ECNU_ICA_1 at SemEval-2021 Task 4: Knowledge-Enhanced Graph Attention Networks for Reading Comprehension of Abstract Meaning

Pingsheng Liu, Linlin Wang*, Qian Zhao, Hao Chen, Yuxi Feng, Xin Lin*, liang he
School of Computer Science and Technology,
East China Normal University, Shanghai 200062, China
{51205901014,51205901129,10175102236}@stu.ecnu.edu.cn
im0qianqian@ica.stc.sh.cn
{llwang,xlin,lhe}@cs.ecnu.edu.cn

Abstract

This paper describes our system ECNU_ICA_1 for SemEval-2021 Task 4: Reading Comprehension of Abstract Meaning. For this task, we utilize knowledge-enhanced Graph Attention Networks with a novel semantic space transformation strategy. It leverages heterogeneous knowledge to learn adequate evidences, and seeks for an effective semantic space of abstract concepts to better improve the ability of a machine in understanding abstract meanings of natural language. Experimental results show that our system achieves strong performance on this task in terms of both imperceptibility and nonspecificity.

1 Introduction

Recent years have witnessed the remarkable success of pre-trained language models in machine reading comprehension (MRC). Nevertheless, new research points out that these dominant approaches rely heavily on superficial text pattern-matching heuristics to achieve shallow comprehension on natural language (Zhang et al., 2020). For humans, the basic ability to represent abstract concepts guarantees an in-depth understanding of natural language. Consequently, teaching machines to better comprehend abstract meaning is a significant and urgent step to push the frontier technique of MRC forward.

If computers can understand passages as human do, we expect them to accurately predict abstract words that people can use in summaries of the given passages. Thus, researchers have recently proposed a reading comprehension of abstract meaning (ReCAM) task in SemEval 2021. Unlike some previous datasets such as CNN/Daily Mail (Hermann et al., 2015) that request computers to predict concrete concepts, e.g., named entities, ReCAM requires machines to fill out abstract words removed from human written summaries. In ReCAM, subtask 1 and subtask 2 respectively evaluate the performance of machines towards imperceptibility and nonspecificity, two formal definitions of abstractness in natural language understanding(Spreen and Schulz, 1966; Changizi, 2008). Specifically, concrete words refer to things, events, and properties that we can perceive directly with our senses (Spreen and Schulz, 1966; Coltheart, 1981; Turney et al., 2011), e.g., donut, trees, and red. In contrast, abstract words refer to the ideas and concepts that are distant from immediate perception. Examples for abstract words include objective, culture, and economy. Subtask 1 requires machines to perform reading comprehension of abstract meaning for imperceptible concepts, while subtask 2 concentrates on hypernyms, which is more abstract and different from the concrete concepts (Changizi, 2008).

To better understand the abstract meaning, we utilize the Knowledge-Enhanced Graph Attention Network (KEGAT) architecture with a novel semantic space transformation strategy for ReCAM. It well incorporates structured knowledge base such as ConceptNet (Speer et al., 2017) and exploits a novel representation transformation strategy to improve the ability of machines in natural language understanding. The main contributions of our system are as follows:

- We utilize the KEGAT architecture to accomplish two subtasks in Reading Comprehension of Abstract Meaning, leveraging heterogeneous knowledge resources to provide adequate evidences and relying on Graph Attention Networks for the better reasoning.
- The proposed semantic space transformation strategy seeks for an effective representation mapping from concrete objects to abstract concepts, enabling machines to better understand the abstract meanings of natural language.

*Equal corresponding authors.

Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pages 183–188
Bangkok, Thailand (online), August 5–6, 2021.

- Extensive experiments show that our system achieves strong performance on this task in terms of both imperceptibility and nonspecificity.

2 Methodology

In this section, we describe the framework of our system and propose some strategies to enhance the reasoning ability of the model. An overview of the architecture is depicted in Figure 1.

2.1 Input Module

We cast the ReCAM task as a classification problem. For each instance, we assume that P is the passage, Q is the question, A is the number of candidate options, and O_i stands for the options, where $i \in \{1, 2, \ldots, A\}$. For a specific training instance, we first replace the "@placeholder" in Q with O_i, and thus the resulting question-answer pair can be denoted as QO_i. Then we concatenate the passage and question-answer pairs as [CLS] P [SEP] QO_i [SEP], and denote this converted input as U_i for convenience. Although various approaches can be exploited to encode this U_i, we primarily adopt the basic way, in which tokens are represented with the one-hot vectors and the positional encoding is added, providing the model with a new embedding as E_{U_i} for every U_i.

2.2 Reasoning Module

Since pre-trained language models have achieved state-of-the-art performance in various NLP tasks (Devlin et al., 2019; Yang et al., 2019; Lan et al., 2020), we adopt the pre-trained architecture to process the embedding E_{U_i} that is obtained from the previous step to get the high-level representation as $\hat{E}^{base}_{U_i}$. Specifically, we use Electra (Clark et al., 2020), a word-sensitive pre-trained language model which is composed of N-layer transformer encoders (Vaswani et al., 2017) depicted in the middle of Figure 1. Then, we utilize a Knowledge-Enhanced Graph Attention Network (KEGAT) component to accomplish the reasoning process based on all relevant entities and the high-level representation of the entire question-answer pair from the pre-trained model. The working principle of our KEGAT model is introduced later.

As shown in Figure 1, our KEGAT model mainly consists of a Graph Attention Network, a self-attention submodule and a multi-layer perceptron (MLP). It enables a multi-level reasoning process from entities to sentences. For the entity level, we utilize some structured knowledge from ConceptNet with a different integration approach to achieve the goal of conducting inferences over new constructed subgraphs. Here, we adopt the N-gram method to extract all entities from the converted input U_i, and use edge weight as the probability to select a maximum of k adjacent nodes from ConceptNet for subgraph construction. Suppose the number of entities is n, we construct n subgraphs in total, and the subgraphs may be connected with edges. Next, we utilize the conceptnet-numberbatch* to obtain the i-th entity embedding as the initial representation $h_i^{(0)}$, which is subsequently refined by the L-layer Graph Attention Network (GAT). In the refinement process, the GAT module automatically learns an optimal edge weight between two entities in these subgraphs based on the ReCAM task, indicating the relevance of adjacent entities to every central entity. In other word, for a central entity, the GAT tries to only assign higher weight values to those edges connected with several most reasonable adjacent entities from the constructed subgraph, and discards some irreverent edges. Thus, the abstract semantic inference ability of our model is highly improved with the knowledge incorporated by the refined subgraphs. The working principle of our GAT is in Eq. 1–3.

$$h_i^{(l+1)} = \sigma\left(\frac{1}{M}\sum_{m=1}^{M}\sum_{j\in\mathcal{N}_i}\alpha_{ij_m}^{(l)}\mathbf{W}_m^{(l)}h_j^{(l)}\right) \quad (1)$$

$$\alpha_{ij}^{(l)} = \mathrm{softmax}_j\Big(\mathrm{f}([\mathbf{W}^{(l)}h_i^{(l)};\mathbf{W}^{(l)}h_j^{(l)}])\Big) \quad (2)$$

We update each entity node based on Eq. 1, where $\sigma(\cdot)$ represents a ELU function (Clevert et al., 2016), $\mathbf{W}$ is the network parameter, $h_i^{(l)}$ is the representation from the l-th layer of GAT, and $\mathcal{N}_i$ stands for all adjacent nodes to the i-th entity. M is the number of independent attention mechanisms in Eq. 2, and $a_{ij}^{(l)}$ is the relevance degree of the j-th adjacent entity with respect to the i-th entity. Besides, $\mathrm{f}(\cdot)$ represents a projection function converting the vector to a real number, and $[;]$ stands for the concatenation operation. Finally, we define

$$\hat{E}^{gnn}_{U_i} = \frac{1}{n}\sum_{i=1}^{n}h_i^{(L)} \quad (3)$$

to be the final representation for entity subgraphs that are obtained from the GAT.

*ConceptNet-Numberbatch: https://github.com/commonsense/conceptnet-numberbatch

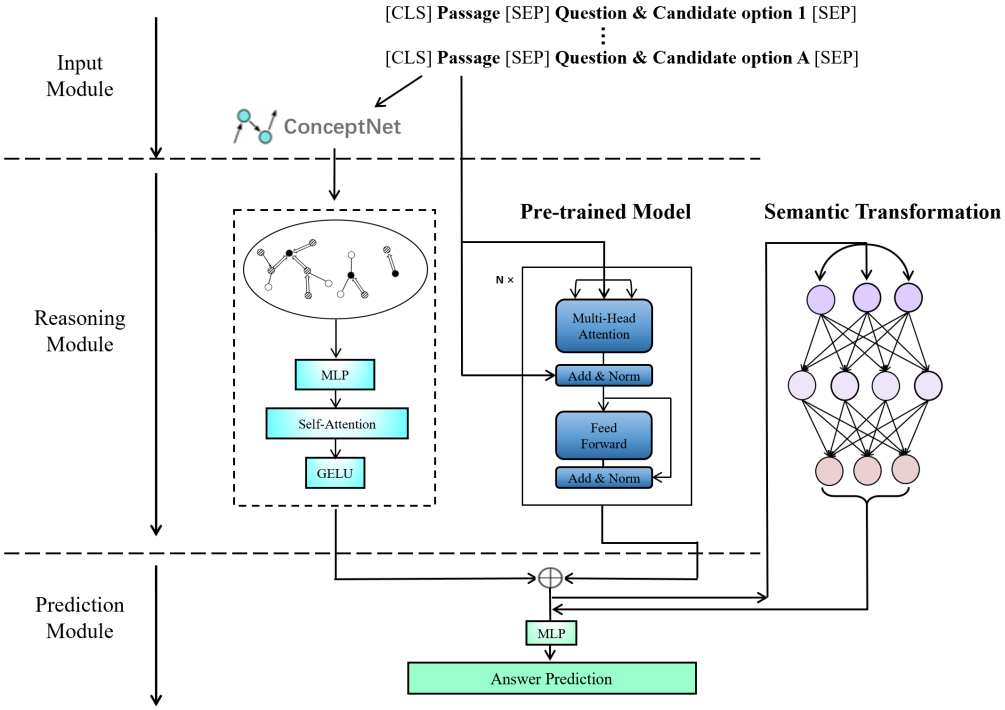

Figure 1: The overview of our system for ReCAM.

From the sentence level, we adopt a self-attention submodule and several MLPs to promote the model to reason over both entities and input sentences. We first utilize a MLP to fuse the symbolic and semantic representations and then take a self-attention operation for refinement. Thus, the entity-level representation can be further refined by taking the question-answer pair as a reference. To sum up, some valuable dimensions can be highlighted to retain the most reasonable information from the fused representations $\hat{E}_{U_i}^{all}$ to improve the reasoning ability. We formulate these steps as Eq. 4 and Eq. 5.

$$\hat{E}_{U_i}^{all} = \text{MLP}([\hat{E}_{U_i}^{base}; \hat{E}_{U_i}^{gnn}]) \quad (4)$$

$$G_{U_i} = \sigma(\text{SelfAttn}(\hat{E}_{U_i}^{all})) \quad (5)$$

where G_{U_i} is the refined representation, SelfAttn$(\cdot)$ represents a self-attention operation, and $\sigma(\cdot)$ is the activation function. Finally, we concatenate G_{U_i} and $\hat{E}_{U_i}^{base}$ to obtain the entire reasoning representation as

$$\hat{E}_{U_i} = [G_{U_i}; \hat{E}_{U_i}^{base}] \quad (6)$$

2.3 Prediction Module

With the previous multi-level reasoning process, we obtain the representation of converted inputs as $\{\hat{E}_{U_i}\}_{i=1}^{A}$ for each instance. In the prediction module, we use a multi-layer perceptron to solve the downstream tasks of ReCAM based on Eq. 7–9.

$$P_i = \text{MLP}(\hat{E}_{U_i}), \quad P' = \text{softmax}(P) \quad (7)$$

$$y = \arg\max(P') \quad (8)$$

$$\mathcal{L} = -\sum_{i=1}^{A} y_i^* \log P_i' \quad (9)$$

where y represents the prediction result, and P_i' stands for the probability of selecting the i-th option label. P is the output of the MLP, where $P \in \mathbb{R}^{A\times 1}$. $\mathcal{L}$ is the training objective to minimize negative log-likelihood and y^* here stands for one-hot vector of the optimal label.

2.4 Adaptive Strategies

Noise Reduction Strategy Previous methods of knowledge integration often lead to inevitable noise (Zhong et al., 2019; Wang et al., 2019), and it is still an open research problem to balance the impact between noise and the amount of incorporated knowledge. (Weissenborn et al., 2018; Khashabi et al., 2017). Our KEGAT can alleviate the noise that is caused by incorporated structured knowledge to a certain extent. This module accomplishes the goal of identifying the most reasonable external entities and discarding the irreverent ones. For

example, we rely on both entity-level and sentence-level inference thoroughly that is discussed in the previous Reasoning Module part to achieve this goal. Furthermore, we remove several unimportant types of edges to avoid unnecessary noises, such as "/r/DistinctFrom", "/r/ExternalURL", etc.

Semantic Space Transformation Strategy Unlike some previous MRC tasks that request computers to predict concrete concepts, ReCAM task here asks models to fill out abstract words removed from human written summaries. Thus, we utilize a semantic space transformation strategy to convert ordinary semantic representation into abstract representation for classification. Specifically, for the final answer prediction, this approach deals with the hidden vector representation V which is obtained ahead of the prediction module. One method is to extend the dimension (ED) of V. For instance, we use a MLP to expand V by 500 dimensions and then perform the downstream classification prediction. The second attempt is to transform V directly with a nonlinear activation function, such as RELU. And another method is to transform V through a simple deep neural network (DNN), which is depicted in the right of Figure 1.

3 Experiments

3.1 Datasets and Metric

In the ReCAM task, it requires the model to fill out abstract words removed from human written summaries. The total number of abstract words that can be selected is five. We utilize Accuracy as a metric to evaluate model performance.

3.2 Experimental Settings

In our experiment, we set the maximum sentence length as 210 and the batch size as 16. During training, we freeze all layers and learn 2 epochs with a learning rate of 0.001 except for the last classification layer, In the fine-tuning phase, we unfreeze all layers and learn 10 epochs with a learning rate of 0.000005. Like the training phase, it is beneficial to use the weights of the pre-trained language model to correct the randomly initialized classification layer. All layers of the entire model in the fine-tuning phase are suitable for classifying downstream tasks with the low learning rate. For each phase, we save model parameters when it reaches the highest accuracy on the dev set, and load it at the beginning of the next phase. In addition, we adopt the Adam optimizer (Kingma and Ba, 2015) and set epsilon to be 0.000001 for the gradient descent. We train our model with Titan XP GPUs.

3.3 Results

Table 1 shows the results of the top five teams from the leaderboard for ReCAM task (by February 10). Our system achieves the 3rd place in Subtask 1 in terms of Accuracy. And it can be concluded from Table 2 that our system has the ability to solve the ReCAM task.

Besides, we test the performance of our system with the strategies mentioned in Section 2.4. Here, "+KEGAT" represents our proposed model with Knowledge-Enhanced Graph Attention Networks, "+ED ", "+RELU", "+DNN" refer to our system with different semantic space transformation strategies. In addition, Dev Acc. and Test Acc. stand for the accuracy on the dev set and test set respectively. Table 2 shows the experimental results of our system on the ReCAM task. In this table, the baseline model GA Reader provided by the competition organizer is not ideal, and its performance is slightly higher than 20% with our actual testing. We conclude that on the dev set, our system respectively achieves the relative improvement of 6.69% and 4.24% on subtask 1 and subtask 2 when adding KEGAT submodule compared with the fine-tuned Roberta large. Moreover, we test the performance of three ensemble models shown in the bottom of 2, and the "Electra-large ED + Electra-large KEGAT-RELU " ensemble obtains the best performance on the dev set, which respectively outperforms the fine-tuned Roberta large model with the relative improvement of 7.41% and 5.29% on subtask 1 and subtask 2. Here, this ensemble framework refers to the combination of two models. Therefore, it can be concluded that the ensemble models with the semantic space transformation strategy greatly improve the reasoning ability of our system, and the single system with multiple strategies performs well in most cases.

3.4 Further Discussion

To further investigate this task, we have additionally assessed the impact of data bias on the model performance. By statistics, the average length of passages in the dev sets of subtask 1 and subtask 2 are 268.8 and 434.6, respectively. In general, longer passages often consist of more noise that greatly influences answer reasoning process of the model. We only select a portion of contents from

Subtask 1			Subtask 2		
Rank	**Team Name**	**Accuracy**	**Rank**	**Team Name**	**Accuracy**
1	Silvilla	95.11	1	PINGAN Omini-Sinitic	95.29
2	PINGAN Omini-Sinitic	93.04	2	Silvilla	94.89
3	**ECNU_ICA_1 (ours)**	**90.47**	3	tt123	93.41
4	tt123	89.98	**4**	**ECNU_ICA_1 (ours)**	**93.01**
5	cxn	88.69	5	cxn	92.91

Table 1: Top 5 results for ReCAM task.

	Subtask 1		Subtask 2	
Model	**Dev Acc.(%)**	**Test Acc.(%)**	**Dev Acc.(%)**	**Test Acc.(%)**
GA Reader	24.61	-	22.79	-
Our Architectures				
-w fine-tuned Roberta-large	85.18	-	87.30	-
-w Electra-large	90.80	89.28	91.07	90.48
-w Electra-large + KEGAT	91.87	89.37	91.54	92.01
-w Electra-large + KEGAT-RELU	92.35	90.37	91.89	92.11
-w Electra-large + ED	91.51	90.12	91.65	90.95
-w Electra-large + DNN	91.40	-	91.77	-
Ensemble Models -w Electra-large				
+ KEGAT	91.99	-	92.36	-
ED + KEGAT	92.47	-	92.48	-
ED + KEGAT-RELU	**92.59**	**90.47**	**92.59**	**93.01**

Table 2: Experimental results of ReCAM task.

	Subtask1	**Subtask2**
AVG length	268.8	434.6
Position	**Dev Acc.(%)**	**Dev Acc.(%)**
0-210	90.80	91.07
211-420	89.31	89.65

Table 3: Performance on different contents of passage.

the given passage for this assessment instead of the whole passage. Specially, in the given dataset, we take a fixed length of 210 as the content interval by intercepting it at two different positions, namely token ID 0 $\sim$ 210 and token ID 211 $\sim$ 420. Then we fine-tune the Electra-large model for each subtask using their own training set and compare the performance of the fine-tuned Electra model on two different passage intervals. It means that we have conducted experiments with different passage contents twice. Table 3 reports the results of our system on these different passage intervals. In this table, compared to the experiment that adopts the passage content with position from 0 to 210, intercepting the one with position from 211 to 420 leads the performance to drop by about 1 $\sim$ 2% on these two subtasks. Thus, we conclude that the positional bias indeed affects model performance to some extent.

4 Conclusion

We utilize a knowledge-Enhanced Graph Attention Network architecture with semantic transformation strategies for machines to better comprehend the abstract meanings of natural language. It well incorporates heterogeneous knowledge and relies on Graph Attention Networks to learn adequate evidences. The subsequent semantic transformation enables an effective representation mapping from concrete objects to abstract concepts. Our system achieves strong performance on this comprehension task in terms of both imperceptibility and non-specificity. We hope this work can shed some lights on the study of in-depth reading comprehension.

Acknowledgements

This work was supported by the National Innovation 2030 Major ST Project of China, the Fundamental Research Funds for the Central Universities, National Natural Science Foundation of China (No.62006077), Shanghai Sailing Program (No.20YF1411800), and the Science and Technology Commission of Shanghai Municipality (No.20511105102).

References

Mark A. Changizi. 2008. Economically organized hierarchies in wordnet and the oxford english dictionary. *Cognitive Systems Research*, 9(3):214–228.

Kevin Clark, Minh-Thang Luong, V. Quoc Le, and D. Christopher Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *ICLR*.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. 2016. Fast and accurate deep network learning by exponential linear units (elus). In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Max Coltheart. 1981. The mrc psycholinguistic database. *The Quarterly Journal of Experimental Psychology Section A*, 33(4):497–505.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, page 1693–1701, Cambridge, MA, USA. MIT Press.

Daniel Khashabi, Tushar Khot, Ashish Sabharwal, and Dan Roth. 2017. Learning what is essential in questions. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 80–89, Vancouver, Canada. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 4444–4451. AAAI Press.

Otfried Spreen and Rudolph W. Schulz. 1966. Parameters of abstraction, meaningfulness, and pronunciability for 329 nouns. *Journal of Verbal Learning and Verbal Behavior*, 5(5):459–468.

Peter Turney, Yair Neuman, Dan Assaf, and Yohai Cohen. 2011. Literal and metaphorical sense identification through concrete and abstract context. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 680–690, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Xiaoyan Wang, Pavan Kapanipathi, Ryan Musa, Mo Yu, Kartik Talamadupula, Ibrahim Abdelaziz, Maria Chang, Achille Fokoue, Bassem Makni, Nicholas Mattei, and Michael Witbrock. 2019. Improving natural language inference using external knowledge in the science questions domain. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):7208–7215.

Dirk Weissenborn, Tomas Kocisky, and Chris Dyer. 2018. Dynamic integration of background knowledge in neural NLU systems.

Zhilin Yang, Zihang Dai, Yiming Yang, G. Jaime Carbonell, Ruslan Salakhutdinov, and V. Quoc Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS 32 (NIPS 2019)*, pages 5754–5764.

Zhuosheng Zhang, Hai Zhao, and Rui Wang. 2020. Machine reading comprehension: The role of contextualized language models and beyond. *arXiv preprint arXiv:2005.06249*.

Wanjun Zhong, Duyu Tang, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. 2019. Improving question answering by commonsense-based pre-training. In *Natural Language Processing and Chinese Computing*, pages 16–28, Cham. Springer International Publishing.