XRJL-HKUST at SemEval-2021 Task 4: WordNet-Enhanced Dual Multi-head Co-Attention for Reading Comprehension of Abstract Meaning

Yuxin Jiang*, **Ziyi Shou***, **Qijun Wang**, **Hao Wu**, **Fangzhen Lin**
HKUST-Xiaoi Robot Joint Lab on Machine Learning and Cognitive Reasoning
Department of Computer Science and Engineering
The Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong
{yjiangcm,zshou,qwangcd,hwubx}@connect.ust.hk,flin@cse.ust.hk

Abstract

This paper presents our submitted system to SemEval 2021 Task 4: *Reading Comprehension of Abstract Meaning*. Our system uses a large pre-trained language model as the encoder and an additional dual multi-head co-attention layer to strengthen the relationship between passages and question-answer pairs, following the current state-of-the-art model DUMA. The main difference is that we stack the passage-question and question-passage attention modules instead of calculating parallelly to simulate re-considering process. We also add a layer normalization module to improve the performance of our model. Furthermore, to incorporate our known knowledge about abstract concepts, we retrieve the definitions of candidate answers from WordNet and feed them to the model as extra inputs. Our system, called **W**ord**N**et-enhanced **DU**al **M**ulti-head Co-**A**ttention (WN-DUMA), achieves 86.67% and 89.99% accuracy on the official blind test set of subtask 1 and subtask 2 respectively.

1 Introduction

Recently, there has been an increasing interest on Machine Reading Comprehension (MRC). While most MRC studies such as CNN/Daily Mail (Hermann et al., 2015) focus on concrete concepts, SemEval 2021 Task 4, **Re**ading **C**omprehension on **A**bstract **M**eaning (ReCAM), targets abstract concept understanding, including *imperceptibility* in subtask 1 and *nonspecificity* in subtask 2. The former, imperceptibility, highlights the abstract words that refer to ideas and concepts that do not correspond directly to human perception. The latter is for hypernyms and abstract concepts such as the class of vertebrate which includes whales as a concrete subclass (Changizi, 2008).

*Equal contribution.

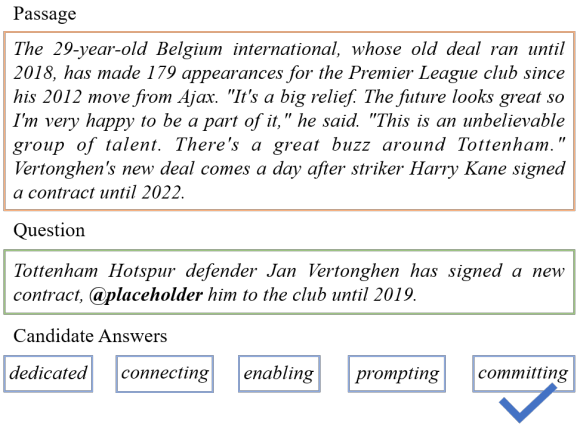

Figure 1: An example of ReCAM subtask 1.

In this task, given news fragments and incomplete abstracts, the machine needs to select the most suitable abstract words from candidate answers. Figure 1 shows one example of ReCAM subtask 1. Passage is the news sections and Question is a human written summary in which abstract words have been removed. Machines are requested to choose abstract words from five candidates for replacing *@placeholder*.

For this shared task, we regard both subtasks as multi-choice MRC tasks. Various deep neural networks and attention mechanisms (e.g. (Dhingra et al., 2017; Wang et al., 2018; Zhang et al., 2020a,b; Jin et al., 2020)) have been proposed to address these tasks. In our work, following the state-of-the-art model DUMA (Zhu et al., 2020), we adopt a Pre-trained Language Model (PrLM) as encoder and extend with an additional dual multi-head co-attention layer to strengthen the relationship between passages and question-answer pairs. For the dual multi-head attention layer, while DUMA builds passage-question and question-passage attention modules in a parallel way to simulate the transposition thinking process,

Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pages 793–798
Bangkok, Thailand (online), August 5–6, 2021.

our model stacks two attention modules in order to simulate the process of re-considering for a deeper understanding of the passage. More details on our attention calculation process can be found in Section 2. Furthermore, we add an additional layer normalization module immediately after the attention module. From our experiments, we found that this additional normalization module definitely improves our model's performance.

Our most significant design decision is to use WordNet (Miller, 1995) to expand on the abstract concepts in the candidate answers. Intuitively, expanding an abstract concept according to its definition in a dictionary should help as it helps relate the abstract concept with others that may occur in the text. A key conclusion that we can draw from our experiments is that this is indeed the case. One problem that we encountered when implementing this idea was that most English words have multiple entries in WordNet. For example, *bank* in WordNet can have Noun definitions as well as Verb definitions. We addressed this problem by using some heuristics and some additional information such as part-of-speech labels. Because of the significant role played by WordNet, we call our system **W**ord**N**et-enhanced **DU**al **M**ulti-head Co-**A**ttention (WN-DUMA).

We remark that our system did not use any additional training data for the tasks. In the final evaluation, our model is ranked 10 out of 23 and 9 out of 28 on the official subtask 1 and subtask 2 blind test set with 86.67% and 89.99% accuracy, respectively. The code for our model is publicly available[1].

The rest of the paper is organized as follows. Section 2 gives the details of our system. Section 3 describes our experimental setup including the datasets and hyper parameters used for training. Section 4 presents experimental results. Section 5 concludes this paper with some final remarks.

2 System Description

In this section, We describe the framework of our end-to-end model WN-DUMA. Figure 2 depicts the detailed architecture of our approach, with inputs at the bottom and outputs at the top.

WordNet-enhanced Encoder We regard both subtask 1 and 2 as multi-choice MRC problems. Such a problem includes a passage, a question with a *@placeholder*, and 5 candidate answers to choose from. First, we replace *@placeholder* in the question with the given 5 candidate answers to form 5 options. In the tasks, the candidate answers are all single words with abstract meanings, so we decided to add some extra knowledge from WordNet (Miller, 1995) to help the system better understanding the abstract meanings. More specifically, for a single candidate answer, we find its part-of-speech tag based on the option it's located in, and extract its definitions under this part-of-speech tag. After tokenization, every instance is cast into the input form: [CLS] *passage* [SEP] *option* + *answer definition* [SEP]. To encode input tokens into representations, we feed them through a PrLM based on Transformer to obtain sequence embeddings, which draws a global relationship between the passage and the option-definition.

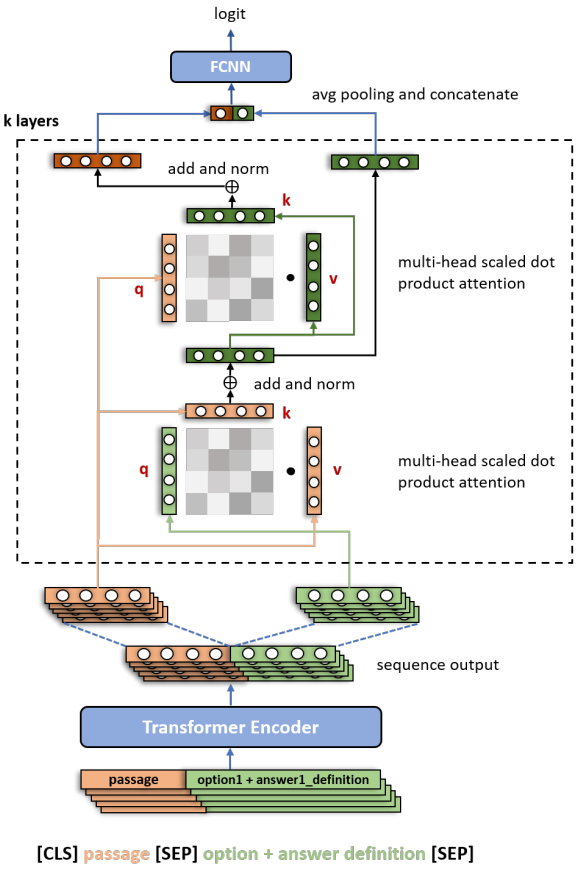

Figure 2: The overall model architecture.

Dual Multi-head Co-Attention Layer Based on the above process, we further separate the output representations from transformer encoder to acquire the passage context embeddings $E^P \in \mathbb{R}^{d_{model} \times l_p}$ and the context embeddings of option-

[1] https://github.com/zzshou/RCAM

definition $E^{OD} \in \mathbb{R}^{d_{model} \times l_{od}}$, where l_p, l_{od} denote the maximum length of passage and option-definition respectively. Then based on the bi-directional matching network of DUMA which is quite similar to the multi-head self-attention module in vanilla transformer block (Vaswani et al., 2017), we first take E^{OD} as Query, E^P as Key and Value to calculate one of the co-attention representations, which simulates the process of human re-reading the passage with impression of option and definition. The formulas are listed as follows:

$$Q_i = E^{OD} W_i^Q \quad (1)$$

$$K_i = E^P W_i^K \quad (2)$$

$$V_i = E^P W_i^V \quad (3)$$

$$head_i = softmax(\frac{Q_i K_i^T}{\sqrt{d_k}}) V_i \quad (4)$$

$$MHA = Concat(head_1, ..., head_h) W^O \quad (5)$$

$$REP_1 = Normalize(E^{OD} + MHA) \quad (6)$$

where $W_i^Q \in \mathbb{R}^{d_{model} \times d_q}$, $W_i^K \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{model} \times d_v}$, $W_i^O \in \mathbb{R}^{hd_v \times d_{model}}$ are linear transformations with learnable parameters, d_q, d_k, d_v denote the dimension of *Query*, *Key* and *Value*, h denotes the number of heads. Different from the structure of DUMA, here we make two changes: 1) apply "Add and Normalize" after getting the multi-head attention representation, which could result in more stable training. 2) compute another co-attention representation by stacking rather than paralleling: take the acquired REP_1 as *Key* and *Value*, E^P as *Query*, which simulates the process of re-considering the option-definition with deeper understanding of the passage. Finally, we obtain REP_1 and REP_2, which have the same dimension as E^{OD} and E^P, respectively. As a result, we can stack the co-attention module for k layers.

Classifier Here the co-attention representations REP_1 and REP_2 are merged and used for final classification:

$$I_1 = AvgPool(REP_1) \quad (7)$$

$$I_2 = AvgPool(REP_2) \quad (8)$$

$$M = Concat(I_1, I_2) \quad (9)$$

$$logits = MW_M \quad (10)$$

where $I_1, I_2 \in \mathbb{R}^{d_{model}}$, $M \in \mathbb{R}^{2d_{model}}$, $W_M \in \mathbb{R}^{d_{model} \times n_{class}}$ denotes the one-layer fully-connected neural network, n_{class} denotes the number of candidate answers. Consequently, for a single instance, we could get as many logits as the candidate answers, which are used to compute the corss-entropy loss by softmax.

	Task 1	Task 2
Train	3,227	3,318
Dev	837	851
Test	2,025	2,017
Avg. passage length	270.3	429.7
Avg. question length	24.6	27.1
Vocabulary size	16,318	17,006
Answer vocabulary size	4,333	4,775

Table 1: Basic statistics of subtask 1 and subtask 2 dataset.

3 Experimental Setup

Data and Metric We used the official datasets (Zheng et al., 2021) provided by SemEval 2021 Task 4 competition. They were collected from BBC News in English language. Some basic statistics are listed in Table 1. According to the requirement of the organizers, participants could only use the corresponding dataset for a specific subtask to build models to ensure fairness. For better performance, technics like multi-task learning (Wan, 2020) are recommended for MRC tasks. In both subtask 1 and subtask 2, we utilize accuracy as the metric to evaluate our model performance.

Hyper Parameters All of our codes are written based on PyTorch[2]. To extract the word definition of candidate answers, we use NLTK toolkit (Bird et al., 2009). The transformer encoder we used is pretrained ALBERT-xxlarge-v2 model[3]. Since the code of DUMA is not open-source, we reimplement it by only using one co-attention layer where the attention heads are 64 and the dimension of *Query*, *Key* and *Value* are all 64, because it is pointed that more co-attention layers do not improve the performance (Zhu et al., 2020). The setting is also applied to our WN-DUMA for fair comparison.

Due to limited resources, the maximum sequence length of input tokens is set to 150 for both subtask 1 and subtask 2. In fact, we found that sequence length longer than 150 can only slightly improve the model performance. We choose mini-batch size equal to 2, and the AdamW optimizer

[2] https://pytorch.org/

[3] https://github.com/huggingface/transformers

(Loshchilov and Hutter, 2018) with an initial learning rate of 5e-06. We use some strategies for more stable training: 1) clip the gradient norm to 10; 2) adopt a linear scheduler with warm up of the first 10% training steps. To avoid overfitting, we apply 0.1 dropout (Srivastava et al., 2014) rate to the co-attention layer. We trained all the models for 3 epochs, evaluate on the dev set at every 200 training steps and save the model with the best dev accuracy. For each single model, we run experiments for 5 times with different random seeds and use the average as the ultimate performance.

4 Results

4.1 Quantitative Analysis

Table 2 summarizes the experimental results. The first three models only have encoder (without enhancement of WordNet) and classifier part. It is clearly seen that ALBERT is much more efficient as encoder for abstract meaning understanding. It is worth noting that by only using the question and answer as input, the ALBERT model can also get pretty good results, as table 2 shows. Intuitively, it may be because the model could utilize syntax and semantics of the question sentence to choose the correct answer without looking through the passage.

Compared to $\text{ALBERT}_{\text{xxlarge}}$, adding DUMA layer obtains around 0.2% improvement in subtask 1, and more than 3% improvement in subtask 2. Besides, our WN-DUMA single model achieves further improvements based on DUMA on both subtasks, +0.83% and +1.3% respectively, without increasing the number of parameters. Using a majority vote scheme, we ensemble our WN-DUMA model with different parameters for more stable predictions. Eventually, our ensemble models which get 87.57% on subtask 1 dev set and 90.01% on subtask 2 dev set acquire the best performance on test sets (86.67% and 89.99%, respectively) among our submissions.

Figure 3 and Figure 4 illustrate the dev accuracy of different models on subtask 1 and subtask 2 as the number of training steps increases. It is interesting to observe that models with co-attention layer (DUMA and WN-DUMA) could get over 70% accuracy with only 10% of training examples. While ALBERT model has to be trained with the full dataset to get relatively high accuracy. Consequently, our WN-DUMA model may be useful when there only exists a small amount of training data.

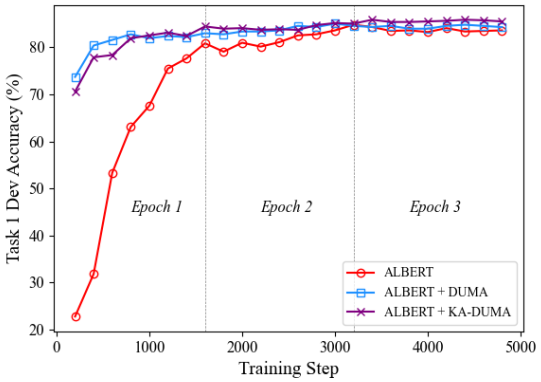

Figure 3: Subtask 1 dev accuracy over number of training steps.

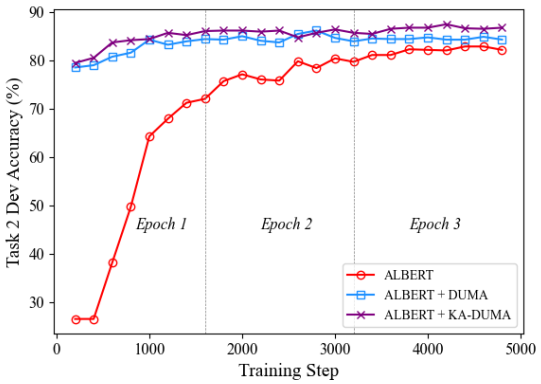

Figure 4: Subtask 2 dev accuracy over number of training steps.

4.2 Error Analysis

In order to further improve our model performance in the future, we analyze some incorrect predictions made by WN-DUMA, and classify them into two categories:

- Candidate answers with similar meanings. In some failure cases, the similarities between candidate answers are too high to distinguish. For example, outstanding and extraordinary, challenge and attempt, etc.
- Lack of commonsense and relying too much on the information of the passage. Due to the fact that the question is the summary of the passage, the machine need to choose the most appropriate answer from a global perspective with some commonsense. However, our model make decisions by only capturing the local information in some cases. A spe-

Model	Task 1 dev	Task 1 test	Task 2 dev	Task 2 test
$\text{BERT}_{\text{large}}$ (Devlin et al., 2019)	67.74	-	69.45	-
$\text{RoBERTa}_{\text{large}}$ (Liu et al., 2019)	74.31	-	74.50	-
$\text{ALBERT}_{\text{xxlarge}}$ (Lan et al., 2020)	84.83	-	82.84	-
$\text{ALBERT}_{\text{xxlarge}}$ + DUMA (Zhu et al., 2020)	85.07	-	86.13	-
$\text{ALBERT}_{\text{xxlarge}}$ (question only)	79.57	-	82.14	-
$\text{ALBERT}_{\text{xxlarge}}$ + WN-DUMA (single)	**85.90**	84.54	**87.43**	86.61
$\text{ALBERT}_{\text{xxlarge}}$ + WN-DUMA (ensemble)	**87.57**	86.67	**90.01**	89.99

Table 2: Model comparison on subtask 1 and subtask 2 dataset.

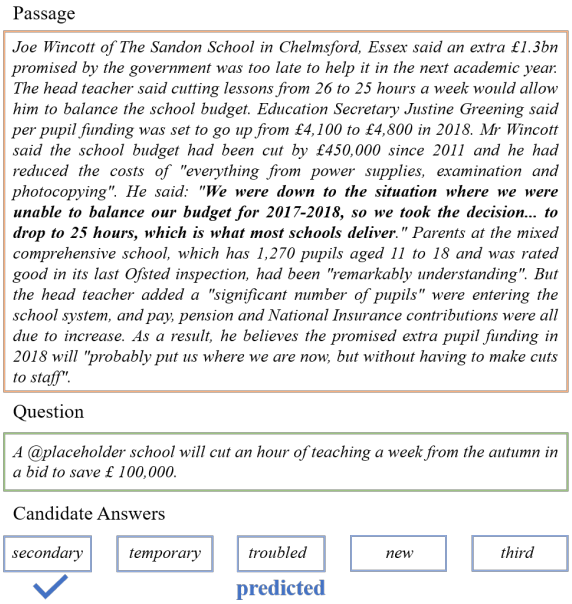

Figure 5: An failure example made by WN-DUMA. The ground true is the answer with a correct mark at the bottom. While the prediction is the answer with "predicted" at its bottom.

cific example can be seen in Figure 5. We can see that the model predicts that the answer is "troubled", most likely because the passage mentions "the school was trapped into financial difficulties".

5 Conclusion

In this paper, we describe our submitted system in SemEval 2021 Task 4 ReCAM. Unlike previous MRC datasets, ReCAM focus more on machine's ability in understanding and representing abstract concepts. In order to provide more knowledge of abstract word, we extract WordNet definitions for each candidate answer based on part-of-speech tags. In addition, our proposed WN-DUMA model consists of a PrLM as the encoder and a dual multi-head co-attention layer to enhance the relationship between passage and question-answer pairs as human's re-considering process. Our WN-DUMA model improves the performance of our baseline model DUMA on these datasets.

There are some limitations in our experiments. Firstly, training data size of this task is limited compared to other MRC tasks, with less than 3400 training pairs in both subtasks. This is understandable as collecting labeled data in many natural language processing tasks is expensive. Secondly, using $\text{ALBERT}_{\text{xxlarge}}$ PrLM, we only set 150 as the maximum text length in our experiments due to device limitation. Important sentences in the passage that are highly relevant to the summary are sometimes not covered. For PrLMs, their performance always improve as the number of their parameters increase. The use of large pre-trained models sometimes requires the sacrifice of context. For our future work, we plan to explore ways to train models more efficiently with limited amount of labeled data, and to design more cost-effective models to deal with long input texts.

References

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly Media.

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

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.

Bhuwan Dhingra, Hanxiao Liu, Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov.

2017. Gated-attention readers for text comprehension. 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 1832–1846. Association for Computational Linguistics.

Karl Moritz Hermann, Tomáš Kočiskỳ, 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*, pages 1693–1701.

Di Jin, Shuyang Gao, Jiun-Yu Kao, Tagyoung Chung, and Dilek Hakkani-tur. 2020. Mmm: Multi-stage multi-task learning for multi-choice reading comprehension. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8010–8017.

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 *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2018. Fixing weight decay regularization in adam.

George A. Miller. 1995. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Hui Wan. 2020. Multi-task learning with multi-head attention for multi-choice reading comprehension. *arXiv preprint arXiv:2003.04992*.

Shuohang Wang, Mo Yu, Jing Jiang, and Shiyu Chang. 2018. A co-matching model for multi-choice reading comprehension. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, pages 746–751. Association for Computational Linguistics.

Shuailiang Zhang, Hai Zhao, Yuwei Wu, Zhuosheng Zhang, Xi Zhou, and Xiang Zhou. 2020a. Dcmn+: Dual co-matching network for multi-choice reading comprehension. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9563–9570.

Zhuosheng Zhang, Yuwei Wu, Junru Zhou, Sufeng Duan, Hai Zhao, and Rui Wang. 2020b. Sg-net: Syntax-guided machine reading comprehension. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9636–9643.

Boyuan Zheng, Xiaoyu Yang, Yuping Ruan, Quan Liu, Zhen-Hua Ling, Si Wei, and Xiaodan Zhu. 2021. SemEval-2021 task 4: Reading comprehension of abstract meaning. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*.

Pengfei Zhu, Hai Zhao, and Xiaoguang Li. 2020. Duma: Reading comprehension with transposition thinking. *arXiv preprint arXiv:2001.09415*.