# Creating Scoring Rubric from Representative Student Answers for Improved Short Answer Grading

Smit Marvaniya
IBM Research - India
smarvani@in.ibm.com

Swarnadeep Saha
IBM Research - India
swarnads@in.ibm.com

Tejas I. Dhamecha
IBM Research - India
tidhamecha@in.ibm.com

Peter Foltz
Pearson
peter.foltz@pearson.com

Renuka Sindhgatta
IBM Research - India
renuka.sr@in.ibm.com

Bikram Sengupta
IBM Research - India
bsengupt@in.ibm.com

## ABSTRACT

Automatic short answer grading remains one of the key challenges of any dialog-based tutoring system due to the variability in the student answers. Typically, each question may have no or few expert authored exemplary answers which make it difficult to (1) generalize to all correct ways of answering the question, or (2) represent answers which are either partially correct or incorrect. In this paper, we propose an affinity propagation based clustering technique to obtain class-specific representative answers from the graded student answers. Our novelty lies in formulating the Scoring Rubric by incorporating class-specific representatives obtained after proposed clustering, selecting, and ranking of graded student answers. We experiment with baseline as well as state-of-the-art sentence-embedding based features to demonstrate the feature-agnostic utility of class-specific representative answers. Experimental evaluations on our large-scale industry dataset and a benchmarking dataset show that the Scoring Rubric significantly improves the classification performance of short answer grading.

## KEYWORDS

Short Answer Grading, Scoring Rubric, Clustering, Supervised Learning, Classification, Sentence Embeddings

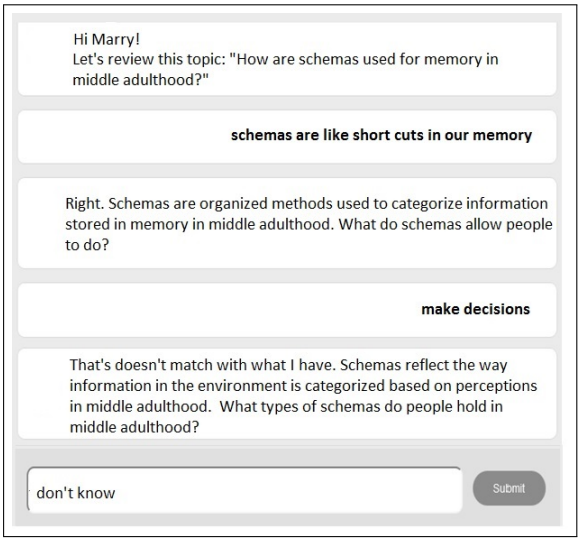
**ACM Reference Format:**
Smit Marvaniya, Swarnadeep Saha, Tejas I. Dhamecha, Peter Foltz, Renuka Sindhgatta, and Bikram Sengupta. 2018. Creating Scoring Rubric from Representative Student Answers for Improved Short Answer Grading. In *The 27th ACM International Conference on Information and Knowledge Management (CIKM '18), October 22–26, 2018, Torino, Italy*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3269206.3271755

## 1 INTRODUCTION

The dialog-based tutoring system is a type of intelligent tutoring system, where learning is driven by a natural language dialog between a student and the tutoring system [13]. The tutoring system guides the student by asking questions, analyzing student responses and providing relevant feedback (as shown in Figure 1). Hence, automatically analyzing short student answers is a critical requirement of any dialog-based intelligent tutoring system (ITS). Student Response Analysis (SRA) [9] is the task of assessing a student answer in comparison to a (set of) reference answer(s) for a given question. A key challenge of SRA is to approximate human tutor performance by interpreting the correctness of student answers. Human tutors are often able to consider a student response as correct even if it does not strictly match a reference answer or is not a semantic alternative of the same. They use their judgments and world knowledge to grade answers. Consider the example in Table 1. The correct responses are not semantic alternatives of the reference answer in the strict sense. However, annotators mark them as correct based on their judgments of student's understanding. Scenarios like this warrant the need for creating additional reference answers.

Note that there are three ways to address this challenge. One is to *create* additional reference answers with the help of Subject Matter Experts (SME). The second is to *collect* additional answers by sampling a large number of students. However, both these options are expensive; first, due to the SME costs and second, the difficulty in collecting a large enough set of student answers to

**Figure 1: Example of student interactions with dialog-based tutoring system.**





*CIKM '18, October 22–26, 2018, Torino, Italy*

ACM ISBN 978-1-4503-6014-2/18/10…$15.00
https://doi.org/10.1145/3269206.3271755

**Table 1: Examples of student answers for a given question representing inter and intra-class variations.**

| | |
|---|---|
| Question | With whom do older adults seek support? |
| Reference Answer | Older adults seek support from people who understand what they experience in old age. |
| Correct Exemplars | People experiencing similar problems |
| | other older adults like them |
| | those in similar situations as them |
| Partial Exemplars | Support groups, peers, relationship's with others |
| | From people who are caring and interested in their lives and well-being. |
| | Friends and family. |
| Incorrect Exemplars | I think they seek support from there children if they have any. |
| | Woman |
| | A Counselor |

**Table 2: Various models for short answer grading. $\delta$: classifier, $q$: question, $a$: student answer, $r$: reference answer, $c_j$, $p_j$, and $i_j$: exemplar correct, partially correct, and incorrect student answers, respectively.**

| | |
|---|---|
| Traditional Model | $\delta(a\|q, r)$ |
| Multiple References Model | $\delta(a\|q, \{r, c_1, c_2, \ldots, c_k\})$ |
| Scoring Rubric Model | $\delta(a\|q, \{\mathbf{c} : \{r, c_1, \ldots, c_k\}, \mathbf{p} : \{p_1, p_2, \ldots, p_k\}, \mathbf{i} : \{i_1, i_2, \ldots, i_k\}\})$ |

cover the variations. The third way is to build matching systems that can capture a general model of the domain by utilizing limited student answers. Such a system should be generalizable not just on straight semantic alternatives, but should also understand varying degrees of correctness across answer variants. Building on this research direction, we present a solution that automatically creates a Scoring Rubric from student answers. As shown in Table 2, the Scoring Rubric contains exemplary student answers at varying degrees of correctness (in this case, `correct`, `partially correct`, and `incorrect`). Intuitively, this modeling contains information that helps the classifiers understand the student answer space holistically, as opposed to traditional and multiple reference models that emphasize on subspace describing the intra-class variations of only correct answers.

This research focuses on leveraging graded student answers for obtaining a Scoring Rubric. Table 1 shows certain class-specific exemplars that are part of a Scoring Rubric. Note the extra information that the exemplars contain compared to the reference answer. It is our assertion that formulating a representative Scoring Rubric, and using it efficiently for feature extraction can significantly improve short answer grading. Overall, this paper makes the following research contributions:

- We propose a method to formulate Scoring Rubric (SR) from graded student answers. This is achieved by clustering the student answers and selecting and ranking cluster representatives for each grade category. Our SR is a generic concept that is independent of the exact grade categories in any short answer grading task. In this work, we formalize and utilize the notion of Scoring Rubric for classification task in SRA.
- We present generic methods to extract features from student answers with respect to the SR. These feature extraction techniques extend traditional techniques of feature representation that use only reference answer(s).
- We experiment on a benchmarking dataset (SemEval-2013 dataset [7]) and a large-scale industry dataset with simple lexical baseline features and sophisticated sentence-embedding based features. Substantial improvements signify the feature-agnostic utility of SR for short answer grading. Empirically, we also show that SR outperforms existing techniques that compare the student answer with only reference answer(s).
- Finally, on the SemEval-2013 dataset, we show that our SR in conjunction with sentence-embedding based features yields better or comparable results to state-of-the-art systems.

## 2 LITERATURE

Researchers have been working on the problem of short answer grading for more than a decade now [20] with the SemEval-2013 challenge on student response analysis [7] further formalizing the problem, benchmarking datasets, and evaluation protocols. Research works have focused on various aspects around the problem, including usage of knowledge-base [20], patterns [25, 29, 30], graph alignment [19], corpus-based similarity [12], combination of simple features [32], clustering [17], and Bayesian domain adaptation [31]. Efforts have been made for effective annotation [2, 3, 6] and content assessment [24]. Further, modern deep learning approaches are also explored for short answer grading [16, 27, 28, 34] and related problems, such as textual entailment [5, 22]. However, majority of the literature utilizes a traditional model $\delta(a|q, r)$, as defined in Table 2, for grading the student answer $a$ in context of the question $q$ and the SME created reference answer $r$.

Limited research has focused on either using multiple alternate representations [25, 26] or obtaining generalizable lexical representations of the reference answer [8]. Broadly, the research in this direction can be considered as following the Multiple References (MR) Model as defined in Table 2. Dzikovska et al. [8] have rules to cover semantic information not encoded in the ontology for evaluating student responses. Ramachandran and Foltz [26] use top scoring student responses to automatically extract patterns. They further use summaries of correct student responses to create alternate reference answers [25]. Often, a student answer is compared with all the reference answers. If any of the comparisons yields a match, the student answer is predicted as correct.

Mitchell et al. [18] propose a semi-automated approach to create a *Marking Scheme* for short answer evaluation. Marking Scheme consists of sets of answers with certain degrees of correctness. Utilization of alternate reference answers can be seen as a special case of creating a marking scheme. The benefits of using alternate reference answers vary depending on the grading technique. Broadly, it is helpful to add those answers as alternate reference answers that the grading technique is not able to match otherwise. In the similar direction, our paper generalizes the idea of having additional reference answers by incorporating representative answers from all grade categories. Representative correct answers by themselves are often not indicative of the answers from other categories, more so when the categories are many and hard to tell apart. We call this

the *Scoring Rubric* (SR) for modeling better inter-class variations for the task of short answer grading.

## 3 PROPOSED APPROACH

Given a set of graded student answers to a question as an input, our proposed approach first outputs a Scoring Rubric consisting of representative answers from each grade category (e.g. correct, partial, incorrect), followed by an efficient feature representation for the end task of short answer grading. Figure 2 illustrates the detailed steps involved in the process. Computing the scoring rubric is a 3-step process involving clustering, representative selection, and ranking. First, the student answers belonging to each grade are clustered independently. The clusters in each grade category represent different ways of answering the question for that grade. Next, a representative student answer is selected from each cluster. It signifies a candidate answer for the scoring rubric of the respective grade category. The scoring rubric consists of a set of ranked student answers that are short and grammatically well-formed for each of the grade categories. For the end classification task, a feature representation for each student answer is computed in context of the scoring rubric. The individual steps are described in detail below.

### 3.1 Clustering Graded Student Answers

Clustering student answers belonging to a grade category identifies various ways of answering the question in that category. For example, clustering all correct student answers identifies different ways of answering the question correctly. The clustering process involves design decisions pertaining to choice of 1) clustering algorithm and 2) similarity metric.

The number of clusters, which translates to the number of ways of answering a question for a grade, is unknown. This challenge rules out utilization of clustering techniques parametric to the number of clusters, e.g. $k$-means. Therefore, we employ affinity propagation [11] based clustering algorithm for clustering the student answers.

Our similarity metric for the clustering algorithm measures the similarity between two student answers. Given two student answers, we devise three novel similarity metrics as described below - 1) Token-based similarity, 2) Sentence-embedding based similarity and 3) Combination of 1) and 2). For factoid questions, the similarity between keyword tokens should suffice as a similarity metric, whereas for a relatively broader question, semantic similarity is more important. Therefore, we employ token and sentence-embedding based similarity metrics.

*Token-based Similarity.* The token-based similarity between two student answers $u$ and $v$ is computed using Dice coefficient [10] which is defined as:

$$\mathcal{T}(\mathbf{u}, \mathbf{v}) = \frac{2 \cdot |\alpha_u \cap \alpha_v|}{|\ \alpha_u\ | + |\ \alpha_v\ |} \tag{1}$$

where, $\alpha_u$ and $\alpha_v$ are the token bags for $u$ and $v$ respectively. A token bag for a student answer is computed after removing stop words from the answer, followed by question demoting [19, 32]. The term, $\alpha_{\mathbf{u}} \cap \alpha_{\mathbf{v}}$ represents the overlapping tokens between both the bags. A word $w_i \in \mathbf{u}$ is considered overlapping with a word $w_j \in \mathbf{v}$, if 1) they are exact matches, 2) they are synonyms, or 3) the cosine distance between their word-vectors [21] is less than a certain threshold $\theta$. In our experiments, we have empirically chosen the value of $\theta$ as 0.7.

*Sentence-Embedding based Similarity.* Our next choice of similarity metric is based on sentence-embeddings that help capture the semantics of sentences. We use state-of-the-art InferSent [5] embeddings for encoding the student answers. InferSent embeddings, in general, are shown to work well across various NLP tasks. Their pre-trained model is a bidirectional LSTM trained on the Stanford Natural Language Inference dataset. Given the embeddings $\mathcal{E}(\mathbf{u})$ and $\mathcal{E}(\mathbf{v})$ for two student answers $\mathbf{u}$ and $\mathbf{v}$ respectively, the similarity between them is computed as:

$$\mathcal{S}(\mathbf{u}, \mathbf{v}) = \frac{1}{2}\left(1 + \frac{\mathcal{E}(\mathbf{u}) \cdot \mathcal{E}(\mathbf{v})}{||\mathcal{E}(\mathbf{u})||\ ||\mathcal{E}(\mathbf{v})||}\right) \tag{2}$$

*Combined Similarity.* Our combined similarity metric is a combination of token based and sentence-embedding based similarities. We compute the similarity score between student answers $\mathbf{u}$ and $\mathbf{v}$ as the weighted sum of their token-based and sentence-embedding based similarity scores, formally given as

$$\mathcal{H}(\mathbf{u}, \mathbf{v}) = \beta \cdot \mathcal{S}(\mathbf{u}, \mathbf{v}) + (1 - \beta) \cdot \mathcal{T}(\mathbf{u}, \mathbf{v}) \tag{3}$$

where $\beta$ is the weight parameter. Intuitively, for a factoid question, the presence of keyword(s) dictates the correctness of the answer. This is captured using the token-based similarity. On the other hand, the sentence-embedding based similarity score is indicative of the overall similarity for broader questions. Value of $\beta = 0.5$ is used in our experiments.

Using the similarity metrics, a set of clusters is obtained for each grade. Figure 3 shows the cluster distributions using token-based, sentence-embedding based similarity and combined similarity methods respectively on our large-scale industry dataset. These clusters are used for representative selection as described in the following subsection.

### 3.2 Representative Selection

Representative selection involves selection of all such student answers, each of which is most representative of a cluster. A cluster representative indicates a distinct way of answering the question. In the simplest of approaches, cluster centroid may be considered as the cluster representative. However, given that the samples are student answers here, the notion of representativeness may not correspond to that of the cluster center. Ideally, the smallest correct answer exhibits what is *sufficient* to answer a question correctly. Moreover, having a grammatically well-formed answer is helpful in varying kinds of grading techniques that rely on parsing [4, 35]. Table 3 shows examples for short and grammatically well-formed answers and otherwise, from our industry dataset.

In light of these two observations, we propose to identify representative answers using sentence construction metric ($\mathcal{C}$), which is a linear combination of 1) sentence length metric ($\mathcal{L}$) and 2) sentence parsing metric ($\mathcal{P}$). Formally, it is defined as follows.

$$C(\mathbf{u}) = \alpha \cdot \mathcal{L}(\mathbf{u}) + (1 - \alpha) \cdot \mathcal{P}(\mathbf{u}) \tag{4}$$

where, $\mathcal{P}(\mathbf{u})$ represents the confidence score of the dependency parse [4] of the student answer. Dependency parse score ($\mathcal{P}(\mathbf{u})$) is

**Figure 2: Our proposed approach for clustering, selection and ranking of student answers and formulating Scoring Rubric for improved short answer grading.**

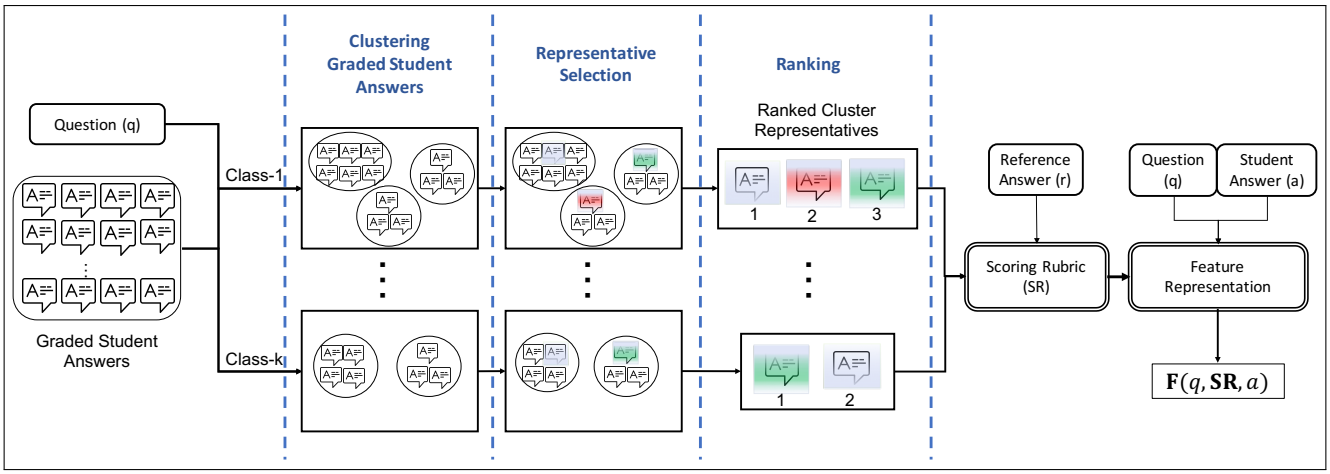
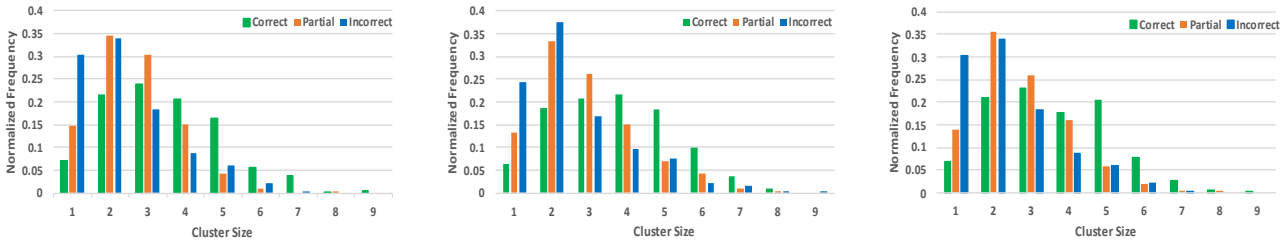



**Figure 3: Cluster size vs Normalized question frequency from our large-scale industry dataset. (a) token-based similarity (b) sentence-embedding based similarity (c) combined similarity.**




(a) token-based similarity (b) sentence-embedding based similarity (c) combined similarity

a likelihood probability which is computed based on inter words dependency and the rarity of the words in the sentence; which can provide a proxy for the complexity of the utterance. This way dependency parse score $\mathcal{P}(\mathbf{u})$ helps in selecting student answers which are relatively simple. Figures 4(a) and 4(b) show examples of dependency parse[1] for short and grammatically well-formed student answer and lengthy and complex student answer from our large-scale dataset respectively. $\mathcal{L}(\mathbf{u})$ represents the length score of a sentence $\mathbf{u}$ which is estimated after question demoting for a sentence $\mathbf{u}$. $\mathcal{L}(\mathbf{u})$ is inverse of the normalized length of the sentence $\mathbf{u}$ which helps in preferring smaller sentences as compared to longer sentences. $\alpha$ is a weight parameter. Note that both length and parsing metric values are normalized to be bounded in $[0, 1]$ before computing the construction metric. The student answer with the highest construction score is identified as the cluster representative. Value of $\alpha = 0.5$ is used in our experiments.

## 3.3 Ranking

There are situations when the number of clusters is large either due to limitations of the clustering algorithms or due to the high intra-class variations. This results in an increased scoring rubric. Presumably, there is an optimal size of the scoring rubric, beyond which elaborating the scoring rubric may not yield any further benefits. In fact, very large scoring rubrics may as well confuse the classifier, thereby degrading the performance. Therefore, it is necessary to create a ranked order of the representatives and to obtain the final scoring rubric. We propose a ranking approach based on two observations - 1) The cluster with more student answers represent a more likely way of answering the question. Therefore, the corresponding representative should be a part of the scoring rubric. 2) Scoring Rubric should include well-formed and sufficient answers. To reflect both these observations in the ranking scheme, we propose the following function ($\mathcal{R}$) to score a representative.

$$\mathcal{R}(\mathbf{u}) = C(\mathbf{u}) \cdot \left( \frac{|\pi| - |\pi_{min}|}{|\pi_{max}| - |\pi_{min}|} \right) \tag{5}$$

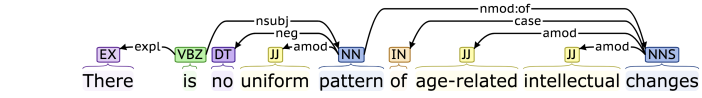
where, $\pi$ is the cluster that the representative answer $\mathbf{u}$ belongs to, $\pi_{min}$ and $\pi_{max}$ represent the smallest and largest clusters, and $C(\cdot)$ is the sentence construction metric as defined in Eq. 4. This way of ranking the cluster representatives helps in identifying the most representative well-formed student answers. Values of minimum cluster size ($\pi_{min}$) and maximum cluster size ($\pi_{max}$) are 1 and 9 used in our experiments.

[1]Figures 4(a) and 4(b) are generated using Stanford CoreNLP (http://corenlp.run)

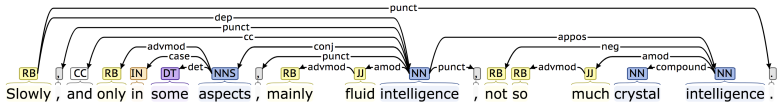
**Table 3: An example of clustering correct and partial student answers from our large-scale industry dataset using token-based clustering method. The cluster representative student answers are shown in boldface.**

| | |
|---|---|
| **Question:** How are schemas used for memory in middle adulthood? | |
| **Reference Answer:** Schemas are organized methods used to categorize information stored in memory in middle adulthood | |
| **Clustered Correct Student Answers** | |
| Cluster 1 | **Categorizing things makes remembering easier.** |
| | they are memory shortcuts to ease the burden of remembering the many things they experience each day |
| | to help retrieve information from past experiences to use in different situations or scenarios |
| Cluster 2 | **schemas are like short cuts in our memory** |
| Cluster 3 | It helps organize information, making it easier to retain. |
| | **Schemas in memory are used to organize and simplify information in our environment.** |
| | schemas are used for memory because they are organized bodies of information stored in your memory. |
| | way people recall information, organized bodies of information stored in memory. |
| **Clustered Partial Student Answers** | |
| Cluster 1 | They are used to help store information. |
| | **To retain information** |
| | recall new information, comprehend and encounter old experiences. |
| Cluster 2 | People are more likely to notice things that fit into their schema or how they think. A schema is a pattern of behavior or thought |
| | Schemas are used for memory in middle adulthood as a past memory can help them be familiar with a new one |
| | **to help remember the past** |
| Cluster 3 | Schemas organize one's thoughts. |
| | **Having an outline and organization are useful** |
| | its referred to on a a daily basis to help organize and prioritize their day |
| | They help organize behavior |

**Figure 4: Examples of dependency parsing. (a) Short and grammatically well-formed student answer (normalized dependency score = 0.90) (b) Lengthy and complex student answer (normalized dependency score = 0.81).**




(a)




(b)

Thus, by clustering the student answers, selecting representatives, and ranking them, we obtain a scoring rubric consisting of ranked sample answers at varying degree of correctness. Next, we explain the proposed methodology to obtain feature representation of a student answer using the scoring rubric.

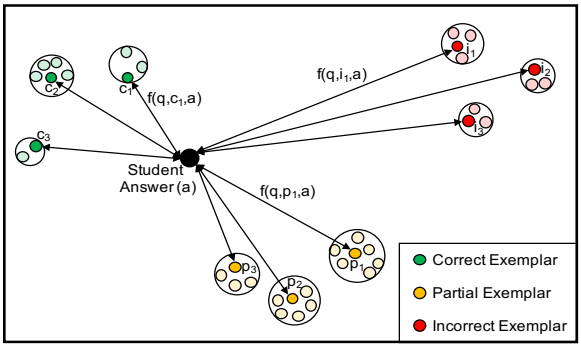
## 4 FEATURE REPRESENTATION

Traditionally, a student answer is compared against the reference answer to obtain its feature representation. In the simplest of forms, it may be textual overlap and other hand-crafted features. For sentence-embedding based representation, it can be the difference between the embeddings of student answer and the reference answer. In either of the cases, the pair-wise representation forms the basis. Let $f$ be the operator that creates the feature representation $f(q, r, a)$ between a student answer $a$ and a reference answer $r$ for a question $q$.

In this work, we utilize a scoring rubric instead of just a reference answer for comparing with the student answer. As described earlier, the scoring rubric consists of representative student answers with varying degrees of correctness. As an example, let us consider three degrees of correctness, `correct`, `partial`, and `incorrect`. Then the scoring rubric consists of a reference answer $r$, a set of correct answers $\mathbf{c}$, a set of partially correct answers $\mathbf{p}$, and a set of incorrect answers $\mathbf{i}$.

As opposed to the traditional approach where the comparison is against a reference answer $r$, given by $f(q, r, a)$, we compare the same against representative sets of correct, partial, and incorrect answers also, represented as $\mathcal{F}(q, \mathbf{c}, a)$, $\mathcal{F}(q, \mathbf{p}, a)$, and $\mathcal{F}(q, \mathbf{i}, a)$ respectively. We add the reference answer $r$ as part of the correct set

**Figure 5: An example of feature extraction by comparing against all representatives sets of correct, partial and incorrect student answers.**




**c**. Thus, the resultant concatenated feature representation w.r.t to the scoring rubric **SR** is obtained from Eq. 6.

$$\mathbf{F}(q, \mathbf{SR}, a) = [\mathcal{F}(q, \mathbf{c}, a), \mathcal{F}(q, \mathbf{p}, a), \mathcal{F}(q, \mathbf{i}, a)] \tag{6}$$

We now define the operator $\mathcal{F}$. We use two different definitions of $\mathcal{F}$ as described below.

(1) **Closest Match:** If the elements of the pair-wise feature representation $f$ express different aspects of similarity, we propose to use following definition of $\mathcal{F}$.

$$\mathcal{F}(q, \mathbf{c}, a) = f(q, r, a) \oplus f(q, c_1, a) \oplus \cdots \oplus f(q, c_k, a) \tag{7}$$

where $\oplus$ is the element-wise max operator, $c_i \in \mathbf{c}$, and $|\mathbf{c}| = k + 1$. Alternatively, the closest matching answer can also be used for feature representation as defined below.

$$\mathcal{F}(q, \mathbf{c}, a) = f(q, c_j, a) \text{ where } c_j = \arg\max_{c_i \in \mathbf{c}} \phi(c_i, a|q) \tag{8}$$

where $\phi(c_i, a|q)$ represents similarity between $c_i$ and $a$ in context of the question $q$. Eqs. 7 and 8 exhibit the closest match notion per element and per sample, respectively. In either ways, it is a form of max aggregation. Note that in this approach the feature dimension of $\mathcal{F}(q, \mathbf{c}, a)$ is same as that of $f(q, c_i, a)$. Thus, the feature dimensionality is independent of the the scoring rubric size.

(2) **Scoring Rubric Preserving:** In this approach, we propose to preserve the representation of a student answer with respect to all the elements of the scoring rubric. Therefore, as opposed to finding the closest match, the feature $\mathcal{F}(q, \mathbf{c}, a)$ is the feature concatenation of all the pair-wise features $f(q, c_i, a)$.

$$\mathcal{F}(q, \mathbf{c}, a) = [f(q, c_1, a), f(q, c_2, a), \cdots, f(q, c_k, a)] \tag{9}$$

where, $|\mathbf{c}| = k$. Since, this approach preserves all the pair-wise features, it allows the classifier to learn what is holistically important across all the $c_1, c_2, \ldots, c_k$. The feature dimensionality of $\mathcal{F}(q, \mathbf{c}, a)$ is $k$ times that of $f(q, c_i, a)$. Therefore, the role of ranking is important in creating succinct scoring rubric to control the feature size.

Figure 5 shows an interpretation of feature representation $f$ of a student answer w.r.t correct, partial and incorrect exemplars. In a broad sense, it encodes the coordinates of the student answer in the space anchored by the exemplars. To compute the feature representation $f$, we use lexical overlap baseline features, as released by the SemEval-2013 task organizers [7, 9] and sentence-embedding based features [5]. We choose these two feature sets as they differ significantly in their ways of encoding the textual similarity; allowing us to evaluate the effectiveness of the proposed approach in the context of feature complexity.

(1) **Baseline Features:** The baseline lexical overlap features, as released [7, 9] include 4 features between the student response and the reference answer - (1) Raw overlap count, (2) Cosine Similarity, (3) Lesk Similarity [1] and (4) F1 score of the overlaps. The 4 features are additionally computed between the reference answer and the question to create an 8-dimensional feature representation. We use these features to compute the representation $f$ between the student answer and each selected representative. The feature representation $\mathcal{F}$ w.r.t a grade category is computed according to Eq. 7. Note that the 4 features between the question and the reference answer are the same for all grade categories and hence taken into consideration only once. Since the individual features are non-overlapping, taking element-wise maximum gives the closest match to a certain grade category. The final feature representation **F** w.r.t the SR is given by Eq. 6.

(2) **Sentence-Embedding based Features:** Sentence-embedding features are obtained based on the sentence representations using InferSent [5]. InferSent provides its embeddings in a $d$ dimensional space. Given a question $q$, student answer $a$ and a reference answer $r$, the feature representation is obtained as

$$f(q, r, a) = [|\mathcal{E}(a) - \mathcal{E}(r)|, \mathcal{E}(a) * \mathcal{E}(r)] \tag{10}$$

where $\mathcal{E}(a)$ and $\mathcal{E}(r)$ are the embeddings of the student answer and reference answer respectively. $-$ and $*$ are element-wise subtraction and multiplication respectively. Note that for limiting the dimensionality of the feature representation, we do not use the question ($q$) embedding as part of our feature representation. The feature representation $\mathcal{F}$ w.r.t a grade category, unlike the baseline features, is computed using Eq. 9. Since the features are dimensions of the embedding space, we chose to concatenate the individual representations from the representatives rather than taking the element-wise maximum. The final representation **F** w.r.t the SR is again given by Eq. 6. We keep the InferSent embedding dimension ($d$) as 4, 096. All experiments using InferSent were performed using the pre-trained model (infersent.snli.pickle) which is trained on SNLI dataset.

## 5 EXPERIMENTS

We perform experiments to evaluate the effectiveness of the proposed Scoring Rubric approach, emphasizing on (1) its benefits over multiple references (MR) model, i.e. using only correct exemplars, (2) its sensitivity to feature representations, and (3) its performance compared to earlier published results. Further, to evaluate its generalizability, we experiment on two datasets - (1) SemEval-2013 [7] dataset and (2) Our large-scale industry dataset. Table 4 shows the train-test splits of both datasets. Overall, we perform experiments on two datasets, with two different feature representations,

**Table 4: Distribution of SemEval-2013 SciEntsBank dataset and our large-scale industry datasets.**

| | Questions | Total Responses | Train | Test |
|---|---|---|---|---|
| **Our dataset** | 483 | 16,458 | 12,317 | 4,141 |
| **SciEntsBank** | 135 | 5,509 | 4,969 | 540 |

by using MR and proposed SR, along with three proposed similarity metrics of clustering.

This section is organized into 4 subsections. In the first two, we describe and analyze the results on both the datasets. The third subsection shows a detailed ablation study on both datasets to better understand the effectiveness of SR. Finally, we compare our sentence-embedding based features in conjunction with SR against state-of-the-art systems on the SemEval-2013 dataset.

## 5.1 SemEval-2013 [7] Dataset

Our first set of experiments is on the SciEntsBank corpus of SemEval-2013 dataset. The corpus contains reference answers and student answers for 135 questions in Science domain. There are three classification subtasks on three different test sets - Unseen Answers (UA), Unseen Questions (UQ) and Unseen Domains (UD). The three classification subtasks include (1) 2-way classification into `correct` and `incorrect` classes, (2) 3-way classification into `correct`, `incorrect` and `contradictory` classes and (3) 5-way classification into `correct`, `partially correct`, `contradictory`, `irrelevant` and `non_domain` classes. Note that the samples in SciEntsBank are same across 2-way, 3-way and 5-way classifications; however, the labels change as the task becomes more granular. Each question in SciEntsBank data has exactly one associated reference answer and is thus a suitable choice for evaluating the effectiveness of SR involving additional class-specific representatives. Furthermore, similar to [26], we also test only on Unseen Answers as the representative answers generated for one question at train time might not be relevant for another question at test time.

We did not use any representative for `non_domain` and `partial` classes in the 5-way classification subtask, as a significant number of questions did not have any student answer from those classes in the train set. Although some questions did not have answers from `contradictory` and `irrelevant` classes also, the number of such questions was relatively less. Therefore, we mitigate this problem in the following way. For questions which did not have any `contradictory` student answer, we used the sentence "I don't know"[2] and for questions lacking irrelevant answers, we use the question itself as an `irrelevant` representative. Note that, these synthetic representations were necessary to evaluate the effect of class-specific exemplars over only correct ones.

We devise two separate experiments using two different sets of features - (1) Baseline features [7, 9] and (2) Sentence-embedding based features using InferSent [5]. These sets of features range from the simple hand-crafted baseline features to deep learning based sentence-embeddings. The choice of our features helps evaluate the feature-agnostic utility of class-specific representatives.

*5.1.1 Baseline Features.* A decision tree with default parameters is learned over the baseline features. We compare against Ramachandran and Foltz [26] as they also show results on the same dataset using the same set of baseline features. Table 5 shows the macro-averaged-F1 and weighted-F1 on 2-way, 3-way, and 5-way test sets.

The first row of the table demonstrates the results with the baseline features computed using only the reference answers. The next two rows are taken from [26] where Ramachandran and Foltz introduce two clustering/summarization techniques on top of the baseline features for incorporating additional reference answers. We compare our Multiple References (MR) and finally our Scoring Rubric (SR) with these state-of-the-art results. Note that our MR is in principle similar to the MEAD and Graph techniques as all of them are based on selecting and using multiple reference (or correct) answers. However, in this work, we introduce a novel method for identifying the student answer representatives with different similarity metrics. Our Scoring Rubric is a novel contribution where we extend MR to generate representative answers for all classes. We list key observations from the results shown in Table 5.

- Using the Multiple References obtained by the proposed clustering approach with sentence-embedding based similarity metric (MR + sent) outperforms both Graph and MEAD. Specifically, we achieve 3 points improvement over MEAD in 5-way and 2 and 3 points improvement over Graph in 3-way and 2-way respectively. This suggests that the proposed approach yields the correct exemplars well suited for the grading task.
- Our Scoring Rubric (SR) shows further improvement after incorporating representative answers for all classes. We achieve 3 points improvement in macro-averaged-F1 in 5-way over our best performing MR. The improvements for 3-way and 2-way are 8 points and 2 points respectively. We believe 5-way results would have improved further, had there been enough samples for clustering representatives from all classes. Nonetheless, this validates the core intuition that providing information of various grade exemplars helps the classifier to better encode the domain holistically.
- Overall, our proposal of using class-specific representatives, i.e. Scoring Rubric, significantly outperforms state-of-the-art MEAD and Graph. Moreover, our best results achieve enormous improvements over just the simple baseline features - 6 points, 14 points and 9 points in 5-way, 3-way and 2-way respectively.

*5.1.2 Sentence-Embedding based Features.* We now show the utility of class-specific representatives on state-of-the-art sentence-embedding based features as well. We use InferSent [5] embeddings of reference answers and student answers to encode our features as described in Section 4.

From the ranked representatives of each grade category, we use top 3 answers (including the reference answer) for `correct` class

[2]The choice of the sentence was motivated by already existing samples of similar meaning in the `contradictory` class.

**Table 5: Comparative evaluation with baseline features on 5-way, 3-way, and 2-way protocols of SciEntsBank dataset and our large-scale industry dataset. MR: Multiple References, SR: Scoring Rubric**

| | | 5-way | | 3-way | | 2-way | | Our dataset | |
|---|---|---|---|---|---|---|---|---|---|
| **Approach** | **Sim.** | **M-F1** | **W-F1** | **M-F1** | **W-F1** | **M-F1** | **W-F1** | **M-F1** | **W-F1** |
| **BF [7]** | - | 0.375 | 0.435 | 0.405 | 0.523 | 0.617 | 0.635 | 0.423 | 0.465 |
| **Graph [26]**‡ | - | 0.372 | 0.458 | 0.438 | 0.567 | 0.644 | 0.658 | - | - |
| **MEAD [26]**‡ | - | 0.379 | 0.461 | 0.429 | 0.554 | 0.631 | 0.645 | - | - |
| **MR** | token | 0.362 | 0.428 | 0.446 | 0.545 | 0.630 | 0.647 | 0.468 | 0.506 |
| | sent | 0.402 | 0.474 | 0.459 | 0.581 | 0.673 | 0.686 | 0.477 | 0.515 |
| | combined | 0.355 | 0.412 | 0.455 | 0.557 | 0.676 | 0.688 | 0.475 | 0.507 |
| **SR** | token | **0.430** | **0.472** | **0.545** | **0.604** | 0.692 | 0.703 | 0.525 | 0.552 |
| | sent | 0.405 | 0.459 | 0.500 | 0.578 | 0.676 | 0.685 | 0.530 | 0.559 |
| | combined | 0.400 | 0.462 | 0.501 | 0.579 | **0.702** | **0.712** | **0.531** | **0.560** |

‡ Results as reported by Ramachandran and Foltz [26]

**Table 6: Comparative evaluation with sentence-embedding based features on 5-way, 3-way, and 2-way protocols of SciEntsBank dataset and our large-scale industry dataset. MR: Multiple References, SR: Scoring Rubric**

| | | 5-way | | 3-way | | 2-way | | Our dataset | |
|---|---|---|---|---|---|---|---|---|---|
| **Approach** | **Sim.** | **M-F1** | **W-F1** | **M-F1** | **W-F1** | **M-F1** | **W-F1** | **M-F1** | **W-F1** |
| **SE-based [5]** | - | 0.497 | 0.541 | 0.594 | 0.672 | 0.725 | 0.731 | 0.586 | 0.629 |
| **MR** | token | 0.578 | **0.621** | 0.610 | 0.681 | 0.735 | 0.747 | 0.631 | **0.658** |
| | sent | 0.557 | 0.584 | 0.627 | 0.703 | 0.754 | 0.766 | 0.627 | 0.654 |
| | combined | 0.557 | 0.604 | 0.599 | 0.682 | 0.744 | 0.754 | 0.627 | 0.657 |
| **SR** | token | **0.579** | 0.610 | **0.637** | 0.710 | 0.752 | 0.767 | **0.634** | 0.655 |
| | sent | 0.568 | 0.600 | 0.621 | 0.688 | 0.745 | 0.756 | 0.629 | **0.658** |
| | combined | **0.579** | 0.610 | 0.636 | **0.719** | **0.773** | **0.781** | 0.633 | **0.658** |

and top 1 answer for `incorrect`, `contradictory` and, `irrelevant` classes as exemplars in the Scoring Rubric.

We learn a multinomial logistic regression classifier on top of the features. The best parameters are learned using k-fold cross-validation. Table 6 compares macro-averaged-F1 and weighted-F1 on 5-way, 3-way, and 2-way with and without using the Scoring Rubric. Note that we could not compare our MR with those of Ramachandran and Foltz [26] as their code was not publicly available. We again list our key observations below.

- Our MR outperforms all the three testing protocols for embedding-based features as well. Specifically, our best MR achieves 8 points better macro-averaged-F1 in 5-way and 3 points better in 3-way and 5-way. This suggests that even when feature representations are semantically rich, using Multiple References is better than only using the SME created reference answer.
- Our Scoring Rubric further improves the results with the addition of representatives from all classes. We obtain 1 point, and 2 points macro-averaged-F1 gains with our best SR over the best MR in 3-way and 2-way respectively. Note that 5-way does not improve much because of lack of representatives from `non_domain` and `partial` classes.
- Overall, incorporating class-specific representatives improves only the sentence-embedding based features with substantially high 10 points, 4 points and 5 points macro-averaged-F1 gains in 5-way, 3-way and 2-way respectively. This demonstrates the utility of Scoring Rubric even when the student answers are represented in a semantic space.

## 5.2 Our large-scale Industry Dataset

We also conducted experiments on our large-scale industry dataset, which consists of questions from Psychology domain. The evaluation is on a 3-way classification task consisting of 3 classes - `correct`, `partial`, and `incorrect`. The ground truth grades of each student answers are provided by subject matter experts.

Similar to SemEval-2013 dataset, we again experiment with both baseline and sentence-embedding based features.

*5.2.1 Baseline Features.* Table 5 compares the macro-averaged-F1 and weighted-F1 of all the configurations on our dataset using baseline features. We compare our MR and SR against the traditional approach of not using any class-specific representatives. We make following key observations.

- Our Multiple References (MR) improve upon the baseline features, with sentence-embedding based similarity metric achieving 5 points better macro-averaged-F1.
- Our Scoring Rubric (SR) shows further improvement. Particularly, SR with combined similarity metric achieves 6 points improvement over MR.

**Table 7: Ablation study of Scoring Rubric showing macro-averaged-F1 scores with baseline features on SciEntsBank-3way and our dataset. CR: Correct, I: Incorrect, P: Partial, CN: Contradictory.**

| Similarity Component | SciEntsBank-3way | | | Our dataset | | |
|---|---|---|---|---|---|---|
| | CR | CR+I | CR+I+CN | CR | CR+P | CR+P+I |
| **Token** | 0.446 | 0.451 | **0.545** | 0.468 | 0.497 | **0.525** |
| **Sentence** | 0.459 | 0.457 | **0.500** | 0.477 | 0.504 | **0.530** |
| **Combined** | 0.455 | 0.468 | **0.501** | 0.475 | 0.513 | **0.531** |

**Table 8: Comparison of our best SR on sentence-embedding based features with state-of-the-art results on SciEntsBank 2-way, 3-way and 5-way testing protocols. ‡Results as reported by Riordan et al. [27].**

| Approach | 5-way | | 3-way | | 2-way | |
|---|---|---|---|---|---|---|
| | M-F1 | W-F1 | M-F1 | W-F1 | M-F1 | W-F1 |
| **Sultan et al. [32]** | 0.412 | 0.487 | 0.444 | 0.570 | 0.677 | 0.691 |
| **ETS [14]** | **0.598** | **0.640** | **0.647** | 0.708 | 0.762 | 0.770 |
| **COMeT [23]** | 0.551 | 0.598 | 0.640 | 0.707 | 0.768 | 0.773 |
| **SOFTCAR [15]** | 0.474 | 0.537 | 0.555 | 0.647 | 0.715 | 0.722 |
| **T & N best [33]‡** | - | 0.521 | - | - | - | 0.670 |
| **T & N tuned [27]‡** | - | 0.533 | - | - | - | 0.712 |
| **SE-based [5]** | 0.497 | 0.541 | 0.594 | 0.672 | 0.725 | 0.731 |
| **SE-based + SR** | 0.579 | 0.610 | 0.636 | **0.719** | **0.773** | **0.781** |

- The overall improvement with respect to just baseline features is again substantial with almost 11 points better macro-averaged-F1.

*5.2.2 Sentence-Embedding based Features.* We use the top 3 correct representatives (including the reference answer), and top 1 for partial and incorrect classes as exemplars in the Scoring Rubric. Our results with sentence-embedding based features are shown in table 6. We again study the effect of MR and SR on the sentence-embedding based features. The salient observations are listed below.

- Using token-based MR obtains 5 points better macro-averaged-F1 over the sentence-embedding based features.
- The SR here does not improve the results much. We believe the sophistication in the feature representation largely limits the substantial improvement that we saw with baseline features.
- The overall improvement is still noteworthy, with token-based SR obtaining 5 points better macro-averaged-F1 over the sentence-embedding features.

## 5.3 Ablation Study of Scoring Rubric

In any short answer grading task, we believe that encoding representatives for each class has its merits. We show this in Table 7 by our ablation study of incorporating class-specific representatives on baseline features. Our experiments are on both the datasets. However, in SciEntsBank dataset, we show results only on 3-way, as for 5-way we did not have representatives for all classes. The results improve as we incrementally add representatives for each of the three classes. Note that the classes in our dataset and in SciEntsBank are different. However, that does not affect the results as our idea of utilizing class-specific representatives to formulate Scoring Rubric is generic. While using representatives for all the three classes in SciEntsBank, the macro-averaged-F1 with token-based similarity metric improves substantial by 8 points as compared to only using correct class representatives. Similarly, in our dataset, the improvement is about 6 points.

## 5.4 Comparison with State-of-the-Art Methods

Our final experiment is one where we compare the sentence-embedding based feature in conjunction with Scoring Rubric against state-of-the-art methods on SciEntsBank Unseen Answers test data. We use the combined similarity metric for the comparisons. We specifically compare against best performing systems in the task[3] –namely, ETS [14], COMeT [23], and SOFTCARDINALITY [15], along with recent research including Sultan et al. [32][4], Taghipour and Ng [33], Riordan et al. [27], and Infersent [5]. Table 8 shows the macro-averaged-F1 and weighted-F1 of all the systems on SciEntsBank dataset.

We observed that the sentence-embedding based features with our SR outperform all systems on the 2-way subtask. On 3-way, we match the state-of-the-art results with 1 point better weighted-F1 but 1 point less macro-averaged-F1 compared to ETS [14]. On 5-way, our results are competitive, where we do better than all systems except ETS. We believe that this could be down to our inability to incorporate representatives for all classes. Also, note that ETS benefits from its underlying domain adaptation. On a direct comparison between the sentence embedding features [5] and its extension to proposed Scoring Rubric, the later yields significant improvements. We could use better features; using Scoring Rubric on top of more sophisticated features should improve the results further.

[3] https://docs.google.com/spreadsheets/d/1Xe3lCi9jnZQiZW97\hBfkg0x4cI3oDfztZPhK3TGO_gw/pub?output$=$html#
[4] All experiments were performed using their publicly available code at https://github.com/ma-sultan/short-answer-grader

## 6 CONCLUSION

Automatic short answer grading for Intelligent Tutoring Systems has been a well-studied problem in NLP community over the years. Traditional approaches have looked into it as a classification task where the student answer is matched against a reference answer for a given question. To overcome challenges involving the variability in student answers, researchers have incorporated automated ways of selecting correct exemplars and use them as multiple reference answers. In this work, we generalize the notion of multiple reference answers to that of a Scoring Rubric that incorporates representative student answers from multiple grade categories. Creation of a Scoring Rubric involves clustering student answers for each grade category, followed by a representative selection from each cluster, and finally ranking them. Extending the feature representation of a student answer w.r.t. to a reference answer, we propose techniques to obtain its feature representation w.r.t the Scoring Rubric. We experiment with simple lexical overlap baseline features as well as sophisticated sentence-embedding based features to demonstrate that the notion of a Scoring Rubric is feature-agnostic. Its effectiveness is empirically evaluated on a benchmarking dataset and our large-scale industry dataset. We report significantly better results on both the datasets compared to existing approaches that compare the student answer against only reference answer(s). Our model involving sentence-embedding based features w.r.t the Scoring Rubric also demonstrates comparable or better results to state-of-the-art models on the benchmarking dataset.

Certain short answer grading tasks that output real-valued scores are modeled as a regression problem too. While our notion of a Scoring Rubric is independent of the individual grade categories, extending it to regression tasks where the classes are not well-defined, remains one of the key future directions to pursue.

## REFERENCES


[1] S. Banerjee and T. Pedersen. An adapted lesk algorithm for word sense disambiguation using wordnet. In *Proceedings of the International Conference on Intelligent Text Processing and Computational Linguistics*, pages 136–145, 2002.

[2] S. Basu, C. Jacobs, and L. Vanderwende. Powergrading: a clustering approach to amplify human effort for short answer grading. *Transactions of the Association for Computational Linguistics*, 1:391–402, 2013.

[3] M. Brooks, S. Basu, C. Jacobs, and L. Vanderwende. Divide and correct: using clusters to grade short answers at scale. In *Proceedings of the ACM Conference on Learning @ Scale*, pages 89–98, 2014.

[4] D. Chen and C. Manning. A fast and accurate dependency parser using neural networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 740–750, 2014.

[5] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 670–680, 2017.

[6] T. I. Dhamecha, S. Marvaniya, S. Saha, R. Sindhgatta, and B. Sengupta. Balancing human efforts and performance of student response analyzer in dialog-based tutors. In *International Conference on Artificial Intelligence in Education*, 2018.

[7] M. Dzikovska, R. Nielsen, C. Brew, C. Leacock, D. Giampiccolo, L. Bentivogli, P. Clark, I. Dagan, and H. T. Dang. SemEval-2013 Task 7: The joint student response analysis and 8th recognizing textual entailment challenge. In *Proceedings of the NAACL-HLT Workshop on Semantic Evaluation*, 2013.

[8] M. Dzikovska, N. Steinhauser, E. Farrow, J. Moore, and G. Campbell. BEETLE II: Deep natural language understanding and automatic feedback generation for intelligent tutoring in basic electricity and electronics. *International Journal of Artificial Intelligence in Education*, 24(3):284–332, 2014.

[9] M. O. Dzikovska, R. D. Nielsen, and C. Brew. Towards effective tutorial feedback for explanation questions: A dataset and baselines. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 200–210, 2012.

[10] W. B. Frakes. Stemming algorithms., 1992.

[11] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315(5814):972–976, 2007.

[12] W. H. Gomaa and A. A. Fahmy. Short answer grading using string similarity and corpus-based similarity. *International Journal of Advanced Computer Science and Applications*, 3(11), 2012.

[13] A. C. Graesser, S. Lu, G. T. Jackson, H. H. Mitchell, M. Ventura, A. Olney, and M. M. Louwerse. Autotutor: A tutor with dialogue in natural language. *Behavior Research Methods, Instruments, & Computers*, 2004.

[14] M. Heilman and N. Madnani. ETS: Domain adaptation and stacking for short answer scoring. In *Proceedings of the Joint Conference on Lexical and Computational Semantics*, volume 2, pages 275–279, 2013.

[15] S. Jimenez, C. Becerra, and A. Gelbukh. SOFTCARDINALITY: Hierarchical text overlap for student response analysis. In *Proceedings of the Joint Conference on Lexical and Computational Semantics*, volume 2, pages 280–284, 2013.

[16] S. Kumar, S. Chakrabarti, and S. Roy. Earth mover's distance pooling over siamese LSTMs for automatic short answer grading. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 2046–2052, 2017.

[17] A. S. Lan, D. Vats, A. E. Waters, and R. G. Baraniuk. Mathematical language processing: Automatic grading and feedback for open response mathematical questions. In *ACM Conference on Learning @ Scale*, pages 167–176, 2015.

[18] T. Mitchell, N. Aldridge, and P. Broomhead. Computerised marking of short-answer free-text responses. In *Manchester IAEA conference*, 2003.

[19] M. Mohler, R. C. Bunescu, and R. Mihalcea. Learning to grade short answer questions using semantic similarity measures and dependency graph alignments. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 752–762, 2011.

[20] M. Mohler and R. Mihalcea. Text-to-text semantic similarity for automatic short answer grading. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*, pages 567–575, 2009.

[21] N. Mrkšic, D. OSéaghdha, B. Thomson, M. Gašic, L. Rojas-Barahona, P.-H. Su, D. Vandyke, T.-H. Wen, and S. Young. Counter-fitting word vectors to linguistic constraints. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016.

[22] J. Mueller and A. Thyagarajan. Siamese recurrent architectures for learning sentence similarity. In *Proceedings of the Association for the Advancement of Artificial Intelligence*, pages 2786–2792, 2016.

[23] N. Ott, R. Ziai, M. Hahn, and D. Meurers. CoMeT: Integrating different levels of linguistic modeling for meaning assessment. In *Proceedings of the Joint Conference on Lexical and Computational Semantics*, volume 2, pages 608–616, 2013.

[24] R. J. Passonneau, A. Poddar, G. Gite, A. Krivokapic, Q. Yang, and D. Perin. Wise crowd content assessment and educational rubrics. *International Journal of Artificial Intelligence in Education*, 2018.

[25] L. Ramachandran, J. Cheng, and P. W. Foltz. Identifying patterns for short answer scoring using graph-based lexico-semantic text matching. In *Proceedings of the NAACL-HLT Workshop on Innovative Use of NLP for Building Educational Applications*, pages 97–106, 2015.

[26] L. Ramachandran and P. W. Foltz. Generating reference texts for short answer scoring using graph-based summarization. In *Proceedings of the NAACL-HLT Workshop on Innovative Use of NLP for Building Educational Applications*, pages 207–212, 2015.

[27] B. Riordan, A. Horbach, A. Cahill, T. Zesch, and C. M. Lee. 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, 2017.

[28] S. Saha, T. I. Dhamecha, S. Marvaniya, R. Sindhgatta, and B. Sengupta. Sentence level or token level features for automatic short answer grading?: Use both. In *International Conference on Artificial Intelligence in Education*, 2018.

[29] J. Z. Sukkarieh and J. Blackmore. C-rater: Automatic content scoring for short constructed responses. In *International Florida Artificial Intelligence Research Society Conference*, pages 290–295, 2009.

[30] J. Z. Sukkarieh and S. Stoyanchev. Automating model building in c-rater. In *Proceedings of the ACL Workshop on Applied Textual Inference*, pages 61–69, 2009.

[31] M. A. Sultan, J. Boyd-Graber, and T. Sumner. Bayesian supervised domain adaptation for short text similarity. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 927–936, 2016.

[32] M. A. Sultan, C. Salazar, and T. Sumner. Fast and easy short answer grading with high accuracy. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1070–1075, 2016.

[33] K. Taghipour and H. T. Ng. A neural approach to automated essay scoring. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1882–1891, 2016.

[34] Y. Zhang, R. Shah, and M. Chi. Deep Learning+ Student Modeling+ Clustering: a recipe for effective automatic short answer grading. In *Proceedings of the International Conference on Educational Data Mining*, pages 562–567, 2016.

[35] M. Zhu, Y. Zhang, W. Chen, M. Zhang, and J. Zhu. Fast and accurate shift-reduce constituent parsing. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 434–443, 2013.