Exploring Distinct Features for Automatic Short Answer Grading

Lucas B. Galhardi[1], Helen C. de Mattos Senefonte[1], Rodrigo C. Thom de Souza[2], Jacques D. Brancher[1]

[1]Graduate Program in Computer Science – Londrina State University (UEL)
10.011 - 86057-970 - Londrina, PR - Brazil

[2]Natural and Scientific Computing Research Group – Federal University of Paraná (UFPR)
86900-000 - Jandaia do Sul, PR - Brazil

{lucasbgalhardi,helen,jacques}@uel.br, thom@ufpr.br

Abstract. *Automatic short answer grading is the study field that addresses the assessment of students' answers to questions in natural language. The grading of the answers is generally seen as a typical classification supervised learning. To stimulate research in the field, two datasets were publicly released in the SemEval 2013 competition task "Student Response Analysis". Since then, some works have been developed to improve the results. In this context, the goal of this work is to tackle such task by implementing lessons learned from the literature in an effective way and report results for both datasets and all of its scenarios. The proposed method obtained better results in most scenarios of the competition task and, therefore, higher overall scores when compared to recent works.*

1. Introduction

Evaluations are used to demonstrate the acquired knowledge in the student's learning process. Despite the importance of evaluation, teachers usually find the task of assessing the respondents' answers very time-consuming. Also, students may have to wait for a long time to receive feedback on their responses and, when they finally get it, the grade can be different from another classmate's, who has given a very similar answer [Santos et al. 2016, Passero et al. 2016].

Computer-based assessment came to address those issues and improve other aspects of learning by automating the evaluation process. Evaluations are often composed of recall and recognition type of questions, which are at different levels of the learning depth. The recognition kind seeks to test the respondent's ability to organize or identify some specific information. In its turn, for recall questions, respondents need to remember external knowledge and write their own answers. Automatic grading is a solved problem for recognition questions, but it is an open problem and research subject to the recall kind [Burrows et al. 2015].

The automatic assessment brings some benefits, such as the formalization of correction criteria, the delivery of faster feedback to both teacher and student and the better use of teachers' time [Liu et al. 2016].

There are many different types of questions that can be required from students. Short answers are the focus of interest in this work. They can range from one sentence to one paragraph (few sentences), must be written in some natural language and

recalls to external knowledge outside the question statement. Moreover, the evaluation is made with a focus on the content rather than style. This research field is defined in [Burrows et al. 2015] as Automatic Short Answer Grading (ASAG), and used in later works like [Roy et al. 2016] and [Zhang et al. 2016]. It consists in automatically assessing short natural language responses using computational methods.

The goal of this work is to propose a new method for automatic short answer grading, being based on what previous researches had found. This is done by incorporating the best of each approach and presenting results for two known datasets released during the SemEval 2013 competition task named "Student Response Analysis" [Dzikovska et al. 2013]. Also, we explore distinct types of features to accomplish better results by getting the best of each one. Especially, we report results for both datasets, for all possible scenarios and using multiple metrics, in order to perform comparisons with works from the original competition to more recent researches.

The remaining of this paper is organized as follows. Section 2 presents the data and experimental scenarios for the datasets. Section 3 reviews works that addressed the same task. In Section 4, the proposed method for this work is described. Section 5 shows the experiments and their results and discussion. Finally, Section 6 exposes our conclusions and future works.

2. Experimental Design

The data addressed in this work was released as two public datasets in 2012 by [Dzikovska et al. 2012] in order to provide a basis for development and evaluation of different automatic grading approaches[1]. Besides, this data was part of the "SemEval-2013 Task 7: The Joint Student Response Analysis" and several researchers submitted their models to the competition [Dzikovska et al. 2013]. Therefore, in this section, the data, tasks, baselines, and metrics are explained to provide a basis for our findings.

2.1. Datasets

The corpora is composed of two distinct datasets. The first one is the Beetle dataset, with data collected from the Beetle II tutorial dialogue system [Dzikovska et al. 2010], designed to teach students about basic electricity and electronics. These questions require a short answer, as seen in the following examples: "Why was bulb C off when switch Z was open?" and "Why does a damaged bulb impact a circuit?". The inter-rater agreement for the assigned labels to student answers was Kappa = 0.69.

The second is a set of answers for science entailment questions, the Science Entailment Bank (SciEntsBank), originated from the Assessing Science Knowledge tests [Nielsen et al. 2008]. Students from 3th to 6th grades across North America have responded to the questions. The subject of the questions consists of physical, life, earth and space sciences, as well as scientific reasoning and technology. Some examples of questions are: "How does the water flow in a creek during a flood compared to normal water flow?" and "A solution is a type of mixture. What makes it different from other mixtures?". The reported inter-rater agreement was Kappa = 0.728.

Both corpus are composed of questions, student answers, reference answers, and a grade defined on a scale. The Beetle dataset has 56 questions, about 3000 student

[1] www.cs.york.ac.uk/semeval-2013/task7/index.html

answers and from 1 to 14 reference answers per question. In its turn, SciEntsBank has 197 questions, about 10000 student answers and, unlike Beetle, only one reference answer per question [Dzikovska et al. 2012]. The labels assigned to each answer are one of the following [Dzikovska et al. 2013]:

- **Correct:** if the student answer is a completely correct answer and similar to the reference answer;
- **Partially_correct_incomplete:** if the answer is only partially correct or there is missing information;
- **Contradictory:** if the answer states the opposite of the reference answer;
- **Irrelevant:** if the student is writing inside the domain of study, but the answer is not applicable to the question;
- **Non_domain:** if there is nothing in the answer related to the expected answer. Examples: "I don't know" and "what the book says".

Answers classified in this way are part of the five-way task as specified by the competition organizers. Another classification is performed using three classes, where the labels *partially_correct_incomplete*, *irrelevant* and *non_domain* become all *incorrect*. Moreover, the labels *correct* and *contradictory* remains the same. Finally, for a two-way setting, the labels *contradictory* and *incorrect* are turned into *incorrect*.

2.2. Competition data split

When defining the competition, the organizers divided the data in order to evaluate different aspects of competitors' performance. The usual train/test split was performed to evaluate the generalization of systems. Additionally, the test part was split into three distinct test sets (see [Dzikovska et al. 2013] for more details):

- **Unseen Answers (UA):** a test set with the goal to assess system performance on predicting in answers from known questions. Every answer from this test set has a known (associated) specific question. The train/test split is performed preserving each answer with its corresponding question;
- **Unseen Questions (UQ):** created to assess the system capability to predict answers to questions never seen before, but that still fall in the knowledge domain of the training data;
- **Unseen Domains (UD):** a set of responses to questions not previously seen in the training data, and from different knowledge domain than the training set. This variation was created only for the SciEntsBank data.

The label distribution across both corpus (Beetle and SciEntsBank), ways (two, three or five) and test sets (UA, UQ, UD) can be seen in Table 1.

2.3. Baselines

In order to provide competitors with some base results, the organizers created two baseline models to measure performance in both Beetle and SciEntsBank. The first one is a simple majority class, i.e. assigning every sample as the most frequent class [Dzikovska et al. 2012].

The second is a lexical similarity baseline, built with a decision tree classifier with default parameters. Eight features were used to model the answers. They were extracted by calculating the similarity between each answer with its correspondent reference

Table 1. Label distribution. Adapted from [Dzikovska et al. 2013]

	Beetle			SciEntsBank			
Label	**Train**	**UA**	**UQ**	**Train**	**UA**	**UQ**	**UD**
correct	1665	176	344	2008	233	301	1917
pc_incorrect	919	112	172	1324	113	175	986
contradictory	1049	111	244	499	58	64	417
irrelevant	113	17	19	1115	133	193	1222
non_domain	195	23	40	23	3	0	20
incorrect-3way	1227	152	231	249	368	2228	2845
incorrect-2way	2276	263	475	307	432	2645	3384

answer and question. The four measures used are the Lesk Score (WordNet-based) and three token-based scores: Cosine, F1 (Sorensen) and Overlap. In case of more than one reference answer, the highest score was considered.

2.4. Evaluation metrics

The competition defined three metrics as the official way to compare results for the datasets [Dzikovska et al. 2013]. In this work, all three metrics will be reported in order to compare the results with the maximum of other researches. They are defined as follows:

- **Accuracy:** defined as the correctly predicted instances over the total of examples;
- **Macro-average F1 score:** is the average of precision, recall, and F1 across each class without weighing the values by class size. A special note is that for the SciEntsBank, in the five-way task, the *non_domain* class is highly underrepresented and, therefore, the results are presented considering only the other four classes;
- **Weighted-average F1 score:** similar to the macro-average, but in this case, each class size is taken into consideration as the weights of the score.

3. Related Work

When first released, the objective of Beetle and SciEntsBank was to provide a means for ASAG researchers to compare their models and stimulate development in the field. Following this idea, we searched for every work done on these datasets. Beyond the nine papers from the original competition, we found seven more that evaluates on the same data.

However, two of them provides no means of comparison, as the test scenario is not properly reported (which test set was taken for evaluation [Aldabe et al. 2015] and [Riordan et al. 2017]). In another work [Kumar et al. 2017], the authors changed the label from a categorical variable type to a continuous variable, turning the problem into a regression task and reported using respective metrics. Considering these conditions, the four left papers alongside with the three best-performing submissions from the original competition are considered here, totalizing seven analyzed works as follows.

1. **Softcardinality [Jimenez et al. 2013] (2013):** the system is based on text overlap through a specific way called Softcardinality and on a weight propagation mechanism. The Softcardinality consists of an aggregation of similarities between word and sentences to generate a final score. The baseline features from

[Dzikovska et al. 2012] were used, the data was preprocessed, and tree bagging classification technique was done to give final predictions.

2. **CoMeT [Ott et al. 2013] (2013):** is a meta-classifier built from three other subsystems developed from the authors: CoMiC, CoSeC and shallow bag approach. CoMiC uses features from the matches between student and reference answers after an annotation process, consisting of enhancing answers with information about POS tags, lemmas, chunks, and dependency parses. In contrast with CoMiC's lexical/syntactical approach, CoSeC is based on semantic similarity, using Lexical Resource Semantics and semantic networks to compute similarity.
3. **ETS [Heilman and Madnani 2013] (2013):** it is a system built on stacking [Wolpert 1992] and domain adaptation to join general text similarity measures to more item-specific ngrams features. The similarity measures consist of the original competitors' baseline features in conjunction with other measures obtained between the student answer and the reference answers alongside with other correct student answers.
4. **ZETEMA [Mancera et al. 2015] (2015):** is a web service for automatic short answer grading. The method employed is an improvement of the Softcardinality system, adapted to the web service format. The results are quite similar, but there are improvements for the SciEntsBank whilst Beetle have some decrease in performance. Then, authors conclude that systems have a certain equivalence.
5. **Magooda *et al.*'s [Magooda et al. 2016] (2016):** in this work, authors experimented with various sentence representation techniques and similarity measures to come up with a final system. Vector representation includes Word2Vec [Mikolov et al. 2013], GloVe [Pennington et al. 2014] and Sense Aware Vector [Neelakantan et al. 2015]. Similarity measures consist of lexical, knowledge and corpus-based types.
6. **Sultan *et al.*'s [Sultan et al. 2016] (2016):** it uses word and sentence alignments by means of lexical and semantic similarity. Moreover, semantic vector similarity is obtained from word embeddings. The authors also use the length ratio between student and reference answers, question demoting to not account for question words and term weighting, in a variety of term frequency-inverse document frequency (tf-idf).
7. **Roy *et al.*'s [Roy et al. 2016] (2016):** the authors also propose an ensemble technique that combines bag-of-words modeling with similarity measures extracted from answers. Furthermore, they employ a canonical correlation analysis based on transfer learning to build an ensemble classifier for questions with no labeled data.

4. Proposed Method

The proposed approach is composed of four sets of features. Each group has distinct characteristics and they are employed together in order to better model the task. Three of them are used in all three variations of test sets (UA, UQ, and UD) and the fourth one is only used for the UA test set, as bag-of-words are specifically defined for the answers of each question. Each group of features is described in the following subsections and can be seen in Figure 1. Section 5 will go over other details of Figure 1.

Preprocessing was applied to Text Statistics (partially) and Semantic Similarity,

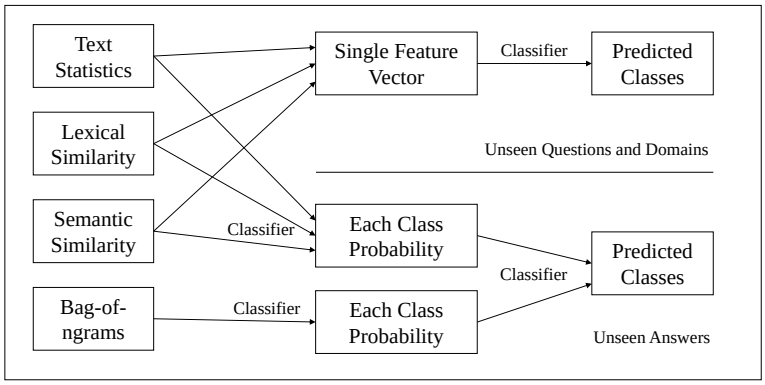

Figure 1. Proposed Method.

described in subsections 4.1 and 4.3. It consists of case normalization, non-alphanumeric character removal, spelling correction, lemmatization and stopword removal.

4.1. Text statistics

This set of features is composed of statistics extracted from each individual student answer and some ratio between them and reference answers alongside with questions. **Spelling Errors:** the count of found spelling errors. The idea is that students with less spelling errors could write better answers. **Length Ratio:** the length ratio between the student answer and the questions. Also, the maximum, minimum and mean of the ratio between the student answer and each reference answer. A large distance between the student and reference answer may indicate an incorrect answer. **Counts:** count per answer of words, sentences, commas, unique words, negation words and each part-of-speech tag (in the universal English tagset). Style of answers by the counts of their components may indicate better writing. **Word Length Average:** the simple average of the length of words in the answer. Can indicate if answers with larger words turn in correct or incorrect grading. **Words per Sentence Average:** the size of each sentence. Another style writing feature to measure if shorter or larger sentences can lead to correct answers.

4.2. Lexical similarity

This set of features is based on the lexical level similarity between the student answers, the question and the reference answers. This type of similarity is widely employed in automatic short answer grading research [Burrows et al. 2015]. In this work, the correct student answers are also considered as reference answers, as was also done in [Heilman and Madnani 2013]. Important to highlight that when measuring correct answers similarity, the correct student answer was not compared to itself, but only with the rest of the other correct student answers.

Four different groups of metrics are considered here to measure similarity, as grouped in the survey of [Vijaymeena and Kavitha 2016]. Each group has one or more metrics for the purpose of getting better results by using each metric's strength. They are described as follows:

1. **Token-based:** measures similarity between two strings by considering the intersection of characters in both texts. Three different metrics were selected: Cosine, Overlap, and Sorensen.

2. **Edit-based:** metrics of this type are based on counting the minimum number of operations performed to transform one string into the other. Levenshtein, Hamming, and Jaro-Winkler were used.
3. **Sequence-based:** unlike token-based, here the order counts and the similarity is based on sequences. One way is to measure the longest common substring between two given strings. The principle is that sentences with the longest shared sequences will be likely to be more similar. A variation of this idea is also employed in this work, the RatcliffObershelp similarity.
4. **Compression-based:** is similar to edit-based but the similarity is extracted from the shortest computer program that can convert one string (in this case, represented as a bit vector) to another. The representative algorithm used was the Normalized Compression Distance.

4.3. Semantic similarity

Semantic similarity in this work is measured using a semantic network (WordNet, as it is the most popular) to retrieve the semantic distance between words. In WordNet [Miller 1995], words are grouped in synsets, which are sets of cognitive synonyms, that represents some concept. The synsets are interlinked by their conceptual-semantic and lexical relations, providing a means to measure semantic similarity.

There are a few established algorithms that can compute word-to-word similarity in WordNet. They do so by walking through the links between synsets and measuring how close or distant they are, if they have hierarchical relationships, among other indicators. Six algorithms were used in this work: Leacock & Chodorow, Wu & Palmer, Lin, Resnik, Jiang & Conrath and Shortest Path (all available in WordNet interface and documentation).

In order to use a word-to-word similarity for measuring answers similarity, an algorithm was implemented as described in [Mohler and Mihalcea 2009]. The idea is to compute the Cartesian product of the synsets from the student and the reference answer, considering only open class words (nouns, verbs, adjectives, and adverbs). Then, the six mentioned algorithms compute the similarities and add to a vector. Finally, the average of each metric is returned.

In the SciEntsBank dataset, the preceding algorithm is directly applied, totalizing six features. However, in the Beetle dataset, is possible to have more than one reference answer per question. In this case, the algorithm is applied to each reference answer and the mean and max functions are applied to each of the six metrics, returning 12 features.

4.4. Bag-of-ngrams

Ngrams are one of the most common ways to model language in ASAG [Burrows et al. 2015]. It is based on the idea that the words' presence or absence can predict the desired output. The problem that arises from modeling text this way is that no order is considered and sentences with opposite meaning using a negation word can be considered very similar. Despite the apparent naivety from using ngrams, it is still one of the most powerful predictors in ASAG context [Magooda et al. 2016, Roy et al. 2016, Heilman and Madnani 2013].

As ngrams works on the principle of presence or absence of pieces of text, it is a question-specific feature. Important words for a question are not important to another.

Hence, each question has its own bag-of-ngrams sparse matrix of features, where each document is represented as a row and each ngram as a column. In this work, the simple binary count (presence or absence) was used, as tf-idf did not perform as good.

Two types of ngrams were extracted: words and characters. To illustrate the difference, consider the sentence "the open circuit". Word 2-grams of this sentence would be: ["the open", "open circuit"] and character 6-grams would be: ["the op", "he ope", "e open", ...]. For word ngrams, n ranged from 1 to 3 and for character ngrams n varied from 5 to 7. For each type of ngram, the top 250 features were kept (concerning its importance, i.e. term frequency in the documents), totalizing 500 features in a sparse matrix.

5. Experiments, Results and Discussion

As stated in the previous section, the bag-of-ngrams features were generated in a different manner from the other groups of features and, therefore, included in a special way. For the Unseen Questions (UQ) and Unseen Domains (UD) scenarios, the features from text statistics, lexical and semantic similarity were joined and scaled to compose the final set of features for further prediction (a single feature vector), as represented in Figure 1.

For the Unseen Answers (UA) scenario, however, the joint features of statistics and similarities were used to predict each class probability instead of a final class (using 10-fold cross-validation). Then, the bag-of-ngram model was also trained to predict probabilities (also using cross-validation, but with 5 folds due to the small number of samples per question), as can be seen in Figure 1. Finally, the class probabilities generated from both models were used as input to a meta-classifier, which predicted the final classes, in a stacking process [Wolpert 1992].

We experimented with some classifiers and the best-performing ones were chosen: Random Forests and Extreme Gradient Boosting (as indicated by recent research to be the best in general [Zhang et al. 2017]). Some parameter tuning was made to each scenario (in the number of estimators and learning rate).

In some of the 15 scenarios (seen in Tables 2 and 3), a simple threshold feature selection algorithm was applied. We performed experiments with automatic feature selection algorithms, but due to their slowness, we opted for a simple threshold algorithm. It works by cutting off features using a threshold value on its importance, usually being the mean, but also the median or an empirical value.

In order to evaluate the system's performance by comparing with as many works as possible, the results are presented in three metrics: accuracy, macro-averaged and weighted-averaged F1 scores, respectively in Tables 2 to 4. In each of these metrics, the mean score is also reported, as it was originally used for comparison among competitors in [Dzikovska et al. 2013].

Table 2 compares the results of the proposed system with two out of three best performing systems from the competition. The third ([Jimenez et al. 2013]) is on the accuracy table because the paper reports an improvement from the original competition submission.

As can be seen in Table 2, our system performs better in general. The cases where it goes worst are in all ways of Beetle UA. Some specific characteristic of this test set disadvantage our model, as it is the only case in both Tables 2 and 3 where we

performed worst. Also, in the three and five-way of Beetle UQ, the proposed system and [Heilman and Madnani 2013] performs almost the same. But in general, the overall results are way ahead.

Table 2. Macro-averaged F1-Score

Two-way						
	Beetle		**SciEntsBank**			
System	UA	UQ	UA	UQ	UD	Mean
Ott-2013	**0,833**	0,695	0,768	0,579	0,670	0,709
Heilman-2013	**0,833**	0,720	0,762	0,688	0,683	0,737
Proposed	**0,829**	**0,774**	**0,792**	**0,761**	**0,758**	**0,784**
Three-way						
	Beetle		**SciEntsBank**			
System	UA	UQ	UA	UQ	UD	Mean
Ott-2013	**0,715**	0,466	0,640	0,380	0,404	0,521
Heilman-2013	**0,710**	**0,585**	0,643	0,459	0,439	0,567
Proposed	0,677	**0,588**	**0,702**	**0,493**	**0,537**	**0,595**
Five-way						
	Beetle (5-way)		**SciEntsBank (4-way)**			
System	UA	UQ	UA	UQ	UD	Mean
Ott-2013	0,569	0,300	0,551	0,201	0,151	0,354
Heilman-2013	**0,619**	**0,552**	0,581	0,372	0,339	0,493
Proposed	0,570	**0,554**	**0,628**	**0,420**	**0,431**	**0,520**

Table 3. Accuracy Results

Two-way						
	Beetle		**SciEntsBank**			
System	UA	UQ	UA	UQ	UD	Mean
Jimenez-2013	0,797	0,725	0,717	0,733	0,726	0,740
Mancera-2015	0,772	0,714	0,744	0,716	0,724	0,734
Proposed	**0,836**	**0,785**	**0,798**	**0,769**	**0,758**	**0,790**
Three-way						
	Beetle		**SciEntsBank**			
System	UA	UQ	UA	UQ	UD	Mean
Jimenez-2013	0,608	0,532	0,656	0,671	0,646	0,623
Mancera-2015	0,414	0,415	0,667	0,652	0,657	0,561
Proposed	**0,695**	**0,592**	**0,744**	**0,708**	**0,706**	**0,687**
Five-way						
	Beetle		**SciEntsBank**			
System	UA	UQ	UA	UQ	UD	Mean
Jimenez-2013	0,572	0,476	0,552	0,520	0,534	0,531
Mancera-2015	0,538	0,449	0,526	0,546	0,519	0,516
Proposed	**0,683**	**0,606**	**0,659**	**0,557**	**0,566**	**0,614**

Next, we have [Jimenez et al. 2013] and [Mancera et al. 2015] reporting accuracy results in Table 3. In this case, our system outperforms in at least 0.011 every test case scenario, with overall accuracy results between 0.049 and 0.083 higher. As well as in Table 2, our proposed method better shows its strength in the overall scenario.

Finally, there are the weighted-averaged F1 scores from six of the seven analyzed works in Table 4, reporting the 5-way task in SciEntsBank (the most commonly reported evaluation). In this case, only three scenarios are reported, as they were the ones present in [Roy et al. 2016], [Magooda et al. 2016], and [Sultan et al. 2016]. Here, our proposed system performs slightly worse in all three cases scenarios against [Roy et al. 2016] and [Sultan et al. 2016]. However, in the overall case (mean score), we get slightly better results than them and way ahead of the others.

Table 4. Weighted-average F1-Score (SciEntsBank 5-way)

System	**UA**	**UQ**	**UD**	**Mean**
[Heilman and Madnani 2013]	0,625	0,356	0,434	0,472
[Ott et al. 2013]	0,598	0,299	0,252	0,383
[Jimenez et al. 2013]	0,537	0,492	0,471	0,500
[Roy et al. 2016]	**0,672**	0,518	0,507	0,566
[Magooda et al. 2016]	0,470	0,510	0,460	0,480
[Sultan et al. 2016]	0,582	**0,554**	**0,545**	0,560
Proposed	**0,666**	0,531	0,524	**0,574**

5.1. Preliminary Error Analysis

We performed a preliminary error analysis, studying some misclassified student answers, in order to get a glimpse at the current limitations of our approach. Also, it provided means to improve this work in the future. Following, we present two examples extracted from the SciEntsBank corpus, UA test set, and binary classification. The student answers from both examples were incorrect, but were predicted as correct.

Example 1. Question 1: *Carrie wanted to find out which was harder, a penny or a nickel, so she did a scratch test. How would this tell her which is harder?* Reference Answer 1 **(R1)**: *The harder coin will scratch the other.* Student Answer 1 **(S1)**: *If she can scratch a penny with her fingernail the nickel is harder.*

In Example 1, S1 is incorrect, but it looks like a correct answer. One reason is that with the simple change from "her fingernail" to "the nickel", the answer would be completely correct. As there are not incorrect answers in the training set using the word "fingernail" and most words from S1 are also used in correct answers, the learning model saw no problem in considering it correct. A possible workaround for this limitation could be the use of Semantic Role Labeling, a technique to correctly characterize the roles in a sentence (in the example, it would expect nickel to be the scratcher object).

Example 2. Question 2: *Diva's father told her she should not eat so many cookies because they were pure sugar. Diva decided to investigate the amount of sugar in Fruity Cream cookies. She performed the sugar test on 4 grams of pure sugar and on 4 grams of Fruity Cream cookies. The results are pictured at the right. Are Fruity Cream cookies pure sugar? What is your evidence?* Reference Answer 2 **(R2)**: *No, Fruity cream cookies are not pure sugar. The cookies did not produce as much gas as was produced by the pure sugar.* Student Answer 2 **(S2)**: *No, because the Fruity Cream cookies is at 150 milliliters.*

In this second example, the student answer got misclassified as correct because it is correct, but it is also incomplete in answering the question. Another student assigned with the *correct* label wrote: *No because if they were pure sugar it would be 200 milliliters not 150 milliliters.* In S2, the student forgot to compare with the pure sugar (or maybe he thought it was obvious, as there was a figure indicating). So, despite the correctness of S1, and the use of many words present in other correct answers, it lacked the most important: comparison words as done in R2.

6. Conclusions and Future Work

This work presented a new system to automatically grade short answers. The evaluation of the system was performed in all scenarios from the original competition that first introduced the used datasets. This was accomplished in order to compare the results with other literature researches and to establish new results for the further development of these specific datasets.

The proposed method incorporated the best from previous literature works, using a modern classifier and achieving better results in most scenarios. Especially, we reported results for the Beetle dataset and for all test case scenarios, usually not addressed in other ASAG researches. Despite the good results, there is still room for improvement and a need to create a model that can perform consistently good across all scenarios.

As future work, we first want to extend the performed error analysis, in order to discover more limitations in our approach. Then, we want to explore other word representation techniques for feature computation, such as word embeddings. Other possibilities include the use of graphs and complex knowledge networks to check on the veracity of student answers, besides the use of deep learning techniques.

References

Aldabe, I., Lacalle, O. L., Maritxalar, M., and Lopez-Gazpio, I. (2015). Supervised Hierarchical Classification for Student Answer Scoring.

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

Dzikovska, M., Nielsen, R., Brew, C., Leacock, C., Giampiccolo, D., Bentivogli, L., Clark, P., Dagan, I., and Dang, H. T. (2013). SemEval-2013 Task 7: The Joint Student Response Analysis and 8th Recognizing Textual Entailment Challenge. *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, 2(SemEval):263–274.

Dzikovska, M. O., Bental, D., Moore, J. D., Steinhauser, N. B., Campbell, G. E., Farrow, E., and Callaway, C. B. (2010). Intelligent tutoring with natural language support in the beetle ii system. In *European Conference on Technology Enhanced Learning*, pages 620–625. Springer.

Dzikovska, M. O., Nielsen, R. D., and Brew, C. (2012). Towards Effective Tutorial Feedback for Explanation Questions: A Dataset and Baselines. *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 200–210.

Heilman, M. and Madnani, N. (2013). ETS: Domain Adaptation and Stacking for Short Answer Scoring. *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, 2(SemEval):275–279.

Jimenez, S., Becerra, C., and Gelbukh, A. (2013). SOFTCARDINALITY: Hierarchical Text Overlap for Student Response Analysis. *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, 2(SemEval):280–284.

Kumar, S., Chakrabarti, S., and Roy, S. (2017). Earth mover's distance pooling over siamese LSTMs for Automatic short answer grading. *International Joint Conference on Artificial Intelligence*, pages 2046–2052.

Liu, O. L., Rios, J. A., Heilman, M., Gerard, L., and Linn, M. C. (2016). Validation of automated scoring of science assessments. *Journal of Research in Science Teaching*, 53(2):215–233.

Magooda, A., Zahran, M. A., Rashwan, M., Raafat, H., and Fayek, M. B. (2016). Vector Based Techniques for Short Answer Grading. *International Florida Artificial Intelligence Research Society Conference Ahmed*, pages 238–243.

Mancera, S., Jimenez, S., and Gonzalez, F. A. (2015). ZETEMA: A web service for automatic short-answer questions grading. *2015 10th Computing Colombian Conference (10CCC)*, pages 504–508.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Miller, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Mohler, M. and Mihalcea, R. (2009). Text-to-text semantic similarity for automatic short answer grading. *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics on - EACL '09*, pages 567–575.

Neelakantan, A., Shankar, J., Passos, A., and McCallum, A. (2015). Efficient non-parametric estimation of multiple embeddings per word in vector space. *arXiv preprint arXiv:1504.06654*.

Nielsen, R. D., Ward, W. H., Martin, J. H., and Palmer, M. (2008). Annotating students' understanding of science concepts. In *LREC*.

Ott, N., Ziai, R., Hahn, M., and Meurers, D. (2013). CoMeT: Integrating different levels of linguistic modeling for meaning assessment. *Second Joint Conference on Lexical and Computational Semantics , and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, 2(SemEval):608–616.

Passero, G., Haendchen Filho, A., and Dazzi, R. (2016). Avaliação do uso de métodos baseados em lsa e wordnet para correção de questões discursivas. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 27, page 1136.

Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Riordan, B., Horbach, A., Cahill, A., Zesch, T., and Lee, C. M. (2017). Investigating neural architectures for short answer scoring. *$Bea17*, pages 159–168.

Roy, S., Bhatt, H. S., and Narahari, Y. (2016). An Iterative Transfer Learning Based Ensemble Technique for Automatic Short Answer Grading. 285:1622–1623.

Santos, J. C. A. d. et al. (2016). Avaliação automática de questões discursivas usando lsa. *Universidade Federal do Pará*.

Sultan, M. A., Salazar, C., and Sumner, T. (2016). Fast and Easy Short Answer Grading with High Accuracy. *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1070–1075.

Vijaymeena, M. and Kavitha, K. (2016). A survey on similarity measures in text mining. *Machine Learning and Applications: An International Journal*, 3(2):19–28.

Wolpert, D. H. (1992). Stacked generalization. *Neural networks*, 5(2):241–259.

Zhang, C., Liu, C., Zhang, X., and Almpanidis, G. (2017). An up-to-date comparison of state-of-the-art classification algorithms. *Expert Systems with Applications*, 82:128–150.

Zhang, Y., Shah, R., and Chi, M. (2016). Deep Learning + Student Modeling + Clustering: a Recipe for Effective Automatic Answer Grading. *Proceedings of the 9th International Conference on Educational Data Mining*, pages 562–567.