Improving the Robustness of Deep Reading Comprehension Models by Leveraging Syntax Prior

Bowen Wu[1], **Haoyang Huang**[2], **Zongsheng Wang**[1], **Qihang Feng**[1],
Jingsong Yu[2], **Baoxun Wang**[1]
[1]Platform and Content Group, Tencent
[2]School of Software & Microelectronics, Peking University, Beijing, China
jasonbwwu, jasoawang, careyfeng, asulewang@tencent.com
huanghaoyang@pku.edu.cn, yjs@ss.pku.edu.cn

Abstract

Despite the remarkable progress on Machine Reading Comprehension (MRC) with the help of open-source datasets, recent studies indicate that most of the current MRC systems unfortunately suffer from weak robustness against adversarial samples. To address this issue, we attempt to take sentence syntax as the leverage in the answer predicting process which previously only takes account of phrase-level semantics. Furthermore, to better utilize the sentence syntax and improve the robustness, we propose a Syntactic Leveraging Network, which is designed to deal with adversarial samples by exploiting the syntactic elements of a question. The experiment results indicate that our method is promising for improving the generalization and robustness of MRC models against the influence of adversarial samples, with performance well-maintained.

1 Introduction

As one of the ultimate goals of natural language processing, Machine Reading Comprehension (MRC) has been attracting much attention from both the academical and industrial institutions (Richardson et al., 2013; Hermann et al., 2015). Recently, most of the outstanding studies have benefited from the rapid development of machine reading competitions with shared datasets, such as SQuAD (Rajpurkar et al., 2016), MS MARCO (Nguyen et al., 2017). According to the competition results, the Deep Learning based approaches have shown significant strength on MRC tasks and achieved most of the top-ranked positions (Wang et al., 2017; Yu et al., 2018).

Nevertheless, the very recent research in MRC indicates that simply chasing the performance improvement on given datasets is unwise, since the generalization and robustness might be weakened due to the great fitting capability of DL models trained on a specific corpus. Especially, the research on adversarial reading comprehension samples conducted by Jia and Liang (2017) has shown that the performances of most of the DL based MRC models decrease significantly on the adversarial samples. These adversarial samples are constructed by simply appending one sentence similar to the question into the paragraph, without changing the original answer. This work indicates that, apparently, there exists quite a gap between the current MRC approaches and the methodologies that really comprehend natural language passages.

In this paper, we attempt to face the challenge brought by the RC adversarial samples and aim at proposing a reading comprehension system with better generalization and robustness. For this purpose, this paper presents a method to improve the answer inferencing process of MRC, by leveraging the probability function for estimating answer using the information related to sentence-question matching. Moreover, to further improve the robustness of the MRC system, we propose a novel model named syntactic leveraging network which exploits the syntax of the question as the prior information to match the answer-contained sentence and question more precisely.

2 Methodology

Most existent MRC methods predict answers by calculating probabilities of answer spans (i, j). For an answer a starts at position i, ends at j and locates in sentence k, we denote it as $a = \{i, j, k\}$. Given a question $\mathbf{q}$ and a paragraph $\mathbf{p}$, the probability of a is computed by:

$$p(a|\mathbf{q},\mathbf{p}) = p_s(i|\mathbf{q},\mathbf{p}) \cdot p_e(j|\mathbf{q},\mathbf{p}) \qquad (1)$$

and:

$$\begin{aligned} p_s(i|\mathbf{q},\mathbf{p}) &= f_s(i|\mathbf{q},\mathbf{p}) \\ p_e(j|\mathbf{q},\mathbf{p}) &= f_e(j|\mathbf{q},\mathbf{p}) \end{aligned} \qquad (2)$$

Proceedings of the Second Workshop on Machine Reading for Question Answering, pages 53–57
Hong Kong, China, November 4, 2019. ©2019 Association for Computational Linguistics

Here functions f_s and f_e are usually implemented by neural networks to predict the probabilities.

In most non-inferencing machine reading comprehension datasets such as SQuAD, all information needed to identify answers can be found inside one single sentence (Raiman and Miller, 2017). In such datasets, given one question and one phrase inside a sentence, overall whether this phrase is the answer depends on two conditions: 1) if the phrase itself generally matches with the question; 2) if the syntactic elements in the sentence are precisely consistent with the syntactic elements in the question.

However, the experiment results in Jia and Liang (2017) have shown that the current MRC systems pay less attention to the second condition, thus can be easily attacked by question-related sentences as adversarial samples. We attribute this deficiency to the fact that the current models solely takes the phrase-level information into account when predicting the probability $p(a|\mathbf{q}, \mathbf{p})$, but fails to exploit the sentence-level matching between the answer-contained sentence and the question, which is of importance on evaluating the second condition. Consequently, we propose a new probability function for estimating answers by considering the sentence level matching degree:

$$\begin{aligned} p^*(a|\mathbf{q}, \mathbf{p}) &= p_s(i|\mathbf{q}, \mathbf{p}) \cdot p_e(j|\mathbf{q}, \mathbf{p}) \cdot p_{sent}(k|\mathbf{q}, \mathbf{p})^\alpha \\ p_{sent}(k|\mathbf{q}, \mathbf{p}) &= f_{sent}(\mathbf{q}, s_k) \end{aligned} \tag{3}$$

where s_k is the $k-th$ sentence in p. In general, p_{sent} predicts if the answer a presents in the $k-th$ sentence from the paragraph, it captures the matching between sentence and question as a leverage to improve the system robustness. α is the leveraging factor for $p_{sent}(k|\mathbf{q}, \mathbf{p})$.

2.1 Syntactic Leveraging Network

Although theoretically f_{sent} can be implemented by any model aiming at evaluating the matching between two sentences, to correctly identify real answer-contained sentences from semantically-closed adversarial sentences, it is necessary to come up with a model which is capable of precisely extracting and comparing the syntactic elements within sentences and questions. Therefore Syntactic Leveraging Network (SLN) is proposed to predict $p_{sent}(k|\mathbf{q}, \mathbf{p})$, so as to improve the robustness of MRC models. The structure of SLN is shown in Figure 1, which consists of the **SRL (Semantic role labeling) extractor**, the **CNN encoder**, the **Matching operator** performing optimal transport (Tam et al., 2019) and a classifier.

2.1.1 SRL Extractor

We utilize SRL (Gildea and Jurafsky, 2002; Khashabi et al., 2018) to analyze the syntax of sentences as prior information. In brief, it automatically produces syntactic analyses by exploiting generalizations from syntax-semantics links and assigns labels to phrases in a sentence based on their syntactic roles.

Given a question $\mathbf{q}$, the SRL extractor separates $\mathbf{q}$ into a sequence of phrases Q, specifically:

$$Q = \text{SRL}(\mathbf{q}) = [q_1, q_2, \ldots, q_n] \tag{4}$$

with corresponding lengths $L = [l_1, l_2, \ldots, l_n]$. Here each q_i represents one syntactic element within the $\mathbf{q}$, and each can also be considered as a condition that answer-contained sentences must satisfy. The SLN model takes such sequence of n-grams as inputs to represent the question.

2.1.2 CNN Encoder

The encoder projects the syntactic elements in Q and s into real-valued vectors. Assuming CNN's filter windows range from w_{min} to w_{max} with each kernel size of k. For q_i in Q, it is only transferred into the filter window of size l_i in CNN:

$$q_i^v = CNN_{l_i}(q_i) \qquad i \in [1, n] \tag{5}$$

This CNN is performed following Kim (2014), so that the size of each q_i^v equals to the kernel size k.

For sentence s of length L, it is first split into m[1] separate phrases $[s_1, s_2, \ldots, s_m]$, which contains all n-grams ($w_{min} \leq n < w_{max}$) in the sentence. Then, each s_i is transferred into s_j^v of size k through CNN filters, such that:

$$\mathbf{s}^v = [s_1^v, s_2^v, \ldots, s_m^v] \tag{6}$$

where s_j^v and q_i^v represent pieces of semantics in the sentence and question.

2.1.3 Matching Operator

The matching operator is designed to evaluate if the sentence generally matches with the syntactic elements of the question. It first computes the cosine-similarity between each q_i^v and s_j^v, which gives a similarity matrix $\mathbf{S} \in \mathbb{R}^{n \times m}$. Then we implement the max pooling across the row of $\mathbf{S}$ to obtain $\mathbf{q}^{sim}$:

$$\mathbf{q}^{sim} = max_{row}(\mathbf{S}) = [q_1^{sim}, q_2^{sim}, \ldots, q_n^{sim}] \tag{7}$$

[1] $m = (w_{max} - w_{min} + 1) * L - \sum_{i=min}^{max}(w_i - 1)$

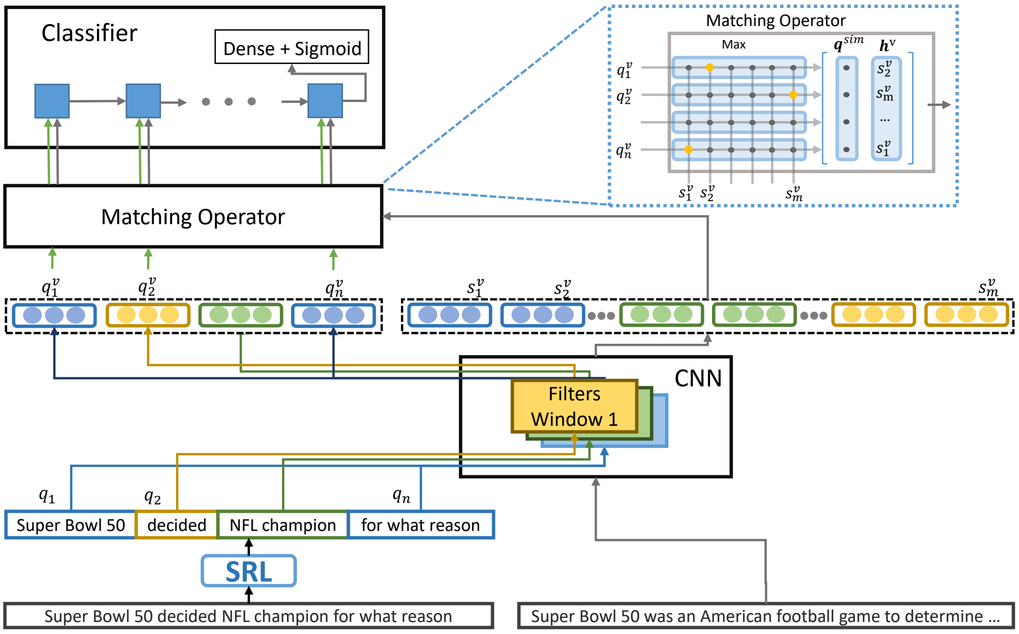

Figure 1: The Architecture of Syntactic Leveraging Network

The value of each q_i^{sim} varies from 0 to 1, which indicates the degree of similarity of each syntactic element q_i in s. Meanwhile, q_i^{sim} equals to 1 if the syntactic element q_i exist exactly in the s, which is a significant signal for the element matching.

Furthermore, given $\mathbf{S}$, for each q_i^v we compute its corresponding h_i^v. Specifically:

$$\begin{aligned} h_i^v &= [s^v_{\arg\max_j S_{ij}}; q_i^{sim}] \\ \mathbf{h}^v &= [h_1^v, h_2^v, \ldots, h_n^v] \end{aligned} \tag{8}$$

where $s^v_{\arg\max_j S_{ij}}$ is the vector representation of the most semantically-similar phase in the sentence given q_i^v, and q_i^{sim} represents the degree of similarity. Overall, h_i^v represents the most matched phase in the sentence for one syntactic element in the question and its corresponding degree of matching. Finally, $\mathbf{h}^v$ is transferred from the Matching Operator as the output.

2.1.4 Classifier

The final classifier of SLN is designed to predict if the sentence matches with the question. It first concatenates the outputs h_i^v from the matching operator with q_i^v as the LSTM inputs, such that:

$$c_i = LSTM(c_{i-1}, [h_i; q_i]) \tag{9}$$

The last LSTM hidden states c_n is then transferred into a dense layer followed by a sigmoid activation function, and binary cross-entropy is adopted as the loss function.

3 Experiments

3.1 Experimental Setups

Data Description. We implement our method on several end-to-end MRC models trained by SQuAD dataset, and evaluate their robustness before and after considering $p_{sent}(k|\mathbf{q}, \mathbf{p})$ using the AddSent adversarial dataset (Jia and Liang, 2017). The training and test sets for MRC models are generated from SQuAD. To compute $p_{sent}(k|\mathbf{q}, \mathbf{p})$, we set those answer-contained sentences in SQuAD as positive samples. For each positive sample, three sentences inside the same paragraph which do not contain answer are randomly chosen as negative samples, so that the positive/negative ratio is 1:3. All sentence-level matching models are trained on above samples as a binary-classification task using cross-entropy loss.

Baselines. Besides of SLN, we use relevance-LSTM and Inner-Attention (Liu et al., 2016) as baselines to compute $f_{sent}(\mathbf{q}, s_k)$. Relevance-LSTM simply takes the last hidden states of the sentence and question for similarity computation, which is also used in the MRC model of Raiman and Miller (2017); while Inner-Attention is the abbreviation for the Bidirectional LSTM encoders with intra-attention, it utilizes the sentence's representation to attend words appearing in itself. BiDAF (Seo et al., 2017) and MneReader (Hu et al., 2017) are chosen as the back-end MRC models, and the results are obtained by our Keras

	SQuAD		AddSent	
	EM	F1	EM	F1
BiDAF				
original	67.7	77.4	26.4	34.2
+Relevance-LSTM	67.8	77.6	26.4	34.2
+Inner-Attention	**68.0**	**77.9**	27.4	35.4
+SLN	67.7	77.5	**28.4**	**36.4**
MneReader				
original	71.1	80.6	36.3	44.7
+Relevance-LSTM	70.8	80.1	36.1	44.3
+Inner-Attention	**71.2**	**80.7**	37.4	46.0
+SLN	70.9	80.3	**37.9**	**46.7**

Table 1: Results on the MRC datasets

	Accuracy	P@1
Random Guess	75.0%	25.0%
Relevance-LSTM	83.2%	80.1%
Inner-Attention	**87.8%**	**86.2%**
SLN	85.6%	82.7%

Table 2: Results on Sentence Matching

implementation (Chollet et al., 2015).

Parameter Settings. For SLN, we utilize the AllenNLP to perform SRL (Gardner et al., 2017), the filter windows are set from 1 to 8, with each kernel size of 128. The hidden size of LSTM is set as 128, while the size of the dense layer is set as 64. Adam (Kingma and Ba, 2014) with learning rate 0.001 is used to optimize SLN, the batch size is set as 8 and the models are trained for 50 epochs, with the early stop when the loss on validation set starts to drop. Dropout rate is set to 0.2 to prevent overfitting (Srivastava et al., 2014). We utilize the pretrained 100-dim GloVe embeddings for all the models and set it as untrainable during training (Pennington et al., 2014). The leveraging factor α are all set as 0.25 for relevance-LSTM, Inner-Attention, and SLN.

For BiDAF and MneReader as back-end MRC models, we follow the exact hyperparameter settings of (Seo et al., 2017; Hu et al., 2017).

3.2 Results of the MRC Task

Table 1 details the performances of models on MRC datasets. The results show that both the performances of BiDAF and MneReader drop significantly on the adversarial dataset, which indicates that current MRC models are not robust enough to distinguish the semantically similar candidates from answers. Concerning robustness, both Inner-Attention and SLN improve the EM and F1 of BiDAF and MneReader on AddSent dataset. This shows evidence that the robustness of MRC models can be improved by properly exploiting the sentence-level matching information. It can be also observed that introducing the sentence-level matching into the models overall is not detrimental to the performances of models on the regular dataset, and the Inner-Attention even slightly increases the EM and F1 on regular SQuAD.

By contrary, Relevance-LSTM fails to improve the performance of current MRC models. We attribute this phenomenon to two reasons: 1) Relevance-LSTM mainly focuses on the semantics of the whole sentence to evaluate the relevance of two sentences, but current MRC models have already captured this information; 2) The word-level or phrase-level correspondence is important in identifying whether two sentences are talking about the same thing, which is also omitted in current End-to-End metric-oriented MRC models.

3.3 Analysis on Sentence Matching

The results of the sentence matching are shown in Table 2. It can be observed that Inner-Attention achieves the best performance. We attribute its high performance to the fact that its attention mechanism helps to capture the semantics clues on detecting answer-related sentences given the question. However, although the Inner-Attention outperforms SLN significantly on sentence matching, the results on Adversarial dataset show that SLN is more effective on robustness-promoting, reflected by the highest EM and F1 achieved by SLN on AddSent. Since most current MRC models have already modeled the high-level semantics in the sentences sufficiently, the attention mechanism in inner-attention might be redundant thus less effective in identifying the adversarial samples. The performance of SLN on robustness-promotion further verifies our hypothesis that introducing the syntax information as leverage on answer prediction is a feasible way to enhance the robustness of MRC systems.

4 Conclusions

In this paper, we exploit the usage of sentence-level information, especially sentence syntax as leverage, on machine reading comprehension task. The experiment results show such approach is capable of improving the robustness of MRC systems against adversarial samples, with the performance on regular datasets well maintained, although currently, the improvements on robustness are relatively moderate.

References

François Chollet et al. 2015. Keras.

Matthew Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew E. Peters, Michael Schmitz, and Luke S. Zettlemoyer. 2017. Allennlp: A deep semantic natural language processing platform. *CoRR*, abs/1803.07640.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in neural information processing systems*, pages 1693–1701.

Minghao Hu, Yuxing Peng, and Xipeng Qiu. 2017. Reinforced mnemonic reader for machine comprehension. *arXiv preprint arXiv:1705.02798*.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031.

Daniel Khashabi, Ashish Sabharwal, Tushar Khot, and Dan Roth. 2018. Question answering as global reasoning over semantic abstractions. In *AAAI-18 AAAI Conference on Artificial Intelligence*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Yang Liu, Chengjie Sun, Lei Lin, and Xiaolong Wang. 2016. Learning natural language inference using bidirectional lstm model and inner-attention. *arXiv preprint arXiv:1605.09090*.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2017. Ms marco: A human-generated machine reading comprehension dataset. *neural information processing systems*.

Jeffrey Pennington, Richard Socher, and Christopher D. 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.

Jonathan Raiman and John Miller. 2017. Globally normalized reader. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1059–1069.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Matthew Richardson, Christopher JC Burges, and Erin Renshaw. 2013. Mctest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 193–203.

Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. *international conference on learning representations*.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

Derek Tam, Nicholas Monath, Ari Kobren, Aaron Traylor, Rajarshi Das, and Andrew McCallum. 2019. Optimal transport-based alignment of learned character representations for string similarity. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5907–5917.

Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. 2017. Gated self-matching networks for reading comprehension and question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 189–198. Association for Computational Linguistics.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V Le. 2018. Qanet: Combining local convolution with global self-attention for reading comprehension. *arXiv preprint arXiv:1804.09541*.