A Semantic Feature-Wise Transformation Relation Network for Automatic Short Answer Grading

Zhaohui Li and **Yajur Tomar** and **Rebecca J. Passonneau**
Department of Computer Science and Engineering
Pennsylvania State University
{zjl5282, yst5012, rjp49}@psu.edu

Abstract

Automatic short answer grading (ASAG) is the task of assessing students' short natural language responses to objective questions. It is a crucial component of new education platforms, and could support more wide-spread use of constructed response questions to replace cognitively less challenging multiple choice questions. We propose a Semantic Feature-wise transformation Relation Network (SFRN) that exploits the multiple components of ASAG datasets more effectively. SFRN captures relational knowledge among the questions (Q), reference answers or rubrics (R), and labeled student answers (A). A relation network learns vector representations for the elements of QRA triples, then combines the learned representations using learned semantic feature-wise transformations. We apply translation-based data augmentation to address the two problems of limited training data, and high data skew for multi-class ASAG tasks. Our model has up to 11% performance improvement over state-of-the-art results on the benchmark SemEval-2013 datasets, and surpasses custom approaches designed for a Kaggle challenge, demonstrating its generality.

1 Introduction

Educators at every level rely on classroom assessments to evaluate students' knowledge, often through quizzes. Multiple choice questions can be graded automatically, but many studies have shown that short answer, constructed response questions provide greater benefit to students (Lee et al., 2011; McDaniel et al., 2007; Butler and Roediger, 2007; Clariana, 2003). Manual assessment of short asnwer questions is time-consuming, and has bias and errors (Galhardi et al., 2020; Bejar, 2012). Automatic Short Answer Grading (ASAG) applies NLP methods to reduce the assessment burden for short answer questions (Burrows et al., 2015), with potential to assist educators in providing more timely feedback to students on a preferred assessment method. With increased reliance on virtual learning environments and educational technology, the potential impact of ASAG has grown.

The most common ASAG approach classifies students' answers into two or more categories. The key challenge is that the classification problem is inherently relational, involving the relation of the student's answer to the question, as well as to one or more reference answers. Another challenge is that existing datasets are relatively small, especially for the kinds of neural network models that perform best on other NLP tasks. Much of the ASAG work is conducted in industry labs with proprietary methods and datasets (Wang et al., 2019; Liu et al., 2019). Creation of benchmark datasets, however, has fostered broader interest in the problem from the NLP community. The main benchmark dataset, SemEval-2013 (Dzikovska et al., 2013), covers multiple STEM domains, and has multiple classification tasks regarding both the number of classes, and the level of generalization required. For example, one task addresses unseen answers to known questions (UA), another addresses unseen questions (UQ) within the same subject domain, and a third is unseen domains (UD) for student answers to topics and questions not seen in the training data. While this is a rich dataset, it is not large enough to support complex models for all the classification tasks. A somewhat larger dataset, ASAP-SAS[1] is less well structured, and contains a range of supporting information other than reference answers, such as rubrics. While several challenge models performed well, they were not described in publications. We use both datasets.

Besides the potential benefits for educational technology, we believe ASAG can push NLP research in new directions due to the relational nature of the task, and the graded difficulty of the different

[1] ASAP-SAS was created for a KAGGLE challenge; https://www.kaggle.com/c/asap-sas/

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 6030–6040
November 7–11, 2021.

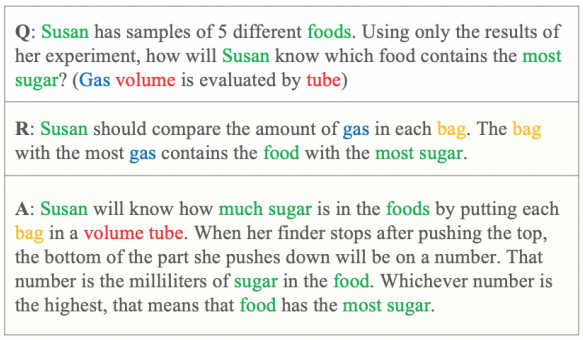

Figure 1: A [Q, R, A] triple example with a question (Q), a reference answer (R), and a student answer (A) from SemEval-2013 SciEntsBank.

classification tasks that ASAG presents. Figure 1 shows a question (Q), reference answer (R), and student answer (A) from SemEval-2013. The color coding of words simulates a Venn diagram of conceptual overlap of all three components (green), versus only Q and R (blue), only R and A (yellow), and only Q and A (red). We hypothesize that a relation network can better exploit these different similarity spaces. The input to our neural network consists of QRA triples, which are then modeled as a 3-way relation.

We present a Semantic Feature-Wise transformation Relation Network (SFRN). It is inspired by vision research using relation networks (Santoro et al., 2017) and feature mapping (Perez et al., 2018) functions, which were both motivated by a desire for greater generalization ability (referred to as reasoning), combined with computational efficiency and low model complexity. SFRN is an end-to-end model with three components. An encoder first encodes each component of a QRA triple, producing vectors for the question, one of a set of reference answers, and the student answer. When there are multiple reference answers, a relation network converts each triple of vectors for a given student answer into a single relation vector, and a learned feature-wise transformation function merges all the relation vectors for the student answer by leveraging the attentions calculated by a QRA triple. Finally, a classifier determines which class the student answer belongs to.

To address data insufficiency and class imbalance, we adopt a simple data augmentation method, back-translation, which was studied for augmentation of paraphrase data (Wieting et al., 2017), and has been used in other NLP problems. Most ASAG datasets are relatively small, especially compared to the typical training sets for large neural networks, and the multiway classification problems have extreme class imbalance. For example, the SemEval-2013 Beetle subset for the 5-way classification task has only 4,146 training samples, and one of the classes has only 195 examples. Here, as in (Xie et al., 2020), we apply back-translation to generate examples from existing ones without changing the label. We find that data augmentation is beneficial for simple models, like logistic regression, and for complex models, including our most complex model, SFRN+BERT. SFRN+BERT combined with data augmentation achieves up to 11% performance improvement over the state-of-the-art.

Our contributions are: SFRN+, a novel relation network that outperforms the state-of-the-art, use of data augmentation to address data insufficiency and imbalance, and ability to learn ASAG relations from either reference answers or rubrics. The next section presents related work on ASAG, relation networks, and data augmentation. Section 3 presents our SFRN and SFRN+ models. Section 4 describes the datasets and classification tasks. Section 5 presents our data augmentation method. Section 6 presents our experiments and results.

2 Related Work

ASAG is generally modeled as a classification problem with two or more classes. Burrows et al. (2015) gives a thorough overview of benchmark datasets and ASAG systems. Early work (Mohler and Mihalcea, 2009) formulated ASAG as a comparison of semantic text similarity between student and reference answers. A wide range of handcrafted features have been used: POS tag, word and character n-gram features (Heilman and Madnani, 2013), context overlap features (Ott et al., 2013), and graph alignment and lexical semantic similarity features (Mohler et al., 2011; Sultan et al., 2016), for input to SVM or other kinds of classifiers. Recent work applies a combination of deep neural networks with data mining. Attention networks have been used on large proprietary datasets (Wang et al., 2019; Liu et al., 2019; Ha et al., 2020). Süzen et al. (2020) used text mining to improve similarity results. Contextualized semantic representations like BERT have also been used (Hassan et al., 2018; Sung et al., 2019; Camus and Filighera, 2020). Saha et al. (2018) leveraged both hand-crafted features and sentence embeddings to achieve high performances on many tasks.

Relation Networks (RN) originated as an alternative to other kinds of graph-based neural models to develop relation-based representations, and were designed to overcome the limitations of CNNs and MLPs for reasoning problems in vision, NLP and symbolic domains such as physics (Santoro et al., 2017). RN performance on a visual question answering dataset CLEVR (Johnson et al., 2017) surpassed human performance. RNs also prove effective at improving object detection models (Hu et al., 2018). Moreover, RNs have became a general framework for few-shot learning (Sung et al., 2018). We adapt RNs for the three-way relation represented by the questions, reference answers, and student answers in ASAG datasets.

RNs typically combine learned representations of relational vectors using vector addition. We combine the relation vectors that represent multiple reference answers for the same question and student answer using a learned relation fusion function. For the fusion function, we use feature-wise transformation, based on its success with multi-modal data, e.g., images and text. Perez et al. (2018) developed FiLM, an approach to merge information from language and visual input. A language vector serves as conditioning input, to control the scaling and shifting of the visual feature map in a feature-wise fashion. Similarly, we train a function with learnable parameters to combine QRA triples for a given student answer. We have not seen feature wise transformation used for vector combination in ASAG research, which typically relies instead on fixed arithmetic operations or concatenation.

Data augmentation is less common in NLP, compared with computer vision, where image data augmentation is standard. Operations on images such as rotating an image a few degrees, or converting it to grayscale, do not change their essential meaning. In general, data augmentation is used both to increase the size of the training data and to add irrelevant noise to examples to improve the robustness of learned models. Recently, data augmentation has been found to significantly improve performance on NLP tasks as various as paraphrasing (Wieting et al., 2017), natural language generation (Kedzie and McKeown, 2019), semantic parsing (Cao et al., 2019), or various sentiment and opinion classification tasks (Kobayashi, 2018). One method is to substitute a random word with a synonym drawn from a lexical database like WordNet (Mueller and Thyagarajan, 2016; Zhang et al., 2015; Wei and Zou, 2019), or to use word embeddings to find synonyms (Wang and Yang, 2015; Jiao et al., 2020). Back-translation leverages machine translation to paraphrase a text while retaining the meaning (Wieting et al., 2017; Edunov et al., 2018; Xie et al., 2020). We adopt back-translation for its ease of use, given that machine translation methods have achieved very high performance.

3 SFRN

A Relation Network (RN) (Santoro et al., 2017) is particularly suitable for ASAG because it is designed to infer higher order generalizations, meaning generalizations that hold across tuples of examples, in a data efficient manner. RNs have been used in vision to efficiently learn generalizations across *pairs* of objects without having to learn individual weights for all possible object pairs in the data. We extend the RN framework to handle relations across vectorized text *triples* using a data fusion function.

In its simplest form, an RN is a composite function:

$$RN(O) = f_\phi(\sum_{i,j} g_\theta(o_i, o_j)) \quad (1)$$

where O is a set of input objects $\{o_1, o_2, \ldots, o_n\}$, $o_i \in \mathbb{R}^m$, and f_ϕ and g_θ are functions with trainable parameters. The inner function g learns a relation over tuples which feeds the outer classifier f with an abstract representation over the tuple objects.

In this section, we first present the SFRN model to convert the three vectors for each Q, R and A into a relational vector, then we introduce the Semantic Feature-wise Transformation unit that fuses relation vectors. Although encoding the textual Q, R and A inputs into vectors is the first step in the model, we postpone discussion of the different encoders we try out until the last subsection.

3.1 Creating the QRA Relation Vectors

The encoded vectors for a given triple are $(q, r_j, a), j \in \{0, 1, \ldots, n\}$, where n is the number of reference answers for a given question q and a student answer a of this question. Corresponding to equation (1) above, the relation vectors l_j are inferred by g_θ, using the concatenation of QRA vectors from the encoding step as the input (left hashed box in Figure 2):

$$l_j = g_\theta([q, r_j, a]) \quad (2)$$

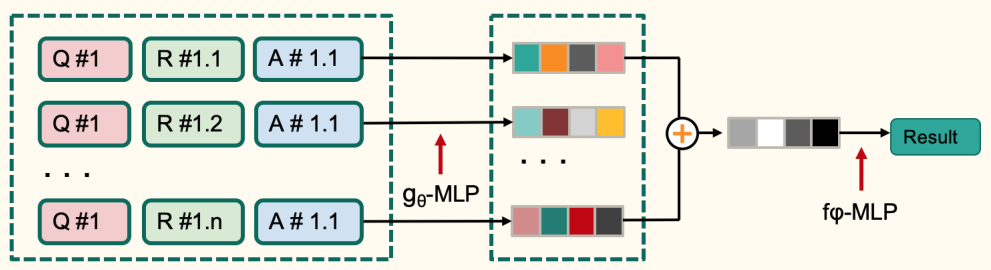

Figure 2: The structure of SFRN. The g_θ-MLP function computes the relation vector for each [Q,R,A] triple. A set of relation vectors is combined (+) using SFT. The f_ϕ-MLP function is the assessment classifier.

where g is a multilayer perceptron (MLP) with learnable parameters θ, and g produces the relation vectors $l_j \in L$ for $j \in \{0, 1, \ldots, n\}$ (one per reference answer; see center hashed box in Figure 2).

3.2 Relation Fusion

After learning the relation vectors, the RN in Santoro et al. (2017) summed them together to feed a single merged relation vector to the classifier f, where the relations were binary. For our purposes, this approach does not have enough capacity to model the subtle relational information in ASAG datasets, where the relations are ternary. Inspired by Perez et al. (2018), we adopt a relation fusion unit, Semantic Feature-wise Transformation (SFT), to learn different weights for fusing each of the learned n relation vectors l for one of the k answers a_{ik} to a question q_i. The concatenation of the three vectors in the QRA triples serves as a conditioning context to learn how to incorporate each of the n relation vectors with its own weights, before classifying the student's answer a_{ik} to a question q_i. For a given q_i and one answer a_{ik}, the input to SFT is thus the set of triples C, the set of learned relations L, and for clarity the size n of these sets:

$$\text{SFT}(C, L, n) = \sum_{j=1}^{n} (\alpha(c_j) \odot l_j + \beta(c_j)) \quad (3)$$

where C is the set of n QRA triples $[q_i, r_{ij}, a_{ik}]$, L is the set of n relation vectors from equation (2) for the n reference answers, c_j is the concatenation of a QRA triple consisting of $[q_i, r_{ij}, a_{ik}]$, l_j is the corresponding learned relation, and α and β are MLPs. The output of SFT is a fused relation vector that represents all the relational information in the input QRA triples for a given question q_i and student answer a_{ik}, relative to the reference answers r_{ij}.

Finally, a f_ϕ MLP function classifies the output of SFT into one or more classes, depending on the ground truth labeling scheme. Combining equations (2) and (3), the composite function becomes:

$$SFRN([\text{Q,R,A}]) = f_\phi\,(\text{SFT}(g_\theta)) \quad (4)$$

where $g_\theta = MLP([q, r, a])$. Overall, SFRN is a relation-based classifier that takes the QRA triple as input. Since the functions are all MLPs, the whole architecture is simple and end-to-end differentiable.

3.3 SFRN Encoder

We experiment with different SFRN encoders. Our baseline SFRN model uses LSTM (Hochreiter and Schmidhuber, 1997), which is relatively easy to train, but prone to overfitting and information loss. We compare LSTM with a BERT-based encoder. BERT is a deep, pre-trained, transformer-based model that has proven to be extremely powerful when fine-tuned for a wide range of NLP tasks (Devlin et al., 2019). We use the last output layer of the BERT base model, and fine-tune it on the ASAG data. We also use the pre-trained BERT base as an encoder in a logistic regression baseline.

We use the bert-base-uncased model pre-trained on the BooksCorpus and Wikipedia, with a 30,000 token vocabulary, and 110 million parameters. For fine-tuning, we pre-process the sentences in our QRA triples by prefixing [CLS] and postfixing [SEP] to the word token lists for each question text, or reference or student answer. Then we take the last layer output (of 12 layers in total) as the vector encodings of the elements of each QRA triple.

4 Datasets

The ASAG datasets we use are SemEval-2013, and the ASAP-SAS Kaggle competition dataset. The two were created for different purposes, and have distinct structures. With ASAP, we use only the components that correspond roughly to the SemEval format, as explained further below.

SemEval-2013 (Dzikovska et al., 2013) provides two training datasets, Beetle and SciEntsBank, that have 2-way, 3-way (*Correct, Contradictory, Incorrect*) and 5-way (*Correct, Partially correct incomplete, Contradictory, Irrelevant, Non domain*) class labels. SciEntsBank has three classification tasks comprising unseen answers (UA), unseen questions (UQ) or unseen domains (UD), and Beetle has the first two of these. The 5-way labels were chosen to potentially provide tutoring feedback. The Beetle dataset consists of 56 questions on basic electricity and electronics, with approximately 3,000 student answers. SciEntsBank contains approximately 10,000 answers to 197 assessment questions across 15 different science domains. Each question has from 1 to 14 reference answers.

To test the generality of our method, we also apply it to a dataset that lacks reference answers, Automated Student Assessment Prize Short Answer Scoring (ASAP-SAS), used in a Kaggle competition (Shermis, 2014). It has 10 prompts from science, biology and English Language Arts (ELA), with 17,207 training examples and 5,224 test examples. Responses are rated by two annotators: four prompts are rated in $\{0, 1, 2, 3\}$, and others are in $\{0, 1, 2\}$. This dataset has a wide array of other information, including rubrics. Therefore we test SFRN on triples that use rubrics in place of reference answers. We exclude all information other than the questions, rubrics, and student answers.

Many secondary and post-secondary STEM courses that use short answer questions rely on rubrics rather than reference answers. We find that SFRN performs as well as the models that use the full resources in ASAP, which shows the generalization ability of SFRN, and makes it potentially more useful to educators. We use the score assigned by the first annotator as the label and evaluate with Quadratic Weighted Kappa (QWK; a method to measure the agreement between the graders), based on the Kaggle competition guidelines.

5 Data Augmentation

Since the SemEval ASAG dataset has limited training data, and high data skew for the 3-way and 5-way tasks, we utilize back-translation to efficiently generate more examples for any given class (Edunov et al., 2018). Back-translation refers to translation from a source language to one or more *pivot* languages, followed by translation from the pivot(s) back to the source. We found good performance using two one-step pivot languages, French and Chinese. We interleaved use of a state-of-the-art neural machine translation (NMT) system, EasyNMT, with the Google Translation API. This provides us with greater control over the scale of new examples–given that Google limits calls to Google Translate–as well as more noise injection for robustness. We randomly select sentence-label pairs to generate variant sentences with the same label. If the EasyNMT back-translation is not different from the source, we call Google Translate.

Old	En: positive battery terminal is separated by a gap from terminal 1
Pivot	Ch: 正极电池端子与端子1隔开一定距离
New	En: The positive battery terminal is separated from terminal 1 by a certain distance
Old	En: positive battery terminal is separated by a gap from terminal 1
Pivot	Fr: la borne positive de la batterie est séparée par un espace de la borne 1
New	En: the positive battery terminal is separated by a space from terminal 1

Figure 3: An example of a Chinese and French back-translation for a sentence from Beetle.

Figure 3 shows two examples of back-translation, one for each pivot language we use. With Chinese as the pivot, the original word *gap* was converted to *distance*, and the two prepositional phrase arguments of *separated* were swapped. With French as the pivot, *space* replaces *gap*, and word order is preserved.

Through trial-and-error, we found that data balancing was not effective unless there was a large enough gap between the original size of the rebalanced class and the largest class. Also, there were limits to the maximum augmentation that worked well. If there was a five-fold difference in the size of a class and the largest one, we doubled the size of the small class. Otherwise, data augmentation either resulted in little improvement or in degradation of performance. We speculate that increasing the diversity of linguistic form along with the number of examples might allow for greater increases in class size.

5.1 Back-translation Experiment

Here we test our back-translation data augmentation on a logistic regression baseline ASAG model with an LSTM encoder (also used in the experiments in section 6). We compare all pairs among

	org	double	triple	org+fr	org+ch	org+ch+fr
mean	68.40	69.70	71.10	70.90	70.70	**72.70**
std	1.02	1.01	0.94	1.22	1.35	1.27
org	x	**-2.724, p=0.014**	**-5.831, p=0.000**	**-4.715, p=0.000**	**-4.087, p=0.001**	**-7.924, p=0.000**
double	x	x	**-3.047, p=0.007**	**2.277, p=0.035**	-1.786, p=0.091	**-5.560, p=0.000**
triple	x	x	x	0.389, p=0.702	0.730, p=0.475	**-3.036, p=0.007**
org+fr	x	x	x	x	0.330, p=0.745	**-3.067, p=0.007**
org+ch	x	x	x	x	x	**-3.244, p=0.005**
org+ch+fr	x	x	x	x	x	x

Table 1: T-test results of training the LR baseline 10 times on *org*, *double*, *triple*, *org+fr*, *org+ch*, and *org+fr+ch*.

five conditions combined with the original Beetle datasets on the 3-way UA task: doubling the data, tripling the data, using French as the pivot language, using Chinese as the pivot language, and combining examples from the French and Chinese back-translations. We find statistically significant improvements of the LR baseline, especially for the comparison of the original dataset with an augmentation that uses both Chinese and French back-translations.

We double and triple the original data to use as controls for comparison with augmentation by back-translation, to verify that it is not size alone that matters. Thus the original and two controls are *org, double, triple*. The augmented datasets *org+fr, org+ch* are the same size as *double*, and *org+ch+fr* is the same size as *triple*.

To get average performance results, we repeated ten iterations of training and testing on the Beetle 3-way UA classification. We trained a logistic regression baseline (LR) on the 6 training sets (see section 6.1 for the LR training details), then applied T-tests on the means to compare mean accuracy for all pairs of conditions. The results are shown in Table 1; we use alpha value $p \leq 0.05$ as the threshold to reject the null hypothesis that the means of two conditions are not different. The first two rows of Table 1 show the means and standard deviations over the ten trials for each condition. The remaining rows show the difference in means and p-values for each pair, comparing rows and columns. Table 1 shows that all the data augmentation conditions are significantly better than *org* (p-values ≤ 0.05). The condition *org+ch+fr* has the best absolute improvement over *org*, with a difference in average performance of 7.92. We concluded that back-translation is useful for data augmentation and re-balancing.

Using the best performing augmentation (*org+ch+fr*), we created new datasets, Beetle+ and SciEntsBank+. Beetle+ (N=12,438) is thrice the size of the original (N=4,146). SciEntsBank+ (N=15,450) is just over thrice the size of the original SciEntsBank (N=5,104).

6 Experiments

The research questions our experiments address are: 1) How well does SFRN perform compared to the state-of-the art? 2) Does data augmentation and re-balancing improve SFRN performance? For both the SemEval-2013 and ASAP-SAS datasets, we compare the two SFRN variants–with the LSTM (SFRN) or BERT encoders (SFRN+)–against multiple baselines. On SemEval, we also compare performance after training on the augmented SemEval-2013+. Performance metrics are accuracy, and macro-averaged F1 (M-F1). SFRN+ performs competitively on most SemEval-2013 tasks without data augmentation. With data augmentation, SFRN+ outperforms the state-of-the-art on all Beetle tasks, and on most SciEntsBank tasks. On ASAP-SAS, performance is measured using quadratic weighted kappa (QWK). SFRN+ outperforms all baselines, including three Kaggle models that use the full datasets, while SFRN+ uses a subset of the data that is more analogous to the SemEval datasets.[2] We calculated 95% confidence intervals for all results for our models in Table 2, 3, 4 and 5: all margins of error were at most 2%. To save space, however, only the point estimates are shown in the tables. The results of ASAP-SAS dataset are the best results of all runs as the benchmark model presented in their paper.

6.1 SemEval-2013 Experiments

On SemEval-2013, we compare SFRN and SFRN+ with eight baselines: 1) LO (Dzikovska et al., 2013), a model based on lexical overlap; 2) ETS (Heilman and Madnani, 2013) and 3) CoMeT (Ott et al., 2013), both of which use handcrafted features; 4) TF+SF (Saha et al., 2018), which com-

[2]Our ASAG code: https://github.com/psunlpgroup/ASAG

Task	Method	UA		UQ	
		Acc	M-F1	Acc	M-F1
2 Way	LO	79	78	**75**	**72**
	ETS	81	80	74	72
	CoMeT	83	83	70	69
	LR	73	71	63	62
	RN	75	74	64	64
	SFRN	81	80	66	65
	LR+	82	82	67	65
	RN+	84	84	68	68
	SFRN+	**89**	**89**	70	70
3 Way	LO	60	55	51	47
	ETS	63	59	55	52
	CoMeT	73	**71**	51	46
	LR	67	60	51	46
	RN	69	61	55	48
	SFRN	70	66	58	50
	LR+	72	63	62	52
	RN+	73	64	60	52
	SFRN+	**78**	67	**63**	**55**
5 Way	LO	51	42	48	41
	ETS	71	**61**	**62**	**55**
	CoMeT	68	56	48	30
	LR	60	50	55	50
	RN	55	52	57	51
	SFRN	65	55	55	51
	LR+	67	56	58	52
	RN+	69	55	57	51
	SFRN+	**75**	56	60	**55**

Table 2: Performance on the Beetle test set.

bines handcrafted features with deep learning; 5) LR, a logistic regression baseline we developed that uses the same LSTM encoder as SRFN; 6) LR+ (Sung et al., 2019), a logistic regression model that uses the pre-trained BERT-base model with fine-tuning as the encoder. Since Sung et al. (2019) report results only for the 3-way SciEntsBank tasks, we re-implemented LR+. 7) RN, a relation network baseline without the relation fusion module, using the same LSTM encoder as SRFN; 8) RN+ , a relation network baseline without the relation fusion module, which uses the pre-trained BERT-base model with fine-tuning as the encoder.

We trained the LR, RN and SFRN models that use an LSTM encoder with batches of size 32 and hidden size 256, using cross entropy loss, the Adam optimizer, a step learning rate from 5e-6 to 5e-4, and dropout of 50% on every function. Word lookup used 300D GloVe embeddings (Wikipedia/Gigaword) (Pennington et al., 2014) as input. For LR+, RN+ and SFRN+ with BERT as the encoder, we also used cross entropy loss with the Adam optimizer, but a smaller learning rate 1e-5 for BERT, and 3e-4 for the g and f functions. We used 20% of the training samples as a dev set. We varied the number of fine-tuning epochs to be from 5 to 10, depending on performance.

Method	UA		UQ		UD	
	Acc	M-F1	Acc	M-F1	Acc	M-F1
2 Way Task						
LO	66	61	66	63	67	65
ETS	72	70	71	68	69	68
CoMeT	77	76	60	57	67	67
TF+SF	**79**	**78**	**70**	**68**	**71**	**70**
LR	65	63	57	53	53	51
RN	70	66	60	55	56	53
SFRN	72	68	62	58	60	59
LR+	70	70	59	57	57	53
RN+	72	72	63	62	63	62
SFRN+	78	**78**	64	64	67	67
3 Way Task						
LO	55	40	54	39	51	41
ETS	72	64	62	42	62	42
CoMeT	71	64	54	38	57	40
TF+SF	71	65	**65**	48	**64**	45
LR	62	55	48	35	50	39
RN	64	57	50	37	52	42
SFRN	67	59	52	40	53	43
LR+	67	60	52	42	54	42
RN+	71	63	54	44	54	44
SFRN+	**73**	**65**	56	**49**	58	**47**
5 Way Task						
LO	43	37	41	32	41	31
ETS	62	**58**	**66**	27	**63**	**39**
CoMeT	60	55	43	20	42	15
TF+SF	62	47	50	31	50	35
LR	49	35	37	25	42	19
RN	58	50	40	30	46	33
SFRN	62	53	46	32	48	35
LR+	61	45	42	30	47	25
RN+	64	46	43	32	50	35
SFRN+	**69**	47	47	**35**	51	35

Table 3: Performance on the SciEntsBank test set.

Table 2 gives the results on Beetle. SFRN+ outperforms all the baselines on the 3-way tasks, and on the 2-way UA task. On the 2-way UQ task, it is bested only by LO and ETS. It performs in the mid-range on the 5-way task. We will see, however, that with data augmentation and rebalancing, SFRN+ outperforms all models on all Beetle tasks.

Table 3 gives the results on SciEntsBank. SFRN+ outperforms all baselines on the 3-way UA tasks, but TF+SF outperforms other models by a large margin on 2-way and 3-way UQ and UD. SFRN+ achieves the highest accuracy on the 5-way UA task, and the highest M-F1 on the 3-way UQ, 3-way UD and 5-way UQ tasks. On the other 5-way tasks, ETS performs best.

Tables 2 and 3 also show that SFRN and SFRN+ outperform RN and RN+ on all the sub-tasks, which indicates that the relation fusion module learns an effective method to combine the relation vectors that boosts the model performance.

In the next subsection, we present results after retraining our models on the augmented datasets

Task	Method	UA		UQ	
		Acc	M-F1	Acc	M-F1
2 Way	SOTAs	85	85	75	72
	LR	80	77	65	63
	LR+	83	83	69	67
	RN	78	74	65	63
	RN+	86	86	75	74
	SFRN	83	81	71	70
	SFRN+	**91**	**90**	**81**	**80**
3 Way	SOTAs	76	71	64	56
	LR	67	62	53	50
	LR+	72	63	60	53
	RN	70	64	59	54
	RN+	75	66	62	55
	SFRN	74	71	62	53
	SFRN+	**83**	**76**	**66**	**63**
5 Way	SOTAs	72	65	62	55
	LR	65	55	58	50
	LR+	74	64	59	54
	RN	65	57	59	52
	RN+	76	62	61	55
	SFRN	70	60	61	54
	SFRN+	**81**	**66**	**64**	**63**

Table 4: Results after retraining on Beetle+.

Method	UA		UQ		UD	
	Acc	M-F1	Acc	M-F1	Acc	M-F1
2 Way Task						
SOTAs	80	80	**70**	**68**	**71**	**70**
LR	68	65	58	54	55	52
LR+	73	73	60	59	58	57
RN	70	70	62	59	56	56
RN+	78	78	65	65	64	63
SFRN	72	71	64	62	62	58
SFRN+	**82**	**82**	67	65	68	66
3 Way Task						
SOTAs	71	65	65	48	**64**	45
LR	63	56	50	40	52	42
LR+	72	65	54	45	55	44
RN	65	58	53	42	53	43
RN+	75	68	56	46	56	46
SFRN	69	62	58	46	56	44
SFRN+	**78**	**72**	**66**	**52**	**64**	**54**
5 Way Task						
SOTAs	62	58	**66**	27	**63**	39
LR	52	39	43	29	42	22
LR+	61	45	45	32	49	25
RN	63	49	48	30	52	25
RN+	69	52	50	31	55	27
SFRN	64	57	53	35	53	36
SFRN+	**73**	**59**	57	**37**	58	**40**

Table 5: Results after retraining on SciEntsBank+.

Beetle+ and SciEntsBank+. This produces large performance gains for all six models we trained.

6.2 SemEval-2013+ Experiments

In this section we report test results after retraining LR, LR+, RN, RN+, SFRN and SFRN+ on the augmented datasets. Results of retraining on Beetle+ appear in Table 4, and results of retraining on SciEntsBank+ are in Table 5. Note that for ease of comparison, both tables have a row showing the best performance on the non-augmented SemEval-2013 datasets (SOTAs). All six re-trained models show at least some performance gains on all tasks. SFRN+ has performance gains of up to 9%, and becomes the top performing model on all Beetle test sets. On SciEntsBank, there are performance gains on nearly all tasks for the six models. SFRN+ becomes the top performer on the 2-way UA task, but remains bested by TF+SF on the UQ and UD task. On the 3-way tasks, SFRN+ beats all baselines, apart from ties with TF+SF on the UD task. On the 5-way tasks, SFRN+ gets the greatest boost with gains of up to 11%, ETS remains the top performer for accuracy on UQ and UD, but otherwise, SFRN+ outperforms all other baselines.

In sum, SFRN+ achieves phenomenal gains on SemEval-2013 by training on augmented data, and outperforms nearly all baselines on all tasks. In addition, the data augmentation and balancing helps all models. It helps much more for Beetle, which we speculate is due to the presence of multiple reference answers in Beetle compared with fewer reference answers for each question in SciEntsBank.

6.3 ASAP-SAS Experiments

As a further test of the generalization ability of SFRN and SFRN+, we run experiments on another large scale ASAG dataset, ASAP-SAS, where the training data has a different structure. As mentioned above, for this dataset we use rubrics in place of reference answers in SFRN's QRA triples. We again train four models: LR, LR+, SFRN, and SFRN+. We compare against four published baselines using QWK: 1) human raters, 2) the Kaggle winner (Tandalla), which relies on regular expression matching; 3) AutoP (Ramachandran et al., 2015), a stacked patterns model; 4) the model in (Riordan et al., 2017) (Rior), an LSTM network with attention. The four published baselines use the full ASAP-SAS resources, whereas our four models were trained only on questions, rubrics, and student answers.

The ASAP-SAS results appear in Table 6. SFRN+ has the best performance, even though it relies on less data than AutoP, Rior or Tandalla. AutoP performs nearly as well as SFRN+. This fact implies that SFRN with BERT as encoder not only has strong generalization ability, but also has the flexibility to learn from triples that contain reference answers or rubrics.

Method	QWK
Human	0.90
AutoP (Ramachandran et al., 2015)	0.78
Rior (Riordan et al., 2017)	0.74
Tandalla (1st place in competition)	0.77
LR (Baseline)	0.68
LR+ (Baseline)	0.71
SFRN (Proposed model)	0.71
SFRN+ (Proposed model)	**0.79**

Table 6: Comparing performance of models on the test set from the Kaggle ASAP competition.

6.4 Error Analysis

We carried out error analysis motivated by the large gaps between accuracy and M-F1 on many tasks. On the 5-way tasks, no model achieved an M-F1 greater than 0.73. Inspection of the per-class performance on the 5-way tasks reveals that our four models all get far worse performance on the *non-domain* class, which is much smaller than three of the other four classes, and consists mainly of very short phrases with little semantic content. For instance, the test M-F1 scores on the Beetle 5-way UA experiment are {0.9, 0.62, 0.92, 0.59, 0.21} with training sample sizes {5610, 2757, 3147, 1170, 1017} for the 5 classes {*correct, partially_correct_incomplete, contradictory, irrelevant, non_domain*}. Examples of *non-domain* include: {*what the book says, I do not know, Because if you see it is that because I chose it*}. We believe that progress on the 5-way classes would depend on a combination of input from domain experts and more sophisticated data-augmentation.

7 Conclusion

We have presented a new type of relation network, SFRN, that learns relational information from QRA triples for automatic short answer grading (ASAG). It can learn from two types of training data, using reference answers or rubrics. SFRN+, the version with the BERT encoder, outperforms previous state-of-the-art by 8-11%, depending on the dataset and classification task, when combined with a simple data augmentation method to compensate for the small and unbalanced training data. As relational meaning is central to NLP, our future work will investigate ways to improve SFRN, to understand its behavior, and to apply it to new problems. Another key avenue we aim to explore, however, is how to improve data augmentation and balancing for ASAG, and for other NLP tasks where data is difficult to come by.

8 Acknowledgements

We thank our lab mate, Vipul Gupta, for interesting discussions about the fusion function. This work was supported under NSF DRK award 2010351.

References

Issac I. Bejar. 2012. Rater cognition: Implications for validity. *Educational Measurement: Issues and Practice*, 31(3):2–9.

Steven Burrows, Iryna Gurevych, and Benno Stein. 2015. The eras and trends of automatic short answer grading. *International Journal of Artificial Intelligence in Education*, 25(1):60–117.

A. C. Butler and H. L. Roediger. 2007. Testing improves long-term retention in a simulated classroom setting. *European Journal of Cognitive Psychology*, 19:514–527.

Leon Camus and Anna Filighera. 2020. Investigating transformers for automatic short answer grading. In *International Conference on Artificial Intelligence in Education*, pages 43–48. Springer.

Ruisheng Cao, Su Zhu, Chen Liu, Jieyu Li, and Kai Yu. 2019. Semantic parsing with dual learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 51–64, Florence, Italy. Association for Computational Linguistics.

Roy B. Clariana. 2003. The effectiveness of constructed-response and multiple-choice study tasks in computer aided learning. *Journal of Educational Computing Research*, 28:395–406.

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.

Myroslava Dzikovska, Rodney Nielsen, Chris Brew, Claudia Leacock, Danilo Giampiccolo, Luisa Bentivogli, Peter Clark, Ido Dagan, and Hoa Trang Dang. 2013. SemEval-2013 task 7: The joint student response analysis and 8th recognizing textual entailment challenge. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 263–274, Atlanta, Georgia, USA. Association for Computational Linguistics.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. *arXiv preprint arXiv:1808.09381*.

Lucas Galhardi, Rodrigo C Thom de Souza, and Jacques Brancher. 2020. Automatic grading of Portuguese short answers using a machine learning approach. In *Anais Estendidos do XVI Simpósio Brasileiro de Sistemas de Informação*, pages 109–124. SBC.

Le An Ha, Victoria Yaneva, Polina Harik, Ravi Pandian, Amy Morales, and Brian Clauser. 2020. Automated prediction of examinee proficiency from short-answer questions. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 893–903, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Sarah Hassan, Aly A Fahmy, and Mohammad El-Ramly. 2018. Automatic short answer scoring based on paragraph embeddings. *International Journal of Advanced Computer Science and Applications*, 9(10):397–402.

Michael Heilman and Nitin Madnani. 2013. ETS: Domain adaptation and stacking for short answer scoring. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 275–279.

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

Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. 2018. Relation networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3588–3597.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. TinyBERT: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, page 4163–4174. Association for Computational Linguistics.

Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. 2017. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2901–2910.

Chris Kedzie and Kathleen McKeown. 2019. A good sample is hard to find: Noise injection sampling and self-training for neural language generation models. In *Proceedings of the 12th International Conference on Natural Language Generation (NLG)*, pages 584–593, Tokyo, Japan. Association for Computational Linguistics.

Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 452–457, New Orleans, Louisiana. Association for Computational Linguistics.

H. Lee, O. L. Liu, and M. C. Linn. 2011. Validating measurement of knowledge integration in science using multiple-choice and explanation items. *Applied Measurement in Education*, 24(2):115–136.

Tiaoqiao Liu, Wenbiao Ding, Zhiwei Wang, Jiliang Tang, Gale Yan Huang, and Zitao Liu. 2019. Automatic short answer grading via multiway attention networks. In *International conference on artificial intelligence in education*, pages 169–173. Springer.

M. A. McDaniel, J. L. Anderson, M. H. Derbish, and N. Morrisette. 2007. Testing the testing effect in the classroom. *European Journal of Cognitive Psychology*, 19:494–513.

Michael Mohler, Razvan Bunescu, and Rada Mihalcea. 2011. Learning to grade short answer questions using semantic similarity measures and dependency graph alignments. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 752–762.

Michael Mohler and Rada Mihalcea. 2009. Text-to-text semantic similarity for automatic short answer grading. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 567–575.

Jonas Mueller and Aditya Thyagarajan. 2016. Siamese recurrent architectures for learning sentence similarity. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, page 2786–2792. AAAI Press.

Niels Ott, Ramon Ziai, Michael Hahn, and Detmar Meurers. 2013. CoMeT: Integrating different levels of linguistic modeling for meaning assessment. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 608–616, Atlanta, Georgia, USA. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. 2018. FiLM: Visual reasoning with a general conditioning layer. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, pages 3942–3951.

Lakshmi Ramachandran, Jian Cheng, and Peter Foltz. 2015. Identifying patterns for short answer scoring using graph-based lexico-semantic text matching. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 97–106.

Brian Riordan, Andrea Horbach, Aoife Cahill, Torsten Zesch, and Chungmin Lee. 2017. Investigating neural architectures for short answer scoring. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 159–168.

Swarnadeep Saha, Tejas I Dhamecha, Smit Marvaniya, Renuka Sindhgatta, and Bikram Sengupta. 2018. Sentence level or token level features for automatic short answer grading?: Use both. In *International conference on artificial intelligence in education*, pages 503–517. Springer.

Adam Santoro, David Raposo, David GT Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. 2017. A simple neural network module for relational reasoning. *arXiv preprint arXiv:1706.01427*.

Mark D Shermis. 2014. State-of-the-art automated essay scoring: Competition, results, and future directions from a United States demonstration. *Assessing Writing*, 20:53–76.

Md Arafat Sultan, Cristobal Salazar, and Tamara Sumner. 2016. Fast and easy short answer grading with high accuracy. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1070–1075.

Chul Sung, Tejas Indulal Dhamecha, and Nirmal Mukhi. 2019. Improving short answer grading using transformer-based pre-training. In *International Conference on Artificial Intelligence in Education*, pages 469–481. Springer.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. 2018. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1199–1208.

Neslihan Süzen, Alexander N Gorban, Jeremy Levesley, and Evgeny M Mirkes. 2020. Automatic short answer grading and feedback using text mining methods. *Procedia Computer Science*, 169:726–743.

Tianqi Wang, Naoya Inoue, Hiroki Ouchi, Tomoya Mizumoto, and Kentaro Inui. 2019. Inject rubrics into short answer grading system. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, pages 175–182.

William Yang Wang and Diyi Yang. 2015. That's so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using# petpeeve tweets. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2557–2563.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388, Hong Kong, China. Association for Computational Linguistics.

John Wieting, Jonathan Mallinson, and Kevin Gimpel. 2017. Learning paraphrastic sentence embeddings from back-translated bitext. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 274–285, Copenhagen, Denmark. Association for Computational Linguistics.

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. 2020. Unsupervised data augmentation for consistency training. In *Advances in Neural Information Processing Systems*, volume 33, pages 6256–6268. Curran Associates, Inc.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.