [12]	80.0
CAFE [11]	83.3
MMN	**84.8**

3.3 Ablation Study

We performe an ablation study on the MMN model to examine the effectiveness of each major component. Table 3 shows the ablation study results on SciTail and SNLI datasets. First, if we remove the attention aware representation fusion (AARF) layer from the model, the performance of the model has dropped slightly on two datasets, from 84.8% to 84.24% on Scaitail, and from 88.2% to 87.9% on SNLI. This indicates the AARF layer is helpful for improving the performance of the model. Second, if we do not use multiple levels of word representations (only use the final word representaions to get word level matching results), the accuracy drops by over 1% on both datasets. According to the results, all of the components are effective for performance improvement.

Table 3: Ablation study on SciTail and SNLI datasets

Model	Scitail	SNLI
MMN	84.8	88.2
- AARF	84.24	87.9
- Multi-level matching	83.34	87.8

3.4 Effect of Number of Matching levels

Fig. 3 shows the effect of number of matching levels on SNLI and Scitail datasets. We observe that the optimal performance is 3 matching levels for SNLI. However, the performance of SNLI declines after 3 matching levels. Similarly, Scitail achieves its best performance at level 2 and then declines after 2 matching levels.

4 CONCLUSION

In this paper, we have presented a novel multi-level matching network (MMN) for text matching, which obtains word level matching results based on multiple levels of word representations to capture more matching information. The MMN model achieves the state-of-the-art performance on two datasets: SNLI and Scitail. In future work, we will extend the proposed MMN to address other tasks, such as question answering and machine reading comprehension.

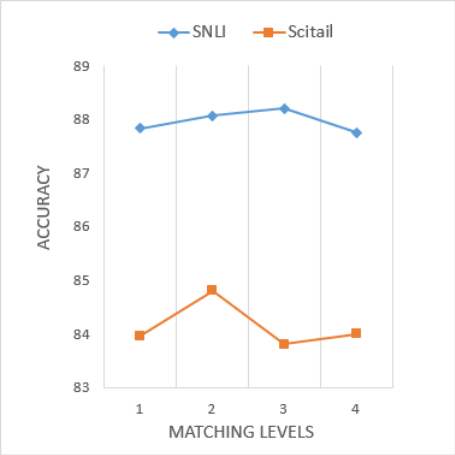

Figure 3: Effect of number of matching levels.

5 ACKNOWLEDGMENTS

This work is partially funded by the EU Horizon 2020 under Grant 690238 for DESIREE Project, under Grant 700381 for ASGARD project, by the UK EPSRC under Grant EP/P031668/1.

REFERENCES

[1] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *EMNLP*. 632–642.
[2] Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced LSTM for Natural Language Inference. In *ACL*, Vol. 1. 1657–1668.
[3] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. 2017. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. In *EMNLP*. 670–680.
[4] Jingjing Gong, Xipeng Qiu, Xinchi Chen, Dong Liang, and Xuanjing Huang. 2018. Convolutional Interaction Network for Natural Language Inference. In *EMNLP*. 1576–1585.
[5] Yichen Gong, Heng Luo, and Jian Zhang. 2017. Natural language inference over interaction space. *ICLR*.
[6] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.
[7] Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. SciTail: A textual entailment dataset from science question answering. In *AAAI*.
[8] Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.
[9] Ankur P Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A Decomposable Attention Model for Natural Language Inference. In *EMNLP*. 2249–2255.
[10] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*. 1532–1543.
[11] 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 *EMNLP*.
[12] Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. 2018. Hermitian Co-Attention Networks for Text Matching in Asymmetrical Domains. In *IJCAI*.
[13] Nam Khanh Tran and Claudia Niedereée. 2018. Multihop Attention Networks for Question Answer Matching. In *SIGIR*. 325–334.
[14] Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral multi-perspective matching for natural language sentences. In *IJCAI*. 4144–4150.
[15] Liu Yang, Minghui Qiu, Chen Qu, Jiafeng Guo, Yongfeng Zhang, W Bruce Croft, Jun Huang, and Haiqing Chen. 2018. Response ranking with deep matching networks and external knowledge in information-seeking conversation systems. In *SIGIR*. 245–254.
[16] Wei Emma Zhang, Quan Z Sheng, Zhejun Tang, and Wenjie Ruan. 2018. Related or Duplicate: Distinguishing Similar CQA Questions via Convolutional Neural Networks. In *SIGIR*. 1153–1156.