Using Natural Language Relations between Answer Choices for Machine Comprehension

Rajkumar Pujari
Department of Computer Science
Purdue University
rpujari@purdue.edu

Dan Goldwasser
Department of Computer Science
Purdue University
dgoldwas@purdue.edu

Abstract

When evaluating an answer choice for Reading Comprehension task, other answer choices available for the question and the answers of related questions about the same paragraph often provide valuable information. In this paper, we propose a method to leverage the natural language relations between the answer choices, such as entailment and contradiction, to improve the performance of machine comprehension. We use a stand-alone question answering (QA) system to perform QA task and a Natural Language Inference (NLI) system to identify the relations between the choice pairs. Then we perform inference using an Integer Linear Programming (ILP)-based relational framework to re-evaluate the decisions made by the standalone QA system in light of the relations identified by the NLI system. We also propose a multitask learning model that learns both the tasks jointly.

1 Introduction

Given an input text and a set of related questions with multiple answer choices, the reading comprehension (RC) task evaluates the correctness of each answer choice. Current approaches to the RC task quantify the relationship between each question and answer choice independently and pick the highest scoring option. In this paper, we follow the observation that when humans approach such RC tasks, they tend to take a holistic view ensuring that their answers are consistent across the given questions and answer choices. In this work we attempt to model these pragmatic inferences, by leveraging the *entailment* and *contradiction* relations between the answer choices to improve machine comprehension. To help clarify these concepts, consider the following examples:

How can the military benefit from the existence of the CIA?
c_1: They can use them
c_2: These agencies are keenly attentive to the military's strategic and tactical requirements (✗)
c_3: The CIA knows what intelligence the military requires and has the resources to obtain that intelligence (✓)

The above example contains multiple correct answer choices, some are easier to capture than others. For example, identifying that c_3 is true might be easier than c_2 based on its alignment with the input text. However, capturing that c_3 entails c_2 allows us to predict c_2 correctly as well. Classification of the answer in red (marked ✗) could be corrected using the blue (marked ✓) answer choice.

Q1: *When were the eggs added to the pan to make the omelette?*
c^1_1: When they turned on the stove
c^1_2: When the pan was the right temperature (✓)
Q2: *Why did they use stove to cook omelette?*
c^2_1: They didn't use the stove but a microwave
c^2_2: Because they needed to heat up the pan (✗)

Similarly, answering Q1 correctly helps in answering Q2. Our goal is to leverage such inferences for machine comprehension.

Our approach contains three steps. First, we use a stand-alone QA system to classify the answer choices as true/false. Then, we classify the relation between each pair of choices for a given question as *entailment*, *contradiction* or *neutral*. Finally, we re-evaluate the labels assigned to choices using an Integer Linear Programming based inference procedure. We discuss different training protocols and representation choices for the combined decision problem. An overview is in figure 1.

We empirically evaluate on two recent datasets, MultiRC (Khashabi et al., 2018) and SemEval-

Proceedings of NAACL-HLT 2019, pages 4010–4015
Minneapolis, Minnesota, June 2 - June 7, 2019. ©2019 Association for Computational Linguistics

2018 task-11 (Ostermann et al., 2018) and show that it improves machine comprehension in both.

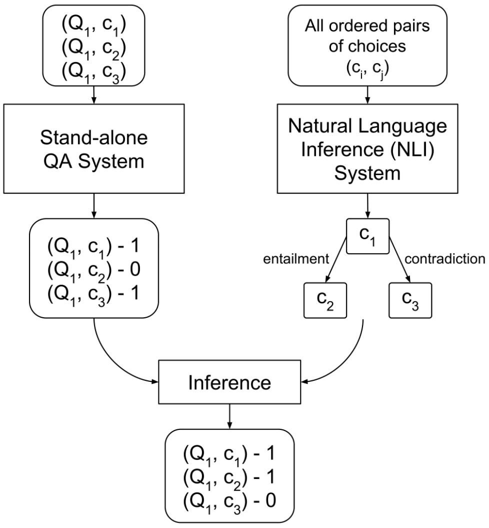

Figure 1: Proposed Approach

2 Related Work

Recently, several QA datasets have been proposed to test machine comprehension (Richardson, 2013; Weston et al., 2015; Rajpurkar et al., 2016; Trischler et al., 2016a; Nguyen et al., 2016). Yatskar (2018) showed that a high performance on these datasets could be achieved without necessarily achieving the capability of making commonsense inferences. Trischler et al. (2016b), Kumar et al. (2016), Liu and Perez (2017), Min et al. (2018) and Xiong et al. (2016) proposed successful models on those datasets. To address this issue, new QA datasets which require commonsense reasoning have been proposed (Khashabi et al., 2018; Ostermann et al., 2018; Mihaylov et al., 2018). Using common sense inferences in Machine Comprehension is a far from solved problem. There have been several attempts in literature to use inferences to answer questions. Most of the previous works either attempt to infer the answer from the given text (Sachan and Xing, 2016; Sun et al., 2018) or an external commonsense knowledge base (Das et al., 2017; Mihaylov and Frank, 2018; Bauer et al., 2018; Weissenborn et al., 2017).

While neural models can capture some dependencies between choices through shared representations, to the best of our knowledge, inferences capturing the dependencies between answer choices or different questions have been not explicitly modeled.

3 Model

Formally, the task of machine comprehension can be defined as: given text $\mathcal{P}$ and a set of n related questions $\mathcal{Q} = \{q_1, q_2, \ldots, q_n\}$ each having m choices $\mathcal{C} = \{c_1^i, c_2^i, \ldots, c_m^i\} \forall q_i \in \mathcal{Q}$, the task is to assign true/false value for each choice c_j^i.

3.1 Model Architecture

Our model consists of three separate systems, one for each step, namely, the stand-alone question answering (QA) system, the Natural Language Inference (NLI) system and the inference framework connecting the two. First, we assign a true/false label to each question-choice pair using the stand-alone QA system along with an associated confidence score s_1. Consequently, we identify the natural language relation (entailment, contradiction or neutral) between each ordered pair of choices for a given question, along with an associated confidence score s_2. Then, we use a relational framework to perform inference using the information obtained from the stand-alone QA and the NLI systems. Each of the components is described in detail in the following sub-sections.

We further propose a joint model whose parameters are trained jointly on both the tasks. The joint model uses the answer choice representation generated by the stand-alone QA system as input to the NLI detection system. The architecture of our joint model is shown in figure 2.

3.1.1 Stand-alone QA system

We use the TriAN-single model proposed by Wang et al. (2018) for SemEval-2018 task-11 as our stand-alone QA system. We use the implementation[1] provided by Wang et al. (2018) for our experiments. The system is a tri-attention model that takes passage-question-choice triplet as input and produces the probability of the choice being true as its output.

3.2 NLI System

Our NLI system is inspired from decomposable-attention model proposed by Parikh et al. (2016). We modified the architecture proposed in Parikh et al. (2016) to accommodate the question-choice pairs as opposed to sentence pairs in the original model. We added an additional sequence-attention layer for the question-choice pairs to allow for the

[1] https://github.com/intfloat/commonsense-rc

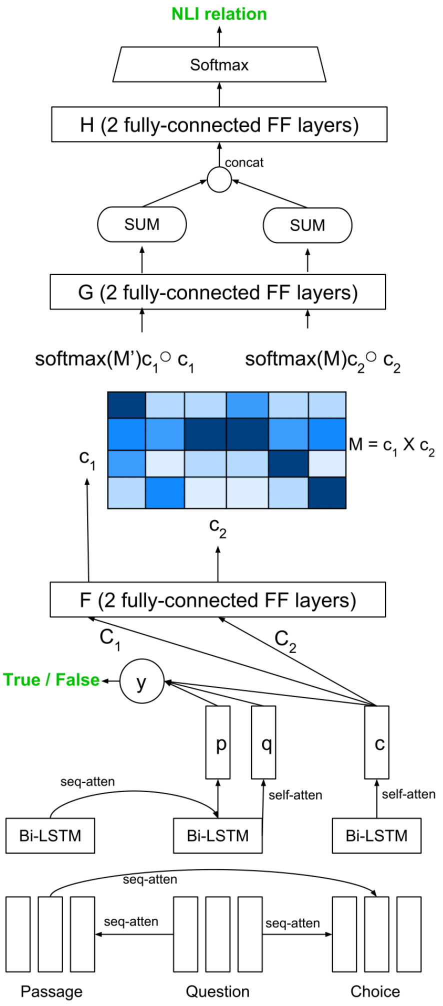

Figure 2: Architecture of the Joint Model

representation of both the answer choice and the question. Sequence-attention is defined in Wang et al. (2018) as:

$$Att_{seq}(\mathbf{u}, \{\mathbf{v}_i\}_{i=1}^{n}) = \sum_{i=1}^{n} \alpha_i \mathbf{v}_i \qquad (1)$$

$$\alpha_i = softmax_i(f(\mathbf{W}_1\mathbf{u})^T f(\mathbf{W}_1\mathbf{v}_i))$$

where $\mathbf{u}$ and $\mathbf{v}_i$ are word embeddings, $\mathbf{W}_1$ is the associated weight parameter and f is non-linearity. Self-attention is Att_{seq} of a vector onto itself.

The embedding of each word in the answer choice is attended to by the sequence of question word embeddings. We use pre-trained GloVe (Pennington et al., 2014) embeddings to represent the words. The question-attended choices are then passed through the decomposable-attention layer proposed in Parikh et al. (2016).

3.2.1 Inference using DRAIL

We use Deep Relational Learning (DRaiL) framework proposed by Zhang et al. (2016) to perform the final inference. The framework allows for declaration of predicate logic rules to perform relational inference. The rules are scored by the confidence scores obtained from the stand-alone QA and the NLI systems. DRaiL uses an Integer Linear Programming (ILP) based inference procedure to output binary prediction for each of the choices. We use the following constraints for our inference:

1. c_i is true & c_i entails $c_j \implies c_j$ is true.
2. c_i is true & c_i contradicts $c_j \implies c_j$ is false.

On the MultiRC dataset, we use the dependencies between the answer choices for a given question. On SemEval dataset, we use the dependencies between different questions about the same paragraph.

3.3 Joint Model

The design of our joint model is motivated by the two objectives: 1) to obtain a better representation for the question-choice pair for NLI detection and 2) to leverage the benefit of multitask learning. Hence, in the joint model, choice representation from stand-alone QA system is input to the decomposable-attention layer of the NLI system.

The joint model takes two triplets (p, q_i, c_i) and (p, q_j, c_j) as input. It outputs a `true/false` for each choice and an NLI relation (entailment, contradiction or neutral) between the choices. The representations for passage, question and choice are obtained using Bi-LSTMs. The hidden states of the Bi-LSTM are concatenated to generate the representation. This part of the model is similar to TriAN model proposed in Wang et al. (2018). The choice representations of c_i and c_j are passed as input to the decomposable attention layer proposed in Parikh et al. (2016). The architecture of the joint model is shown in figure 2.

3.4 Training

We train the stand-alone QA system using the MultiRC and SemEval datasets for respective experiments. We experiment with 2 different training settings for the NLI system. In the first setting, we use SNLI dataset (Bowman et al., 2015) to train the NLI system. The sequence-attention layer is left untrained during this phase. Hence, we only use the answer choice and do not consider the question for NLI detection.

Self-Training: Subsequently, to help the system adapt to our settings, we devise a self-training protocol over the RC datasets to train the NLI system. Self-training examples for the NLI system were obtained using the following procedure: if the SNLI-trained NLI model predicted entailment and the gold labels of the ordered choice pair were `true-true`, then the choice pair is labeled as *entailment*. Similarly, if the SNLI-trained NLI model predicted contradiction and the gold labels of the ordered choice pair were `true-false`, then the choice pair is labeled as *contradiction*. This is noisy labelling as the labels do not directly indicate the presence of NLI relations between the choices. The NLI model was additionally trained using this data.

Model	**Entailment**	**Contradiction**	**Overall**
NLI_{SNLI}	40.80	74.25	55.11
$NLI_{MultiRC}$	57.30	69.22	66.31

Table 1: Accuracy of entailment and contradiction detection on the development set of self-training data for NLI model trained on SNLI data (NLI_{SNLI}) vs training set of self-training data ($NLI_{MultiRC}$)

To train the joint model we use ordered choice pairs, labeled as *entailment* if the gold labels are `true-true` and labeled as *contradiction* if the gold labels are `true-false`. This data was also used to test the effectiveness of the self-training procedure. The results on the development set of MultiRC dataset are in table 1.

The NLI model trained on SNLI dataset achieves 55.11% accuracy. Training the NLI model on the data from MultiRC data increases the overall accuracy to 66.31%. Further discussion about self-training is provided in section 5.

4 Experiments

We perform experiments in four phases. In the first phase, we evaluate the stand-alone QA system. In the second phase, we train the NLI system on SNLI data and evaluate the approach shown in figure 1. In the third phase, we train the NLI system using the self-training data. In the fourth phase, we evaluate the proposed joint model. We evaluate all models on MultiRC dataset. The results are shown in table 2. We evaluate the joint model on SemEval dataset, shown in table 3.

4.1 Datasets

We use two datasets for our experiments, MultiRC dataset[2] and the SemEval 2018 task 11 dataset[3]. MultiRC dataset consisted of a training and development set with a hidden test set. We split the given training set into training and development sets and use the given development set as test set.

Each question in the MultiRC dataset has approximately 5 choices on average. Multiple of them may be `true` for a given question. The training split of MultiRC consisted of 433 paragraphs and $4,853$ questions with $25,818$ answer choices. The development split has 23 paragraphs and 275 questions with $1,410$ answer choices. Test set has 83 paragraphs and 953 questions with $4,848$ answer choices.

SemEval dataset has 2 choices for each question, exactly one of them is `true`. The training set consists of $1,470$ paragraphs with $9,731$ questions. The development set has 219 paragraphs with $1,411$ questions. And the test set has 430 paragraphs with $2,797$ questions.

4.2 Evaluation Metrics

For MultiRC dataset, we use two metrics for evaluating our approach, namely $EM0$ and $EM1$. $EM0$ refers to the percentage of questions for which all the choices have been correctly classified. $EM1$ is the the percentage of questions for which at most one choice is wrongly classified. For the SemEval dataset, we use *accuracy* metric.

4.3 Results

Results of our experiments are summarized in tables 2 & 3. $EM0$ on MC task improves from 18.15% to 19.41% when we use the NLI model trained over SNLI data and it further improves to 21.62% when we use MultiRC self-training data. Joint model achieves 20.36% on $EM0$ but achieves the highest $EM1$ of 57.08%. Human $EM0$ is 56.56%.

Method	**EM0**	**EM1**
Stand-alone QA	18.15	52.99
QA + NLI_{SNLI}	19.41	56.13
QA + $NLI_{MultiRC}$	21.62	55.72
Joint Model	20.36	57.08
Human	56.56	83.84

Table 2: Summary of the results on MultiRC dataset. $EM0$ is the percentage of questions for which all the choices are correct. $EM1$ is the the percentage of questions for which at most one choice is wrong.

Results of SemEval experiments are summarized in table 3. TriAN-single results are as re-

[2] http://cogcomp.org/multirc/

[3] https://competitions.codalab.org/competitions/17184

ported in (Wang et al., 2018). The results we obtained using their implementation are stand-alone QA results. With the same setting, joint model got 85.4% on dev set and 82.1% on test set. The difference in performance of the models in tables 2 and 3 is statistically significant according to McNemar's chi-squared test.

Model	Dev	Test
TriAN-single (Wang et al., 2018)	83.84%	81.94%
Stand-alone QA	83.20%	80.80%
Joint Model	85.40%	82.10%

Table 3: Accuracy of various models on SemEval'18 task-11 dataset

5 Discussion

We have shown that capturing the relationship between various answer choices or subsequent questions helps in answering questions better. Our experimental results, shown in tables 2 & 3, are only a first step towards leveraging this relationship to help construct better machine reading systems. We suggest two possible extensions to our model, that would help realize the potential of these relations.

1. Improving the performance of entailment and contradiction detection.
2. Using the information given in the text to identify the relations between choices better.

As shown in table 1, identification of entailment/contradiction is far from perfect. Entailment detection is particularly worse because often the system returns *entailment* when there is a high lexical overlap. Moreover, the presence of a strong negation word (*not*) causes the NLI system to predict *contradiction* even for *entailment* and *neutral* cases. This issue impedes the performance of our model on SemEval'18 dataset as roughly 40% of the questions have *yes/no* answers. Naik et al. (2018) show that this is a common issue with state-of-the-art NLI detection models.

Self-training (table 1) results suggest that there are other types of relationships present among answer choice pairs that do not come under the strict definitions of *entailment* or *contradiction*. Upon investigating, we found that although some answer hypotheses do not directly have an inference relation between them, they might be related in context of the given text. For example, consider the sentence, '*I snack when I shop*' and the answer choices: c_1: '*She went shopping this extended weekend*' and c_2: '*She ate a lot of junk food recently*'. Although the sentences don't have an explicit relationship when considered in isolation, the text suggests that c_1 might entail c_2. Capturing these kinds of relationships could potentially improve MC further.

6 Conclusion

In this paper we take a first step towards modeling an accumulative knowledge state for machine comprehension, ensuring consistency between the model's answers. We show that by adapting NLI to the MC task using self-training, performance over multiple tasks improves.

In the future, we intend to generalize our model to other relationships beyond strict entailment and contradiction relations.

Acknowledgements

We would like to thank the reviewers for their insightful comments. This work was partially supported by the NSF through grant NSF-1814105.

References

Lisa Bauer, Yicheng Wang, and Mohit Bansal. 2018. Commonsense for generative multi-hop question answering tasks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4220–4230, Brussels, Belgium. Association for Computational Linguistics.

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. Association for Computational Linguistics.

Rajarshi Das, Manzil Zaheer, Siva Reddy, and Andrew McCallum. 2017. Question answering on knowledge bases and text using universal schema and memory networks. *CoRR*, abs/1704.08384.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *NAACL*.

Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. 2016. Ask me anything: Dynamic memory networks for natural language processing. In *International Conference on Machine Learning*, pages 1378–1387.

Fei Liu and Julien Perez. 2017. Gated end-to-end memory networks. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, volume 1, pages 1–10.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*.

Todor Mihaylov and Anette Frank. 2018. Knowledgeable reader: Enhancing cloze-style reading comprehension with external commonsense knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 821–832, Melbourne, Australia. Association for Computational Linguistics.

Sewon Min, Victor Zhong, Richard Socher, and Caiming Xiong. 2018. Efficient and robust question answering from minimal context over documents. *CoRR*, abs/1805.08092.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. *CoRR*, abs/1611.09268.

Simon Ostermann, Michael Roth, Ashutosh Modi, Stefan Thater, and Manfred Pinkal. 2018. Semeval-2018 task 11: Machine comprehension using commonsense knowledge. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 747–757. Association for Computational Linguistics.

Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255, Austin, Texas. Association for Computational Linguistics.

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, Doha, Qatar. Association for Computational Linguistics.

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

Matthew Richardson. 2013. Mctest: A challenge dataset for the open-domain machine comprehension of text.

Mrinmaya Sachan and Eric Xing. 2016. Machine comprehension using rich semantic representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 486–492.

Yawei Sun, Gong Cheng, and Yuzhong Qu. 2018. Reading comprehension with graph-based temporal-casual reasoning. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 806–817. Association for Computational Linguistics.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2016a. Newsqa: A machine comprehension dataset. *CoRR*, abs/1611.09830.

Adam Trischler, Zheng Ye, Xingdi Yuan, Jing He, Phillip Bachman, and Kaheer Suleman. 2016b. A parallel-hierarchical model for machine comprehension on sparse data. *CoRR*, abs/1603.08884.

Liang Wang, Meng Sun, Wei Zhao, Kewei Shen, and Liu Jingming. 2018. Yuanfudao at semeval-2018 task 11: Three-way attention and relational knowledge for commonsense machine comprehension. *CoRR*, abs/1803.00191.

Dirk Weissenborn, Tomas Kocisky, and Chris Dyer. 2017. Reading twice for natural language understanding. *CoRR*, abs/1706.02596.

Jason Weston, Antoine Bordes, Sumit Chopra, and Tomas Mikolov. 2015. Towards ai-complete question answering: A set of prerequisite toy tasks. *CoRR*, abs/1502.05698.

Caiming Xiong, Victor Zhong, and Richard Socher. 2016. Dynamic coattention networks for question answering. *arXiv preprint arXiv:1611.01604*.

Mark Yatskar. 2018. A qualitative comparison of coqa, squad 2.0 and quac. *arXiv preprint arXiv:1809.10735*.

Xiao Zhang, Maria Leonor Pacheco, Chang Li, and Dan Goldwasser. 2016. Introducing DRAIL - a step towards declarative deep relational learning. In *Proceedings of the Workshop on Structured Prediction for NLP@EMNLP 2016, Austin, TX, USA, November 5, 2016*, pages 54–62.