Yuanfudao at SemEval-2018 Task 11: Three-way Attention and Relational Knowledge for Commonsense Machine Comprehension

Liang Wang **Meng Sun** **Wei Zhao** **Kewei Shen** **Jingming Liu**
Yuanfudao Research
Beijing, China
{wangliang01,sunmeng,zhaowei01,shenkw,liujm}@fenbi.com

Abstract

This paper describes our system for *SemEval-2018 Task 11: Machine Comprehension using Commonsense Knowledge* (Ostermann et al., 2018b). We use **Three**-way **A**ttentive **N**etworks (*TriAN*) to model interactions between the passage, question and answers. To incorporate commonsense knowledge, we augment the input with relation embedding from the graph of general knowledge *ConceptNet* (Speer et al., 2017). As a result, our system achieves state-of-the-art performance with 83.95% accuracy on the official test data. Code is publicly available at `https://github.com/intfloat/commonsense-rc`.

1 Introduction

It is well known that humans have a vast amount of commonsense knowledge acquired from everyday life. For machine reading comprehension, natural language inference and many other NLP tasks, commonsense reasoning is one of the major obstacles to make machines as intelligent as humans.

A large portion of previous work focus on commonsense knowledge acquisition with unsupervised learning (Chambers and Jurafsky, 2008; Tandon et al., 2017) or crowdsourcing approach (Singh et al., 2002; Wanzare et al., 2016). *ConceptNet* (Speer et al., 2017), *WebChild* (Tandon et al., 2017) and *DeScript* (Wanzare et al., 2016) etc are all publicly available knowledge resources. However, resources based on unsupervised learning tend to be noisy, while crowdsourcing approach has scalability issues. There is also some research on incorporating knowledge into NLP tasks such as reading comprehension (Lin et al., 2017; Yang and Mitchell, 2017) neural machine translation (Zhang et al., 2017a) and text classification (Zhang et al., 2017b) etc. Though experiments show performance gains over baselines, these gains are often quite marginal over the state-of-the-art system without external knowledge.

In this paper, we present **Three**-way **A**ttentive **N**etworks(*TriAN*) for multiple-choice commonsense reading comprehension. The given task requires modeling interactions between the passage, question and answers. Different questions need to focus on different parts of the passage, attention mechanism is a natural choice and turns out to be effective for reading comprehension. Due to the relatively small size of training data, *TriAN* use word-level attention and consists of only one layer of LSTM (Hochreiter and Schmidhuber, 1997). Deeper models result in serious overfitting and poor generalization empirically.

To explicitly model commonsense knowledge, relation embeddings based on ConceptNet (Speer et al., 2017) are used as additional features. ConceptNet is a large-scale graph of general knowledge from both crowdsourced resources and expert-created resources. It consists of over 21 million edges and 8 million nodes. ConceptNet shows state-of-the-art performance on tasks like word analogy and word relatedness.

Besides, we also find that pretraining our network on other datasets helps to improve the overall performance. There are some existing multiple-choice English reading comprehension datasets contributed by NLP community such as *MCTest* (Richardson et al., 2013) and *RACE* (Lai et al., 2017). Although those datasets don't focus specifically on commonsense comprehension, they provide a convenient way for data augmentation. Augmented data can be used to learn shared regularities of reading comprehension tasks.

Combining all of the aforementioned techniques, our system achieves competitive performance on the official test set.

Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018), pages 758–762
New Orleans, Louisiana, June 5–6, 2018.

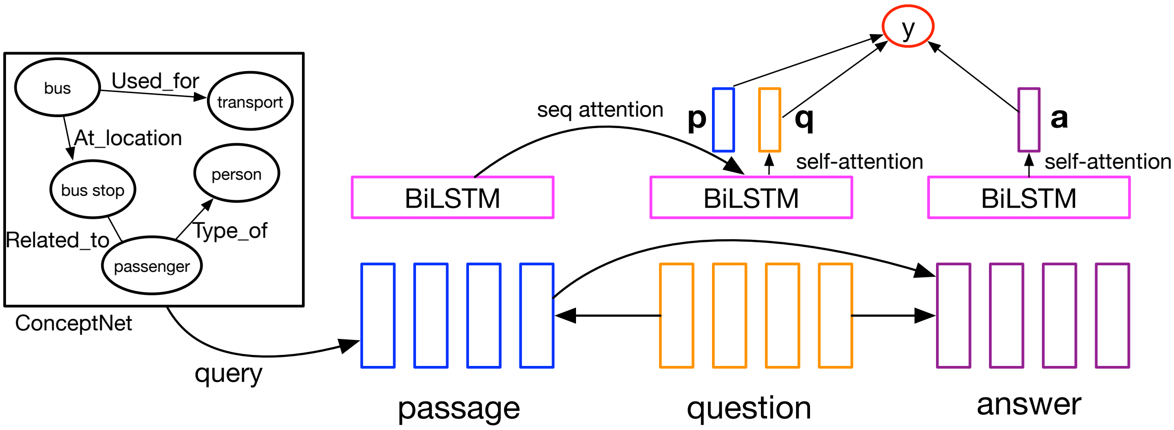

Figure 1: *TriAN* Model Architecture.

2 Model

The overall architecture of *TriAN* is shown in Figure 1. It consists of an input layer, an attention layer and an output layer.

Input Layer. A training example consists of a passage $\{P_i\}_{i=1}^{|P|}$, a question $\{Q_i\}_{i=1}^{|Q|}$, an answer $\{A_i\}_{i=1}^{|A|}$ and a label $y^* \in \{0, 1\}$ as input. P, Q and A are all sequences of word indices. For a word P_i in the given passage, the input representation of P_i is the concatenation of several vectors:

- **GloVe embeddings.** Pretrained 300-dimensional *GloVe* vector $\mathbf{E}_{P_i}^{glove}$.
- **Part-of-speech and named-entity embeddings.** Randomly initialized 12-dimensional part-of-speech embedding $\mathbf{E}_{P_i}^{pos}$ and 8-dimensional named-entity embedding $\mathbf{E}_{P_i}^{ner}$.
- **Relation embeddings.** Randomly initialized 10-dimensional relation embedding $\mathbf{E}_{P_i}^{rel}$. The relation is determined by querying ConceptNet whether there is an edge between P_i and any word in question $\{Q_i\}_{i=1}^{|Q|}$ or answer $\{A_i\}_{i=1}^{|A|}$. If there exist multiple different relations, just randomly choose one.
- **Handcrafted features.** We also add logarithmic term frequency feature and co-occurrence feature $\mathbf{f}_{P_i}$. Term frequency is calculated based on English Wikipedia. Co-occurrence feature is a binary feature which is true if P_i appears in question $\{Q_i\}_{i=1}^{|Q|}$ or answer $\{A_i\}_{i=1}^{|A|}$.

The input representation for P_i is $\mathbf{w}_{P_i}$:

$$\mathbf{w}_{P_i} = [\mathbf{E}_{P_i}^{glove}; \mathbf{E}_{P_i}^{pos}; \mathbf{E}_{P_i}^{ner}; \mathbf{E}_{P_i}^{rel}; \mathbf{f}_{P_i}] \quad (1)$$

In a similar way, we can get input representation for question $\mathbf{w}_{Q_i}$ and answer $\mathbf{w}_{A_i}$.

Attention Layer. We use word-level attention to model interactions between the given passage $\{P_i\}_{i=1}^{|P|}$, the question $\{Q_i\}_{i=1}^{|Q|}$ and the answer $\{A_i\}_{i=1}^{|A|}$. First, let's define a sequence attention function (Chen et al., 2017):

$$\begin{aligned} Att_{seq}(\mathbf{u}, \{\mathbf{v}_i\}_{i=1}^{n}) &= \sum_{i=1}^{n} \alpha_i \mathbf{v}_i \\ \alpha_i &= \mathrm{softmax}_i(f(\mathbf{W}_1\mathbf{u})^T f(\mathbf{W}_1\mathbf{v}_i)) \end{aligned} \quad (2)$$

$\mathbf{u}$ and $\mathbf{v}_i$ are vectors and $\mathbf{W}_1$ is a matrix. f is a non-linear activation function and is set to $ReLU$.

Question-aware passage representation $\{\mathbf{w}_{P_i}^q\}_{i=1}^{|P|}$ can be calculated as: $\mathbf{w}_{P_i}^q = Att_{seq}(\mathbf{E}_{P_i}^{glove}, \{\mathbf{E}_{Q_i}^{glove}\}_{i=1}^{|Q|})$. Similarly, we can get passage-aware answer representation $\{\mathbf{w}_{A_i}^p\}_{i=1}^{|A|}$ and question-aware answer representation $\{\mathbf{w}_{A_i}^q\}_{i=1}^{|A|}$. Then three BiLSTMs are applied to the concatenation of those vectors to model the temporal dependency:

$$\begin{aligned} \mathbf{h}^q &= \mathrm{BiLSTM}(\{\mathbf{w}_{Q_i}\}_{i=1}^{|Q|}) \\ \mathbf{h}^p &= \mathrm{BiLSTM}(\{[\mathbf{w}_{P_i}; \mathbf{w}_{P_i}^q]\}_{i=1}^{|P|}) \\ \mathbf{h}^a &= \mathrm{BiLSTM}(\{[\mathbf{w}_{A_i}; \mathbf{w}_{A_i}^p; \mathbf{w}_{A_i}^q]\}_{i=1}^{|A|}) \end{aligned} \quad (3)$$

$\mathbf{h}^p, \mathbf{h}^q, \mathbf{h}^a$ are the new representation vectors that incorporates more context information.

Output Layer. Question sequence and answer sequence representation $\mathbf{h}^q, \mathbf{h}^a$ are summarized into fixed-length vectors with self-attention (Yang et al., 2016). Self-attention function is defined as follows:

$$Att_{self}(\{\mathbf{u}_i\}_{i=1}^{n}) = \sum_{i=1}^{n} \alpha_i \mathbf{u}_i$$
$$\alpha_i = \text{softmax}_i(\mathbf{W}_2^T \mathbf{u}_i) \tag{4}$$

Then we have question representation $\mathbf{q} = Att_{self}(\{\mathbf{h}_i^q\}_{i=1}^{|Q|})$, answer representation $\mathbf{a} = Att_{self}(\{\mathbf{h}_i^a\}_{i=1}^{|A|})$ and passage representation $\mathbf{p} = Att_{seq}(\mathbf{q}, \{\mathbf{h}_i^p\}_{i=1}^{|P|})$. The final output y is based on their bilinear interactions:

$$y = \sigma(\mathbf{p}^T \mathbf{W}_3 \mathbf{a} + \mathbf{q}^T \mathbf{W}_4 \mathbf{a}) \tag{5}$$

Model Learning. We first pretrain *TriAN* on *RACE* dataset for 10 epochs. Then our model is fine-tuned on official training data. Standard cross entropy function is used as the loss function to minimize.

3 Experiments

3.1 Setup

Data. For data preprocessing, we use *spaCy* [1] for tokenization, part-of-speech tagging and named-entity recognition. Statistics for official dataset *MCScript* (Ostermann et al., 2018a) are shown in Table 1. *RACE* [2] dataset is used for network pretraining. English stop words are ignored when computing handcrafted features. Input word embeddings are initialized with 300-dimensional *GloVe* (Pennington et al., 2014) vectors [3].

	train	dev	test
# of examples	9731	1411	2797

Table 1: Official dataset statistics.

Hyperparameters. Our model *TriAN* is implemented based on *PyTorch* [4]. Models are trained on a single GPU(Tesla P40) and each epoch takes about 80 seconds. Only the word embeddings of top 10 frequent words are fine-tuned during training. The dimension of both forward and backward LSTM hidden state is set to 96. Dropout rate is set to 0.4 for both input embeddings and BiLSTM outputs (Srivastava et al., 2014). For parameter optimization, we use *Adamax* (Kingma and Ba, 2014) with an initial learning rate 2×10^{-3}. Learning rate is then halved after 10 and 15 training epochs. The model converges after 50 epochs. Gradients are clipped to have a maximum L2 norm of 10. Minibatch with batch size 32 is used. Hyperparameters are optimized by random search strategy (Bergstra and Bengio, 2012). Our model is quite robust over a wide range of hyperparameter configurations.

3.2 Main Results

The experimental results are shown in Table 2. Human performance is shared by task organizers. For *TriAN-ensemble*, we average the output probabilities of 9 models trained with the same datasets and network architecture but different random seeds. *TriAN-ensemble* is the model that we used for official submission.

model	dev	test
Random	50.00%	50.00%
TriAN-RACE	64.78%	64.28%
TriAN-single	83.84%	81.94%
TriAN-ensemble	**85.27%**	**83.95%**
HFL	–	84.13%
Human	–	98.00%

Table 2: Main results. *TriAN-RACE* only use *RACE* dataset for training; *HFL* is the 1st place team for *SemEval-2018 Task 11*. The evaluation metric is accuracy.

From Table 2, we can see that even though *RACE* dataset contains nearly $100k$ questions, *TriAN-RACE* achieves quite poor results. The accuracy on development set is only 64.78%, which is worse than most participants' systems. However, pretraining acts as a way of implicit knowledge transfer and is beneficial for overall performance, as will be seen in Section 3.3. The accuracy of our system *TriAN-ensemble* is very close to the 1st place team *HFL* with 0.18% difference. Yet there is still a large gap between machine learning models and human.

We also compared the performances of shallow and deep TriAN models. On datasets such as *SQuAD* (Rajpurkar et al., 2016), deep models typically works better than shallow ones. Notice that the attention layer in our proposed *TriAN* model can be stacked multiple times if we treat the output vectors of BiLSTMs as new input representations.

Maybe a little bit surprising, Table 3 shows that *2-layer TriAN* model performs worse than *1-layer*

[1] https://github.com/explosion/spaCy
[2] http://www.cs.cmu.edu/~glai1/data/race/
[3] http://nlp.stanford.edu/data/glove.840B.300d.zip
[4] http://pytorch.org/

model	dev	test
1-layer TriAN-single	**83.84%**	**81.94%**
2-layer TriAN-single	82.71%	80.55%

Table 3: Accuracy comparison of shallow and deep TriAN models.

TriAN. One possible explanation is that the labeled dataset is relatively small and deeper models tend to easily overfit.

3.3 Ablation Study

The input representation consists of several components: part-of-speech embedding, relation embedding and handcrafted features etc. We conduct an ablation study to investigate the effects of each component. The results are in Table 4.

model	dev	test
TriAN-single	**83.84%**	**81.94%**
w/o pretraining	82.71%	80.51%
w/o ConceptNet	82.78%	81.08%
w/o POS	82.84%	81.27%
w/o features	82.92%	81.35%
w/o NER	83.60%	81.87%

Table 4: Ablation study for input representation.

Pretraining on *RACE* dataset turns out to be the most important factor. Without pretraining, the accuracy drops by more than 1% on both development and test set. Relation embeddings based on ConceptNet make approximately 1% difference. Part-of-speech and named-entity embeddings are also helpful. In fact, combining input representations from multiple sources has been a standard practice for reading comprehension tasks.

At attention layer, our proposed *TriAN* involves applying several attention functions to model interactions between different text sequences. It would be interesting to examine the importance of each attention function, as shown in Table 5.

model	dev	test
TriAN-single	**83.84%**	81.94%
w/o passage-question attention	83.51%	**82.20%**
w/o passage-answer attention	83.07%	81.39%
w/o question-answer attention	83.23%	81.84%
w/o attention	81.93%	80.65%

Table 5: Ablation study for attention. The last one "*w/o attention*" removes all word-level attentions.

Interestingly, removing any of the three word-level sequence attentions does not seem to hurt the performance much. In fact, removing passage-question attention even results in higher accuracy on test set than *TriAN-single*. However, if we remove all word-level attentions, the performance drastically drops by 1.9% on development set and 1.3% on test set.

3.4 Discussion

Even though our system is built for commonsense reading comprehension, it doesn't have any explicit knowledge reasoning component. Relation embeddings based on ConceptNet only serve as additional input features. Methods like event calculus (Mueller, 2014) are more rigorous mathematically and resemble the way of how human brain works. The problem of event calculus is that it requires large amounts of domain-specific axioms and therefore doesn't scale well.

Another limitation is that our system relies on hard-coded commonsense knowledge bases, just like most systems for commonsense reasoning. For humans, commonsense knowledge comes from constant interactions with the real-world environment. From our point of view, it is quite hopeless to enumerate all of them.

There are a lot of reading comprehension datasets available. When the size of training data is relatively small like this *SemEval-2018 task*, transfer learning among different datasets is a useful technique. This paper shows that pretraining is a simple and effective method. However, it still remains to be seen whether there is a better alternative approach.

4 Conclusion

In this paper, we present the core ideas and design philosophy for our system *TriAN* at *SemEval-2018 Task 11: Machine Comprehension using Commonsense Knowledge*. We build upon recent progress on neural models for reading comprehension and incorporate commonsense knowledge from ConceptNet. Pretraining and handcrafted features are also proved to be helpful. As a result, our proposed model *TriAN* achieves near state-of-the-art performance.

Acknowledgements

We would like to thank SemEval 2018 task organizers and several anonymous reviewers for their helpful comments.

References

James Bergstra and Yoshua Bengio. 2012. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305.

Nathanael Chambers and Dan Jurafsky. 2008. Unsupervised learning of narrative event chains. *Proceedings of ACL-08: HLT*, pages 789–797.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

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

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*.

Hongyu Lin, Le Sun, and Xianpei Han. 2017. Reasoning with heterogeneous knowledge for commonsense machine comprehension. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2032–2043.

Erik T Mueller. 2014. *Commonsense reasoning: an event calculus based approach*. Morgan Kaufmann.

Simon Ostermann, Ashutosh Modi, Michael Roth, Stefan Thater, and Manfred Pinkal. 2018a. MCScript: A Novel Dataset for Assessing Machine Comprehension Using Script Knowledge. In *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan.

Simon Ostermann, Michael Roth, Ashutosh Modi, Stefan Thater, and Manfred Pinkal. 2018b. SemEval-2018 Task 11: Machine Comprehension using Commonsense Knowledge. In *Proceedings of International Workshop on Semantic Evaluation(SemEval-2018)*, New Orleans, LA, USA.

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.

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

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.

Push Singh, Thomas Lin, Erik T Mueller, Grace Lim, Travell Perkins, and Wan Li Zhu. 2002. Open mind common sense: Knowledge acquisition from the general public. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 1223–1237. Springer.

Robert Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *AAAI*, pages 4444–4451.

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

Niket Tandon, Gerard de Melo, and Gerhard Weikum. 2017. Webchild 2.0: fine-grained commonsense knowledge distillation. *Proceedings of ACL 2017, System Demonstrations*, pages 115–120.

Lilian DA Wanzare, Alessandra Zarcone, Stefan Thater, and Manfred Pinkal. 2016. Descript: A crowdsourced corpus for the acquisition of high-quality script knowledge. In *The International Conference on Language Resources and Evaluation.*

Bishan Yang and Tom Mitchell. 2017. Leveraging knowledge bases in lstms for improving machine reading. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1436–1446.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489.

Jiacheng Zhang, Yang Liu, Huanbo Luan, Jingfang Xu, and Maosong Sun. 2017a. Prior knowledge integration for neural machine translation using posterior regularization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1514–1523.

Ye Zhang, Matthew Lease, and Byron C Wallace. 2017b. Exploiting domain knowledge via grouped weight sharing with application to text categorization. *arXiv preprint arXiv:1702.02535*.