Tackling Adversarial Examples in QA via Answer Sentence Selection

Yuanhang Ren[1], **Ye Du**[2], **Di Wang**[2,3]
[1]University of Electronic Science and Technology of China
[2]Southwestern University of Finance and Economics, China
[3]Vector Lab, JD Finance, Beijing, China
{ryuanhang,henry.duye,albertwang0921}@gmail.com

Abstract

Question answering systems deteriorate dramatically in the presence of adversarial sentences in articles. According to Jia and Liang (2017), the single BiDAF system (Seo et al., 2016) only achieves an F1 score of 4.8 on the ADDANY adversarial dataset. In this paper, we present a method to tackle this problem via answer sentence selection. Given a paragraph of an article and a corresponding query, instead of directly feeding the whole paragraph to the single BiDAF system, a sentence that most likely contains the answer to the query is first selected, which is done via a deep neural network based on Tree-LSTM (Tai et al., 2015). Experiments on ADDANY adversarial dataset validate the effectiveness of our method. The F1 score has been improved to 52.3.

1 Introduction

Question answering is an important task in evaluating the ability of language understanding of machines. Usually, given a paragraph and a corresponding question, a question answering system is supposed to generate the answer of this question from the paragraph. By comparing the predicted answer with human-approved answers, the performance of the system can be assessed. Recently, many systems have achieved great results on this task (Shen et al., 2017b; Wang and Jiang, 2016; Hu et al., 2017). However, Jia and Liang (2017) show that these systems are very vulnerable to paragraphs with adversarial sentences. For instance, the single BiDAF system (Seo et al., 2016), which achieves an F1 of 75.5 on Standford Question Answering Dataset (SQuAD), deteriorates significantly to an F1 of 4.8 on the ADDANY adversarial dataset. Besides the single BiDAF, the single Match LSTM, the ensemble Match LSTM, and the ensemble BiDAF achieve an F1 of 7.6, 11.7, and 2.7 respectively in question answering on ADDANY adversarial dataset (Jia and Liang, 2017). Therefore, question answering with adversarial sentences in paragraphs is a prominent issue and is the focus of this study.

In this paper, we propose a method to improve the performance of the single BiDAF system[1] on ADDANY adversarial dataset. Given a paragraph and a corresponding question, our method works in two steps to generate an answer. In the first step, a deep neural network named the QA Likelihood neural network is deployed to predict the likelihood of each sentence in the paragraph to be an *answer sentence*, i.e., the sentence that contains the answer. The architecture and the loss of the QA Likelihood neural network follow the neural network for semantic relatedness proposed by Tai et al. (2015). Its main ingredient is the Tree-LSTM model. While the neural network for semantic relatedness is used to predict the similarity between sentence A and B, the QA Likelihood neural network is used to predict if sentence A contains the answer to query B. In the second step, only the sentence with the highest likelihood is paired with the question and passed to the single BiDAF to further output an answer. In summary, compared to the original BiDAF that is an end-to-end question answering system, our method first selects a sentence that is most likely to be an answer sentence. Since adversarial sentences are not supposed to contain the answer, they can be screened out. Therefore, the distractions of adversarial sentences are reduced. Experiments on ADDANY adversarial dataset demonstrates the effectiveness

[1] Since all the QA systems tested in Jia and Liang (2017) deteriorate on the ADDANY adversarial dataset, we arbitrarily choose one of them, the single BiDAF, as the benchmark.

Proceedings of the Workshop on Machine Reading for Question Answering, pages 31–36
Melbourne, Australia, July 19, 2018. ©2018 Association for Computational Linguistics

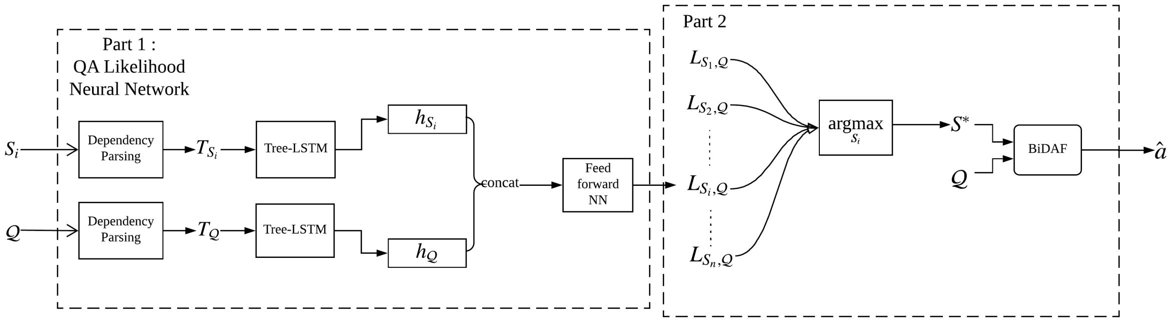

Figure 1: The Architecture of Our Approach

of our method. The F1 has been significantly improved from 4.8 to 52.3.

The contributions of this study are in three folds. First, to the best of our knowledge, it's the first work that tries to address the problem of *Question Answering with Adversarial Examples*. Our results show the effectiveness of answer sentence selection to tackle adverserial sentences in ADDANY dataset. Second, the power of sentence representation of Tree-LSTM has been demonstrated in different NLP tasks, such as semantic relatedness computation, machine translation evaluation and natural language inference (Tai et al., 2015; Gupta et al., 2015; Chen et al., 2017); meanwhile, multiple methods have been proposed for answer sentence selection (Wang and Nyberg, 2015; Rao et al., 2016; Wang et al., 2017; Shen et al., 2017a; Choi et al., 2017). We are the first to design a framework that illustrates the effectiveness of Tree-LSTM in answer sentence selection. Third, two sampling methods are implemented to build the training set for the QA Likelihood neural network. We show that different sampling methods do influence the performance of question answering in this scenario.

2 Methods

Given a paragraph $\mathcal{C}$ and a corresponding query $\mathcal{Q}$, the paragraph is split into a bunch of sentences $\mathcal{C} = \{S_i | i = 1, 2, \ldots, |\mathcal{C}|\}$. By combining each sentence S_i with the query $\mathcal{Q}$, a set of sentence pairs $\mathcal{P}_{\mathcal{C},\mathcal{Q}} = \{(S_i, \mathcal{Q}) | i = 1, 2, \ldots, |\mathcal{C}|\}$ is obtained. Then, the dependency parsing (Manning et al., 2014) is used to get the tree representation T_{S_i} for S_i and $T_{\mathcal{Q}}$ for $\mathcal{Q}$. Based on T_{S_i} and $T_{\mathcal{Q}}$, two Tree-LSTMs, **Tree-LSTM**$_{S_i}$ and **Tree-LSTM**$_{\mathcal{Q}}$, are built respectively (Tai et al., 2015). The inputs to the leafs of both Tree-LSTMs are GloVe word vectors generated by Pennington et al. (2014). The output hidden vectors of the Tree-LSTM for S_i and $\mathcal{Q}$ are h_{S_i} and $h_{\mathcal{Q}}$ respectively. Then, h_{S_i} and $h_{\mathcal{Q}}$ are concatenated and passed to a feed forward neural network to output the likelihood that S_i contains the answer to $\mathcal{Q}$. The architecture and the loss of the feed forward neural network follows the neural network for semantic relatedness (Tai et al., 2015). During training, the likelihood is supervised by 1 if S_i contains the answer and 0 otherwise. The procedure above is summarized as the QA Likelihood neural network that is illustrated in Part 1 of Figure 1. Following that, the sentence that is most likely to be an answer sentence,

$$S^* = \underset{S_i \in \mathcal{C}}{\operatorname{argmax}} \, L_{S_i,\mathcal{Q}},$$

is selected, where L stands for the likelihood predicted by the QA Likelihood neural network. After that, a pair of sentences S^* and $\mathcal{Q}$ are passed to the pre-trained single BiDAF(Seo et al., 2016) to generate an answer $\hat{a}$ to $\mathcal{Q}$. This process is illustrated in Part 2 of Figure 1.

3 Experiments

Dataset for Training. As Figure 1 shows, the input of our system is a pair of sentences. Thus, the training instances for the QA Likelihood neural network are in the form of sentence pairs. They are sampled from the training set of SQuAD v1.1 (Rajpurkar et al., 2016) that contains no adversarial sentences. Specifically, there are 87,599 queries of 18,896 paragraphs in the training set of SQuAD v1.1. While each query refers to one paragraph, a paragraph may refer to multiple queries.

For the k-th query $\mathcal{Q}^k$, by splitting its corresponding paragraph $\mathcal{C}^k$ into separate sentences and combining them with the query, a set of sentence pairs is obtained,

$$\mathcal{D}_k = \{(S_i^k, \mathcal{Q}^k) | i = 1, 2, \ldots, m_k\}$$

where $\mathcal{D}_k$ represents the set of sentence pairs for the k-th query, m_k is the number of sentences in the paragraph $\mathcal{C}^k$, S_i^k is the i-th sentence in $\mathcal{C}^k$. A sentence pair $(S_i^k, \mathcal{Q}^k)$ is called a *positive instance* if S_i^k contains the answer to $\mathcal{Q}^k$; otherwise, it is called a *negative instance*. Then, the union of the sets $\mathcal{D}_k$ for all the 87,599 queries in SQuDA is

$$\mathcal{D} = \bigcup_{k=1}^{d} \mathcal{D}_k$$

where d=87,599 is the number of queries. The set $\mathcal{D}$ contains 440,135 sentence pairs, among which 87,306 are positive instances and 352,829 are negative instances.

In order to train our model properly and efficiently, both downsampling of $\mathcal{D}$ and undersampling of negative instances must be done. In this paper, we implement two different sampling methods: **pair-level sampling** and **paragraph-level sampling**. In pair-level sampling, 45,000 positive instances and 45,000 negative instances are randomly selected from $\mathcal{D}$ as the training set. By contrast, in paragraph-level sampling, we first randomly select a query $\mathcal{Q}^k$ without replacement, then one positive instance and one negative instance are randomly sampled from the set of sentence pairs $\mathcal{D}_k$. This operation is repeated until we get 45,000 positive instances and 45,000 negative instances. Finally, two different training sets are generated by pair-level sampling and paragraph-level sampling. Each set has 90,000 instances. The validation set with 3,000 instances are sampled through these two methods as well.

Dataset for Testing. Our test set is Jia and Liang (2017)'s ADDANY adversarial dataset. It includes 1,000 paragraphs and each paragraph refers to only one query, i.e., 1,000 $(\mathcal{C}, \mathcal{Q})$ pairs. By splitting and combining, 6,154 sentence pairs are obtained.

Experimental Settings. The dimension of GloVe word vectors (Pennington et al., 2014) is set as 300. The sentence scoring neural network is trained by Adagrad (Duchi et al., 2011) with a learning rate of 0.01 and a batch size of 25. Model parameters are regularized by a 10^{-4} strength of per-minibatch L_2 regularization.

QA System	F1	Precision	Recall
QA Likelihood (pair-level sampling)	50.6	51.4	53.0
QA Likelihood (paragraph-level sampling)	**52.3**	**53.1**	**54.9**
Single BiDAF	4.8	4.8	6.2

Table 1: Results of QA with Adversarial Examples

QA System	F1	Accuracy	Precision	Recall
QA Likelihood (pair-level sampling)	62.5	87.4	64.8	60.2
QA Likelihood (paragraph-level sampling)	**63.4**	**87.7**	**65.8**	**61.2**
Single BiDAF	17.0	72.0	17.6	16.4

Table 2: Results of Answer Sentence Selection

4 Results

The performance of question answering is evaluated by the Macro-averaged F1 score (Rajpurkar et al., 2016; Jia and Liang, 2017). It measures the average overlap between the predicted answer $\hat{a}$ and real answers on token-level. We also compute the Macro-averaged Precision and Recall following the same procedure. The results are in Table 1. As it shows, both the systems based on pair-level sampling and paragraph-level sampling significantly outperform the single BiDAF system[2]. The Macro-averaged F1 has been improved from 4.8 to 52.3. Besides, the paragraph-level sampling achieves better results than the pair-level sampling.

In order to analyze the source of performance improvements, we further evaluate the performance of the QA Likelihood neural network and the single BiDAF system on answer sentence selection[3]. Here, we consider the problem as a binary classification problem. In the test set, positive instances are labeled with 1 and negative ones are labeled with 0. A sentence pair selected by a QA system (QA Likelihood neural network or the single BiDAF) has a predicted label 1, while the others have a predicted label 0. The results are shown in Table 2. It shows that both of our systems outperform the single BiDAF on all of the four metrics in the table.

We further evaluate the performance of the QA Likelihood neural network and the single BiDAF system on answer sentence selection from another perspective. Here, we consider three types of sentences: adversarial sentences, answer sentences, and the sentences that include the answers returned by the single BiDAF system. Given a QA

[2] Since Jia and Liang (2017) and we are evaluating the systems on the same test set, the results of the single BiDAF in our paper are derived from the results published by them on https://worksheets.codalab.org/worksheets/0xc86d3ebe69a3427d91f9aaa63f7d1e7d/

[3] The sentences which include the answers generated by the single BiDAF are regarded as the answer sentences selected by the single BiDAF.

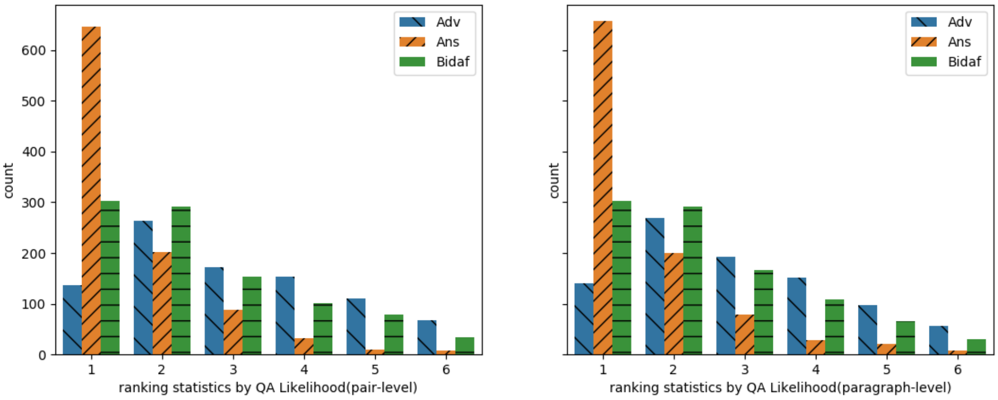

Figure 2: Sentences Ranking Statistics

Likelihood neural network, we draw a histogram in which the x-axis denotes the ranked position for each sentence according to its likelihood score [4], while the y-axis is the number of sentences for each type ranked at this position. The results are presented in Figure 2. It shows that among the 1,000 $(\mathcal{C}, \mathcal{Q})$ pairs, 647 and 657 answer sentences are selected by the QA Likelihood neural network based on pair-level sampling and paragraph-level sampling respectively, but only 136 and 141 adversarial sentences are selected by the QA Likelihood neural network. It indicates the effectiveness of the QA Likelihood neural network to reduce the impact of adversarial sentences.

5 Related works

With the help of deep learning, many techniques have been investigated to achieve exciting results on answer sentence selection and QA. Wang and Nyberg (2015) measure the relevance between sentences through a stacked bidirectional LSTM network. They show that these scores are effective in answer sentence selection. He et al. (2015) embed sentences with CNN at multiple levels of granularity to model the similarity between sentences. Rao et al. (2016) extend the method of Noise-Contrastive Estimation to questions paired with positive and negative sentences. Based on that, they present a pairwise ranking approach to select an answer from multiple candidate sentences. Wang et al. (2017) propose a bilateral multi-perspective matching model which achieves rivaling results in the task of answer sentence selection. Shen et al. (2017a) measure the similarity between sentences by utilizing the word level similarity matrix. This approach is validated in answer selection. To efficiently tackle question answering for long documents, Choi et al. (2017) propose a method based on answer sentence selection to first narrow down a document and then use RNN to generate an answer.

However, following the idea of adversarial examples in image recognition(Goodfellow et al., 2014; Kurakin et al., 2016; Papernot et al., 2016), Jia and Liang (2017) point out the unreliability of existing question answering models in the presence of adversarial sentences. In this study, we propose a method to tackle this problem through answer sentence selection. The main component of our system is Tree-LSTM which is a powerful variant of Tree-RNN. Therefore, studies about Tree-RNN(Pollack, 1990; Goller and Kchler, 1996; Socher et al., 2011, 2012, 2013; Zhang et al., 2016) are also related.

6 Conclusions

In this paper, we propose a method to address the problem of question answering with adversarial sentences in paragraphs. Specifically, our system via the QA Likelihood neural network based on Tree-LSTMs successfully boost the performance of the single BiDAF on ADDANY adversarial dataset. Experiments show the F1 score has been largely improved from 4.8 to 52.3. To the best of our knowledge, we are the first to apply Tree-LSTMs in answer sentence selection and the first to tackle question answering with adversarial examples on ADDANY adversarial dataset.

However, Jia and Liang (2017) also present the deterioration of QA systems on another dataset, ADDSENT adversarial dataset. Question answer-

[4] The x-axis is truncated to save the space.

ing on this dataset remains unsolved. We leave it as a future work.

7 Acknowledgments

We would like to thank the anonymous reviewers for their valuable comments. This work is supported in part by the National Natural Science Foundation of China grants 11501464 and 11761141007.

References

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)*. volume 1, pages 1657–1668.

Eunsol Choi, Daniel Hewlett, Jakob Uszkoreit, Illia Polosukhin, Alexandre Lacoste, and Jonathan Berant. 2017. Coarse-to-fine question answering for long documents. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*. pages 209–220. https://doi.org/10.18653/v1/P17-1020.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research* 12(Jul):2121–2159.

Christoph Goller and Andreas Kchler. 1996. Learning task-dependent distributed representations by backpropagation through structure. In *In Proc. of the ICNN-96*. IEEE, pages 347–352.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* .

Rohit Gupta, Constantin Orasan, and Josef van Genabith. 2015. Reval: A simple and effective machine translation evaluation metric based on recurrent neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. pages 1066–1072.

Hua He, Kevin Gimpel, and Julie Qiaojin Lin. 2015. Multi-perspective sentence similarity modeling with convolutional neural networks. In *EMNLP*.

Minghao Hu, Yuxing Peng, and Xipeng Qiu. 2017. Reinforced mnemonic reader for machine comprehension. *CoRR, abs/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, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*. pages 2021–2031. https://aclanthology.info/papers/D17-1215/d17-1215.

Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. 2016. Adversarial examples in the physical world. *CoRR* abs/1607.02533. http://arxiv.org/abs/1607.02533.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*. pages 55–60. http://www.aclweb.org/anthology/P/P14/P14-5010.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. 2016. Practical black-box attacks against deep learning systems using adversarial examples. *arXiv preprint* .

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.

Jordan B Pollack. 1990. Recursive distributed representations. *Artificial Intelligence* 46(1-2):77–105.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250* .

Jinfeng Rao, Hua He, and Jimmy Lin. 2016. Noise-contrastive estimation for answer selection with deep neural networks. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, pages 1913–1916.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603* .

Gehui Shen, Yunlun Yang, and Zhi-Hong Deng. 2017a. Inter-weighted alignment network for sentence pair modeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. pages 1179–1189.

Yelong Shen, Po-Sen Huang, Jianfeng Gao, and Weizhu Chen. 2017b. Reasonet: Learning to stop reading in machine comprehension. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, pages 1047–1055.

Richard Socher, Brody Huval, Christopher D Manning, and Andrew Y Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 joint conference on empirical* *methods in natural language processing and computational natural language learning*. Association for Computational Linguistics, pages 1201–1211.

Richard Socher, Cliff C Lin, Chris Manning, and Andrew Y Ng. 2011. Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the 28th international conference on machine learning (ICML-11)*. pages 129–136.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*. pages 1631–1642.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*. pages 1556–1566. http://aclweb.org/anthology/P/P15/P15-1150.pdf.

Di Wang and Eric Nyberg. 2015. A long short-term memory model for answer sentence selection in question answering. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 2: Short Papers*. pages 707–712. http://aclweb.org/anthology/P/P15/P15-2116.pdf.

Shuohang Wang and Jing Jiang. 2016. Machine comprehension using match-lstm and answer pointer. *arXiv preprint arXiv:1608.07905* .

Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral multi-perspective matching for natural language sentences. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*. pages 4144–4150. https://doi.org/10.24963/ijcai.2017/579.

Xingxing Zhang, Liang Lu, and Mirella Lapata. 2016. Top-down tree long short-term memory networks. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*. pages 310–320. http://aclweb.org/anthology/N/N16/N16-1035.pdf.