



Interactive Learning Environments



ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/nile20

Automatic short answer grading by encoding student responses via a graph convolutional network

Hongye Tan, Chong Wang, Qinglong Duan, Yu Lu, Hu Zhang & Ru Li

To cite this article: Hongye Tan, Chong Wang, Qinglong Duan, Yu Lu, Hu Zhang & Ru Li (2023) Automatic short answer grading by encoding student responses via a graph convolutional network, *Interactive Learning Environments*, 31:3, 1636-1650, DOI: [10.1080/10494820.2020.1855207](https://doi.org/10.1080/10494820.2020.1855207)

To link to this article: <https://doi.org/10.1080/10494820.2020.1855207>



Published online: 11 Dec 2020.



[Submit your article to this journal](#)



Article views: 411



[View related articles](#)



[View Crossmark data](#)



Citing articles: 6 [View citing articles](#)



Automatic short answer grading by encoding student responses via a graph convolutional network

Hongye Tan ^a, Chong Wang ^a, Qinglong Duan^b, Yu Lu ^b, Hu Zhang^a and Ru Li^a

^aSchool of Computer and Information Technology, Shanxi University, Taiyuan, People's Republic of China;

^bAdvanced Innovation Centre for Future Education, Beijing Normal University, Beijing, People's Republic of China

ABSTRACT

Automatic short answer grading (ASAG) is a challenging task that aims to predict a score for a given student response. Previous works on ASAG mainly use nonneural or neural methods. However, the former depends on handcrafted features and is limited by its inflexibility and high cost, and the latter ignores global word cooccurrence in a corpus and global interaction among the samples in datasets. However, ASAG requires this global information to learn the different expressions conveying the same meaning and the relations between the expressions and the grading labels. To address these limitations, we explore the use of a two-layer graph convolutional network (GCN) to encode the undirected heterogeneous graph of all student responses. The graph has sentence-level and word/bigram-level nodes. An edge is constructed between two nodes according to their inclusion or cooccurrence relationship. The sentence-level TF-IDF value or the PMI value is calculated as the edge weights to reflect the correlation degree between two nodes. Experimental results on the SemEval-2013 benchmark dataset and a two-subject dataset show that our model performs better.

ARTICLE HISTORY

Received 20 January 2020

Accepted 20 November 2020

KEYWORDS

Automatic short answer grading; graph convolution networks; graph representation

1. Introduction

Compared with multiple-choice questions or yes/no questions, short-answer questions can activate the more complicated reconstructive cognitive process and further promote student learning. However, short-answer questions are less utilized in large-scale education environments due to the relatively low accuracy of automatic short answer grading (ASAG), which aims to predict a score for a given student response.

Most existing methods regard ASAG as a classification or regression task. Some methods build models based on human-designed features (Mohler et al., 2011; Sultan et al., 2016; 2008); others utilize deep learning methods such as convolution neural networks (CNNs) and long short-term memory networks (LSTMs) to learn the representation of student responses and to avoid designing features manually (Alikaniotis et al., 2016; Hassan et al., 2018; Huang et al., 2018; Kumar et al., 2017; Riordan et al., 2017; Yang et al., 2018). However, these deep learning models can capture semantic and grammatical information in local consecutive word sequences but may ignore global word cooccurrence in a corpus (Peng et al., 2018).

Currently, the main challenge of ASAG is that reference answers and the limited training datasets cannot cover all the expressions appearing in student responses, leading to incorrect predictions in ASAG systems. In the student responses in Table 1, the distinct words of “dried up” and “evaporated” (in Example 1) and the different grammatical forms of “B if A” and “A B” (in Example 2) are

Table 1. Examples from the SemEval2013 Dataset.

Example1	<p>Question: How did you separate the salt from the water?</p> <p>Ref. Answer: The water was evaporated, leaving the salt.</p> <p>Correct std. responses:</p> <p>(1) Evaporated and left kind of salt.</p> <p>(2) The water dried up and left the salt.</p>
Example2	<p>Question: Andi and Scott decided to investigate solar water heaters What does the graph tell you about the effect of using a cover?</p> <p>Ref. Answer: Water heats up faster when covered.</p> <p>Correct std. responses:</p> <p>(1) Covered it heats up faster. (syntax form: A B)</p> <p>(2) Water heats up faster if the lid is covered (syntax form: B if A)</p>

used to convey the same meaning. To make correct predictions, the system needs to know “dried up” and “evaporated” have the same meaning in this context, and the two grammatical structures of “B if A” and “A B” are paraphrases of each other.

We propose that if ASAG models can consider the global word cooccurrence and global interaction among student responses, they can capture the relations between the various expressions and the grading labels. Additionally, we believe that the general syntax knowledge, such as “B if A” and “A B” are equivalent, can be learned from examples in different domains, as long as they include similar syntax structures.

Recently, some researchers have used the GCNs (graph convolutional networks) in many natural language processing (NLP) tasks and attained excellent effects (Marcheggiani & Titov, 2017; Sahu et al., 2019; Zhang et al., 2018). The reason is that the GCNs utilize graph embedding to preserve the global graph structure information and realize message passing within a graph.

Inspired by these works, we explore the use of the GCNs for ASAG. We construct an undirected heterogeneous graph and utilize the GCN model to encode the graph and predict grades. The graph has two types of nodes to capture the content information of responses. One type is sentence-level nodes, corresponding to reference answers, human-scored responses, and the student responses to be scored. The other type is word/bigram-level nodes, corresponding to the words and bigrams existing in answers and responses. The graph uses edges to capture the relations among the responses. Two nodes are connected by an edge according to their inclusion or cooccurrence relationships. The edge weight is calculated as the sentence-level TF-IDF (term frequency–inverse document frequency) value or the PMI (pointwise mutual information) value, reflecting the correlation degree between the nodes. Since the graph involves all student responses in the dataset, the GCN model captures the global word cooccurrence and global interaction among the student responses via graph embedding.

Additionally, to avoid overfitting problems caused by the size limitation of datasets, we augment the data by applying the method of back translation, which is an effective practice in data augmentation because it can obtain new data that are greatly different from the original data and improve the model performance.

This work thus makes the following key contributions:

- (1) We explicitly identify the limitation of the existing ASAG models, and first propose to capture the missing global information from the student responses by leveraging on the graph convolutional network.
- (2) We propose a novel two-layer GCN that models the global word co-occurrence and global interaction among students’ responses to properly handle diverse student answers.

- (3) Using the student data from two datasets, we have conducted the comprehensive evaluations and the results show that the proposed model outperforms the baselines, where the data argument technique is utilized to avoid overfitting.

The remainder of this article is organized as follows: We first review the related work of ASAG. Then, we illustrate our model based on the GCNs and validate the model on two datasets. Last, we present the discussion and our conclusions.

2. Related work

2.1. ASAG

The study of ASAG began in 1966 (Page, 1966), and early ASAG works mostly used rule-based methods. For example, (Bachman et al., 2002) utilized reference answers to generate regular expression rules, and each rule corresponds to a score. (Mitchell et al., 2003) proposed the IAT model, in which templates were manually created for both correct answers and wrong answers. However, rule-based methods have low generalizability and scalability due to the rules' limited expressions.

With the development of machine learning, researchers have begun to predict scores with textual classification or regression models. Most traditional machine learning methods focus on extracting various features from human-scored responses to construct models. For example, (Sultan et al., 2016) proposed an ASAG system with the features of text similarities, term weights and length ratios. (Mohler et al., 2011) utilized graph alignment and lexical semantic similarity features for scoring. (Bailey & Meurers, 2008) built the CAM content evaluation module with 13 kinds of features, such as word-level and phrase-level features. To use different subsets of features, (Heilman & Madnani, 2013) adopted ensembles of classifiers. These traditional machine learning approaches generally depend on handcrafted features and are limited by their inflexibility and high costs.

Driven by recent advances in deep learning, researchers have also begun to use word vectors and deep neural networks for ASAG. (Alikaniotis et al., 2016) proposed a model that forms word representations by learning the extent to which specific words contribute to the text's score and used LSTMs to represent the meaning of texts. (Hassan et al., 2018) presented a supervised learning approach for ASAG based on paragraph embeddings. (Kumar et al., 2017) built a neural framework for ASAG with three layers: the Siamese bi-LSTM layer, the Earth mover's distance pooling layer and the regression layer. (Riordan et al., 2017) applied CNNs and LSTMs for ASAG. (Yang et al., 2018) proposed a deep auto-encoder grader (DAGrader) for scoring. (Huang et al., 2018) combined the continuous bag-of-words model (CBOW) with the LSTMs to predict scores. However, most existing ASAG works neglect global word cooccurrence in the dataset and do not model global interaction among student responses.

2.2. Graph neural networks

The graph neural networks (GNNs) were first proposed by (Scarselli et al., 2009) and extend existing neural networks for processing the data represented in graph domains. However, the original GNN is oriented to the simplest graph, consisting of nodes with label information and undirected edges (Zhou et al., 2019). Later, some GNN variants were designed to model different kinds of graphs (such as directed graphs and heterogeneous graphs), and extended the representation capability of the original model to solve many problems in different fields. For example, convolution operators were extended from traditional signal processing to graphs (Ortega et al., 2018; Shuman et al., 2013). Several definitions of the frequency representation of the graphic signal were proposed based on spectral theory and wavelet theory (Bruna et al., 2013; Hammond et al., 2011). The GNNs effectiveness of message-passing in quantum chemistry was also studied by (Gilmer et al., 2017). (Garcia & Bruna, 2017) showed how to use GNNs to learn classifiers on image datasets in a few-shot manner.

The GCN is a variant of GNNs and was proposed by (Kipf & Welling, 2017), who used GCNs for classification on citation networks and knowledge graph datasets. The GCN limits the layer wise convolution operation to alleviate the overfitting problem on local neighbourhood structures for graphs and scales linearly in the number of graph edges and learns hidden layer representations that encode both local graph structure and features of nodes. Later, the GCN was used for more NLP tasks. (Marcheggiani & Titov, 2017) proposed a syntactic GCN for semantic role labelling, operating on the dependency graph and learn latent feature representations. (Zhang et al., 2018) proposed an extension of GCN for intra-sentence relation extraction by encoding dependency trees. (Sahu et al., 2019) built a labelled-edge GCN model on a document-level graph using various inter- and intra-sentence dependencies to capture local and nonlocal dependency information for inter-sentence relation extraction. (Yao et al., 2019) built a text graph for a corpus based on word co-occurrence and document-word relations and learn the TextGCN model for text classification. Our work is inspired by (Yao et al., 2019), but different from theirs in these aspects: (1) We focus on the ASAG task, in which student responses are often short and require fine-grained semantic analysis for classification, while they aim at long text classification. (2) Apart from word nodes, we introduce bigram nodes to use word order information, which are important for ASAG because short-answer questions often examine the relations between two concepts. (3) We calculate the PMI value as the edge weight within a response because student responses are short, while they calculate it in a sliding window.

3. Methodology

We formulate the ASAG task as a classification problem: given the reference answer r and the student response s , the learned model predicts the best grade label g^* that maximizes the conditional likelihood as Equation (1):

$$g^* = \operatorname{argmax} P(g|r, s) \quad (1)$$

The procedure of the ASAG system based on the GCN can be divided into these steps: graph building, graph representation and grade prediction. First, we construct an undirected heterogeneous text graph with the sentence-level nodes, the word/bigram-level nodes and the edges between nodes. Then, we use a two-layer GCN model to encode the graph structure and obtain the graph representation, aggregating the content information and global interaction information of student responses and realizing message passing via their neighbouring nodes. Finally, based on the graph representation, the scores of the responses are predicted. Our model architecture is shown in Figure 1.

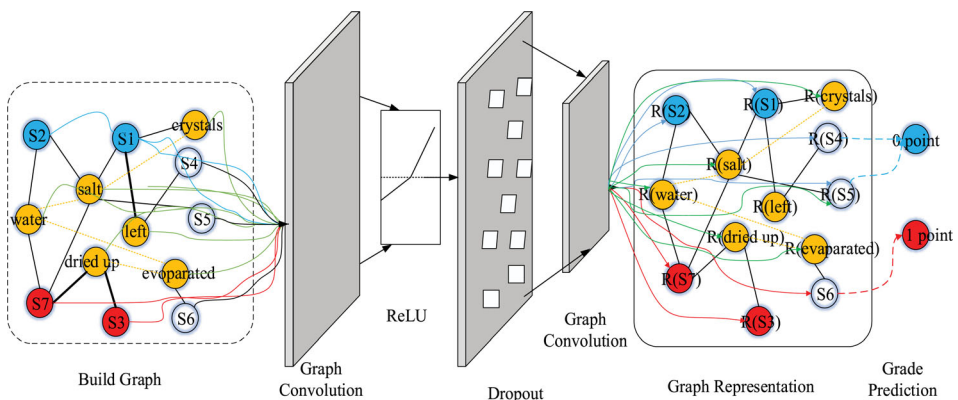


Figure 1. The architecture of the ASAG model.

3.1. Graph building

Generally, an ASAG system needs to recognize good responses from other responses, so it's necessary to model the features of good responses. Good responses usually have these features (Madnani et al., 2017): (1) they contain the correct concepts; and (2) the concepts' syntactic relations are correct; and (3) they may have different expressions to deliver the same meaning because of the nature of natural language. To capture these features, we construct a text graph, which contains the following nodes and edges, as shown with different colours in Figure 2.

Sentence-level nodes: These nodes involve the following sentence-form objects: the reference answers, the human-scored student responses and the student responses to be scored.

Word/bigram-level nodes: These nodes correspond to words and bigrams appearing in reference answers and student responses, and obtain the content information of student responses. Specifically, words reflect the concepts contained in the responses, and bigrams approximately capture the concepts expressed by two successive words, which reflect the syntactic relations between two concepts to some extent.

Sentence-word/bigram edges: This type of edge is used to connect a sentence-level node with a word/bigram-level node if the sentence contains the word/bigram. The edge weight is the sentence-based TF-IDF value of the word/bigram w , calculated by taking a sentence as a document, shown as Eq. (2):

$$tf-idf(w) = \frac{N_w}{N_{wt}} \log \frac{N_s}{N_{s_w}} \quad (2)$$

where N_w is the number of w occurring in sentence S , N_{wt} is the total number of the words in S , N_s is the total number of the sentences in the dataset, and N_{s_w} is the number of the sentences containing w in the

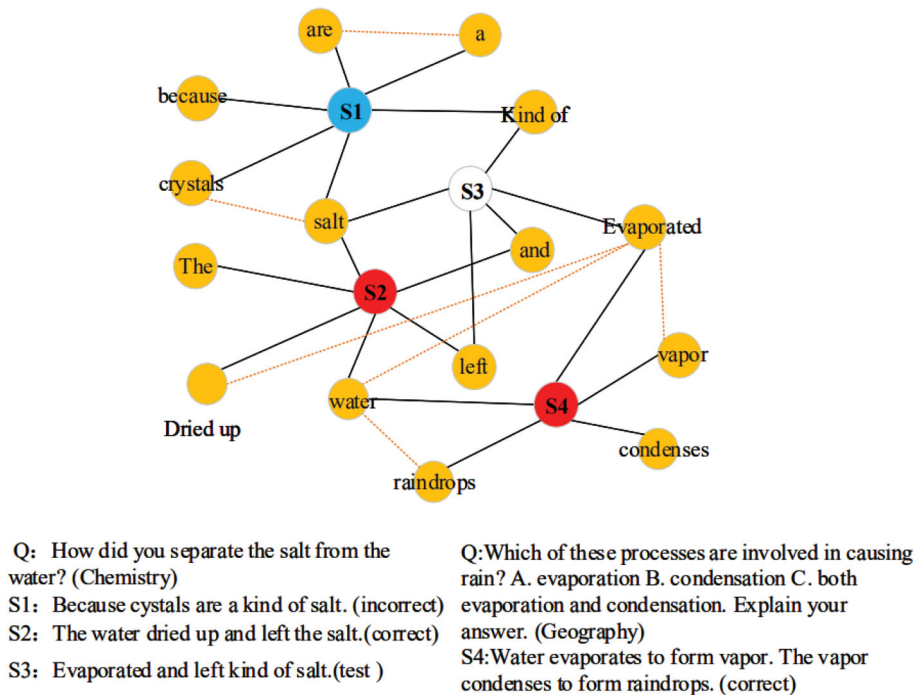


Figure 2. The constructed graph. Yellow nodes indicate word/bigram-level nodes, and nodes of other colours are sentence-level nodes (blue nodes represent incorrect responses, red nodes represent correct responses, and colourless nodes indicate the responses to be scored). A solid black line indicates the edge between a word/bigram-level node and a sentence-level node. A yellow dotted line indicates the edges between two word/bigram-level nodes.

dataset. The content information of a response is captured via the sentence-word/bigram edges and the TF-IDF weights. The grading label information of human-scored responses can be passed via these connected nodes.

Word/bigram-word/bigram edges: To capture the relations among the constituents of responses, we connect two word/bigram nodes if the corresponding two words' (or bigrams') cooccurrence weight is large enough. Specifically, the weight is the PMI value between the two words w_i and w_j , calculated as Eq. (3)

$$PMI(w_i, w_j) = \log \frac{P(w_i, w_j)}{P(w_i)P(w_j)} \quad (3)$$

where $P(w_i, w_j)$ indicates the probability of both w_i and w_j appearing in a student response. $P(w_i)$ is the probability of w_i appearing in a response. When the PMI value is positive (negative), the relevance of the two words is high (low). Here, we only built the edges between the two word/bigram nodes whose PMI value is positive.

3.2. Graph representation

Following the above idea, we construct a graph $G = (V, E)$, where V and E are sets of nodes and edges. Let $X \in R^{n \times m}$ be a matrix containing all nodes with their features, where n is the number of nodes, and m is the dimension of the feature vectors.

We encode the graph by applying the GCN proposed by (Kipf & Welling, 2017) to obtain the graph representation. The GCN is an efficient variant of CNNs. It operates directly on graphs to learn the semantic representation for the graph nodes, and updates the graph representation by information propagation between nodes while preserving the graph structural information. The layer wise propagation rule is shown as Eq. (4) and Eq. (5):

$$H^{(l+1)} = \sigma(\tilde{A}H^{(l)}W^{(l)}) \quad (4)$$

$$\tilde{A} = D^{-\frac{1}{2}}AD^{-\frac{1}{2}} \quad (5)$$

where $H^{(l)}$ is the matrix of hidden states in the l -th layer of the neural network. $W^{(l)}$ is the layerwise weight matrix, and $\sigma(\cdot)$ is the activation function. \tilde{A} is the normalized adjacent matrix, A is the symmetric adjacent matrix of G , and D is its degree matrix with $D_{ii} = \sum_j \tilde{A}_{ij}$. $H^{(0)} = X$ is the input vector.

3.3. Grade prediction

A multiple GCN layer can be stacked to obtain the representation of node v , which accumulates information from distant neighbouring nodes and capture high-order relations in the graph. It can be used for predicting the grading label. In our implementation, we use a two-layer GCN for ASAG. As shown in Figure 1, through the GCN model, the graph structure is preserved, and the representation of every node aggregates various information, including the content information, the label information and the interaction information from its first-order and second-order neighbours. The model adopts the hidden representation to predict grade labels with the form shown as Eq. (6):

$$p = \text{softmax}(\tilde{A} \text{ReLU}(\tilde{A}XW^{(0)})W^{(1)}) \quad (6)$$

where X is the input matrix, $W^{(0)}$ is the input-to-hidden weight matrix, $W^{(1)}$ is the hidden-to-output weight matrix, and $\text{ReLU}(\cdot)$ is the activation function. The softmax function classifies the output of the GCN, defined as Eq. (7):

$$\text{softmax}(x_i) = \frac{\exp(x_i)}{\sum_i \exp(x_i)} \quad (7)$$

where $x_i \in R^n$ corresponds to the representation of a node. Here, similar to (Yao et al., 2019), we simply set $X=I$, indicating that every word or sentence is represented as a one-hot vector to input to the model.

We use the cross-entropy loss as the classification training loss:

$$L = - \sum_{i=1}^I g_i \cdot \log(\hat{g}_i) \tag{8}$$

where g_i is the ground-truth grade label distribution, \hat{g}_i is the predicted grade label distribution and I is the number of grades.

4. Experimental settings

4.1. Datasets

We conduct experiments on two datasets.

SemEval-2013 Dataset. This benchmarking dataset was released as part of the SemEval-2013 Shared Task 7 (Dzikovska et al., 2013). It includes two subsets: Beetle and SciEntsBank. In this paper, we use SciEntsBank, which contains the approximately 10,000 student responses to 197 questions from 15 different science domains. The dataset has 3 versions of 2-way, 3-way and 5-way classifications. The labels of the 2-way classification are “correct” and “incorrect”, those of the 3-way classification are “correct”, “incorrect” and “contradictory”, and the 5-way classification labels are “correct”, “partially correct/incomplete”, “contradictory”, “irrelevant” and “not in the domain”. The dataset has three distinct test sets: (1) unseen answers (UA) (9%), including the responses to questions contained in the training set; (2) unseen questions (UQ) (13%), containing the responses to previously unseen questions but still in the domains presented in the training set; and (3) unseen domain (UD) (78%), including the responses to topics unseen in the training set. Texts in SciEntsBank are in English.

Two-subject Dataset. This dataset includes the questions for junior school students in China, and all texts in the dataset are in Chinese. It has two subsets. One is the math dataset with four mathematical questions to assess students’ understanding of mathematical definitions and theorems. The other is the literature dataset with four literature reading comprehension questions. The student responses in the training set are scored by two independent teachers, and their consistency is evaluated by QWKappa (Cohen’s kappa with quadratic weight). The average QWKappa value is 0.963, showing that the annotation consistency is strong.

Table 2 shows the details about the datasets.

4.2. Data augmentation

Generally, small datasets are not sufficient to train a complex deep learning model and will cause overfitting problems. Data augmentation is a technique to avoid overfitting by generating more training samples or adding data noise. The common methods of data augmentation in NLP are random transformations such as swapping two words, dropping words and replacing words with their synonyms. However, these methods may cause significant semantic changes. Back translation

Table 2. Experimental datasets

Dataset	Responses	Train	Test		
			UA	UQ	UD
SemEval-2013 dataset	10804	4969	540	733	4562
Math Dataset	17248	13798	3450		
Literature Dataset	10104	8083	2021		

from the target language to the source is another common practice in data augmentation, which is effective if a good back translation model can be obtained. Due to the different logical orders of language, back translation can often obtain new data considerably different from the original data (Xie et al., 2020).

We applied the method of back translation for data augmentation because our datasets are in English or in Chinese, which are not low-resource languages, and machine translation systems between Chinese and English perform well.

4.3. Baselines

We compare our system with the following models:

SEMILAR. SEMILAR is a semantic similarity tool, widely used in many NLP tasks, such as paraphrase recognition, question answering, and ASAG (Rus et al., 2013).

Sultan's System. It is a fast, simple and high-performance ASAG system, trained using a random forest classifier with 500 trees (Sultan et al., 2016).

ETS. One of the best systems in the SemEval-2013 task. It uses stacking and domain adaptation techniques to integrate item-specific n-gram features and more general features (Heilman & Madhani, 2013).

Saha's System. It combines some token-level features with sentence embedding-based features and improves ASAG performance (Saha et al., 2018).

SOFTCAR. It's based on text overlap through soft cardinality and a new mechanism for weight propagation, and performs particularly well in the SemEval-2013 task (Jimenez et al., 2013).

CNN + LSTM. CNNs and LSTMs have achieved state-of-the-art results on many NLP tasks. We adopt the model combining CNNs and LSTMs since they have different architectures. LSTMs are sequential, while CNNs are hierarchical for data processing, resulting in crucial complementary information for each other (Yin et al., 2017).

BERT. Bert has achieved outstanding results on many NLP tasks. It realizes the dynamic vector representations for a word based on its context, by utilizing the transformer architecture and self-attention mechanisms (Devlin et al., 2019).

4.4. Experimental settings

Settings for Neural Networks and Baselines. The experimental models of LSTM, CNN, and GCN are all implemented based on TensorFlow framework. In all experiments, the mini-batch value is 32, the dropout is 0.5, and the learning rate is 0.001. The loss function is the cross-entropy loss function, and the optimizer is the Adam optimizer. The word vectors used in LSTM and CNN are obtained with the word2vec tool in the GenSim toolkit, and the dimension is set to 300. We randomly select 10% of the training set as the verification set. If the verification loss is not reduced by 20 consecutive epochs, then the training is stopped.

For the BERT baseline, we chose BERTBASE (an uncased pretrained model) for the experiments on the English datasets. We use the Chinese pretrained BERT (<https://github.com/google-research/bert>) model for the experiments on the Chinese datasets.

For the results on the SemEval-2013 Dataset, we directly report the performances of Sultan's System, ETS, Saha's System and SOFTCAR in the corresponding paper. The baselines of SEMILAR and Sultan's System provide source code, so we use their parameter settings and provide the results on the entire three test sets of UA, UQ and UD, and the two-subject dataset.

Key Concepts. To strengthen the concepts that the questions aim to examine, we built a small vocabulary including the concept words contained in the correct responses. If an edge involves these concepts, we increase the corresponding weights by a certain multiple, defined by a hyperparameter. Here, the hyperparameter is set to 1.25 after many tests.

Some tools. For experiments on the two-subject dataset, we use the Jieba Chinese word segmentation tool to segment the student responses and the reference answers. We conduct back

translation for data augmentation by using the Youdao translator, a popular translation system in China. In this way, the final datasets we obtained are twice as large as the original datasets.

5. Results

In all experiments, we employ accuracy (Acc), macro-average F1 (M-F1) and weighted-F1 (W-F1) as evaluation metrics. Two variants of our model are provided: “Our GCN Model” and “Our GCN-DA Model”. Both are based on the two-layer GCN, but the latter uses back translation-based data augmentation.

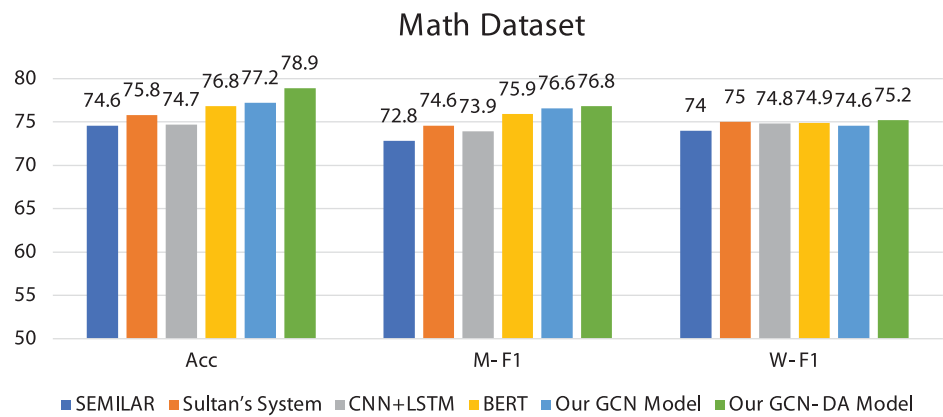
The results on the SemEval-2013 Dataset are shown in Table 3, where the column “All” indicates the results on all the test sets, including UA, UQ and UD. Figure 3 shows the results on the two-subject dataset, a similar scenario to the UA test set.

From Table 3 and Figure 3, we see that our GCN-DA model achieves quite good results in most cases. On the UD test set, it obtains the best performance for all metrics on the 2-way and 5-way tasks and achieves comparable results on the 3-way task. On the “All” test set, our model outperforms other baselines for all metrics. On the math dataset, Our GCN-DA model also outperforms all baselines. We also observe that BERT performs well in most cases, showing the effectiveness of the dynamic word vector representations based on the contexts.

Table 3. Experimental results on SemEval-2013 datasets.

(a) 2-way						
Model	UD			All		
	Acc	M-F1	W-F1	Acc	M-F1	W-F1
SEMILAR	71.1	70.5	67.7	54.6	38.8	52.2
Sultan’s System	71.3	70.4	71.2	63.2	62.5	62.8
ETS	62.3	54.3	57.4	—	—	—
Saha’s System	72.0	70.9	71.8	—	—	—
SOFTCAR	71.1	70.5	71.2	—	—	—
CNN + LSTM	72.0	71.1	71.6	63.6	45.5	61.5
BERT	72.6	71.0	72.4	64.7	63.9	63.4
Our GCN Model	71.0	69.6	70.5	65.5	64.8	63.8
Our GCN-DA Model	73.2	71.6	72.5	65.8	65.2	64.2
(b) 3-way						
Model	UD			All		
	Acc	M-F1	W-F1	Acc	M-F1	W-F1
SEMILAR	65.2	46.9	63.4	46.5	30.7	44.2
Sultan’s System	62.7	45.1	60.3	48.0	35.1	46.7
ETS	54.3	33.3	46.1	—	—	—
Saha’s System	64.0	47.9	61.2	—	—	—
SOFTCAR	63.7	48.6	62.0	—	—	—
CNN + LSTM	59.2	54.1	56.2	48.5	39.2	47.2
BERT	62.0	55.9	61.4	53.1	42.8	51.0
Our GCN Model	63.1	52.4	61.4	52.2	42.3	50.4
Our GCN-DA Model	63.4	56.6	62.52	53.6	43.2	51.3
(c) 5-way						
Model	UD			All		
	Acc	M-F1	W-F1	Acc	M-F1	W-F1
SEMILAR	46.2	30.0	47.1	40.4	39.4	40.6
Sultan’s System	50.6	34.4	48.4	42.1	40.0	42.2
ETS	44.1	38.0	41.4	—	—	—
Saha’s System	51.1	35.7	49.2	—	—	—
SOFTCAR	51.2	30.0	47.1	—	—	—
CNN + LSTM	47.2	36.2	46.8	42.2	40.2	39.2
BERT	48.4	44.2	46.9	42.8	43.0	41.7
Our GCN Model	50.3	45.6	47.7	43.5	44.3	42.6
Our GCN-DA Model	51.2	46.2	49.2	44.6	42.1	42.7

(a) Math Dataset



(b) Literature Dataset

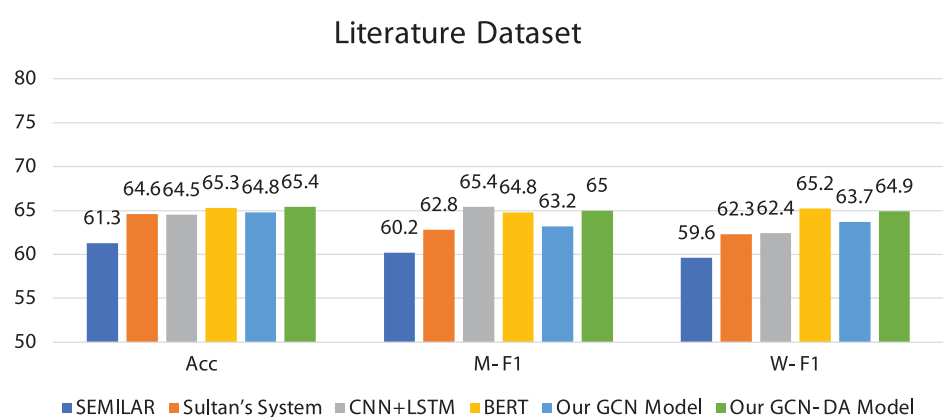


Figure 3. Experimental results on the two-subject dataset.

The main reasons why our systems work well are as follows: (1) the global response-word relations and global interaction among responses can be captured by the graph, and (2) the grade label information can be passed via the neighbouring word nodes and propagated to the entire graph. In this way, relations between the various expressions and the grade labels are obtained.

Table 4 lists some examples in which our system can make correct predictions, but other baselines (e.g. SEMILAR, Sultan’s System, and LSTM + CNN) cannot. We see that in Example1, our system successfully learns that the distinct words of “dried up” and “evaporated” have the same meaning. In Example2, the syntax structure of “if ... stick, B; if not stick, C” is the paraphrase of “It will stick to B and not C”. In Example (3), our system also learns the syntax expression “A. B” is equivalent to “first A then B”.

However, in Figure 3, we see that our system’s performance on the literature dataset is not as good as that on the math dataset. We find that the questions in the literature dataset have relatively higher openness, and students can freely express their ideas from their individual perspective. For example, one question in the literature dataset is “what appeals the most to you about the book ‘Twenty Thousand Leagues under the Sea?’”. The students answer the question from various perspectives, such as the characters, the storyline, the colourful language, ..., and so on. For this

Table 4. Examples correctly predicted by our systems.

Example1	Question: How did you separate the salt from the water? Ref. Answer: The water was evaporated, leaving the salt. Correct std. Response: The water dried up and left the salt. ()
Example2	Question: How can you use a magnet to determine if the key is iron or aluminum? Ref. Answer: If the key sticks, the key is iron; if the key does not stick, the key is aluminum. (syntax form: if.. stick, B; if not stick, C) Correct std. Response: It will stick to iron and not aluminum. (syntax form: It will sick to B and not C)
Example3	Question: Which of these processes are involved in causing rain? Explain your answer. A. evaporation B. condensation C. both evaporation and condensation. Ref. Answer: Water evaporates to form vapour. The vapour condenses to form raindrops. (syntax form: A. B) Correct std. responses: Because first it evaporates then it turns into condensation then it rains. (syntax form: first A then B)

kind of open-ended question, the patterns of good responses are more implicit and are too difficult to learn.

Additionally, we know that when our model adopts data augmentation, nearly all the metrics are improved, meaning that the technique of data augmentation can boost the system’s performance.

Ablation analysis. As shown in Figure 4, we perform an ablation study on the impact of “key concept weights”, “bigram nodes” and “edges related to bigram nodes”. We observe that the performance falls with each exclusion. When the “bigram nodes” and “edges related to bigram nodes” are removed from the graph, the system performance drops dramatically, indicating that (1) fewer graph nodes and edges will limit message passing among the nodes; and (2) word order information is very important for ASAG. Additionally, we find that the exclusion of “key concept weights” has a considerable influence on the model performance, showing that the key concepts that the questions aim to examine are very helpful for ASAG.

6. Discussion

Scalability. While the GCNs have achieved good performance in previous studies, the scalability issue still exists. In this work, we have adopted the two-layer GCN as (Kipf & Welling, 2017) to make the prediction, and the computational complexity of Eq. (6) is $O(ECHF)$, where E is the number of edges, $W^{(0)} \in R^{C \times H}$, and $W^{(1)} \in R^{H \times F}$. In other words, the algorithm’s computational

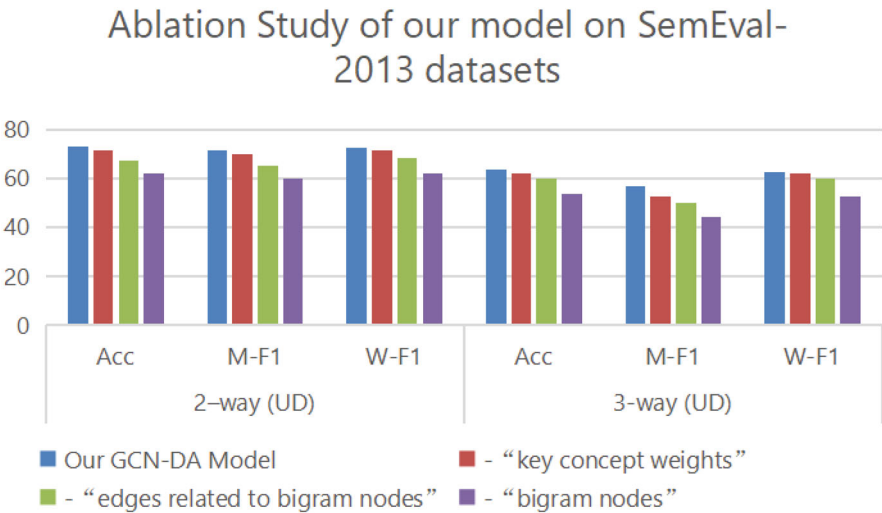


Figure 4. Ablation Study of our model on SemEval-2013 datasets.

complexity is linear to the number of graph edges, indicating that the proposed model could properly perform when the graph stays in a normal size.

On the other hand, the networks are still difficult to be extended and handle the large graphs. The main reason is that when multiple convolutional layers are stacked, the final state of a node involves the hidden states of a large number of its neighbouring nodes, which makes the back-propagation highly complex. Although some new methods, such as inductive representation learning (Hamilton et al., 2017), have been proposed to address this problem, it is still challenging and not fully resolved. Besides that, in reality, a student response may be submitted at any time and need immediate feedback from the model, which requires not only static graphs, but also dynamic graphs. To the best of our knowledge, it is still hard to effectively handle dynamic graphs (Zhou et al., 2019), which are the important topics for our future study.

Risk and limitation. In recent years, we have seen a great success in building different AI-driven intelligent systems, but most systems come with some common risks and limitations, including vulnerability, interpretability and unfairness. Similarly, the existing ASAG systems also encounter such potential risks and negative impact, which should be properly controlled and managed. Comparing with other AI-driven systems, the ASAG systems might face more challenges and risks when they are deployed for real-world applications. For example, (Azad et al., 2020) describe the deployment of an ASAG system on a high-stake educational environment to support an exam in a large-enrollment college programming course. To alleviate the student potential dissatisfaction caused by the grading errors of the ASAG system, the system has to permit students submit their answers multiple times, and meanwhile provide students an opportunity to appeal and request a manual re-scoring. (Filighera et al., 2020) study the issue of “How difficult to fool the ASAG systems” on the setting of exams. They find that the short token sequences can be prepended to students’ responses to artificially improve their grade assigned by the ASAG systems.

In short, we need more efforts to minimize the potential negative impacts and risks of the ASAG systems, especially for their deployment in high-stakes exams. It is also possible to explore the human-in-the-loop AI to tackle the potential risks and limitations of today’s ASAG systems.

7. Conclusions

We explore the application of the GCN model to capture the global relations among student responses for ASAG. We construct a graph for all student responses and build a GCN model to encode the graph structure, obtaining its graph embedding and the relations between the grading labels and the student responses with different expressions. To alleviate the overfitting problem, we augment the data by using back translation. We validate our model on two ASAG datasets and show its effectiveness.

In the future, we will consider more usage of word order information and key concept information in graphs for ASAG, and further study how to define proper feedback for open-ended questions and help the models more effectively learn the implicit semantic information of the responses.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This research is supported by the National Key Research and Development Program of China (No. 2018YFB1005103), the National Natural Science Foundation of China (No. 61673248), and the National Social Science Fund of China (No. 18BYY074).

Notes on contributors

Hongye Tan is an associate professor at Shanxi University. Her research interests include AI in education and NLP.

Chong Wang studied at Shanxi University for his MS degree. He received the MS degree from Shanxi University in 2020.

Qinglong Duan is a research engineer in Advanced Innovation Centre for Future Education, Beijing Normal University. He received the MS degree from Shanxi University in 2017.

Yu Lu is currently an associate professor with the School of Educational Technology, and the Director of the Artificial Intelligence (AI) Laboratory, Advanced Innovation Center for Future Education, Beijing Normal University. His research interests include data mining, pervasive computing, and AI in education.

Hu Zhang is an associate professor at Shanxi University. His research interests include AI in education and NLP.

Ru Li is a professor at Shanxi University. Her research interests include AI in education, NLP.

ORCID

Hongye Tan  <http://orcid.org/0000-0002-5858-899X>

Chong Wang  <http://orcid.org/0000-0002-1668-7914>

Yu Lu  <http://orcid.org/0000-0003-2378-4971>

References

- Alikaniotis, D., Yannakoudakis, H., & Rei, M. (2016, August 7–12). *Automatic text scoring using neural networks*. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016), Berlin, Germany (pp. 715–725).
- Azad, S., Chen, B., Fowler, M., West, M., & Zilles, C. (2020, July 6–10). *Strategies for deploying unreliable AI graders in high-transparency high-stakes exams*. Proceedings of the 21th International Conference on Artificial Intelligence in Education (AIED 2020), Cyberspace (pp. 16–28).
- Bachman, L. F., Carr, N., Kamei, G., Kim, M., & Sawaki, Y. (2002, August 24–September 1). *A reliable approach to automatic assessment of short answer free responses*. Proceedings of the 17th International Conference on Computational Linguistics (COLING 2002), Taipei, Taiwan (pp. 1–4).
- Bailey, S., & Meurers, D. (2008, June 19–20). *Diagnosing meaning errors in short answers to reading comprehension questions*. Proceedings of the Third ACL Workshop on Innovative Use of NLP for Building Educational Applications, Columbus, Ohio, United States (pp. 107–115).
- Bruna, J., Zaremba, W., Szlam, A., & LeCun, Y. (2013). *Spectral networks and locally connected networks on graphs*. arXiv:1312.6203.
- Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019, June 2–7). *Bert: Pre-training of deep bidirectional transformers for language understanding*. Proceedings of Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019), Minneapolis, Minnesota, United States (pp. 4171–4186).
- Dzikovska, M. O., Nielsen, R. D., Brew, C., Leacock, C., Giampiccolo, D., Bentivogli, L., Clark, P., & Dagan, I. (2013, June 14–15). *SemEval-2013 task 7: The joint student response analysis and 8th recognizing textual entailment challenge*. Proceedings of Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Seventh International Workshop on Semantic Evaluation (SemEval 2013), Atlanta, Georgia, United States (pp. 263–274).
- Filighera, A., Steuer, T., & Rensing, C. (2020, July 6–10). *Fooling automatic short answer grading systems*. Proceedings of the 21th International Conference on Artificial Intelligence in Education (AIED 2020), Cyberspace (pp. 177–190).
- Garcia, V., & Bruna, J. (2017). *Few-shot learning with graph neural networks*. arXiv: 1711.04043.
- Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., & Dahl, G. E. (2017, August 6–11). *Neural message passing for quantum chemistry*. Proceedings of the 34th International Conference on Machine Learning (ICML 2017), Sydney, Australia (pp. 1263–1272).
- Hamilton, W. L., Ying, Z., & Leskovec, J. (2017, December 3–9). *Inductive representation learning on large graphs*. Proceedings of 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, California, United States (pp. 1024–1034).
- Hammond, D. K., Vandergheynst, P., & Gribonval, R. (2011). Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2), 129–150. <https://doi.org/10.1016/j.acha.2010.04.005>
- Hassan, S. M., Fahmy, A. A., & Elramly, M. (2018). Automatic short answer scoring based on paragraph embeddings. *International Journal of Advanced Computer Science and Applications*, 9(10), 397–402. <https://doi.org/10.14569/IJACSA.2018.091048>

- Heilman, M., & Madnani, N. (2013, June 14–15). *ETS: Domain adaptation and stacking for short answer scoring*. Proceedings of Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Seventh International Workshop on Semantic Evaluation (SemEval 2013), Atlanta, Georgia, United States (pp. 275–279).
- Huang, Y., Yang, X., Zhuang, F., Zhang, L., & Yu, S. (2018, June 3–7). *Automatic Chinese reading comprehension grading by LSTM with knowledge adaptation*. Proceeding of the Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), Melbourne, Australia (pp. 118–129).
- Jimenez, S., Becerra, C., & Gelbukh, A. (2013, June 14–15). *SOFTCARDINALITY: Hierarchical text overlap for student response analysis*. Proceedings of International Workshop on Semantic Evaluation (SemEva), Atlanta, Georgia, United States (pp. 280–284).
- Kipf, T. N., & Welling, M. (2017, April 24–26). *Semi-supervised classification with graph convolutional networks*. Proceedings of International Conference of Learning Representation (ICLR), Toulon, France.
- Kumar, S., Chakrabarti, S., & Roy, S. (2017, August 19–25). *Earth mover's distance pooling over Siamese LSTMs for automatic short answer grading*. Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), Melbourne, Australia (pp. 2046–2052).
- Madnani, N., Loukina, A., & Cahil, A. (2017, September 8). *A large scale quantitative exploration of modeling strategies for content scoring*. Proceedings of Broadcasting Education Association Convention (BEA), Copenhagen, Denmark (pp. 457–467).
- Marcheggiani, D., & Titov, I. (2017, September 7–11). *Encoding sentences with graph convolutional networks for semantic role labeling*. Proceedings of 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP), Copenhagen, Denmark (pp. 1506–1515).
- Mitchell, T., Aldridge, N., & Broomhead, P. (2003, September 16–17). *Computerised marking of short-answer free-text responses*. Proceedings of Manchester International Atomic Energy Agency (IAEA) Conference, Vienna, Austria.
- Mohler, M., Bunesco, R., & Mihalcea, R. (2011, June 19–24). *Learning to grade short answer questions using semantic similarity measures and dependency graph alignments*. Proceedings of 49th Annual Meeting of the Association for Computational Linguistics (ACL), Portland, Oregon, United States (pp. 752–762).
- Ortega, A., Frossard, P., Kovacevic, J., Moura, J. M. F., & Vandergheynst, P. (2018). Graph signal processing: Overview, challenges, and applications. *Proceedings of the Institute of Electrical and Electronics Engineers (IEEE)*, 106(5), 808–828. <https://doi.org/10.1109/JPROC.2018.2820126>
- Page, E. B. (1966). The imminence of grading essays by computer. *Phi Delta Kappan*, 47(5), 238–243.
- Peng, H., Li, J., He, Y., Liu, Y., Bao, M., Wang, L., Song, Y., & Yang, Q. (2018, April 23–27). *Large-scale hierarchical text classification with recursively regularized deep graph-cnn*. Proceedings of 27th International World Wide Web Conference (WWW), Lyon, France (pp. 1063–1072).
- Riordan, B., Horbach, A., Cahill, A., Zesch, T., & Lee, C. (2017, September 8). *Investigating neural architectures for short answer scoring*. Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications (BEA), Copenhagen, Denmark (pp. 159–168).
- Rus, V., Lintean, M., & Banjade, R. (2013, August 4–9). *Semilar: The semantic similarity toolkit*. Proceedings of 51th Annual Meeting of the Association for Computational Linguistics (ACL), Sofia, Bulgaria (pp. 163–168).
- Saha, S., Dhamecha, T. I., & Marvaniya, S. (2018, June 25–30). *Sentence level or token level features for automatic short answer grading?* Proceedings of 19th International Conference on Artificial Intelligence in Education (AIED), London, United Kingdom (pp. 503–517).
- Sahu, S. K., Christopoulou, F., Miwa, M., & Ananiadou, S. (2019, July 28–August 2). *Inter-sentence relation extraction with document-level graph convolutional neural network*. Proceedings of 57th Annual Meeting of the Association for Computational Linguistics (ACL), Florence, Italy (pp. 4309–4316).
- Scarselli, F., Gori, M., Ah Chung, T., Hagenbuchner, M., & Monfardini, G. (2009). The graph neural network model. *Institute of Electrical and Electronics Engineers (IEEE) Transactions on Neural Networks*, 20(1), 61–80. <https://doi.org/10.1109/TNN.2008.2005605>
- Shuman, D. I., Narang, S. K., Frossard, P., Ortega, A., & Vandergheynst, P. (2013). The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *Institute of Electrical and Electronics Engineers (IEEE) Signal Processing Magazine*, 30(3), 83–98. <https://doi.org/10.1109/MSP.2012.2235192>
- Sultan, M. A., Salazar, C., & Sumner, T. (2016, June 12–17). *Fast and easy short answer grading with high accuracy*. Proceedings of 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL), San Diego, California, United States (pp. 1070–1075).
- Xie, Q., Dai, Z., Hovy, E., Luong, M., & Quoc, V. Le. (2020, December 6–12). *Unsupervised data augmentation*. Proceeding of 34th Conference and Workshop on Neural Information Processing Systems (NeurIPS), Online.
- Yang, X., Huang, X., Zhuang, F., Zhang, L., & Yu, S. (2018, June 25–30). *Automatic Chinese short answer grading with deep autoencoder*. Proceedings of 19th International Conference on Artificial intelligence in education (AIED), London, United Kingdom (pp. 399–404).
- Yao, L., Mao, C., & Luo, Y. (2019). Graph convolutional networks for text classification. *Proceedings of the Association for the Advance of Artificial Intelligence Conference (AAAI) on Artificial Intelligence*, 33, 7370–7377. <https://doi.org/10.1609/aaai.v33i01.33017370>

- Yin, W., Yin, W., Kann, K., Yu, M., & Schütze, H. (2017). *Comparative study of CNN and RNN for natural language processing*. arXiv:1702.01923.
- Zhang, Y., Qi, P., & Manning, C. D. (2018, October 31–November 4). *Graph convolution over pruned dependency trees improves relation extraction*. Proceedings of 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP), Brussels, Belgium (pp. 2205–2215).
- Zhou, J., Cui, G., & Zhang, Z. (2019). *Graph neural networks: A review of methods and applications*. arXiv:1812.08434.