Noname manuscript No.
(will be inserted by the editor)

The Joint Student Response Analysis and Recognizing Textual Entailment Challenge: making sense of student responses in educational applications

Myroslava O. Dzikovska · Rodney D. Nielsen · Claudia Leacock

Received: date / Accepted: date

Abstract We present the results of the Joint Student Response Analysis and 8th Recognizing Textual Entailment Challenge. The goal of this challenge was to bring together researchers from the Educational Natural Language Processing and Computational Semantics communities. The goal of the Student Response Analysis (SRA) task is to assess student responses to questions in the science domain, focusing on correctness and completeness of the response content.

Nine teams took part in the challenge, submitting a total of 18 runs using methods and features adapted from previous research on Automated Short Answer Grading, Recognizing Textual Entailment and Semantic Textual Similarity. We provide an extended analysis of the results focusing on the impact of evaluation metrics, application scenarios and the methods and features used by the participants. We conclude that additional research is required to be able to leverage syntactic dependency features and external semantic resources for this task, possibly due to limited coverage of scientific domains in existing resources. However, each of three approaches to using features and models adjusted to application scenarios achieved better system performance, meriting further investigation by the research community.

The research reported here was supported by the US ONR award N000141410733 and by the Institute of Education Sciences, U.S. Department of Education, through Grant R305A120808 to the University of North Texas. The opinions expressed are those of the authors and do not represent views of the Institute of Education Sciences or the U.S. Department of Education.

M.O. Dzikovska
School of Informatics, University of Edinburgh, Edinburgh, United Kingdom
E-mail: m.dzikovska@ed.ac.uk

R.D. Nielsen
University of North Texas, Denton, TX, USA
E-mail: Rodney.Nielsen@UNT.edu

Claudia Leacock
McGraw Hill Education/CTB, USA
E-mail: claudia.leacock@ctb.com

Keywords student response analysis · short answer scoring · recognizing textual entailment · semantic textual similarity

1 Introduction

Accurate assessment and feedback is a critical requirement of any effective educational program. Teachers spend large amounts of time providing feedback on homework and grading student responses to assessment questions. In recent years, the number of online educational applications has been growing steadily - including intelligent tutoring systems, e-learning environments, distance education and massive open online courses (MOOCs). To operate effectively, such applications require automated feedback and grading support, which requires processing natural language input.

The student's natural language input can be evaluated on multiple levels, differentiating, in particular between style and content, and between summative assessment and formative feedback (see Section 2). Student response analysis (henceforth SRA) is the task of evaluating the content of natural language student responses, and labeling them with categories that could help an educational application generate either formative or summative feedback.

In a typical setting, instructional designers create a repertoire of questions that the system can ask a student, together with reference answers (see Figure 1 for examples). For each student response, the system needs to decide on the appropriate tutorial feedback, either confirming that the student was correct, or providing additional help to indicate how their response is flawed and help the student improve. This task requires semantic inference, for example, to detect when the students are explaining the same content but in different words, and is linked to Recognizing Textual Entailment (RTE) and Semantics Textual Similarity (STS) tasks. Thus, Dzikovska et al (2012b) proposed SRA as a new shared task for the NLP community that would allow computational semantics methods to be tested in the context of a novel practical task.

This paper reports on the results of the shared student response analysis task at SemEval-2013 (SemEval Task 7), aiming to bring together the educational NLP and the semantic inference communities, in the context of the SRA task.[1] We describe the task, datasets and results, and provide an analysis of the results focusing on different application scenarios and features used by the participants. This work builds upon an initial analysis of the evaluation data (Dzikovska et al 2013b), which showed that it is not possible to determine a single "best" system, as different systems performed best on different test sets and metrics. We extend this initial analysis by examining the impact of evaluation metrics, application scenarios and the methods used by the participants. This detailed analysis can provide a basis for future evaluations of similar educational NLP tasks.

[1] There was also a partial entailment task, which is outside the scope of this paper - see Dzikovska et al (2013b) for details.

We begin by describing previous shared tasks in automated writing evaluation (Section 2) and discussing the novel features of the SRA task compared to previous shared tasks. We also briefly discuss the relationship between SRA and other semantic inference tasks. We describe our evaluation corpus, metrics and baselines (Section 3). We then analyze evaluation results along three dimensions (Section 4). The original SRA task proposed three different candidate evaluation metrics with different theoretical properties (Dzikovska et al 2012b). We show that in practice the system rankings were correlated across metrics (Section 4.2) and across task variants using different numbers of targeted feedback classes (Section 4.3). Next, we compare system performance across different application task settings (fully in-domain data versus transfer to new questions and domains, Section 4.4), and across different implementation features used by participants (syntactic features, semantic features and adaptation to the application setting, Section 4.5). We discuss the implications of our results for future NLP research on student response analysis in Section 5.

2 Background

2.1 Shared Tasks in Automated Writing Evaluation

The goals of automated writing evaluation (AWE) vary by application: automated grammatical error detection and correction (GEC), automated essay scoring (AES), and automated short answer grading (ASAG).

While GEC applications have been developed with the goal of helping writers, from the *Unix Writer's Workbench* (MacDonald et al 1982) to the present, GEC is also used to evaluate an essay writer's grasp of grammar, usage, mechanics and other writing conventions (Burstein et al 2013). In recent years, GEC's primary focus has shifted from errors made by native speakers to those most in need of it: English language learners (Leacock et al 2014) which has led to four shared tasks in the NLP community that focus on grammatical error detection for English language learners (Dale and Kilgarriff 2011; Dale et al 2012; Ng et al 2013, 2014). As a sideline to GEC for language learners, a new shared task for Native Language Identification (NLI) was introduced (Tetreault et al 2013) – since knowing the native language of a writer can help predict the kinds of errors likely to be made. For example, if the native language contains no articles (*a, an, the*), then the writer is likely to make article errors in their written English.

In the AES task, the goal is typically to evaluate the quality of writing, in particular essay structure and coherence, and on writing proficiency represented by correct spelling and grammar. The earliest program for scoring essays was developed by Ellis Page in 1996 (Page 1996). By now automated essay scoring systems are being used by most, if not all, major educational assessment companies (Shermis and Burstein 2013).

To aid the effective comparison of AES methods, the William and Flora Hewlett Foundation announced a series of shared tasks for automated scoring in 2012 – offering $100,000 in prizes for each task. The competition was called ASAP (Automated Student Assessment Prize) that was administered by Kaggle (`https://www.kaggle.com/c/asap-aes`). The first phase was for scoring long-form constructed responses (essays). The initial essay data set consisted of six extended-essay prompts: three for "traditional" genres (persuasive, expository and narrative essays) and three for source-based essays, where the essay has to be developed based on several reading passages (sources) – usually written in different styles and often presenting conflicting opinions. These competitions opened up AES to the world at large, including many machine-learning experts with little or no previous experience with NLP, providing shared data and annotations necessary to facilitate progress in the field. However, especially for the "traditional" genres, the essay prompts did not have an expected "correct" answer, and thus the factual correctness and completeness of the specific content were of lower importance for the AES ASAP task.

In contrast, it has long been understood that, especially in STEM (science, technology, engineering and mathematics) subjects, deeper semantic processing is required in order to score student short-text responses (Leacock and Chodorow 2003; Pulman and Sukkarieh 2005; Nielsen et al 2009; Mohler et al 2011). In applications such as assessment scoring and intelligent tutoring systems, each question posed to the student comes with one or more expected correct answers provided by the content experts ("reference answer"). The student is required to cover the relevant points in the reference answer set completely and correctly for their response to receive full credit. Therefore, a system that analyzes student responses to such questions needs to assess the semantic content of the response in some way, in order to establish entailment or similarity between the content of the student response and the content of the reference answer.

This task is called automatic short answer grading task (ASAG) and involves assessing factual correctness and completeness of student answers. A detailed review of ASAG research is presented in Burrows et al (2015a). The paper identifies three key dimensions of ASAG tasks that differentiates them from AES tasks: they assess recall of facts, focus on content and not on style, and expect the answer to be one phrase to one paragraph long. The SRA task presented in this paper is a typical example of an ASAG task. In the rest of the section, we discuss the ASAG task administered as part of the ASAP competition and on the contribution that the SRA task makes to the field.

The second phase of the ASAP competition focused on scoring short-form constructed responses (short answers). The data set consisted of ten short-text prompts covering general science, biology, and English. Sample sizes ranged from about 2,100 to 3,000 student responses per item. With such large amounts of data, statistically-based machine learning methods can be used to score short answers relying on the large set of training samples to cover the possible range of student responses.

The SRA task attempts to address two important issues that were not covered by the previous shared tasks, motivated by the needs of educational applications such as homework assessment or intelligent tutoring. The numeric scores in The Hewlett Foundation competitions represent summative assessment needs – that of grading large-scale assessments.

In formative assessments, such as homework grading or intelligent tutoring, detailed information that goes beyond numeric scores is required for providing effective feedback (Nielsen et al 2008b; Dzikovska et al 2012b). Thus, the SRA task focuses on associating student responses with categorical labels that can be used in providing feedback, labeling student responses as correct, partially correct but incomplete, contradictory to the expected answer, in-domain but not providing content relevant to the answer, and out-of-domain and garbled responses (e.g., swear words, random characters typed). These categories are described in Section 3. They provide a basis for giving students more specific directive feedback, for example, that their response is substantially correct but needs to be extended, or there are factual errors that need to be addressed (Dzikovska et al 2012b, 2013a). [2]

Secondly, the SRA task is set up to evaluate the application performance in scenarios where limited data are available. Having a shared and annotated data set with many student responses to the same question, as in the short answer grading task, is obviously of great value to the community, allowing for comparison and improvement of existing techniques. However, outside the summative exam assessment community, much of the work on providing detailed feedback is motivated by the needs of intelligent tutoring systems and e-learning systems (Graesser et al 1999; Glass 2000; Pon-Barry et al 2004; Jordan et al 2006a; VanLehn et al 2007; Dzikovska et al 2010). If one wanted to use SRA techniques in courses authored and conducted in typical classroom settings, large quantities of sample responses as seen in the Hewlett challenge are difficult or impossible to obtain. Moreover, instructors may want to author new questions or change existing ones as a course develops.

Thus, in addition to a common assessment scenario where multiple possible student responses are collected for each questions, there is need for systems that can operate on newly added questions in the same domain for which graded student responses are not yet available. An ideal assessment system would also be able to function in new domains, for example, by using textual entailment or semantic similarity between the student answer and the reference answer, without the need to collect large amounts of question-specific data.

The SRA challenge at SemEval 2013 was set up to evaluate the feasibility of building systems that could operate more or less unchanged across a range of domains and question-types, requiring only a question text and a reference answer for each question. It therefore provides test sets that evaluate three different application scenarios. Each system's performance is evaluated on

[2] It is easy to imagine a numeric grading scheme that converts such categorical labels into numeric scores, making the SRA labels equally useful for supporting summative assessment.

an in-domain task where labeled data are provided for each question, and on transfer tasks where we look at system performance on previously unseen ("newly authored") questions and domains. The corresponding test sets are discussed in more detail in Section 3.4.

2.2 Relation to other computational semantics tasks

When the SRA task was first proposed, we noted that it has a strong relationship to the Recognizing Textual Entailment task. The RTE community has a tradition of evaluating the RTE tasks within application contexts. Initially, information retrieval, question answering, machine translation and information extraction tasks were used (Dagan et al 2006; Giampiccolo et al 2008), followed by Summarization and Knowledge Base Population (Bentivogli et al 2009, 2010, 2011).

In a typical response analysis scenario, we expect that a correct student response would entail the reference answer, while an incorrect response would not. More precisely, since students often skip details that are mentioned in the question text, the combination of the question text and student response text should entail the reference answer. An initial feasibility study concluded that labels assigned by human annotators in educational settings align sufficiently well with entailment judgments of annotators previously involved in RTE tasks (Dzikovska et al 2013b).

We therefore decided to formally engage the RTE community by providing an additional set of subtasks which used simplified 2- and 3-class labeling schemes similar to 2- and 3-way entailment judgments used in the previous RTE challenges. These labeling subtasks are discussed in more detail in Section 3. The challenge for the textual entailment community was to address the answer analysis task at varying levels of granularity, using textual entailment techniques, and explore how well these techniques can help in this real-world educational setting.

The task participants were obviously not restricted to only using RTE methods. Another clearly related task is the Semantic Textual Similarity (STS) task (Agirre et al 2012, 2013). Clearly, any correct student response is expected to be semantically similar to the reference answer, or, more precisely, the combination of question and student response should be similar to the reference answer. However, the challenge for STS systems is to map the numeric judgments of textual similarity into categorical judgments of correctness. This may not always be obvious - for example, for a simple similarity metric based on LSA approach, two contradictory sentences may appear to be very similar if they share most of the content words.

In practice, most of the systems entered into the task used some measure of semantic similarity as one of the features, and several systems designed for STS participated in this task as well. We discuss the range of NLP techniques used by participants in Section 4.1, and analyze the impact of commonly used features in Section 4.5.

QUESTION	The sand and flour in the gray material from mock rocks is separated by mixing with water and allowing the mixture to settle. Explain why the sand and flour separate.	
REFERENCE ANSWER	The sand particles are larger and settle first. The flour particles are smaller and therefore settle more slowly.	
STUDENT RESPONSE 1	The sand and flour separate because sand floats to the top and the flour stays on the bottom.	*contradictory*
STUDENT RESPONSE 2	Because sand is heavier than flour.	*partially_correct_incomplete*
QUESTION	Explain why you got a voltage reading of 1.5 for terminal 1 and the positive terminal.	
REFERENCE ANSWER	Terminal 1 and the positive terminal are separated by the gap	
STUDENT RESPONSE 1	Because there was not direct connection between the positive terminal and bulb terminal 1	*correct*
STUDENT RESPONSE 2	Voltage is the difference between a positive and negative end on a battery.	*irrelevant*
STUDENT RESPONSE 3	tell me the answer	*non_domain*

Fig. 1 Example questions and answers, with labels assigned

3 Evaluation Data

3.1 Student Response Analysis Corpus

We used the Student Response Analysis corpus (henceforth SRA corpus) (Dzikovska et al 2012b) as the basis for our data set creation. The corpus contains manually labeled student responses to explanation and definition questions typically seen in practice exercises, tests, or tutorial dialogue.

Specifically, given a question, a known correct 'reference answer' and a 1- or 2-sentence 'student response', each student response in the corpus is labeled with one of the following judgments:

- 'Correct', if the student response is a complete and correct paraphrase of the reference answer;
- 'Partially_correct_incomplete', if it is a partially correct response containing some but not all information from the reference answer;
- 'Contradictory', if the student response explicitly contradicts the reference answer;
- 'Irrelevant' if the student response is talking about domain content but not providing the necessary information;
- 'Non_domain' if the student utterance does not include domain content, e.g., "I don't know", "what the book says", "you are stupid".

The examples for all five classes are shown in Figure 1.

The SRA corpus consists of two distinct subsets: BEETLE data, based on transcripts of students interacting with BEETLE II tutorial dialogue system (Dzikovska et al 2010), and SCIENTSBANK data, based on the corpus of student responses to assessment questions collected by Nielsen et al (2008b).

The BEETLE corpus consists of 56 questions in the basic electricity and electronics domain requiring 1- or 2- sentence answers, and approximately 3000 student responses to those questions. The SCIENTSBANK corpus contains approximately 10,000 responses to 197 assessment questions in 15 different science domains (after filtering, see Section 3.2)

Student utterances in the BEETLE corpus were manually labeled by trained human annotators using a scheme that straightforwardly mapped into SRA annotations. The annotations in the SCIENTSBANK corpus were converted into SRA labels from a substantially more fine-grained scheme by first automatically labeling them using a set of question-specific heuristics and then manually revising them according to the class definitions (Dzikovska et al 2012b). We further filtered and transformed the corpus to produce training and test data sets as discussed in the rest of this section.

3.2 Data Preparation and Training Data

In preparation for the task, four of the organizers examined all questions in the SRA corpus, and decided to remove some of the questions to make the dataset more uniform. We observed two main issues. First, a number of questions relied on external material, e.g., charts and graphs. In some cases, the information in the reference answer was sufficient to make a reasonable assessment of student response correctness, but in other cases the information contained in the questions was deemed insufficient and the questions were removed.

Second, some questions in the SCIENTSBANK dataset could have multiple possible correct answers, e.g., a question asking for any example out of two or more unrelated possibilities. For example, consider the following question: '*Q: Sue, Linda, and Arletta, each got matching plant pots and the same amount of the same kind potting soil. Each girl planted 8 sunflower seeds in the soil in her pot. Then each girl took her pot home to watch the growth of the plants. After 10 days they brought their plant pots back to school and compared the results. The picture to the left shows their results. Describe 2 ways the results are different.*' Its reference answer is '*The number of plants is different. The plant heights are different. The number of leaves is different.*' To answer correctly, the student is expected to name only two out of three components of the reference answer; thus, the student response and the question will neither entail nor be semantically similar to all parts of the reference answer. Such questions present additional challenges for RTE and STS methods, beyond those outlined in Section 2.2. They were therefore removed from the test data for our task, though they should be considered in the future SRA work.

Finally, parts of the data were re-checked for reliability. In BEETLE data, a second manual annotation pass was carried out on a subset of questions to check for consistency. In SCIENTSBANK, we manually re-checked the test data. The automatic conversion from the original SCIENTSBANK annotations into SRA labels was not perfectly accurate (Dzikovska et al 2012b). We did not have the resources to check the entire data set. However, four of the organizers jointly hand-checked approximately 100 examples to establish consensus, and then one organizer hand-checked all of the test data set.

3.3 Simplified labeling subtasks

The 5-label annotation scheme used in the SRA corpus is based on previous work on annotating student correctness in tutorial dialogue (Campbell et al 2009; Nielsen et al 2008b) and attempting to create a shared set of labels that can align with common feedback decisions. However, less complicated labeling schemes that highlight relationships to other computational semantics tasks are also of interest to the computational linguistics community.

We therefore created two subtasks based on the same set of data that are strongly similar to previous RTE challenges, a 3-way and 2-way labeling subtasks. The data for those tasks were obtained by automatically collapsing the 5-way labels. In the 3-way task, the systems were required to classify the student responses as either (i) *correct*; (ii) *contradictory*; or (iii) *incorrect* (combining the categories partially correct but incomplete, irrelevant and not in the domain from the 5-way classification).

In the two-way task, the systems were required to classify the student responses as either correct or incorrect (combining the categories contradictory and incorrect from the 3-way classification)

We discuss these subtasks briefly in Section 4.3. The results there show that system rankings and performance was consistent across the subtasks. Therefore, we focus on the 5-way labeling for most of our data analyses.

3.4 Test Data and Evaluation Tasks

We followed the evaluation methodology of (Nielsen et al 2008a) for creating the test data. Since our goal is to support systems that generalize across problems and domains, we created three distinct test sets:

1. **Unseen Answers (UA)**: a held-out set to assess system performance on the responses to questions contained in the training set (for which the system has seen other example student responses). It was created by setting aside a subset of randomly selected learner answers to each question included in the training data set.
2. **Unseen Questions (UQ)**: a test set to assess system performance on responses to questions for which it has not seen any student responses in the training data, but which still fall within the science domains represented

Table 1 Question distribution in training and test data

	BEETLE	SCIENTSBANK	Total
Training	47	135	182
Total Test	56	196	252
Test Subset:			
Unseen Answers (UA)	47	135	182
Unseen Questions (UQ)	9	15	24
Unseen Domains (UD)	–	46	46

Table 2 Student response distribution over the training and three distinct test sets

Test Set	BEETLE	SCIENTSBANK	Total
Training	3941	4969	8910
Unseen Answers (UA)	439 (35%)	540 (9%)	979 (14%)
Unseen Questions (UQ)	819 (65%)	733 (13%)	1532 (22%)
Unseen Domains (UD)	– –	4562 (78%)	4562 (64%)

in the training data. It was created by holding back all student answers to a subset of randomly selected questions in each dataset.

3. **Unseen Domains (UD)**: a domain-independent test set of responses to topics not seen in the training data, available only in the SCIENTSBANK dataset. It was created by setting aside the complete set of questions and answers from three science modules from the fifteen modules in the SCIENTSBANK data.

These test sets correspond to the common educational NLP tasks discussed in Section 2.1. The UA test sets represent application settings where question-specific training data can be collected and annotated. The UQ and UD test set represent transfer performance to new questions and new domains respectively, in situations, for example, where a course instructor authors a new question during course development, but does not have the resources to collect a labeled data set and re-train the classifier. We will refer to these different situations as "scenarios". Some of the participating systems attempted to adapt their models to different scenarios, and we evaluate the effectiveness of such adaptation in Section 4.5

The statistics for questions and student responses in the different test sets are shown in Tables 1 and 2. The final label distribution for train and test data is shown in Table 3. It shows that the training and test data have very similar label distributions. Approximately the same percentage was also seen across the UA, UQ and UD test sets.

3.5 Evaluation Metrics and Baselines

We used two baselines: the majority (most frequent) class baseline and a word overlap baseline described in detail in Dzikovska et al (2012b). The performance of the baselines is presented jointly with system scores in the results tables.

Table 3 Label distribution in the BEETLE and SCIENTSBANK training and test sets. Percent of the corresponding train/test set in parentheses.

	BEETLE		SCIENTSBANK	
Label	Train (%)	Test (%)	Train (%)	Test (%)
correct	1665 (42)	520 (41)	2008 (40)	2451 (42)
partially_correct_incomplete	919 (23)	284 (23)	1324 (27)	1274 (22)
contradictory	1049 (27)	355 (28)	499 (10)	539 (9)
irrelevant	113 (3)	36 (3)	1115 (22)	1548 (27)
non_domain	195 (5)	63 (5)	23 (.5)	23 (.4)
Total	3941(100)	1258(100)	4969(100)	5835(100)

For each evaluation data set (test set), we computed the per-class precision, recall and F_1 score. We also computed three main summary metrics: accuracy, macro-average F_1 and weighted-average F_1.

Accuracy is computed as the percent of all test set responses classified correctly.

Macro-average, for a given metric, is computed as a simple average of the per-class values for that metric.

$$\frac{1}{k}\sum_{c=1}^{k} metric_c \tag{1}$$

where k is the number of classes[3] and $metric_c$ is the value for class c of the metric being averaged (P, R, or F_1). A macro-averaged metric favors systems that perform comparatively well across all classes, as each class contributes equally to the final value of the metric. For example, the word overlap baseline significantly outperforms the majority class baseline on this metric, because the latter scores zero on each class other than the most frequent class.

Weighted Average, for a given metric, is computed by multiplying each per-class value for that metric by the proportion of examples having that label in the gold standard.

$$\sum_{c=1}^{k} \frac{n_c}{N} metric_c \tag{2}$$

where n_c is the number of test set examples labeled as class c in the gold standard and N is the total number of examples in the test set. Relative to the macro-average F_1 score, the weighted-average metric favors systems that perform better on the most frequent class, as its weight dominates all others.

Seeing the substantial differences in scores and rankings assigned by the different evaluation metrics to the two baselines that we noted above, one could also expect to see differences in system rankings between the metrics. This motivated using all three metrics in the original evaluation. However, in

[3] In the 5-way task, we excluded the "non-domain" class from the calculation of the macro-average on the SCIENTSBANK dataset because it had so few examples and was severely underrepresented with only 23 out of 4335 total examples and, hence, could have had a significant random effect.

practice the rankings assigned to different systems were quite similar across the different metrics. We analyze the rankings assigned by different metrics in Section 4.2.

4 Results

4.1 Participants

The participants were invited to submit up to three runs for any combination of the tasks. Nine teams participated in the task, most choosing to attempt all subtasks (5-way, 3-way and 2-way labeling), with one team entering only the 5-way and one team entering only the 2-way subtask. Several teams submitted multiple runs, aiming to better understand the contributions of different features in the overall system performance.

Four of the teams submitted systems custom built for educational applications, either from scratch (CU, EHU-ALM), or by adapting an existing system (ETS, CoMeT). Four teams submitted systems relying on existing RTE (CELI, UKP-BIU) or STS (CNGL, SoftCardinality) systems. One team built a system using a paraphrase identification algorithm (LIMSIILES) that was later adapted to question answering. In addition, some of the features used by the ETS system were also used by an ETS entry in the STS shared task (Heilman and Madnani 2013b) as well. Thus, a wide range of techniques, with roots in different strands of computational semantics research, has been used for this challenge.

In order to better understand the types of features used by the systems, and their potential impact on performance, we classified the systems according to three dimensions:

- Use of **Syntactic dependencies**: Does the system use a bag-of-words approach, or attempt to utilize syntactic dependency features? It has been argued previously that parsing (rather than just using bag of words features) is necessary for analyzing student responses for correctness in tutoring systems (Wolska and Kruijff-Korbayová 2004; Jordan et al 2006b). Examples of syntactic dependency features used are dependency n-grams (UKP-BIU) or syntactic structure overlap (EHU-ALM).
- Use of **Semantic resources**: Does the system attempt to use semantic similarity features that rely on external resources? These could be either in the form of external dictionaries such as WordNet, or clustering similar words together based on additional corpora. While such resources should be useful in all scenarios, we would expect them to be particularly useful for Unseen Questions and Unseen Domains test sets, where there may not be sufficient training data to derive domain-specific lexicalized features. The most commonly used feature was a WordNet similarity score, with some systems using features based on Wikipedia or Europarl corpora, as discussed below.

Table 4 Summary of features from different system runs. Syn: uses syntactic dependencies; Sem: uses semantic classes; Scn: provides scenario-specific features

System	Syn	Sem	Scn	Approach
$CELI_1$	-	+	-	RTE
$CELI_2$	-	+	-	RTE
$CNGL_1$	+	+	-	STS/MT
$CNGL_2$	+	+	-	STS/MT
$CoMeT_1$	+	+	+	Educational/Second language learning
CU_1	-	-	-	Educational/Essay Scoring
CU_2	-	-	-	Educational/Essay Scoring
$EHU\text{-}ALM_1$	+	+	+	Educational/Custom
$EHU\text{-}ALM_2$	+	+	+	Educational/Custom
$EHU\text{-}ALM_3$	+	+	+	Educational/Custom
ETS_1	-	+	+	Educational/Essay Scoring
ETS_2	-	-	+	Educational/Essay Scoring
ETS_3	-	+	+	Educational/Essay Scoring
$LIMSIILES_1$	+	+	-	Paraphrase Identification/QA
$SoftCardinality_1$	-	-	-	STS
$UKP\text{-}BIU_1$	-	+	-	RTE
$UKP\text{-}BIU_2$	+	+	-	RTE
$UKP\text{-}BIU_3$	+	-	-	RTE

– **Scenario-specific features**: Does the system use different features or algorithms depending on the test set type (Unseen Answer, Unseen Question, Unseen Domain) or otherwise attempt to adjust its model to the evaluation scenario? This category covers a range of diverse features. For example, the ETS_3 system has two copies of each syntactic and semantic feature it uses: a "generic" copy with weight trained on the entire data set, and a "specific" feature trained only on scenario-specific data. We discuss this further in Section 5.

Table 4 displays the summary of all submitted runs, and classifies them according to the above criteria. In the remainder of this section, we briefly review the participating systems, focusing on how they align with our classification criteria. For each team, we summarize how the runs differ and what kind of syntactic, semantic or scenario-specific features are used. Most teams also used other features, such as word overlap, which we do not discuss due to space limitations.

CELI (CELI S.R.L.): The CELI team submitted two runs (Kouylekov et al 2013). Both use features based on edit distance to compute word overlap between the student response and the expected answer. A corpus-based distributional approach based on Wikipedia corpus is used to provide a similarity metric for word matching. The difference between runs is in the details of edit distance calculation.

CNGL (Centre for Next Generation Localisation): The CNGL team submitted two runs (Bicici and van Genabith 2013) using machine translation features. The features are computed over n-grams or over head-modifier dependencies

extracted using an unsupervised syntactic parser. An n-gram language model trained on a Europarl corpus is used as an additional feature.[4] The difference between the two runs is the classifier used in training.

CoMeT (Universitat Tubingen): The CoMeT team submitted a single system using a combination of syntactic dependencies, corpus based similarity features and WordNet similarity features (Ott et al 2013). In addition, the system performs scenario adaptation by choosing (based on a development set created by the authors from the training data) which feature subset to use, depending on whether the system is presented with an UA, UQ or UD question. In particular, bag-of-word features are not used for UQ and UD questions, since they appeared to have a negative impact on results on the development set.

CU (University of Colorado): The CU team submitted two runs (Okoye et al 2013). Both runs use only bag of words, lemmas and parts of speech features, but do not attempt to use either syntactic dependencies or semantic similarity classes. The difference between the two runs is in the cost parameter of the LibSVM (Chang and Lin 2011) binary classifier. This team only entered the 2-way subtask.

EHU-ALM (University of Basque Country and University of Edinburgh): The EHU-ALM team submitted three runs, using features based on syntactic dependencies, WordNet similarities and corpus-based similarity measures, as well as scenario adaptation (Aldabe et al 2013). All the runs use syntactic and semantic features, but EHU-ALM$_1$ trained to Unseen Answers vs. other scenarios; EHU-ALM$_2$ provides separate classifiers for UA, UQ and UD scenarios; EHU-ALM$_3$ adds adaptation to question type ("what", "why", "how" etc.).

ETS (Educational Testing Service): The ETS team submitted three runs, using different combinations of n-gram, text similarity and domain adaptation features (Heilman and Madnani 2013a). All three runs accomplish scenario tailoring via domain adaptation, having general and specific copies of features (Daume III 2007). Two of the runs (ETS$_1$ and ETS$_3$) also use a textual similarity feature (PERP) that relies on WordNet as part of the similarity calculation (Heilman and Madnani 2012), while ETS$_2$ uses only features computed from the training data without external resources. No syntactic dependency features are used by any of the submitted runs.

LIMSIILES (Basic English Substitution for Student Answer Assessment): The LIMSIILES team submitted a single run (Gleize and Grau 2013). The system uses a paraphrase model based on syntactic features, acquired from Simple Wiktionary as a source of semantic equivalence. In addition, features based on Stanford Parser dependencies and WordNet similarity scores are used. No scenario tailoring is performed.

[4] We decided to treat it as a semantic resource feature for purposes of our analysis, since it relies on an external corpus, though its exact use is not very clear in the prior literature.

SoftCardinality: The SoftCardinality team submitted a single run, using "soft cardinality" features based on character and word overlap, without additional syntactic or semantic processing, and without any scenario-specific features (Jimenez et al 2013).

UKP-BIU (Ubiquitous Knowledge Processing Lab and Bar-Ilan University): The UKP-BIU team submitted three runs with features based on word overlap, WordNet similarity and syntactic dependency-based entailment features from the BIUTEE system (Levy et al 2013). UKP-BIU$_1$ uses WordNet features only. UKP-BIU$_2$ uses both WordNet and dependency features. UKP-BIU$_3$ uses dependency features only. No scenario-specific features are used.

Overall, there were eight runs from five teams using syntactic dependency information, 11 runs from six teams using external semantic resources, and seven runs from three teams using scenario-specific features. In section 4.5, we use this classification to compare how the different features and resources that were brought to bear upon the task correlate to system performance.

4.2 Evaluation metrics and system rankings

As shown in Section 3.5, we calculated three different evaluation metrics for the system performance: accuracy, macro-average F_1 and weighted-average F_1. The evaluation results for all metrics and all participant runs are provided online.[5] To better understand how different metrics serve in terms of ranking system performance, we computed and compared rankings for each system according to each metric on the 5-way labeling.

However, the rankings assigned by different metrics to the same system on the same test set are highly correlated. When comparing weighted-average F_1 and macro-average F_1, the average Kendall's τ is 0.84. Similarly, comparing macro-average F_1 and accuracy, Kendall's τ has a mean value of 0.65. In other words, all three metrics result in similar system rankings for our data set, presumably because participants did not tune their system to a single metric.

Given this similarity, we decided to focus on a single metric, weighted-average F_1, as the evaluation metric in the rest of this paper. We chose F_1 over accuracy, due to the imbalance in the label distribution, and chose the weighted-average in preference to macro-average because the imbalance was so severe in the SCIENTSBANK training data (just 23 of 4,969 training examples were labeled 'non_domain') that it made learning extremely challenging, and so severe in the test data (there were only single digit numbers in some test sets) that it resulted in chance and variance playing a significant role in system performance on that class.

The system ranking according to the weighted-average F_1 is shown in Table 5. While rankings did not vary significantly depending on the metric, individual system rankings can differ substantially for different domains and

[5] `http://bit.ly/11a7QpP`

scenarios, as can be seen in the table. For example, ETS_2 is ranked first for BEETLE UA and UQ test sets, second for the SCIENTSBANK UA test set, but only 13th for the SCIENTSBANK UQ test set, and sixth for the SCIENTSBANK UD test set.

4.3 Comparing results across labeling subtasks

Tables 6, 7, 8 present the F_1 scores for the best system runs. Results are shown separately for each test set (TS). In addition, we report mean UA performance (obtained by averaging BEETLE Unseen Answers and SCIENTSBANK Unseen Answers sets), mean UQ performance (averaging BEETLE and SCIENTSBANK Unseen Questions sets), and the overall mean on all five test sets.

For reasons of clarity, we report the single run with the best average TS performance for each participant, identified by the subscript in the run title, with the exception of ETS. With all other participants, there was almost always one run that performed best for a given metric on *all* the test sets. In the small number of cases where another run performed best on a given TS, we instead report that value and indicate its run with a subscript (these changes never resulted in meaningful changes in the performance rankings). However, ETS_1 and ETS_2 use distinct feature sets that result in very different performance patterns across the five test sets. Therefore, we decided to report ETS_1 and ETS_2 separately.

The top performing system and systems with performance that was not statistically different from the best results for a given TS are shown in **bold** (significance was not calculated for the mean columns). Systems with performance statistically better than the lexical baseline are displayed in *italics*. Statistical significance tests were conducted using an approximate randomization test (Yeh 2000) with 10,000 iterations; $p \leq 0.05$ was considered statistically significant.

Not surprisingly, evaluation scores on 2-way labeling were the highest and on 5-way labeling the lowest, reflecting the additional difficulty of differentiating between a larger number of small classes.

In all tasks, all the systems performed significantly better than the majority class baseline. However, the lexical baseline was somewhat harder to beat on simpler tasks. On the 5-way task, six out of nine systems outperformed the lexical baseline on the overall mean score. On the 3-way labeling subtask, five of the eight systems outperformed the lexical baseline on the mean TS results, and on the 2-way labeling, four out of nine systems outperformed the lexical baseline on the mean. This may indicate that the more sophisticate features introduced by different systems are helping the most in differentiating the small additional classes introduced by the 5-way SRA labeling.

The participating systems performed consistently across the tasks, even though there was not a single best performer across different test sets. For the BEETLE UA and SCIENTSBANK UA tests sets, $CoMeT_1$ and ETS_2 were consistently the two teams outperforming the lexical baseline, with ETS_2 the

Table 5 System rankings on 5-way labeling, using weighted-average F_1

Dataset:	BEETLE		SCIENTSBANK			
Run	UA	UQ	UA	UQ	UD	Mean
$CELI_1$	14	12	14	11	13	12
$CELI_2$	17	14	17	18	18	18
$CNGL_1$	12	13	18	16	17	16
$CNGL_2$	9	4	15	15	14	13
$CoMeT_1$	3	7	3	14	15	8
$EHU\text{-}ALM_1$	6	11	10	5	4	6
$EHU\text{-}ALM_2$	4	11	8	5	4	5
$EHU\text{-}ALM_3$	8	9	7	10	7	7
ETS_1	7	3	6	2	2	3
ETS_2	1	1	2	13	5	2
ETS_3	2	2	1	6	8	1
$LIMSIILES_1$	10	8	13	3	6	9
$SoftCardinality_1$	5	6	5	1	1	4
$UKP\text{-}BIU_1$	13	15	4	12	9	11
$UKP\text{-}BIU_2$	15	17	9	8	12	14
$UKP\text{-}BIU_3$	16	18	11	9	11	15
Baselines:						
Lexical	11	5	12	7	10	10
Majority	18	16	16	17	16	17

Table 6 Five-way task weighted-average F_1

Dataset:	BEETLE		SCIENTSBANK			Mean		
Run	UA	UQ	UA	UQ	UD	UA	UQ	All
$CELI_1$	0.423	0.386	0.372	0.389	0.367	0.398	0.388	0.387
$CNGL_2$	*0.547*	0.469	0.266	0.297	0.294	0.407	0.383	0.375
$CoMeT_1$	***0.675***	0.445	***0.598***	0.299	0.252	0.637	0.372	0.454
$EHU\text{-}ALM_1$	*0.566*	0.416_3	0.525_3	0.446	*0.437*	0.538	0.421	0.471
ETS_1	*0.552*	*0.547*	*0.535*	***0.487***	*0.447*	0.543	0.517	0.514
ETS_2	***0.705***	***0.614***	***0.625***	0.356	*0.434*	0.665	0.485	0.547
$LIMSIILES_1$	0.505	0.424	0.419	***0.456***	*0.422*	0.462	0.440	0.445
$SoftCardinality_1$	*0.558*	0.450	*0.537*	***0.492***	***0.471***	0.548	0.471	0.502
$UKP\text{-}BIU_1$	0.448	0.269	***0.590***	0.397_2	0.407	0.519	0.323	0.418
Median	0.552	0.445	0.535	0.397	0.422	0.539	0.422	0.454
Baselines:								
Lexical	0.483	0.463	0.435	0.402	0.396	0.459	0.433	0.436
Majority	0.229	0.248	0.260	0.239	0.249	0.245	0.243	0.245

top performer in all cases. For BEETLE UQ, ETS_2 was the top performer on all subsets; ETS_1 and $SoftCardinality_1$ were consistently the top performers on SCIENTSBANK UQ, and $SoftCardinality_1$ on SCIENTSBANK UD. In general, the rankings for other systems were consistent across the tasks as well.

Thus, for the rest of this paper we will concentrate on the systems' performance on the SRA 5-way labeling task only.

4.4 Comparing Performance Between Scenarios

Recall, from Section 3.4, that the UA test sets represent questions for which there were student responses included in the training data, while the UQ

Table 7 Three-way task weighted-average F_1

Dataset:	BEETLE		SCIENTSBANK			Mean		
Run	UA	UQ	UA	UQ	UD	UA	UQ	All
$CELI_1$	0.519	0.463	0.500	0.555	0.534	0.510	0.509	0.514
$CNGL_2$	0.592	0.471	0.383	0.367	0.360	0.488	0.419	0.435
$CoMeT_1$	***0.728***	0.488	***0.707***	0.522	0.550	0.718	0.505	0.599
ETS_1	0.619	0.542	*0.603*	***0.631***	*0.600*	0.611	0.586	0.599
ETS_2	***0.723***	***0.597***	***0.709***	0.537	0.505	0.716	0.567	0.614
$LIMSIILES_1$	0.587	0.454	0.532	0.553	0.564	0.560	0.504	0.538
$SoftCardinality_1$	0.616	0.451	*0.647*	***0.634***	***0.620***	0.631	0.543	0.594
$UKP\text{-}BIU_1$	0.472	0.313	*0.670*	*0.573*	0.577_2	0.571	0.443	0.521
Median	0.604	0.467	0.625	0.554	0.557	0.591	0.507	0.566
Baselines:								
Lexical	0.578	0.500	0.523	0.520	0.554	0.551	0.510	0.535
Majority	0.229	0.248	0.260	0.239	0.249	0.245	0.244	0.245

Table 8 Two-way task weighted-average F_1

Dataset:	BEETLE		SCIENTSBANK			Mean		
Run	UA	UQ	UA	UQ	UD	UA	UQ	All
$CELI_1$	0.638	0.659	0.594	0.629	0.617	0.616	0.644	0.627
$CNGL_2$	0.806	0.675	0.609_1	0.575	0.572	0.688	0.625	0.647
$CoMeT_1$	***0.839***	***0.702***	***0.773***	0.597	0.677	0.806	0.650	0.718
CU_1	0.786	***0.704***	0.623	0.658	*0.686*	0.705	0.681	0.691
ETS_1	*0.810*	***0.732***	*0.714*	*0.703*	*0.694*	0.762	0.718	0.731
ETS_2	***0.840***	***0.715***	***0.770***	0.622	0.574	0.805	0.668	0.704
$LIMSIILES_1$	0.732	0.656	0.602	0.652	0.662	0.667	0.654	0.661
$SoftCardinality_1$	0.782	0.652	*0.722*	***0.745***	***0.712***	0.752	0.699	0.723
$UKP\text{-}BIU_1$	0.642	0.524	*0.734*	0.678	0.677_2	0.688	0.601	0.651
Median	0.786	0.675	0.714	0.652	0.677	0.705	0.654	0.691
Baselines:								
Lexical	0.797	**0.735**	0.635	0.653	0.665	0.716	0.694	0.697
Majority	0.449	0.426	0.412	0.437	0.426	0.431	0.432	0.430

and UD test sets represent transfer performance to new questions and new domains, respectively.

As can be seen from Table 6, the average performance on the UA task was consistently better than on UQ and UD tasks. The two UA test sets had more systems that performed statistically better than the lexical baseline, six systems, than did the UQ test sets, where only two or three systems performed statistically better than the lexical baseline. Twice as many systems outperformed the lexical baseline on UD as on the UQ test sets.

The top performers on the UA test set were $CoMeT_1$ and ETS_2. Their performance was not significantly different on either BEETLE or SCIENTSBANK UA test sets. However, there was not a single best performer on the transfer tasks. ETS_1 performed best on UQ on average, while $SoftCardinality_1$ performed statistically better than all other systems on SCIENTSBANK UD. The best performers on the two UQ test sets differed as well, with ETS_2 performing statistically *better* than all other systems on BEETLE UQ, but performing statistically *worse* than the lexical baseline on SCIENTSBANK UQ, resulting in no overlap in the top performing systems on the BEETLE and SCIENTSBANK

UQ test sets. SoftCardinality$_1$ was the top performer on SCIENTSBANK UQ as well as UD, but was not among the best performers on the other three test sets.

The difference in performance on the different scenarios is not entirely surprising, as the transfer scenarios would be expected to be more difficult and require different analysis to perform well. As mentioned in Section 4.1, several systems attempted to use external semantic resources or scenario-specific features, which should mitigate the coverage issues to a certain extent. We analyze the effects of such features in the next section.

4.5 Comparing Impact of System Features

As discussed in Section 4.1, the task participants used a wide range of methods and features in the task. We analyzed the data to see if any patterns emerged with respect to using different features in different scenarios. We are interested in which type of features were the most helpful in achieving the best performance: syntactic dependency features, inclusion of external semantic resources, or scenario-specific features.

To answer this question, we split the participant systems into groups depending on whether a given feature type is used, and compare performance between groups.

We made two simplifications to see patterns in the data more easily. First, while we want to differentiate between the system runs that use different features, four teams with multiple runs (CELI, CNGL, EHU-ALM, and ETS) submitted systems using the same type of features, but a different learning algorithm or calculation approach. For those teams, we only used their top performing run in the analysis (CELI$_1$, CNGL$_2$, EHU-ALM$_2$, and ETS$_1$), as was the case in Section 4.3.

Second, we focus on the mean Unseen Answers (UA) and Unseen Questions(UQ) figures as defined in Section 4.3, and the Unseen Domains (UD) performance on SCIENTSBANK.[6] We leave the understanding of the difference in performance between individual UA and UQ test sets for future work.

This analysis is not definitive because the number of systems in the competition is relatively small, and we cannot account for feature interactions. However, it provides a pattern for future data analysis and research.

4.5.1 Syntactic Dependency Features

Six out of 11 runs included in our analysis used syntactic features, which are defined as features based on parsing student responses with a dependency parser. Five runs did not use syntactic dependencies.[7] The performance of these two system groups is summarized in Table 9.

[6] Recall that there was no Unseen Domains test set in BEETLE data.

[7] For purposes of this analysis, we focused on the use of dependency features, and did not take into account the use of part of speech tags.

Table 9 Performance comparison for system groups with and without syntactic dependency features.

Test set		N	Best System	Best score	Median
UA	With synt. feat.	6	$CoMeT_1$	0.637	0.448
	Without synt. feat.	5	ETS_2	0.665	0.544
UQ	With synt. feat.	6	$LIMSIILES_1$	0.440	0.378
	Without synt. feat.	5	ETS_1	0.517	0.471
UD	With synt. feat.	6	$EHU\text{-}ALM_2$	0.437	0.374
	Without synt. feat.	5	$SoftCardinality_1$	0.471	0.434

There was not a single best system in either group: different systems performed best for UA, UQ and UD test sets. However, in all cases the top systems in the "without syntactic dependencies" group performed better than the top systems in the "with syntactic dependencies" group, and the median group performance was higher as well.

The three top systems in the "without syntax" group were ETS_1, ETS_2 and $SoftCardinality_1$. Only ETS_1 used external semantic resources, but both ETS systems used scenario-specific features.

This indicates that even though theoretically there are cases where syntactic structure can make a difference in student response analysis, the participating systems were not able to leverage such features effectively. We discuss this in more detail in Section 5.

4.5.2 External Semantic Resources

Recall from Section 4.1 that we are interested in the effectiveness of using semantic features based on external semantic resources, including hand-coded dictionaries such as WordNet and word classes or other features based on word cooccurrence in corpora external to the task. Eight of the runs we analyzed were from system variants using features based on external semantic resources, while three were from system variants not utilizing semantic resources. Group performance for these groups is summarized in Table 10.

Again, there was not a single best performer in either of the groups. For the "with external semantic resources" group, $CoMeT_1$ performed best on the UA set, while ETS_1 was the best performer for UQ and UD sets. In the group "without semantic resources", ETS_2 performed best on the UA and UQ sets, while $SoftCardinality_1$ was best for UD.

However, similar to the syntactic dependencies case, there was no obvious advantage for the systems that used external semantic resources. Median group performance was higher in the "without semantics" group compared to the "with semantics" group. The best systems in the "without semantics" group also outperformed the best systems in the "with semantics" group on two of the three test sets (UA and UD). Thus, the systems participating in the challenge have not yet found an effective way to leverage the external computational semantics resources for this task. We discuss this in more detail in Section 5.

Table 10 Performance comparison for system groups with and without external semantic resource features.

Test set		N	Best System	Best score	Median
UA	With sem. feat.	8	$CoMeT_1$	0.637	0.491
	Without sem. feat.	3	ETS_2	0.665	0.548
UQ	With sem. feat.	8	ETS_1	0.517	0.385
	Without sem. feat.	3	ETS_2	0.485	0.471
UD	With sem. feat.	8	ETS_1	0.447	0.389
	Without sem. feat	3	$SoftCardinality_1$	0.471	0.441

Table 11 Performance comparison for system groups with and without scenario specific features.

Test set		N	Best System	Best score	Median
UA	With scen. feat.	4	ETS_2	0.665	0.590
	Without scen. feat.	7	$SoftCardinality_1$	0.548	0.434
UQ	With scen. feat.	4	ETS_1	0.517	0.454
	Without scen. feat.	7	$SoftCardinality_1$	0.471	0.383
UD	With scen. feat.	4	ETS_1	0.447	0.436
	Without scen. feat.	7	$SoftCardinality_1$	0.471	0.376

4.5.3 Scenario Specific Features

Four of the runs in our analysis set come from systems attempting to use different features or parameters depending on the scenario (test set). Seven runs are from configurations where no scenario-specific configuration was used. The performance of these two system groups is summarized in Table 11.

The ETS_1 and ETS_2 systems were the best performers in the "scenario specific" group, while $SoftCardinality_1$ was the best performer in the "not scenario-specific" group. For the UQ and UA test sets, ETS_2 and ETS_1 outperformed $SoftCardinality_1$. Whereas, for the UD test set, $SoftCardinality_1$ performed best.

However, the median group performance was better for systems performing the adaptation on all test sets, unlike the case for semantic features, where median performance was better for the "without external resources" group. Overall, this indicates that using scenario-specific features can be helpful in improving system performance. We discuss these results in more detail in the next section.

5 Discussion and Future Work

We originally proposed student response analysis as a shared task to the computational linguistics community as a way to test out semantic inference algorithms in an interesting and relevant application setting. The systems entered in the challenge brought in a wide variety of computational linguistic approaches, including methods adapted from previous work on textual entailment, semantic textual similarity, and educational NLP. There was wide variability in terms of the features used. Most systems used easy to compute

features based on word overlap, and many used additional resources, such as dependency parsers and semantic similarity features derived either from WordNet or from additional textual corpora.

For simplicity, we used a single evaluation metric, weighted-average F_1, in our analysis. We originally proposed three possible metrics (see Section 3.5) with different theoretical properties. In principle, the metrics could differ substantially because of the treatment of minority classes, but in practice the relative rankings they assigned to the systems were reasonably consistent. We chose to use weighted-average F_1 in our analysis, due to the nature of the data imbalance as described in Section 4.2. Ideally, the evaluation metric performance should be correlated with practical application indicators. For example, where the system is used to support learning rather than just assessment, the ultimate best metric is the one that results in feedback most beneficial for learning. Dzikovska et al (2012a, 2014) carried out an initial investigation into the relationship between BEETLE II system performance and learning. More such studies in multiple domains are necessary to settle the question of which metric is best to use in different applications.

While all of the systems consistently outperformed the most frequent class baseline, beating the word overlap baseline proved to be more challenging. It was achieved by just over half of the results with about half of those being statistically significant improvements. This underscores the fact that there is still a considerable opportunity to improve student response analysis systems, and in determining which features and resources are useful.

The systems were evaluated on three types of scenarios: a fully in-domain analysis task, with training data specific to the questions ("Unseen Answers"); transfer to new questions in the same subject area ("Unseen Questions"), which would be expected to share the domain vocabulary and concepts; and transfer to new questions outside the domain, with previously unseen vocabulary and domain structure ("Unseen Domains"). As discussed in Section 2.1, these test sets reflect different deployment scenarios, because the difficulty of acquiring question-specific data varies depending on the intended application. In addition, the issues of domain dependence are important to the computational linguistics community as a whole, as many NLP tools such as parsers depend heavily on in-domain supervised training data, and are difficult and expensive to re-train for new domains and applications.

In theory, the use of syntactic dependencies should benefit systems in all scenarios, by allowing them to better handle long-distance dependencies that may be important for precise answer analysis. The use of external semantic resources should be beneficial in all cases, but particularly helpful for transfer scenarios where it is not possible to train lexicalized features adequately. The use of scenario-specific features should result in systems that perform better on a given scenario than the systems not adapted to it. In our data set, however, two of those three hypotheses were not supported.

In our analysis, using syntactic dependency features or external semantic resources did not provide obvious benefits regardless of the scenario. Given the small number of data points, this cannot be taken to mean that such

features are not useful for this task in general. However, additional work is clearly required to successfully incorporate them in the student response analysis task. One issue is that none of the parsers used in the task were adapted to the domain. A pilot study in the BEETLE II domain several years ago indicated that dependency parsers trained on newspaper text were unable to provide reasonable parses for a large number of student utterances, primarily because they did not do a good job of handling the fragments and incorrect punctuation common in (spoken or written) dialogue (McConville and Dzikovska 2008). In the future, it would be helpful to investigate how well the parsers used in this task actually performed on the task data, and whether improved performance would make syntactic features more beneficial.

Similarly, semantic resources used by several other systems (WordNet, Europarl) are likely to have coverage issues in all scientific domains. Wikipedia is another source of semantic information that is likely to have good coverage for STEM topics. However, the systems using it in the task (CELI, LIMSIILES) performed below median in our tests. Thus, clearly more work is needed before out-of-domain semantic resources can be most effectively used for this task. Additional research is also required to determine whether the systems would perform better if in-domain semantic resources were available.

For the Unseen Answers and Unseen Question test sets, systems that used scenario-specific features outperformed systems that did not when looking both at top performers and at median group performance. Thus, we can tentatively conclude that scenario-specific features are beneficial. However, on the Unseen Domains test set, the best performing system was SoftCardinality, which is a system that does not use any syntactic or semantic features, but relies instead on a novel character and word overlap metric which is considerably more sophisticated that the word overlap features used in other systems.

As noted above, external semantic resources should be particularly useful, in theory, for transfer tasks such as "Unseen Domains" by providing missing semantic information that is otherwise (implicitly or explicitly) learned during training. The fact that a word-overlap system significantly outperformed all other systems on the Unseen Domains scenario underscores the limitations of the state of the art computational semantics resources. In the future, it could be particularly interesting to explore whether scenario-specific features would be more effective for the "Unseen Domains" task if combined with domain-specific semantic resources. While training data in the form of student answers can be difficult to obtain for new domains, other resources, such as course textbooks, are often more readily available and this could allow for building semantic classes or distributional semantics spaces useful in such applications.

It is also important to consider how consistent the system groups are based on the different feature types used. Our current conclusions are limited by the relatively small number of systems in the analysis. As a result, some of the systems in each category might not be ideally comparable. For example, in the syntactic dependency category, most systems used features

based on supervised dependency parsers, but one system (CNGL) used features based on the output of an unsupervised dependency parser. Most importantly, the systems in the scenario-specific features group show the most differences. These systems attempt to learn the appropriate features to use in very different ways. The ETS systems start with a base set of features, and add domain-specific and domain-independent copies of each, learning different weights based on the feature's relevance across domains or strictly within a single domain. The CoMeT system uses question and domain IDs as features, attempting to learn which features generalize across the three different scenarios. Finally, the EHU-ALM system trains three different classifiers, using identical feature sets but different training data sets, and then applies a different classifier based on the scenario.

Our analysis currently does not take such differences into account and simply compares system groups with and without a specified features. Our current analysis provides an example and a starting point for data exploration, and for evaluation data analysis if another similar shared challenge is organized in the future. Our tentative conclusions about the utility of different feature types need to be confirmed in further research, and a basis for the SRA systems to classify the features they use, and plan out feature ablation experiments. Many of the systems submitted to the challenges carried out such experiments in different forms, and a similar but finer-grained feature classification was proposed in Burrows et al (2015b). It could be used for similar analyses in the future, when more systems are available to compare in the more specific feature types.

6 Conclusions

We described the data set, participant systems and evaluation results from the Joint Student Response Analysis and 8th Recognizing Textual Entailment challenge. This has proven to be a useful, interdisciplinary task using a realistic dataset from the educational domain. In almost all cases the best systems significantly outperformed the word overlap baseline, sometimes by a large margin, showing that computational linguistics approaches can contribute to educational tasks. However, there is still significant room for improvement in the absolute scores, reflecting the interesting challenges that educational data present to computational linguistics.

The resulting data set and code are available at `http://www.cs.york.ac.uk/semeval-2013/task7/` and can be used for computational linguistics purposes, advancing further research into computational semantics and educational NLP.

Acknowledgements We would like to thank Chris Brew for the discussion and suggestions related to the paper organization. We thank the three anonymous reviewers for their helpful comments.

References

Agirre E, Cer D, Diab M, Gonzalez-Agirre A (2012) Semeval-2012 task 6: A pilot on semantic textual similarity. In: *SEM 2012: The First Joint Conference on Lexical and Computational Semantics, Association for Computational Linguistics, Montréal, Canada, pp 385–393

Agirre E, Cer D, Diab M, Gonzalez-Agirre A, Guo W (2013) *sem 2013 shared task: Semantic textual similarity. In: Second Joint Conference on Lexical and Computational Semantics (*SEM), Association for Computational Linguistics, Atlanta, Georgia, USA, pp 32–43

Aldabe I, Maritxalar M, Lopez de Lacalle O (2013) EHU-ALM: Similarity-feature based approach for student response analysis. In: Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), Association for Computational Linguistics, Atlanta, Georgia, USA, pp 580–584

Bentivogli L, Dagan I, Dang HT, Giampiccolo D, Magnini B (2009) The fifth PASCAL recognizing textual entailment challenge. In: Proceedings of Text Analysis Conference (TAC) 2009

Bentivogli L, Clark P, Dagan I, Dang HT, Giampiccolo D (2010) Thesixth PASCAL recognizing textual entailment challenge. In: Notebook papers and results, Text Analysis Conference (TAC)

Bentivogli L, Clark P, Dagan I, Dang HT, Giampiccolo D (2011) The seventh PASCAL recognizing textual entailment challenge. In: Notebook papers and results, Text Analysis Conference (TAC)

Bicici E, van Genabith J (2013) CNGL: Grading student answers by acts of translation. In: Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), Association for Computational Linguistics, Atlanta, Georgia, USA, pp 585–591

Burrows S, Gurevych I, Stein B (2015a) The eras and trends of automatic short answer grading. International Journal of Artificial Intelligence in Education 25(1):60–117, DOI 10.1007/s40593-014-0026-8, URL `http://dx.doi.org/10.1007/s40593-014-0026-8`

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

Burstein J, Tetreault J, Madnani N (2013) The e-rater essay scoring system. In: Shermis MD, Burstein J (eds) Handbook of automated essay evaluation: Current applications and new directions, Taylor and Francis

Campbell GC, Steinhauser NB, Dzikovska MO, Moore JD, Callaway CB, Farrow E (2009) The DeMAND coding scheme: A "common language" for representing and analyzing student discourse. In: Proceedings of 14th International Conference on Artificial Intelligence in Education (AIED), poster session, Brighton, UK

Chang CC, Lin CJ (2011) LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2:27:1–27:27

Dagan I, Glickman O, Magnini B (2006) The pascal recognising textual entailment challenge. In: Quiñonero-Candela J, Dagan I, Magnini B, d'Alché Buc F (eds) Machine Learning Challenges, Lecture Notes in Computer Science, vol 3944, Springer

Dale R, Kilgarriff A (2011) Helping our own: The HOO 2011 pilot shared task. In: Proceedings of the Generation Challenges Session at the 13th European Workshop on Natural Language Generation, Association for Computational Linguistics, pp 242–249

Dale R, Anisimoff I, Narroway G (2012) HOO 2012: A report on the preposition and determiner error correction shared task. In: Proceedings of the Seventh Workshop of Building Educational Applications Using NLP, Association for Computational Linguistics

Daume III H (2007) Frustratingly easy domain adaptation. In: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, Association for Computational Linguistics, Prague, Czech Republic, pp 256–263

Dzikovska M, Steinhauser N, Farrow E, Moore J, Campbell G (2014) 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, DOI 10.1007/s40593-014-0017-9

Dzikovska MO, Moore JD, Steinhauser N, Campbell G, Farrow E, Callaway CB (2010) Beetle II: a system for tutoring and computational linguistics experimentation. In: Proc. of ACL 2010

System Demonstrations, pp 13–18
Dzikovska MO, Bell P, Isard A, Moore JD (2012a) Evaluating language understanding accuracy with respect to objective outcomes in a dialogue system. In: Proc. of EACL-12 Conference, pp 471–481
Dzikovska MO, Nielsen RD, Brew C (2012b) Towards effective tutorial feedback for explanation questions: A dataset and baselines. In: Proc. of 2012 Conference of NAACL: Human Language Technologies, pp 200–210
Dzikovska MO, Farrow E, Moore JD (2013a) Combining semantic interpretation and statistical classification for improved explanation processing in a tutorial dialogue system. In: Proceedings of the 16th International Conference on Artificial Intelligence in Education (AIED 2013), Memphis, TN, USA
Dzikovska MO, Nielsen R, Brew C, Leacock C, Giampiccolo D, Bentivogli L, Clark P, Dagan I, Dang HT (2013b) Semeval-2013 task 7: The Joint Student Response Analysis and 8th Recognizing Textual Entailment Challenge. In: Proceedings of the 6th International Workshop on Semantic Evaluation (SEMEVAL-2013), Association for Computational Linguistics, Atlanta, Georgia, USA
Giampiccolo D, Dang HT, Magnini B, Dagan I, Cabrio E, Dolan B (2008) The fourth PASCAL recognizing textual entailment challenge. In: Proceedings of Text Analysis Conference (TAC) 2008, Gaithersburg, MD
Glass M (2000) Processing language input in the CIRCSIM-Tutor intelligent tutoring system. In: Papers from the 2000 AAAI Fall Symposium, Available as AAAI technical report FS-00-01, pp 74–79
Gleize M, Grau B (2013) LIMSIILES: Basic english substitution for student answer assessment at semeval 2013. In: Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), Association for Computational Linguistics, Atlanta, Georgia, USA, pp 598–602
Graesser AC, Wiemer-Hastings K, Wiemer-Hastings P, Kreuz R (1999) Autotutor: A simulation of a human tutor. Cognitive Systems Research 1:35–51
Heilman M, Madnani N (2012) Ets: Discriminative edit models for paraphrase scoring. In: *SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012), Association for Computational Linguistics, Montréal, Canada, pp 529–535
Heilman M, Madnani N (2013a) ETS: Domain adaptation and stacking for short answer scoring. In: Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), Association for Computational Linguistics, Atlanta, Georgia, USA, pp 275–279
Heilman M, Madnani N (2013b) HENRY-CORE: Domain adaptation and stacking for text similarity. In: Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity, Association for Computational Linguistics, Atlanta, Georgia, USA, pp 96–102
Jimenez S, Becerra C, Gelbukh A (2013) SOFTCARDINALITY: Hierarchical text overlap for student response analysis. In: Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), Association for Computational Linguistics, Atlanta, Georgia, USA, pp 280–284
Jordan P, Makatchev M, Pappuswamy U, VanLehn K, Albacete P (2006a) A natural language tutorial dialogue system for physics. In: Proceedings of the 19th International FLAIRS conference, pp 521–527
Jordan PW, Makatchev M, Pappuswamy U (2006b) Understanding complex natural language explanations in tutorial applications. In: Proceedings of the Third Workshop on Scalable Natural Language Understanding, ScaNaLU '06, pp 17–24
Kouylekov M, Dini L, Bosca A, Trevisan M (2013) Celi: EDITS and generic text pair classification. In: Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), Association for Computational Linguistics, Atlanta, Georgia, USA, pp 592–597
Leacock C, Chodorow M (2003) C-rater: Automated scoring of short-answer questions. Computers and the Humanities 37(4):389–405

Leacock C, Chodorow M, Gamon M, Tetreault JR (2014) Automated Grammatical Error Detection for Language Learners, Second Edition. Synthesis Lectures on Human Language Technologies, Morgan & Claypool Publishers

Levy O, Zesch T, Dagan I, Gurevych I (2013) UKP-BIU: Similarity and entailment metrics for student response analysis. In: Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), Association for Computational Linguistics, Atlanta, Georgia, USA, pp 285–289

MacDonald NH, Grase LT, Gingrich PS, Keenan SA (1982) The writer's workbench: Computer aids for text analysis. IEEE Transactions on Communications 30

McConville M, Dzikovska MO (2008) Deep grammatical relations for semantic interpretation. In: Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation, pp 51–58

Mohler M, Bunescu R, Mihalcea R (2011) Learning to grade short answer questions using semantic similarity measures and dependency graph alignments. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Portland, Oregon, USA, pp 752–762, URL `http://www.aclweb.org/anthology/P11-1076`

Ng HT, Wu SM, Wu Y, Tetreault J (2013) The CoNLL-2013 shared task on grammatical error correction. In: Proceedings of the 17th Conference on Computational Natural Language Learning, Association for Computational Linguistics

Ng HT, Wu SM, Briscoe T, Hadiwinoto C, Susanto RH, Bryant C (2014) The CoNLL-201r shared task on grammatical error correction. In: Proceedings of the 18th Conference on Computational Natural Language Learning: Shared Task, Association for Computational Linguistics, pp 1–14

Nielsen RD, Ward W, Martin JH (2008a) Learning to assess low-level conceptual understanding. In: Proc. of 21st Intl. FLAIRS Conference, pp 427–432

Nielsen RD, Ward W, Martin JH, Palmer M (2008b) Annotating students' understanding of science concepts. In: Proceedings of the Sixth International Language Resources and Evaluation Conference, (LREC08), Marrakech, Morocco

Nielsen RD, Ward W, Martin JH (2009) Recognizing entailment in intelligent tutoring systems. The Journal of Natural Language Engineering 15:479–501

Okoye I, Bethard S, Sumner T (2013) CU: Computational assessment of short free text answers - a tool for evaluating students' understanding. In: Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), Association for Computational Linguistics, Atlanta, Georgia, USA, pp 603–607

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

Page E (1996) The imminence of grading essays by computer. Phi Delta Kappan pp 238–243

Pon-Barry H, Clark B, Schultz K, Bratt EO, Peters S (2004) Advantages of spoken language interaction in dialogue-based intelligent tutoring systems. In: Proc. of ITS-2004 Conference, pp 390–400

Pulman SG, Sukkarieh JZ (2005) Automatic short answer marking. In: Proceedings of the Second Workshop on Building Educational Applications Using NLP, Association for Computational Linguistics, Ann Arbor, Michigan, pp 9–16

Shermis MD, Burstein J (eds) (2013) Handbook on Automated Essay Evaluation: Current Applications and New Directions. Routledge

Tetreault J, Blanchard D, Cahill A (2013) 12:10 a report on the first native language identification shared task. In: Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications, Association for Computational Linguistics, pp 48–57

VanLehn K, Jordan P, Litman D (2007) Developing pedagogically effective tutorial dialogue tactics: Experiments and a testbed. In: Proc. of SLaTE Workshop on Speech and Language Technology in Education, Farmington, PA

Wolska M, Kruijff-Korbayová I (2004) Analysis of mixed natural and symbolic language input in mathematical dialogs. In: ACL-2004, Barcelona, Spain

Yeh A (2000) More accurate tests for the statistical significance of result differences. In: Proceedings of the 18th International Conference on Computational linguistics (COLING 2000), Association for Computational Linguistics, Stroudsburg, PA, USA, pp 947–953