

University of Memphis

University of Memphis Digital Commons

Electronic Theses and Dissertations

12-2-2014

DeepEval: An Integrated Framework for the Evaluation of Student Responses in Dialogue Based Intelligent Tutoring Systems

Rajendra Banjade

Follow this and additional works at: <https://digitalcommons.memphis.edu/etd>

Recommended Citation

Banjade, Rajendra, "DeepEval: An Integrated Framework for the Evaluation of Student Responses in Dialogue Based Intelligent Tutoring Systems" (2014). *Electronic Theses and Dissertations*. 1069.
<https://digitalcommons.memphis.edu/etd/1069>

This Thesis is brought to you for free and open access by University of Memphis Digital Commons. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of University of Memphis Digital Commons. For more information, please contact khggerty@memphis.edu.

DEEPEVAL: AN INTEGRATED FRAMEWORK FOR THE EVALUATION OF
STUDENT RESPONSES IN DIALOGUE BASED INTELLIGENT TUTORING
SYSTEMS

by

Rajendra Banjade

A Thesis

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

Major: Computer Science

The University of Memphis

December, 2014

Copyright © 2014 Rajendra Banjade

All rights reserved

Acknowledgments

I am very grateful to my academic advisor and thesis committee chair Dr. Vasile Rus for his invaluable guidance, pleasant discussions, and suggestions. I am equally thankful to committee members, Dr. Dipankar Dasgupta and Dr. Scott D. Fleming for their valuable time, support, and suggestions. Indeed, it was a great learning opportunity for me.

I would like to thank my colleagues Nobal Niraula, Dipesh Gautam, Nabin Maharjan, Dr. Dan Stefanescu, Dr. Vivek V. Datla, and Dr. Mihai Lintean for their help and encouragement. Also, I appreciate the direct or indirect help of people in Computer Science department and Institute for Intelligent Systems.

Finally, I owe my gratitude to my family for all their love and encouragement.

Abstract

Banjade, Rajendra. MS. The University of Memphis. December 2014.
DEEPEVAL: An Integrated Framework For The Evaluation Of Student Responses In
Dialogue Based Intelligent Tutoring Systems. Major Professor: Vasile Rus, Ph.D.

The automatic assessment of student answers is one of the critical components of an Intelligent Tutoring System (ITS) because accurate assessment of student input is needed in order to provide effective feedback that leads to learning. But this is a very challenging task because it requires natural language understanding capabilities. The process requires various components, concepts identification, co-reference resolution, ellipsis handling etc. As part of this thesis, we thoroughly analyzed a set of student responses obtained from an experiment with the intelligent tutoring system DeepTutor in which college students interacted with the tutor to solve conceptual physics problems, designed an automatic answer assessment framework (DeepEval), and evaluated the framework after implementing several important components. To evaluate our system, we annotated 618 responses from 41 students for correctness. Our system performs better as compared to the typical similarity calculation method. We also discuss various issues in automatic answer evaluation.

Contents

List of Tables	vii
1 Introduction.....	1
1.1 Intelligent Tutoring Systems	5
1.2 DeepTutor.....	7
1.3 Natural Language Understanding.....	8
1.3.1 Paraphrase Identification, Textual Entailment, and Semantic Similarity	9
1.3.2 SEMILAR Toolkit	10
1.4 Student Response Analysis in ITS	11
1.4.1 Scoring Policy	13
1.5 Major Components of DeepEval System	14
1.6 Motivation	15
1.7 Goal	17
1.8 Contribution	17
1.9 Roadmap.....	18
2 Related works.....	19
2.1 Automatic Essay Grading.....	19
2.2 Assessment of Non-Cognitive Factors.....	20
2.3 Situated Assessments	21
2.4 Short Answer Assessment.....	22
2.5 Competitions and Shared Tasks	26
2.6 Answer Grading in Various Languages	28
2.7 Domain Specific Solutions.....	28
2.8 Assignments Evaluation in MOOC Systems	30
2.9 Common Corpora for Benchmarking.....	31
2.10 Summary.....	32
3 DeepEval Framework	33
3.1 Design Principles.....	33
3.2 A closer Look at Student Responses	34
3.3 DeepEval Components	37

3.4	Models Development	44
3.4.1	Grading Policy Model	44
3.5	Other Challenges	45
4	DataSet	48
4.1	Data Collection Process	48
4.2	Annotation	50
4.3	Statistics	53
4.4	Summary	54
5	Experiments and Results	55
5.1	Experiments	55
5.1.1	Experiment Design	55
5.1.2	Preprocessing	56
5.1.3	Speech-Act Classification	59
5.1.4	Negation Handling	59
5.1.5	Concept Extraction	60
5.1.6	Implicit Coreference Resolution and Ellipsis Handling	62
5.2	Scoring Model Development	64
5.2.1	Features	64
5.2.2	Learning Similarity Threshold Values	70
5.3	Evaluation Metrics	71
5.4	Results	71
6	Discussions	74
7	Conclusions and Future Work	79
	References	80

List of Tables

1. Some of the student responses to the question given in Example 3.1 (the spelling error in (1) is left intentionally).	35
2. Confusion matrix for inter annotator agreement (100 instances)	52
3. Summary statistics of DeepEval dataset	53
4. Improvement in spelling correction after considering the context	57
5. Correlation (measured as Pearson correlation) between similarity scores given by different methods and the human annotated scores in Simlex-999 corpus	67
6. Results (Pearson correlation and Root Mean Square Errors) after combining scores obtained from different methods. The effective features are enclosed in bracket.	68
7. Scoring results using DeS word metric (The results obtained by 10-fold cross validation (first) and 70/30 split are separated by /).	72
8. Scoring results obtained by using linear combination of word metrics (The results obtained by 10-fold cross validation (first) and 70/30 split are separated by /).....	73

1 Introduction

Automatic student answer evaluation has many benefits such as time saving for teacher graders, more effective intelligent tutoring systems, better diagnostic feedback, consistency, and working at scale. Research in this area has the potential to evolve into an automated scoring application that would be appropriate for evaluating short-answer constructed responses in online instruction and assessment applications in virtually all disciplines. However, the automation also poses operational and technical challenges. Much of the work has been conducted in the field of automatic assessment of students' knowledge based on objective questions such as multiple choice or fill-in-the-blank questions. The grading of such questions is convenient. The real challenge is designing the choices which should be close enough to the right answer but still wrong. A drawback of multiple-choice questions is that sometimes students pick the correct answer for the wrong reasons. To fully assess students' knowledge level, we must prompt the students to explain their reasoning. Indeed, in order to fully assess the students' actual progress these should be complemented with open-ended questions (Whittington & Hunt, 1999). Students who cannot explain the logical flaw in a persuasive message may find it easy to identify the flaw when it is presented as one of four or five possibilities. With the advancement in Natural Language Processing (NLP; briefly described in Section 2.3), research in educational field is moving towards automatic evaluation of constructed or free-text responses. These techniques can be applied to assess students' responses in a dialogue based Intelligent Tutoring Systems (ITS) or in other offline settings in the context of homework or exam questions similar to what teachers typically do in traditional schools. In conversational ITSs, students are allowed to express their answers

in natural form using natural language text, and the computer tutor has to evaluate these answers (typically by comparing them with expert answers) and provide appropriate feedback. This thesis presents the design of a unified framework to systematically evaluate students' natural language responses in dialogue based intelligent tutoring systems (ITS) and other contexts, e.g. large-scale evaluations, where automated assessment is desirable.

The manual answer evaluation process is very expensive in time and money and suffers to some degree by inconsistencies introduced by the many human graders who have their own personal biases. A potential solution to this is the development of pedagogically adequate, psychometrically sound, and socially acceptable machine assessment and tutorial feedback mechanism. Methods for such automatic scoring of test essays are already in use (Burstein, 2003; Rich, C. S., Harrington, H., Kim, J., & West, B., 2008). In United States, an increasing number of states have adopted Automatic Essay Scoring (AES) programs in school- and classroom-based writing assessment as well as in state summative writing assessment (Rich et al., 2008). Such methods have numerous benefits compared to the time consuming manual and repetitive process which involves thousands of different human graders. For example, the methods are consistent (fair), faster, cost effective, scalable, flexible, and so on. As a byproduct, it facilitates data collection and analysis. For instance, comparing data across the nation, states or region becomes easier.

Moreover, the computer aided grading is also used to assess short answers (one or couple of lines). Some large scale experiments have proved the feasibility of such systems (Graesser, Lu, Jackson, Mitchell, Ventura, Olney, & Louwerse, 2004; Leacock,

2004; Leacock, & Chodorow, 2003; Rus, D'Mello, & Graesser, 2013). The assessment of student responses in conversational ITS is the central topic of this thesis. The proposed solution can be ported to different contexts. In recent years, the widespread use of Massive Open Online Courses (MOOC) posing a challenge on evaluating and providing feedback to thousands of users (Kulkarni, Wei, Le, Chia, Papadopoulos, Cheng, Koller, & Klemmer, 2014; Perelman, Gulwani, & Grossman, 2014; Shah, Bradley, Balakrishnan, Parekh, Ramchandran, & Wainwright, 2014; Shah, Bradley, Parekh, Wainwright, & Ramchandran, 2013) where automatic evaluation techniques of the kind we propose here can potentially fulfill the need. Naturally, a question comes in mind: what is stopping us using such next-generation technologies? The answer is the limitation of current natural language processing systems to handle wide varieties of texts (including noisy data). The NLP is the backbone of automatic answer assessment technologies. However, the encouraging progress in NLP and research on educational technologies is reducing the gap significantly. This thesis work is also a result of a careful conceptual analysis as well as practical experience working on a successful intelligent tutoring system that is based on natural language conversational interface (in the form of text).

The research in education field is moving towards building artificial agents, such as intelligent tutoring systems. The assessment of student answers is one of the critical components of a conversational ITS because accurate assessment of student input is needed in order to provide effective feedback that leads to learning. The students might get frustrated or discouraged using such intelligent systems by inappropriate feedback and repeated questioning because of incorrect evaluation of their responses to previous

questions. In the recent years, a number of competitions¹, workshops, and shared tasks (Dzikovska et al., 2013) on automatic short answer grading were organized to streamline the research on automatic answer assessment. A study of published works on this problem (Dzikovska et al., 2013; Landauer & Dumais, 1997; Mohler, Bunescu, & Mihalcea, 2011; Mohler & Mihalcea 2009; Pérez, Gliozzo, Strapparava, Alfonseca, Rodríguez, & Magnini, 2005; Rus, & Lintean, 2012) has shown that the idea of semantic similarity and textual entailment has been borrowed directly such that the student answer is compared with the expected answer. However, taking the student response and reference answer as an isolated pair of texts, and measuring their similarity for the purpose of grading does not fully work because their implied assumption is that the texts are self-contained. But the evaluation of student responses requires a lot of additional work, for example preprocessing, co-reference resolution, ellipsis handling etc. Also, it is hard to get diagnostic or explanation based results by applying purely data mining based solutions, which are more commonly adopted similarity or entailment solutions for the purpose of answer assessment.

As part of this thesis work, we identified various components needed to handle different natural language phenomena and designed a framework as a combination of these components for the purpose of student response evaluation. We implemented some of the important components that are capable of handling different natural language phenomena and are potentially useful for diagnostic evaluation. We annotated for correctness a set of student answers provided by students during their interactions with an intelligent tutoring system while solving conceptual physics problems. The questions are

¹ For example, competitions on short answer grading were organized by HP in 2012 and 2013 (<https://www.kaggle.com/c/asap-sas>)

not completely open ended such as asking about the favorite book but have a clearly defined target response (we call reference answer or expected answer). However, the learner may convey the meaning of the target in multiple ways. We tested our system with the annotated data. The DeepEval produces better results as compared to the results produced by applying a typical semantic similarity method.

The following sections in this section include an introduction to Intelligent Tutoring Systems, an introduction to DeepEval system, motivation, goal, and contribution of this thesis.

1.1 Intelligent Tutoring Systems

An Intelligent Tutoring System (ITS), such as DeepTutor (Rus et al., 2013), AutoTutor (Graesser et al., 2004), or CircSim (Evens & Michael, 2006), is computer software designed to simulate human tutor. Students can interact with intelligent tutors through some medium and tutors assist students solving problems by implementing appropriate instructional strategies and provide feedback. The most natural way to interact with the tutor is via natural language dialogues – written and/or spoken. The intelligent tutoring systems are different from general computer based tutoring systems in that they can interpret students' input and adapt themselves in real time as per the student's need. For example, they can change instructional strategies when the learner has difficulty in solving the problem.

There are many examples of ITSs being used in both formal education and professional settings in which they have demonstrated their capabilities and limitations. There is a close relationship between intelligent tutoring and other domains, such as

cognitive science, educational sciences. There are ongoing researches on improving the effectiveness of ITS.

For instance, studies revealed that students perform very poorly when faced with qualitative/conceptual Physics problems even though the same students can master the skills required to solve quantitative physics problems (Hake, 1998; Halloun & Hestenes, 1985). That is, students can mechanically manipulate and apply formulas without fully understanding the underlying concepts. Tutoring on conceptual aspects of science topics is therefore much needed. Conceptual reasoning fits well with a conversational form of interaction.

Dialogue based Intelligent tutoring systems are special kind of tutoring systems where the tutor and the student communicate one to one using the natural language interface. Interactive tutoring systems have been designed for a variety of domains and applications. Dialog-based tutoring systems, such as Why2-Atlas (VanLehn et al., 2002), AutoTutor (Graesser et al., 2004), CircSim (Evens & Michael, 2006), and DeepTutor (Rus et al., 2013) interact with students via questions and answers. These conversational ITSs are based on explanation-based constructivist theories of learning and the collaborative activities that occur during human tutoring when the tutor and the student collaboratively construct a solution. They have proven to promote student learning gains up to an impressive effect of 1.64 sigma when compared to students learning the same content in a canned text remediation condition that focuses on the desired content (VanLehn et al. 2007). However, the true impact of conversational tutoring on learning is still not settled empirically (Rus et al., 2013).

1.2 DeepTutor

The research presented in this thesis was conducted with the intelligent tutoring system DeepTutor ²(Rus et al., 2013). DeepTutor is a state-of-the-art dialogue-based intelligent tutoring system (ITS) that tutors students on science topics (to be specific, it currently focuses on conceptual physics) through problem-solving conversations. It is a web based intelligent tutoring system (accessible 24/7) where students interact with the system via natural language texts. DeepTutor promotes deep learning of complex science topics through a combination of advanced domain modeling methods, deep language and discourse processing algorithms, and advanced tutorial strategies. DeepTutor has been developed as a web service and is fully accessible through a browser from any Internet-connected device, including regular desktop computers and mobile devices such as tablets.

During the interaction with the system (see Figure 1), students are challenged to solve qualitative Physics problems. They must provide an answer in the form of a short essay. Their solutions are automatically evaluated using the natural language assessment methods. The dialogue follows whose goal is to coach students in finding the solution by them.

DeepTutor has been deployed and tested by hundreds of high school and college students. About 90% of the times, the system responded correctly to the students (based on the analysis of about 2500 responses given to the students during a summer 2014 experiment with high school students (41 students) conducted at the University of Memphis. We used some of the records of student-tutor conversations for our independent use (please see Section 4 for the details about the dataset we used in our

² <http://deeptutor.org>

experiments). The detail description about the system is beyond the scope of this thesis (for details see Rus, Niraula, Lintean, Banjade, Stefanescu, & Baggett, 2013)

DeepTutor

Reset Session
Move To Next Task

Current Task

A rocket pushes a meteor with constant force. Assume that the meteor is initially at rest and is far enough away from the sun and the planets to neglect gravity. How will the meteor move?

Please begin by briefly answering the above question. After briefly answering the above question, please go on to explain your answer in as much detail as you can.

Dialogue History

Tutor: We now analyze how an object moves when a single force is being applied on it. Please read the problem and then solve the mystery by using Newton's laws.

You: *the meteor will move with constant velocity in a straight line*

Tutor: It can't be. According to Newton's first law, the meteor would move with constant velocity if a ZERO net force were acting on the meteor. That is not the case here as the rocket exerts a constant NON-zero net force on the meteor. Which of Newton's laws gives the general relationship between net force, mass, and acceleration?

You: *newtons second law*

Tutor: Excellent! Why is Newton's second law relevant?

You: *there is net force acting on the meteor*

Tutor: Right! The second law is relevant because the problem describes a non-zero NET force, which is the push from the rocket, being applied on an object, which is the meteor in this case.

Tutor: Can you articulate a principle or definition which justifies your answer?

You: *net force equals mass times acceleration*

Tutor: Excellent! Newton's second law says that NET FORCE equals MASS times ACCELERATION.

Tutor: Given the non-zero net force acting on the meteor, what does Newton's second law say about the acceleration of the meteor?

Student Response

acceleration is |

Submit

Copyright@ DeepTutor.org,2014

Figure 1: DeepTutor tutoring interface

1.3 Natural Language Understanding

Natural language processing (NLP) is a field of computer science, artificial intelligence, and linguistics concerned with the interactions between computers and human (natural) languages. As such, NLP is related to the area of human–computer interaction. Many challenges in NLP involve natural language understanding, that is,

enabling computers to derive meaning from human or natural language input, and others involve natural language generation (Jurafsky & Martin, 2009). The significant progress has been made in NLP research area. However, to a great extent the human-level understanding process is yet to be fully understood.

Natural language processing techniques are the basis for automatic open answer assessment methods. For example, spelling correction, grammatical error correction, speech act classification, ellipsis handling, negation handling, and co-reference resolution are some of the core NLP tasks that are needed for text understanding. In fact, there are a myriad of techniques applicable to automatic answer evaluation which more or less imitate the process of human cognition. A very important point to note is that all the methods which are developed and tested in different domain do not directly fit into the answer evaluation system. The domain specific concepts, naturally occurring texts (includes noise), and contextual information in conversations are characteristics of ITS data.

1.3.1 Paraphrase Identification, Textual Entailment, and Semantic Similarity

Text similarity is a bidirectional, continuous function which operates on pairs of texts of any length and returns a numeric score of how similar one text is to the other (Bhagat, & Hovy, 2012; Gabrilovich, & Markovitch, 2007; Landauer, Foltz, & Laham, 1998; Mihalcea, Corley, & Strapparava, 2006; Rus, Banjade, & Lintean, 2014).

Paraphrase is a specific case where a pair of texts is compared whether their meaning is same (or almost same). Textual entailment is to recognize whether the meaning of a target natural language statement H (H for hypothesis) can be inferred from another piece of text T (T for text). The bi-directional relation may not always hold true for textual

entailment. Apparently, these core tasks underlie semantic inference in many text applications. The task of analyzing student responses is one such example. By assigning the student's answer as T and the reference answer as H, we are basically asking whether one can infer the correct (reference) answer from the student's response.

Though these methods are widely adapted for automatic answer grading (Dzikovska et al., 2013), they are not perfectly suitable for the purpose. For instance, textual similarity and textual entailment applications assume that the texts are clean, self-contained, and the decision is binary. In contrast, answer assessment systems (especially in ITS environment) requires dealing with naturally occurring texts, and give two-way or multi-way decisions. Moreover, similar to human conversation, they need to infer meaning in context.

There are wide varieties of methods for measuring semantic similarity and detecting textual entailment. But the approach we are taking potentially supports more diagnostic evaluation as opposed to purely data mining approach where giving explanation for the results is a big challenge.

1.3.2 SEMILAR Toolkit

The SEMILAR (the SEMantic simILARity; Rus, Lintean, Banjade, Niraula, & Stefanescu, 2013) toolkit includes a set of tools and implementation of a number of algorithms proposed over the years to measure the semantic similarity of texts at different levels of granularities. It is available for free download in the form of a Java library at www.semanticsimilarity.org. We used various word to word similarity methods and optimal alignment solutions provided by SEMILAR API (in Section 5). Our goal is more ambitious and aims at augmenting SEMILAR with more powerful methods.

1.4 Student Response Analysis in ITS

The development of assessment technologies has a history of about half century. In 1966, Page and Fisher worked on an automatic essay grader called “Project Essay Grade (PEG)”. He looked for correlations between features of student texts, such as number of common words, average word length, number of commas etc. and the corresponding teacher grades. Another line of research targets the evaluation of short descriptive or explanatory answers against the reference answer(s). With the development in the field of Natural Language Processing, many techniques have been developed to evaluate the textual input. As a culmination of progress in natural language processing, automatic answer grading has been possible or is at least promising.

In dialogue based intelligent tutoring systems, such as DeepTutor, the student response evaluation is one of the very critical components because the quality of this component has major consequences for the effectiveness of the system to promote learning gains. Failure to accurately assess the student answer has severe consequences such as inappropriate feedback. It not only frustrates students but also diminishes the value of assistive learning technologies. The purpose of tutoring systems is to help student understand concepts, tackle problems, and correct misconceptions. Encoding every bit of knowledge and inference are very important to understand the meaning (semantic) of text. However, these become intractable at scale, i.e., when the coverage has to be increased. Alternatively, the lexical and syntactic features have been used widely, and they have achieved significant progress.

We illustrate next the problem of student answer assessment in dialogue-based intelligent tutoring systems.

Example 1.1: An example showing a problem solving dialogue in progress and the expected answer (reference answer). The spelling errors and grammatical errors in the student response are intentionally left as typed by the student.

Problem description: A rocket is pushing a meteor with constant force. At one moment the rocket runs out of fuel and stops pushing the meteor. Assume that the meteor is far enough away from the sun and the planets to neglect gravity

Dialogue History:

TUTOR (question 1): How will the meteor move after the rocket stops pushing?

STUDENT: *The metor will move toward the rocket at a constant velocity based on Newton's 1st law. Since there is no forces acting on the metor except its normal force which is upward.*

TUTOR (feedback 1): Great! The meteor will move with constant velocity in a straight line.

TUTOR (question 2): Can you name which of Newton's laws is relevant to this problem and why?

Reference answers (for question 2):

- a. The first law is relevant because the problem involves an object on which no forces are acting.
- b. The first law is relevant because the problem asks about motion when no forces (that is, zero net force) are acting on an object.
- c. The first law is relevant because there is a zero net force acting on the meteor.

STUDENT: *1st law. The 1st law because since there are no forces acting on the object then it will move at a constant velocity.*

TUTOR (feedback 2): Bravo! Newton's first law says that if the net force on an object equals ZERO the object is either AT REST (zero velocity) OR moves with a CONSTANT velocity in a STRAIGHT LINE.

The Example 1.1 illustrates the problem solving dialogue in progress. Let's suppose that tutor is asking the second question and the student gives the answer. Then the computer tutor has to compare the student response with the reference answer(s). The reference answer set contains various ways of telling the same (correct) answer. If the student answer matches with any of the reference answers, the tutor gives positive

feedback. Otherwise, tutor may decide to give hints. The student has used short form *1st* for first law. He/she also did not explicitly mention *relevant* but humans can understand from the context that the student is talking about the relevant law. The student response contains more than what is in the reference answers - *it will move at a constant velocity*. So, student can give incomplete, perfectly correct, partially correct, incorrect, contrasting, etc., and the system should be able to understand and decide accordingly. These are the few things that give some sense about what is answer evaluation and why it is challenging. As this thesis is centered on this topic, each Section discusses this problem from various perspectives.

1.4.1 Scoring Policy

The objective of assessment is to determine how correct the student answer is, such as correct, incorrect, partially correct, unrelated, etc. Typically, the scoring of conceptual answers is content based only as opposed to essay scoring where essays are marked based on content and style. In conceptual short response questions, style is less of a concern. Style could be an indication of learner's language skills which could be used by the system to guide its response.

The numeric score gives the holistic evaluation result. On the other hand, in analytic approach, the student answer is assigned a category (two-way, or multi-way). In two-way scoring, the decision is binary – answer is correct or incorrect. In multi-way scoring, the categories are finer grained. For example: correct, incorrect, partially correct, contradiction, unrelated etc. The choice of scoring categories depends on the need. In some cases, the two-way judgment is enough; in general it is not sufficient because the two-way scoring is not very informative compared to the multi-way scoring. In tutoring

systems, the scoring policy has to be aligned with the feedback mechanism. There is a chain effect. The accuracy of evaluation helps provide more appropriate feedback.

For instance, the student answers in SemEval-2013 shared task on “The Joint Student Response Analysis and 8th Recognizing Textual Entailment Challenge” (Dzikovska et al., 2013) are assigned to one of the five different labels, as follows.

1. **‘Correct’** - if the student answer is a complete and correct paraphrase of the reference answer.
2. **‘Partially_correct_incomplete’** - if it is a partially correct answer containing some but not all information from the reference answer.
3. **‘Contradictory’** - if the student answer explicitly contradicts the reference answer
4. **‘Irrelevant’** - if the student answer is talking about domain content but not providing the necessary information;
5. **‘Non_domain’** - if the student utterance does not include domain content, e.g., “I don’t know”.

1.5 Major Components of DeepEval System

From the analysis of tutor-student interactions, it has been observed that various linguistic phenomena are present in the student responses. In fact, it was expected. They all carry some meaning more or less. To addresses those phenomena and make judgments on which label to assign for the given answer, we have designed multiple components and organized them in the framework. Those components are: Preprocessing (tokenization, normalization, spelling correction, lemmatization, tagging, parsing etc.), speech act classification, gaming detection, co-reference resolution, ellipsis handling,

negation handling, concepts extraction and filtering, scoring feature generation, and score calculation or labeling. We implemented some of these components. However, for various reasons we have not implemented others. Apart from that, we developed various models required by these components; most importantly, we developed a scoring policy model which decides which label to assign given various feature values.

1.6 Motivation

Many of the modern day's educational technologies require automatic answer grading or they work at scale with less human power when assessment part is automated. But the automated assessment technology is not fully developed. The performances of systems participated in SemEval 2013 shared task and other systems (discussed in related works section; Section 2) also suggest that FULLY automated (fully in the sense that the system is scalable and more general) assessment is still a mystery. It requires natural language understanding capability which is still a growing field of research. So, the real motivation working on this problem came from within. The specific factor that triggered was the observation of students' reactions about the intelligent tutoring system when everything went well versus when tutor did not understand their responses.

Let's look at what some of the anonymous comments students have made after using DeepTutor system in summer 2014 (the data collection and analysis is presented in Section 4). *"I HOPE TO SEE PROGRAMS LIKE THIS INTEGRATED INTO DAILY EDUCATION PROCESS". "I REALLY LIKED IT! It was much more effective than I expected! "*. *"The deep tutor is both friendly and helpful". "Deep Tutor was pretty interesting. It's pretty cool". "great way to facilitate and complement physics learning",* and so on.

On the other hand, a few of them shared some negative experiences as well. For example: *"It was a good experience, but I did feel that Deep Tutor did not understand all of my responses", "I articulated the correct answer and was told said answer was incorrect", "Deep tutor was helpful, but she misunderstood some things that I said", " It was a good program and recognized most of my responses, although with some questions it did not understand the answer I gave, even though it was correct".*

As mentioned earlier in this section (in Section 1.2), the DeepTutor responded correctly about 90% of the times. However, when the tutor missed some cases, based on their feedback, it seems that they had some negative impression about the system which could discourage them using such systems or their lack of interest would lead to less gain. In turn, it can have negative impact towards achieving the goal of research on educational technologies. Also, students rightfully demand comprehensible explanations when their solution is rejected (or accepted) by the system. The current systems do not have that capability. So, we firmly believe that the improvement in such automatic assessment techniques will take educational technologies to new heights.

To re-iterate, the evaluation of student responses which are open ended in nature (even if limited to a specific domain and they have specific targeted answer) is an extremely challenging problem. From a technical standpoint, it is difficult mainly because it requires natural language understanding ability which is still far from the understanding capability of human. To achieve some improvement, more systematic approach is needed. For instance, various linguistic phenomena (see Section 3) are present in the student input and they need to be addressed. Also, the evaluation model should align with the feedback model of tutoring system.

1.7 Goal

The ultimate goal of this line of research is to have an accurate and reliable automatic student response evaluation model for intelligent tutoring systems. In other words, the far goal is to have an evaluation system that is as capable as an expert human tutor.

However, there are myriad of issues the intelligent tutor must handle in order to fully understand the semantics (i.e., meaning) of student natural language responses. The goal of this thesis work is to take a deeper look at student-tutor interactions, design the important components to handle various linguistic phenomena, and integrate them to produce an end to end system.

1.8 Contribution

The contribution of this thesis has been summarized in the following points.

1. Studied various linguistic phenomena prominently present in real student responses.
2. Designed a framework to integrate various language and knowledge processing components to systematically handle different linguistic phenomena for improved student response assessment.
3. Reducing human effort by using simpler form of concept representation which is neither too specific nor too shallow, and automatic concepts extraction methods.
4. Stepping towards diagnostic evaluation through the selection of potentially useful solutions for diagnostic evaluation, feedback generation, and follow up question generation.

1.9 Roadmap

In the next section (Section 2) we present a literature review where we describe about major trends in this area of research. In Section 3, we present the design of various components of DeepEval system and their integration. After then, in Section 4, we discuss the data collection and correctness annotation process. Experiment design, implementation, and results are presented in Section 5. A section (Section 6) on discussion follows where we discuss different factors that contribute positively or negatively on the evaluation system. Finally, in Section 7, we conclude the thesis highlighting future avenues on this line of research.

2 Related works

In this section we discuss various works in automatic answer assessment (or grading). Though our work is focused on short answer evaluation, we have included different aspects of evaluation to offer a broader perspective.

2.1 Automatic Essay Grading

Automated essay scoring is a measurement technology in which computers evaluate written work. In 1973, the late Ellis Page and colleagues at the University of Connecticut programmed the first successful automated essay scoring engine, “Project Essay Grade (PEG)” (1973). For example, a model in the PEG system might be formed by taking five intrinsic characteristics of writing (content, creativity, style, mechanics, and organization) and linking process. Now there are many essay scoring systems where some of them are commercial. The Educational Testing Service (ETS) has e-rater (e for essay). Vantage Learning has developed Intellimetric. Similarly, Pearson Knowledge Technologies supports the Intelligent Essay Assessor which is used by a variety of proprietary electronic portfolio systems. Landauer (2003) briefly presents the survey of essay grading technology, and then describes one such system, the Intelligent Essay Assessor (IEA). Apex (for an Assistant for Preparing Exams; Lemaire, & Dessus, 2001) is a tool for evaluating student essays based on their content. Their semantic text analysis relies on LSA. Bethard, Hang, Okoye, Martin, Sultan, and Sumner (2012) present initial steps towards an interactive essay writing tutor that improves science knowledge by analyzing student essays for misconceptions and recommending science web-pages that help correct those misconceptions.

They essay grading technique can be used to grade the summary writings. Several techniques such as Latent Semantic Analysis (LSA), n-gram co-occurrence and BLEU have been proposed to support automatic evaluation of summaries (Miller, 2003; Pérez et. al, 2005; Wild, Stahl, Stermsek, & Neumann, 2005). To improve the performance, He, Hui, and Quan (2009) proposed an ensemble approach that integrates LSA and n-gram co-occurrence based methods. Franzke and Streeter (2006) at the University of Colorado at Boulder developed Summary Street, an automated tool to evaluate the content of students' summaries. Summary Street grades students writing by comparing it with the actual text, evaluating it based on content knowledge, writing mechanics, redundancy and relevancy. Graesser and his team (2004) developed Coh-Metrix to analyze the text characteristics, such as cohesion. There are many other works related to essay analysis and grading.

Automatic essay scoring is related but different from short answer evaluation. Essays are typically long, open ended, and scoring is performed based on both content and the broader style or organization of writings (such as coherence, cohesion). On the other hand, in short answer evaluation the content is evaluated more precisely where the syntax is also important (i.e. the grammatical relation among words is also important).

2.2 Assessment of Non-Cognitive Factors

In recent years, there has been increasing interest in automated assessment of students in a broader range of contexts, and for a broader range of constructs, than traditional assessment achieves (Baker, Goldstein, & Heffernan, 2011; Conati & maclaren, 2009; Matthew & Stemler, 2013; Sabourin, Mott, & Lester, 2011). For instance, as early as 1948, the first President of Educational Testing Services suggested

measuring personal drive, motivation, conscientiousness, interpersonal skill, and interest (Lemann, 1995), and there were serious attempts to measure personality and motivation starting from the 1960s. One construct that has emerged as a focus of research in the last decade is boredom. Miller, Petsche, Baker, Labrum, and Wagner (2013) present automated sensor free assessment that can infer boredom, a key non-cognitive factor during student learning. They assessed student boredom across the year using sensor-free automated detectors developed using a combination of quantitative field observations and data mining, validated to generalize across students and school contexts.

As student's motivational factors are linked to the learning gain, assessment of non-cognitive factors such as boredom, frustration, confusion, etc. is equally important. The sensor free assessment of affective state could mean that the student's response is analyzed to infer those states. However, we have not worked on this aspect.

2.3 Situated Assessments

The assessments of the ability to apply scientific methodology focus on practical side of learning. Techniques have been developed where learners apply their practical knowledge in a simulated environment. Sil, Ketelhut, Shelton, and Yates (2012) presents a project called SAVE in which two virtual worlds that each have a mystery or natural phenomenon requiring scientific explanation are created. The students' behavior as they investigate the mystery is recorded in order to assess their understanding of scientific methods.

It is quite different from the traditional evaluation approaches, such as multiple choice questions, essay writing etc. The theory of situated cognition suggests the lack of contexts in current standardized tests of science on many grounds. The motivation of

these kinds of approaches is help applying classroom-based learning and to engage students more. There are several other works on situated assessments (Choi & Hannafin, 1995; Nidumolu, Subramani, & Aldrich, 2001; Young, 1995).

2.4 Short Answer Assessment

The short answer assessment is to measure how much of the targeted concept is present in the student answer (ranging in length from a few to approximately 100 words). Since scalable techniques that deeply understand the text has not been developed yet, the advances in computational linguistic techniques opened up the possibility of being able to automate the answer assessment process.

Martin and Vanlehn (1995) proposed an assessment system OLAE using Bayesian nets. In OLAE, assessment produces a student model, i.e. a collection of correct and incorrect rules from the domain model known and used by a particular student. A student model is essentially a rule-based computer program that computes answers to actual problems in the same way as the student does. OLAE uses such an approach because assessments of which rules a student uses are necessarily uncertain. Though their solution is distinctive, the problem with this approach is that the human must generate the Bayesian network for each problem; this is why the approach does not scale.

The short answer grading has reached commercial levels as well. The C-rater system (Leacock & Chodorow, 2003; Sukkarieh & Blackmore, 2009) is one of the ETS's automatic scoring technologies (e-rater, c-rater, m-rater, and SpeechRater for essay scoring, content scoring, math scoring, and Speech input scoring respectively). C-rater is used for automatic analytic-based content scoring of short free-text responses. Analytic-based content is the kind of content that is predefined by a test developer in terms of main

ideas or concepts. These concepts form the evidence that a student needs to demonstrate as her/his knowledge in his/her response. C-Rater matches the syntactical features of a student response (subject, object, and verb) to that of a set of correct responses. Their system breaks the reference answers into constituent concepts that must individually be matched for the answer to be considered fully correct. The c-rater system has been used within many domains, including biology, English, mathematics, information technology literacy, business, psychology, and physics.

The C-rater requires that the reference answer be broken down into a set of concepts in the form of simple sentences. Then, it applies textual entailment techniques based on syntax, lexical semantics, and simple semantic roles to identify whether the concept is present or not. However, the process is time consuming and requires more human effort. As they mentioned (Sukkarieh & Stoyanchev, 2009), the knowledge engineering process of building a model for a question took at least 12 hours. They proposed automatic model building for C-rater.

LSA (Landauer & Dumais, 1997) and machine translation evaluation methods are also applied for short answer grading. Pérez et al. (2005) applied the combination of Bleu-inspired algorithm and LSA. Their idea was to perform both syntactic and semantic analysis. They did some syntactic analysis such as stemming, closed-class word removal, word sense disambiguation and synonyms treatment procedures etc. They combined LSA method with syntax based methods where LSA captures the semantics. Despite the simplicity of these shallow NLP methods, they achieved significant correlations to the teachers' scores while keeping language-independence and without requiring any domain specific knowledge.

Another short answer grading system, used in AutoTutor system (Graesser, Wiemer-Hastings, Harter, Tutoring Research Group, & Person) applied LSA approach. Later work on AutoTutor seeks to expand upon the original bag-of-words approach.

Mohler and Mihalcea (2009) explore unsupervised text similarity techniques for the task of automatic short answer grading answers to the introductory computer science assignments. They applied a number of knowledge-based (for example, WordNet) and corpus-based measures (LSA and ESA) of text similarity. They explored the impact of domain and size of the model development corpus on the accuracy. To evaluate the domain impact, they developed LSA model from Computer Science articles and compared with the LSA models developed from general Wikipedia articles. They found higher correlation of similarity score with human ratings when domain specific (i.e. model developed computer science articles) model was used. Mohler et al. (2011) applied semantic similarity methods and dependency graph alignments to grade short answer questions. Similarly, Murrugarra, Lu, and Li (2013) proposed using domain general methods, bag-of-words approach, LSA representation, textual entailment, and others.

Rus and Lintean (2012) presented a novel, optimal semantic similarity approach based on word-to-word similarity metrics to solve the important task of assessing natural language student input in dialogue-based intelligent tutoring systems. The optimal matching is guaranteed using the sailor assignment problem, also known as the job assignment problem, a well-known combinatorial optimization problem. They compared the optimal matching method with a greedy method as well as with a baseline method on data sets from two intelligent tutoring systems, AutoTutor and iSTART.

Creating a good set of reference answers is one point where the human involvement is needed in automatic answer grading. Student can express the same concept using various ways. The automatic answer grading would be done more confidently when the student answer is expressed similar to the reference answer(s). Utilizing the student answers to increase the pool of reference answer is a possibility. Also, grouping the similar answers and evaluating them in a group requires less effort. Basu, Jacobs and Vanderwende (2013) have proposed a model on that called ‘powergrading’. They utilize the similarity methods used in answer grading to form clusters and sub-clusters. The answers in the same cluster are evaluated by teacher in a single action and similar feedback is given to the whole group.

As semantic similarity and textual entailment are closely related to the problem of automatic answer evaluation, virtually every text to text similarity and entailment method could be framed into this task. These two fields are themselves big research areas. We leave further exploration of these fields to the reader. Though various results show that the similarity based methods can be potentially used in the answer grading tasks, they made assumptions that the text available are standard texts with noise filtered, and they did not consider any contextual information, whereas we work on naturally occurring texts.

Noisy data

It is difficult to apply the standard tools and techniques that are applicable for less noisy texts to disfluent student answers (i.e. texts with various breaks, irregularities, or non-lexical vocables) that prominently occur in the elementary students' writing. Leeman-Munk, Wiebe, and Lester (2014) proposed a domain independent method to

assess elementary students' science competency using soft cardinality. This method is robust to grammatical errors. It allows evaluating word similarity across misspellings. Soft cardinality (Jimenez, Becerra, Gelbukh, Bátiz, & Mendizábal, 2013) uses decompositions of words into character sequences, known as q-grams, to gauge similarity between two words. They evaluated this technique with the 4th grade student's writing. Though this technique is simple and can handle noisy data, it does not consider the word order and compositional semantics.

2.5 Competitions and Shared Tasks

Many conferences and workshops, such as Intelligent Tutoring Systems conferences, educational data mining, AI in education, workshop on building educational applications using NLP or innovative use of NLP for building educational applications, KDD¹ workshop on Data Mining for Educational Assessment and Feedback (ASSESS 2014) are being organized. The Hewlett Foundation: Short Answer Scoring competitions (phase I and phase II, 2012)² were organized where 153 teams participated from around the world. Their focus in phase I was essay grading whereas they focused on grading short answer responses. The competitors (administered through kaggle.com) had the challenge to develop a scoring algorithm for student-written short-answer responses. These responses consisted of essays of approximately fifty words which were written by 10th grade students answering questions that cover a broad range of disciplines (from English Language Arts to Science). Many of the participating teams performed above the given baseline. The high ranking algorithms applied tf-idf weighted vectors after a set of

¹ KDD for Knowledge Discovery in database. KDD conferences and workshops are organized every year.

² <https://www.kaggle.com/c/asap-sas>

preprocessing techniques, regular expressions, grouping data and stacking machine learning algorithms. Those short essays were purely open-ended. There were no reference answers to compare with.

A shared task, The Joint Student Response Analysis and 8th Recognizing Textual Entailment Challenge, was organized as a part of the Semeval 2013 (Dzikovska et al., 2013) to promote and streamline research in this area. The task was to label (2-way, 3-way, and 5-way) the student answers by comparing with the correct reference answers. The corresponding questions were also provided. Instances in the dataset were labeled with one of the following labels: Correct, Partially_Correct_Incomplete, Contradictory, Irrelevant, or Non_domain. The 2-way (correct or incorrect) and 3-way (correct, incorrect, or contradictory) labels were created by collapsing the 5-way labels. To test whether the system generalizes across problems and domains, the three different test set were created: unseen answers, unseen questions, and unseen domains. They also organized a pilot task on partial entailment. It aimed at recognizing when specific parts of a reference answer are expressed in the student answer, even if the reference answer is not entailed as a whole. The systems were required to recognize whether the semantic relation between specific parts of the Hypothesis is expressed by the Text, directly or by implication, even though entailment might not be recognized for the Hypothesis as a whole, based on the SCIENTSBANK facet annotation (Nielsen, Ward, Martin, & Palmer, 2008). The detailed annotation was thought to be useful to improve accuracy and providing analytic feedback. As mentioned in the Semeval shared task report (Dzikovska et al., 2013), nine teams participated in main task. At least 6 of the teams used some form of syntactic features. At least 5 systems used a system combination

approach, with several components feeding into a final decision made by some form of stacked classifier. The majority of the methods used some kind of text to text similarity with lexical and syntactic features. The result (some of the teams performed well above the baseline) showed the possibility of computational linguistics in answer grading.

However, only one team participated in the pilot task. This may be due to the difficulty of the task, lack of interest, or the unfamiliarity with such kind of data.

2.6 Answer Grading in Various Languages

Perez-Marin, Alfonseca, Rodriguez, and Pascual-Nieto (2006) developed a scoring system called WILLOW where the students can select either Spanish or English language; i.e. the system adapts to the students' language, so they can write their answer in the language that they choose (Spanish or English). Using automatic Machine Translation techniques, the texts are translated to the language in which the teachers' references are written and compared with the reference answer. Vantage Learning (2001; 2002) studied their grading tool, called IntelliMetric, in Hebrew, and Malay languages. Similarly, there are many other systems developed for scoring answers in different languages, such as Japanese (Kawate-Mierzejewska, 2003), Arabic, French, etc.

2.7 Domain Specific Solutions

Bailey and Meurers (2008) proposed a content assessment model for meaning errors in short answers to reading comprehension questions for English as second language learners. Perry and Shan (2010) described Prograder, a software package for automatic checking of requirements for programming homework assignments. It lets instructors specify requirements in natural language and explains grading results to students in natural language.

The student answer can contain mathematical expressions. Since 1985 the Computer Aided evaluation of mathematical expressions has been developed (Beevers & Paterson, 2003; Rasila, Harjula, & Zenger, 2007; Sangwin, 2004). One philosophical objection to this research is that mathematical work is not about obtaining the ‘correct answer’. In learning and teaching, the method used forms essential evidence for a student’s understanding of the processes involved. Sangwin (2004) presents the case study of STACK, a computer aided assessment tool for mathematics in which a computer algebra system (CAS) is used to help assess students’ responses to elementary algebra questions, to explain the difficulties of using a general purpose computer algebra system to assess elementary algebra questions. In mathematical answer evaluation systems, typically Computer Algebra System CAS (e.g. Maxima, Maple, and Mathematica) is used to grade the student answers and give relevant feedback.

Though general purpose semantic similarity and textual entailment methods have been applied for automatic answer evaluation, domain adaptation can exploit the characteristic features of that specific domain. Within that framework, some domain adaptation techniques have been developed. One such system from Educational Testing Services (ETS) uses an approach called “domain adaptation and stacking” (Heilman & Madnani, 2013) where they use item-specific features as well as general features. To be specific, they generate a copy of a given feature for grading answers to seen questions, answers to unseen questions in seen domain, and answers to questions in unseen domains, and each of these has a separate weight. An item represented in the training data uses all three of these feature copies, and an item from another domain will only use the latter, “generic” feature copy.

2.8 Assignments Evaluation in MOOC Systems

One problem that arises with the increasing numbers of students in Massive Open Online Courses (MOOCs; such as Coursera, and EdX) is that of student evaluation. The large number of students makes it infeasible for humans to grade all assignments. As a result, there has recently been a great push for employing peer-grading, where students grade each other. However, in practice, peer-grading has been observed to have high error rates and has come under serious criticism (Kulkarni, Socher, Bernstein, & Klemmer, 2014; Kulkarni, Wei, Le, Chia, Papadopoulos, Cheng, Koller, & Klemmer, 2013). Now the things have been changing. Most recently, Perelman, Gulwani, and Grossman (2014) proposed automatic evaluation and feedback generation for introductory computer science assignments. Their approach is an adaptation of Test-Driven Synthesis (TDS; Perelman, Gulwani, Grossman, & Provost, 2014). They compare the student response with the reference answer to measure the accuracy. They have devised a data mining approach on adding more reference solutions by selecting different but correct approaches in student responses. They also used their tool to produce hints for the educational programming game Code Hunt.

From statistical analysis, Shah et al. (2014) found that peer grading in MOOC systems does not scale. Also, the research has shown (Kulkarni et al., 2013; Kulkarni et al., 2014) that current auto-grading and peer grading systems make a large number of mistakes. So Shah et al. (2014) considered a hybrid approach that combines peer-grading with auto-grading. In this setting, an automated approach is used for ‘dimensionality reduction’, a classical technique in statistics and machine learning, and peer-grading is employed to evaluate this lower dimensional set of answers. Similarly, Kulkarni et al.

(2014) proposed a hybrid approach. They show that this alternative approach has the potential to scale.

Aggarwal, Minds, Srikant, and Shashidhar (2014) presented a case study on using machine learning for the assessment of programming tasks for job candidate selection. They described important steps or principles to consider while designing such a system, namely, choosing a useful response format, creation of a robust rubric, capturing features which correspond to the developed rubric and choosing a machine learning model to predict human expert's grades.

2.9 Common Corpora for Benchmarking

Although a considerable amount of work has been done in this area, less work has been done on creating common benchmarks and evaluation measures in order to perform a comparative evaluation or progress tracking of this application across systems. However, there has been progress on that avenue. Sukkarieh and Bolge (2010) have introduced an Educational Testing Service-built test suite that makes a step towards establishing such a benchmark. The suite helps us identify the missing phenomena, which phenomena our system fails to capture, and account for rare phenomena. To apply this kind of model, the reference answer has to be divided into multiple concepts. In that sense, though it serves as a good benchmarking test suite, it restricts the systems to break the reference answers into multiple sentences each representing a specific concept. Similarly, the dataset published with shared tasks such as SEMEval 2013 (Dzikovska et al., 2013) can serve as the benchmark data.

2.10 Summary

In this section, we presented related works in automatic assessment of natural language responses and the different aspects that are linked to learning. It is by no means an exhaustive discussion but it represents different variations of tools and techniques applied for automatic assessment.

In summary, the methods applied to evaluate the natural language responses (in text format) borrowed ideas from semantic similarity and textual entailment. Apparently, there is no system that addresses the specific needs of intelligent tutoring systems, such as taking contextual information into account, handling more casual conversations, developing more transparent systems suitable for follow up question generation and diagnostic feedback generation, etc. Our DeepEval system, on the other hand, is designed considering these factors. The closely related system, C-rater (Leacock & Chodorow, 2003; Sukkarieh & Blackmore, 2009), developed by ETS requires that the reference answer be broken into smaller concepts they are themselves simple sentences. But the knowledge engineering process is time consuming and requires more human effort. As they mentioned (Sukkarieh & Stoyanchev, 2009), the process of building a model for a question took at least 12 hours. They also proposed automatic model building for C-rater (Sukkarieh & Stoyanchev, 2009) where those concepts are extracted automatically. However, given the conversations between tutor and students in intelligent tutoring environment which are more like human to human conversations and more informal, we represent concepts in simpler form so that it is possible to automate extraction and inference. We also deal with various linguistic phenomena make an end-to-end system.

3 DeepEval Framework

There are many factors influencing the grading decision when assessing a student answer. The process is complex and developing an automated solution requires integrating ideas from different disciplines including psychology, computational linguistics, educational sciences etc. However, understanding the natural language responses from the students is a big technical challenge. It is a long standing problem. We focus on taking a systematic approach on addressing the natural language understanding task. We first design a set of components needed to handle various linguistic phenomena and to make an end-to-end system in such a way that each component more or less mimics what a human grader would do when he/she has to evaluate the given student answer. For example, preprocessing, speech act classification, co-reference resolution, ellipsis handling, concepts extraction, concepts mapping, grading features generation, and applying grading policy are some of the important processes for answer evaluation.

3.1 Design Principles

The design principles that we followed are:

- a. Methods should facilitate numeric scoring as well as assigning qualitative labels (such as, correct, incorrect, partially correct etc.).
- b. Move towards formative or diagnostic assessment where the system can explain the scores given. Diagnostic assessment is more transparent giving systems capabilities to explain the scores or grades. These are important for giving feedback and follow up question generation.

- c. Domain independent solution is a far goal as in many AI programs. Domain transferable solutions should be embraced.
- d. Handle various linguistic phenomena that carry the meaning.
- e. Reduce human effort (i.e. focus on FULLY automated solutions).
- f. Student answers are good sources of learning for developing various models.
- g. Understand the benefits and obstacles before going too deep. Deep semantic understanding is the ultimate AI goal but we are not there yet.
- h. Do not induce more errors than corrections. There are many components and the chances of error propagation are very high.

3.2 A closer Look at Student Responses

In this section, we look at different types of responses and linguistic phenomena that are more prominent for understanding the meaning of the texts and then we discuss about how human experts would evaluate those responses. To facilitate the analysis, we refer to Example 1 below. A set of responses (representative responses; which is by no means exhaustive) for the problem given in Example 3.1 are shown in Table 3.1. These responses do not cover all possible variations in answers for that question or all important linguistic phenomena.

Example 3.1 (problem: # FM_LV04_PR10.FCI-16)

Problem Description: *To rescue a child who has fallen down a well, rescue workers fasten him to a rope, the other end of which is then reeled in by a machine. The rope pulls the child straight upward at steady speed.*

Question: *How does the amount of tension in the rope compare to the downward force of gravity acting on the child?*

Reference Answers:

- *The amount of tension in the rope is the same as (equal to) the magnitude of the*

downward force of gravity.

- *When velocity is constant, the acceleration is zero; therefore the sum of the forces will equal zero*
- *When an object moves in constant velocity, sum of the force is 0*
- *Since the child is being raised straight upward at a constant speed, the net force on the child is zero and all the forces balance. That means that the tension in the rope balances the downward force of gravity.*
- *Gravity and tension balance.*
- *The amount of tension in the rope is the same as (equal to) the child's weight.*

Table 1: Some of the student responses to the question given in Example 3.1 (the spelling error in (1) is left intentionally).

ID	Student response
1	<i>The force excrted by gravity and tension of the rope are equal.</i>
2	<i>these forces balance each other</i>
3	<i>The tension is equal to the force of gravity</i>
4	<i>They are equal.</i>
5	<i>Equal</i>
6	<i>the tension force is balanced by the weight of the person</i>
7	<i>The tension in the rope is greater than the downward force of gravity.</i>
8	<i>the tension in the rope is greater than gravity in order to raise the child upwards</i>
9	<i>they are equal and opposite in direction</i>
10	<i>The tension in the rope is equal to the mass of boy times gravity. Newton's second law states the force is equal to mass times acceleration. In this case, the tension is the force. Gravity is the acceleration.</i>

By looking at the student responses in Table 1, we can see that the student answers can vary substantially with each other and they do not perfectly overlap with the reference answers (given in Example 3.1) even when they are conceptually correct.

What would a human tutor typically do when she sees a student response? We roughly decompose it into the following ways:

1. If it is meta-cognitive or communicative, such as greetings, she would handle accordingly.
2. If student expresses an affective state, such as boredom or frustration, she would handle accordingly.
3. If student is trying to play with the system, such as unusually seeking more help from the tutor, she would handle it accordingly.
4. When she is mainly looking at whether the answer is conceptually correct, she would treat the misspelled word (if any) as if it was a correctly spelled word suitable in that context.
5. Similar to (point 4), she would ignore any grammatical errors (if not substantially changing the meaning of the text)
6. She would understand what entity the student is referring to after seeing pronouns or other referents. In the response (4), student used “*they*” to refer to *the tension and force of gravity*.
7. She would accept the answer as correct if student says something using different words which are not perfectly matching but acceptable in the given context. For example, in answer (6) the student says *person* which refers to the *child*.
8. If student did not give the complete answer but provides certain concepts, she would try to understand what student meant to say by looking at the context, such as question, previous utterances, problem description etc. For example, the answer (5) can be accepted as correct by looking for the missing words in the question. Otherwise, she would mark the answer as incorrect or incomplete based on the grading scheme.

9. She would detect if the answer presents wrong concepts (misconceptions). For example, answer (8) is incorrect because the student said *greater* though the rest of the answer is similar to the reference answer.
10. If student says something correct but other things incorrect, she would either give partial credit or no credit depending on the policy.
11. She would ignore minor details in the answer that are not conceptually incorrect or something that explains the answer. For example, the answer (10) where student gives explanations but question is not asking for it.
12. If the response is related but incorrect, she would identify them and grade them based on the grading policy. The tutor should respond with correcting the misconception or incorrect answer, or give hint.
13. If the response is completely out of domain, she would probably say “let’s not switch topics”.
14. She would explain (if needed) the score she gives or the overall judgment label she assigns to the student answer.

3.3 DeepEval Components

The ultimate goal of DeepEval system is to mimic the grading process followed by human tutors which was described in the previous section. We design components or group of components which more or less mimic this process. However, not all of the components are implemented for various reasons (discussed in Section 3.5) but we discuss them here. The implementation status is given in the bracket (I - Implemented, PI - partially implemented). Otherwise, the component has not been implemented. The following components are identified as the components required at minimum to build an

end-to-end student response evaluation system. For a particular instance, all or some of them will be applicable.

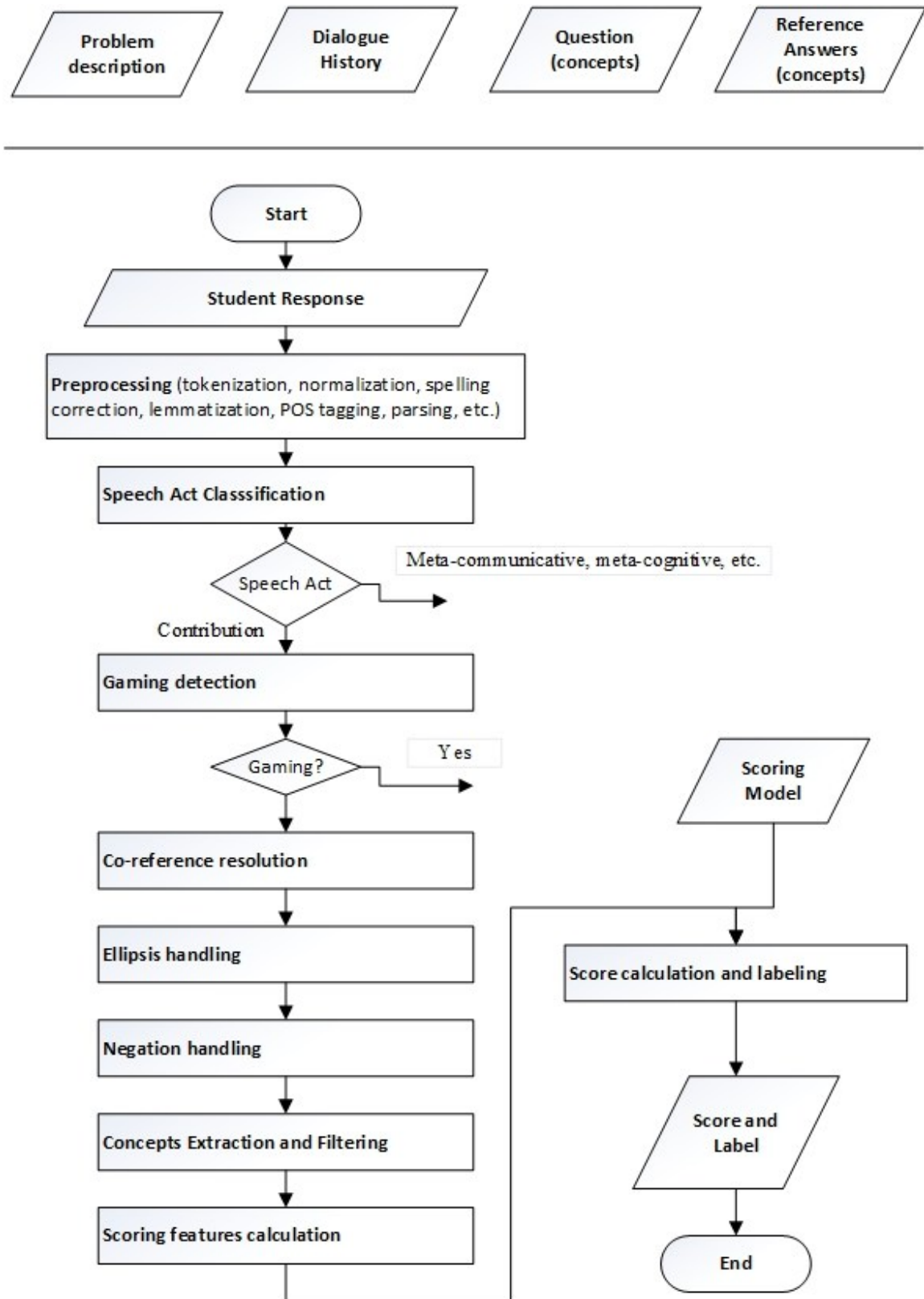


Figure 2: Components of DeepEval framework

The Figure 2 shows various components of DeepEval framework where they represent the sequence of steps to be followed in answer evaluation.

1. Preprocessing

The preprocessing component itself is comprised of a set of sub-components. They include text clean up and application of fundamental processes to facilitate further processing and to enhance the representation of text. For example, tokenization (I), parts of speech tagging (I), parsing (I), data normalization(I), spelling errors correction (I), grammatical errors correction, lemmatization (I), stop word detection and removal (I), and so on. The implementation details are described in Section 5.1.2.

2. Speech Act classification (I)

Speech act is the basic unit of intent in communication, such as greetings, questioning, response to the question etc. the speech act classifier indicates the general intention beyond a student response; example of speech acts are: meta-cognitive, meta-communicative, or contribution. Only the contributions (i.e. answers that address the question and contain real content with respect to the problem to be solved) are graded. The implementation is described in Section 5.1.3.

3. Evaluation of Non-Cognitive Factors

In addition to the traditional assessment, there has been increasing interest in the sensor free assessment of non-cognitive factors during learning, such as boredom, frustration, etc. What they express as part of their response is also important.

Students even use obscene language which clearly indicates their anger, frustration, or lack of interest.

4. Gaming Detection

Students can misuse the Intelligent Tutoring systems. Baker et al. (2004) describes gaming as the behavior of performing well in educational task by systematically taking advantage of properties and regularities in the system to complete that task, rather than by thinking about the material. For instance, they can abuse help or perform trial-and-error systematically. Gaming has to be detected in order to devise strategies to address them.

5. Coreference Resolution (I)

Co-reference resolution is the task of finding all expressions that refer to the same entity in a discourse. For example, the '*they*' in answer (4) in Table 1 refers to *the tension on the string and the force of gravity*. This is commonly occurring phenomena and those references should be solved to properly assess the student response. To be specific, the co-references is resolved before comparing the response with the reference answer. The implementation details are provided in Section 5.1.6.

6. Ellipsis Handling (I)

Incomplete utterances are common in communication between humans.

Similarly, they are also common in tutor-student interactions in ITSs. Though the tutoring systems are mostly coded to provide semantically and syntactically more complete utterances, the student utterances are often elliptical. This phenomenon occurs naturally despite explicit suggestions given to students to write complete

sentences. In many cases the student utterances cannot be understood in isolation but make sense when interpreted within the context. For example, the answer (5) in Table 3.1 is missing important details but they can be inferred from the context (question in that case). Precisely handling elliptical utterances is a difficult problem for natural language systems. However, we apply indirect approach in handling elliptical utterances together with Co-reference resolution. More details are provided in Section 5.1.6.

7. Negation handling (PI)

Negation is a frequent and complex phenomenon in natural language. In an analysis of a large number of student utterances (about 25,000) in dialogues collected from the Intelligent Tutoring System (ITS) DeepTutor, it has been found that 9.36% of the student responses contained negation. For example, if a question is asking about net force acting on an object, the student can write *there is no net force acting on the object*. The presence of negation (*no* is called negation cue word in this example) changed the meaning of the whole sentence. Similarly, the presence of negation can totally change the meaning of a sentence or part of the sentence in the student response. So, the response should be interpreted carefully. It is partially implemented and described in Section 5.1.4.

8. Concept extraction and Filtering (I)

The notion of a concept is defined very loosely with multiple definitions being proposed. In contemporary philosophy, one way to define a *concept* is as a *mental representation*; that is, *concepts are entities that exist in the brain* (cf.

Wikipedia). A concept can be realized at various levels – more specific to more general.

Based on compositional theory of meaning representation, the meaning of bigger text can be represented in terms of meaning of its constituents and their rules. We consider these elements to be the principal phrasal chunks forming the sentence, and the dependencies among them. The phrasal representation is more specific or more informative than words. For example, the concept *net force* is more specific than *force*. On the other hand, the phrases are less general than bigger concepts in that domain. For instance, a bigger concept in physics (here a principle) is that *"when the net force is zero, the velocity of object remains constant"*. However, because of the difficulty in representing larger domain concepts and the need of providing partial credit and explanations for the score or qualitative judgment label assigned to the student response, we represent meaningful phrases or chunks as concepts. The concept extraction and filtering is explained in Section 5.1.5.

9. Scoring features generation (I)

To assign numeric scores or to classify the answers to appropriate categories of evaluation, different features are extracted from various sources. For example, how much of the reference answer or expectation is covered by the student response, how many concepts remained uncovered in the reference answer and in the student response, whether there is any contradictory concept present in the student answer, relevancy, etc. The scoring features we used are described in Section 5.2.

10. Scoring (I)

Finally the answer is scored or a label is assigned based on a grading model. The grading model is first developed from the annotated data or by learning the grading policy. Depending on the scoring model, one value is assigned to the student response from the set of values in either a two-way or multi-way labeling set. The scoring model development is described in Section 5.2.

11. Explanation Generation

In addition to the grade, when needed, an explanation should be generated which is important for follow up question generation and feedback generation.

Uncovered concepts in the reference answer or wrong concepts in the student answer could be potential reasons of incorrect answers.

3.4 Models Development

The set of components (described in Section 3.3) require various models (machine learning models, rule based systems, or some other models) which should be developed before the system is in place or as an online process. The kind of model(s) to use is primarily dependent on the choice of implementation of a particular component or group of components. For example, the negation scope detection model could be built independently and added in the negation handling component of our framework. So, not all of the components are described in this section. However, we briefly describe the grading policy model.

3.4.1 Grading Policy Model

Scoring policy model is developed using the development dataset so that given various features extracted from different sources, such as problem domain, question,

reference answers, or student response, dialogue history, etc., it produce the final score or assigns a judgment label. The model depends on the scoring rubric. For example, the judgment labels could be two, three, or more. The process is similar but the set of features can vary so that they capture the level of granularity of judgment labels.

We designed a two-way scoring model. In two-way scoring model, the answer is either marked as correct (1) or incorrect (0). We selected the following features,

1. Expectation coverage score (ECS)
2. Presence of contradicting concept(s) in student response (PCC)
3. Uncovered concepts in the reference answer (UCRA):
4. Uncovered concepts in the student answer (UCSA)

A logistic function is fit using different combinations of features and best model is selected for unseen answers evaluation. The implementation detail of this scoring model is described in Section 5.2.

3.5 Other Challenges

As human judgment level is simply impossible to model, obviously there are numerous issues left untouched in the DeepEval design such as grammatical error corrections and domain modeling.. Though significant progress has been made on these challenges, it requires a careful judgment on whether to use them at this moment or about the way to integrate them. We have left them for future work. Here, we discuss a couple of issues.

Grammatical Error Corrections

The grammatical errors are of different types, including articles, prepositions, determiners, noun form, verb form, subject-verb agreement, punctuation, capitalization,

etc. The grammatical error correction has attracted much recent research interest, with different shared tasks organized in the last couple of years, such as Helping Our Own 2011 and 2012 (Dale, Anisimoff, & Narroway, 2012; Dale & Kilgariff, 2011), and CoNLL 2013 and 2014 (Ng, Wu, Hadiwinoto, & Tetreault, 2013; Ng, Wu, Briscoe, Hadiwinoto, Susanto, & Bryant, 2014). As Ng et al. (2014) summarized in the shared task report on grammatical error correction, language model based approaches, machine translation based approaches, and rule based approaches were more common and best performing systems which achieved $F_{0.5}$ score (its different from F_1 score) of about 45%. The task is challenging since many grammatical error correction system do not achieve high performance and there is still much room for improvement. There is high chance of introducing errors while correcting others. The grammatical error correction is essential, especially for handling non-native English speakers' writing and when the style of writing is important. However, most of the tutoring systems that focus on science concepts should discount grammatical errors to some extent as style is less important.

Moreover, modern day parsers are able to accurately parse somewhat noisy text from where we extract concepts. So, the methodology we are following is less sensitive to the grammatical errors. In addition, the common grammatical errors are nullified after removing stop words and doing preprocessing. Prepositions, determiners, articles etc. are the common sources of grammatical errors but most of them are stop words.

Knowledge Extraction, Representation, and Reasoning

Without defining knowledge extraction, representation, and reasoning, we would like to discuss what we can learn from projects like HALO (Friedland et al., 2004; Gunning et al., 2010). For instance, they hand-crafted formal knowledge bases for question-

answering in biology, called "*knowledgeable textbook*" such that users can not only browse, but also ask questions and get reasoned or retrieved answers back. While their previous work relied on hand crafted knowledge, their new effort has shifted towards automatic knowledge extraction (Clark, Harrison, Balasubramanian, & Etzioni, 2012). The finer level knowledge acquisition, representation, and reasoning are big challenges when applying them at scale.

4 DataSet

The dataset we used (called DeepEval data) was extracted from anonymized records of tutor-student conversations in one of the DeepTutor (introduced in Section 1.2) experiments where they solved conceptual physics problems; no quantitative problem solving was involved. The interactions happened through natural language interface in the form of written text. The tutor automatically assessed the correctness of responses by comparing the student input with the reference answers given by domain experts. We randomly selected a subset of anonymously recorded dialogue interactions, and annotated student answers for correctness in two-ways (correct/incorrect). That is, the dataset contains naturally occurring texts. In this section, we just describe the dataset and the collection process in brief without looking at other theoretical and practical aspects of the experiment.

4.1 Data Collection Process

In summer 2014, forty-one summer school students at the University of Memphis used DeepTutor system where each of them solved nine different Newtonian physics problems from a set of eighteen problems. They were conceptual physics problems. The DeepEval data is the subset of recorded tutor-student interactions during the experiment. The experiment was conducted in lab at The University of Memphis where each student was given enough time to read and solve those conceptual physics problems through the natural language interactions (in the written form) with the tutor. For each task, a problem description (consisting of a couple of sentences) along with image describing the problem visually was shown. The tutor asked questions and student provided answers by typing in the answer in free form. Some questions required

sentential input whereas other questions, especially during scaffolding, required keywords input. For each question, a list of reference answers was provided by the experts. These reference answers are the different ways of answering the same question and are set by domain experts. However, the lists are not exhaustive. Once the tutor received the student response, it assessed the correctness of that answer by comparing the answer with the reference answers for that particular question. If needed, the tutor provided appropriate feedback - positive, negative, or neutral and some hints to help understand the concept. The process repeats until all the expectations (concepts) of that particular problem were covered. Similarly other tasks were solved. The entire tutor-student interaction was recorded anonymously.

There was no human intervention at any time during the experiment. At the end of the session, students voluntarily submitted demographic information including academic level, gender etc. and their feedback about their experiences using the system. The only related suggestion given to the students was to write as complete as possible during the interactions with the tutor.

We randomly extracted 50% of the problem solving dialogues. That is, for each student and dialogues associated with a problem, either that full conversation is included or excluded. Since each student solved 9 different problems, all students and task solving dialogues are represented in the extracted subset. In total 198 dialogues are in the subset. From that subset, all questions requiring short answers (i.e. keywords) are filtered out as they require less linguistic analysis during assessment. Also, the DeepTutor specific information such as the type of feedback it provided to the students is excluded. In addition, only the contributions (the student answers that address questions) are taken.

The resulting dataset contains 618 question answer pairs. The Table 3 provides the summary of the dataset.

4.2 Annotation

We annotated student responses for correctness in binary form. There are different possibilities, two-way annotation (correct/incorrect) or multi-way annotation such as correct, incorrect, partially correct, etc. Also, an exact numerical score could be given for each answer. Though multi-way annotations are more informative, we need a bigger dataset to see the significance of results. It also requires more time and domain expertise which is costly. Similarly, providing exact numerical score requires domain expertise. So, as part of this thesis, we annotated answers either as correct or incorrect leaving multi-way annotation for future work.

During annotation, the annotators looked at whether the answer is conceptually correct. They annotated the answer as correct if it fully covers the expectation. The partially correct answers are annotated as incorrect. The notable point is that the annotators annotated the answers based on their correctness without looking at the linguistic features. In this case, even if the answer is highly overlapped or lexically similar to reference answer(s) but conceptually different, the answer is marked as incorrect. On the other hand, even if the answer is not lexically similar with any of the reference answers but conceptually correct, the answer is annotated as correct. However, the answer is marked as incorrect if it does not address the question but the concept present in the student response is true in the context of problem or domain. Additionally, the annotators looked at the context whenever needed and made decisions accordingly. For example, the co-references were resolved in mind and the student answer is annotated

as such. To do so, the problem description and previous dialogue utterances were given to the annotator for their reference.

The examples showing the annotation instances where the answer is correct (in Example 4.1) and answer is incorrect (in Example 4.2) are shown below.

Example 4.1: Correct answer (#DTSU041_FF_LV02_PR02.sh)¹

Problem Description: A basketball player is dribbling a basketball (continuously bouncing the ball off the ground).

Question: Because the ball's velocity is upward while the ball is moving upward and its acceleration is downward, what is happening to the ball's velocity?

Student Response: SLOWING DOWN

Reference Answers:

- The ball's velocity is decreasing.
- Since the ball's velocity is upward and its acceleration is downward, the ball is slowing down.
- The ball is slowing down at a constant rate.
- Since the ball's acceleration is in the opposite direction of its velocity, the ball is SLOWING DOWN.
- Since the ball's acceleration is in the opposite direction of its velocity, the ball's velocity is decreasing.

Example 4.2: Incorrect Answer (#DTSU021_FM_LV04_PR10.FCI-16)

Problem Description: To rescue a child who has fallen down a well, rescue workers fasten him to a rope, the other end of which is then reeled in by a machine. The rope pulls the child straight upward at steady speed.

Question: How does the amount of tension in the rope compare to the downward force of gravity

¹ An image is accompanied with the problem description which is not shown in the examples.

acting on the child?

Student Response: *the tension in the rope is greater than the downward force of gravity*

Reference answers:

- The amount of tension in the rope is the same as (equal to) the magnitude of the downward force of gravity.
- Gravity and tension are balanced.
- Gravity and tension have equal magnitudes
- The amount of tension in the rope is the same as (equal to) the child's weight.
- The amount of tension in the rope is the same as (equal to) the magnitude of the downward force of gravity.

Inter-Annotator Agreement

The 100 instances were annotated separately by two annotators A1 and A2 (annotators were graduate students) and agreement was measured. The disagreement was resolved by the domain expert. Table 3 shows the confusion matrix for inter-annotator agreement based on the annotation of 100 instances of DeepEval data. The agreement measured as Kappa was 0.83 (almost perfect agreement; Viera, & Garrett, 2005). As the agreement was high enough the rest of the data was annotated by a single annotator.

Table 2: Confusion matrix for inter annotator agreement (100 instances)

	A2 (1)	A2 (0)
A1 (1)	39	5
A1 (0)	3	53

4.3 Statistics

The Table 3 shows the summary statistics of DeepEval dataset. As described in Section 4.2 (Data collection process), a subset of task solving dialogues were extracted from the collection of all student-tutor interactions recorded in the experiment. However, the sampling was done in such a way that all 41 students and 18 different tasks are represented in the DeepEval dataset.

Table 3: Summary statistics of DeepEval dataset

Parameter	Value
Total number of students	41
Total number of tasks	18
Number of tasks solved by each student	9
Number of task solving dialogues	198
Total number of instances (question-answer pairs)	618
Number of correct answers	358 (57.92%)
Number of incorrect answers	260 (42.07%)
Average number of reference answers per question	9
(The list of reference answers for the very first question includes reference answers for all expectations which is making the average number of reference answers high)	
Average number of words in problem description	25.96
Average number of words in questions	15.77
Average number of words in student answers	14.93
Average number of words in expected answers	17.07

4.4 Summary

In this section, we described DeepEval dataset including the statistics and collection process. In summary,

- We extracted question-answer pairs and reference answers from the records of DeepTutor experiment conducted in summer 2014 with high school.
- Answers were annotated as correct or incorrect irrespective of lexical similarity with the reference answer.
- Total instances 618.
- The dataset represents 18 problems and 41 students' writings.

5 Experiments and Results

This section describes the various experiments performed in different settings and their results. As mentioned in the framework design section (in Section 3), various factors should have contributed to the final performance of an automated grading model. So, the experiments were designed to look at the contribution of the main components and also see the overall improvement in answer grading as compared to the standard text-to-text similarity methods. For baseline, we use the results produced by applying optimal word-to-word alignment based text similarity method (Rus & Lintean, 2012). The dataset we used is DeepEval data which is described in Section 4. The answers were classified either as correct or incorrect and results are presented in terms of precision, recall, F-score, and kappa. The results are promising.

5.1 Experiments

5.1.1 Experiment Design

We first implemented various components as described in their corresponding sections in Section 4. We applied baseline method which is optimal word to word similarity based method (Rus and Lintean, 2012), a typical text similarity method. The same basic preprocessing steps were performed for the baseline method as well. The word to word similarity methods applied to the baseline are: WordNet based methods LIN (for verbs and nouns), and LESK (for adjectives and adverbs).

We then compared the performance of our system with different combinations of features for scoring. We also compared the results with and without implementing certain important components, such as co-reference resolution and ellipsis handling, etc.

5.1.2 Preprocessing

The tokenization, POS tagging, parsing, lemmatization were performed using Stanford CoreNLP 3.4.1¹. When it comes to stop words (words that are present everywhere and are generally considered less important in NLP applications, such as auxiliaries, prepositions, etc.), we removed some of the words from the stop words list as they are very important to express or represent the concept. In another word, some of the words in commonly used stop words list were found important and avoided removing them as stop words. For example, *first*, *second*, *third*, etc. were in the stop words list but we removed them from the stop words list as they are important in the physics domain (Newton's first law and Newton's law are different). The words were not changed initially as they are useful in parsing and concept extraction. The stop words were removed only after extracting the concepts.

5.1.2.1 Spelling Correction

We used Jazzy² spell checker for spelling correction. Jazzy is a widely used edit distance based spell checker. However, the tool does not consider context and suggests a list of words in the descending order of probability. We have provided a domain dictionary (i.e. all the words present in the tasks, questions, and expected answers) as the main dictionary as well as the default general purpose dictionary of Jazzy. This was needed because some domain words that are not found in the general dictionary are not inadvertently changed and the possibility of a wrong word from the general dictionary being at the top in the suggestion list is high. For example, the suggestion for *frctional*

¹ Downloaded from <http://nlp.stanford.edu/software/corenlp.shtml>

² <http://jazzy.sourceforge.net/>

(intentionally left misspelled) is *fractional* and *frictional* is not in the suggestion list as it was missing in the general purpose dictionary. Now, after adding domain words, *frictional* appears in the suggestion list.

But still sometimes it suggested unsuitable words as the most probable word keeping the contextually fit word lower in the list. For example, *fractional* came before *frictional* for the misspelled word *frctional* but *frictional* is contextually fit. To address this issue, we first find the most probable list of words using Jazzy and check them in the domain dictionary. Whichever is found first in the domain dictionary, we use that word. Otherwise, the one suggested by the Jazzy as the most probable one is used.

Table 4: Improvement in spelling correction after considering the context

Parameter	Value ³
Total number of student responses checked	2277
Total number of tokens	27864
Total number of responses with spelling mistakes suggested by the spell checker.	1000 (43.91%)
Tokens with spelling mistake (as suggested by spell checker)	1343 (4.81%)
False positive (spell checker suggested as spelling error but it was not, i.e. missing in the dictionary)	69
Accuracy of spell checker after adding domain dictionary	61.53%
Accuracy of spell checker after selecting the contextual fit word (by going down up to seventh word in the suggestion list)	76.28%

³ These values were generated from bigger corpus (from where DeepEval data were extracted). The data included short answers (keywords) as well.

There is about 15% increment in the spelling correction after considering the contextually fit words. The alternative approach would be to apply language model based techniques, but we did not explore on collecting domain specific resources to develop the model. However, the framework allows integrating an updated component (if any) easily.

5.1.2.2 *Extracting Merged Words*

Some words are merged when students forget to type space in between. For example, *netforce*, *externalforce*, *eachother* (intentionally left misspelled) showed up in the student response as *net* and *force*, *external* and *force*, and *each* and *other* are merged together as typing errors. We have found fourteen unique composite tokens in our dataset. To break them up, from the beginning of the token, we take the subset of characters and look up the dictionary. If the first part and the rest are both found in dictionary, it means that these two valid words formed the composite word. Even if a single word is found valid, we used that word in place of unknown word. We have now assumed that maximum two valid words form the composite token but the technique is applicable for different number of possible words. The spelling checker usually does not suggest any correction for such tokens because the composite word forms a very bizarre token. So, the process is to first try to correct the spelling whenever an unknown word or token appears, and then if the spelling correction cannot suggest a valid word, it tries to find or break into valid words (if any) in it.

5.1.2.3 *Data Normalization*

The irregularities in the text make comparison difficult. For example, 2nd and second are same thing but one could write either 2nd or second. Similarly, abbreviations, phrasal words, etc. could be used. For instance, the phrase "*come apart*" can be replaced

with "*separate*". One could argue that the current tools should be able to handle those variations in writing but still the challenge is to incorporate such automatic handling methods. Normalization can be done in different levels, such as replacing phrases with a single word etc. However, the phrase extraction is still an unsolved problem. We have created a lookup table to accurately replace a token with more standard word or set of words. It is performed for question, student answer, and reference answers. For example: 1st – first, 2nd – second, 0 – zero, etc. There are only few entries at this moment.

5.1.3 Speech-Act Classification

The speech act of the student response is obtained using a speech act classifier (Moldovan, Rus, & Graesser, 2011). This classifier considers four broad speech act categories: contribution (a student response rich in target content), meta-cognitive ("*I don't know*"), meta-communicative ("*I already said that.*"), and question. Only the contributions were included in the DeepEval dataset. We did not analyze the performance of the speech-act classifier.

5.1.4 Negation Handling

No and Not (in the form of *Not*, *not*, *n't*, *NOT*) are the most frequently negation cues found in DeepEval corpus. These are the most frequent negation cues in other domains also (Konstantinova, de Sousa, Díaz, López, Taboada, & Mitkov, 2012). For example, Konstantinova et al. (2012) found that *not*, and *no* were the most frequent negation cues in product review corpus (*not* and *no* appeared 40.23%, and 14.85% respectively). Our focus is to handle them. Though machine learning algorithms, such as CRF (Conditional Random Field) are widely used for negation scope and focus detection and have achieved great success, the interpretation seems to be difficult as they can give

discontinuous scope. In that case, we need to apply some heuristics to decide what to do if some words in the concept, and/or some of the concepts in a phrase (true scope) are labeled as out of scope. Instead, we have directly applied the rules (Rule 1 and Rule 2 shown below) to find the scope of the two negation cue words: No and Not.

Rule 1

If *no* is present as a determiner in front of the noun or adjective phrase, it is replaced with *zero*. For example, there is *no net force* is changed to there is *zero net force*. This replacement is done in student response, question, and the reference answers as well.

Rule 2

If *not* is present, the clause where it presents is treated as its scope. This is a typical annotation rule found in negation annotation guidelines. The only difference is that human can identify the clause boundary whereas it is difficult for machine. However, the student writing is either short or is more straightforward as opposed to cynical or literature style texts. We treat the end or beginning of sentence, coordinating conjunctions, and certain prepositions as clause boundaries. The words in the scope are marked as in-scope. This is used while calculating the expectation coverage score described below (Section 5.2).

5.1.5 Concept Extraction

As discussed in section 3.3, our generalized assumption is that the syntactic constituents are the actual manifestation of the semantic constituents of the sentence. We extract chunks from the text by using Stanford Parser 3.4.1. Chunks are the meaningful groups of syntactically related words. We present principal chunks (chunks that we

consider are important) as concepts. To extract those chunks, we apply shallow parsing or text chunking approach which consists in dividing a text into phrases or chunks so that syntactically related words become members of the same phrase. The example 5.1 illustrates the concepts extracted from the text. The chunks are presented in the format such that the type of chunk is given first (such as NP) and it is followed by a group of words along with their lemma, POS tag, and whether the chunk is in the scope of some negation cue (Y - the chunk is in the scope, N – otherwise).

Example 5.1: Extracting concepts from the text

Sentence: *The net force is greater than the frictional force.*

Chunks: *[NP the/the/DT/N net/net/JJ/N force/force/NN/N] [VP is/be/VBZ/N] [ADJP greater/greater/JJR/N] than [NP the/the/DT/N frictional/frictional/JJ/N force/force/NN/N]*

Refined chunks: *[NP net/net/JJ/N force/force/NN/N] [ADJP greater/greater/JJR/N] [NP frictional/frictional/JJ/N force/force/NN/N]*

We first parse the text using Stanford Parser 3.4.1, and starting from the parse tree, our algorithm extracts the principal syntactic constituents of the sentence, considering all noun and adverbial phrases of maximum length, as long as there is no change in the type of the phrase. Thus, from an annotation such as (NP1 (NP2 ...) (NP3 ...)), our algorithm would select NP1 as a principal chunk, while from an annotation like (NP1 (NP2 ...) (PP (...) (NP3 ...))), NP2 and NP3 would be considered principal chunks. Each verb is considered a singular verb phrase (VP), but the auxiliaries are removed. This approach is similar to our work on chunk extraction for paraphrase identification (Stefanescu, Banjade, & Rus, 2014). By extracting chunks ourselves rather than using

some other tools gives us more control over the type of concepts we want (now and in the future) and the texts annotation (POS tags, information about whether the word is in the scope of negation, etc.).

In this current version of our algorithm, we decided not to consider prepositions and complementizers (e.g. if, although, while, even though, in case, so that), even if though they may have their own contribution to the meaning of a sentence. Nevertheless, for the purpose of computing sentence similarity, we believe that their role is not crucial. We also got rid of any existing annotations representing punctuations.

5.1.6 Implicit Coreference Resolution and Ellipsis Handling

As discussed in Section 3.3, co-reference should be resolved and students' elliptical responses should be completed to fully assess the response. But both phenomena are extremely hard problems to solve accurately. To address these two problems, we have applied an indirect approach (as illustrated in the Figure 3) which is found very helpful (please see Section 5.4 for the results).

Problem description (FM_LVxx_PR02.sh): A rocket is pushing a meteor with constant force. At one moment the rocket runs out of fuel and stops pushing the meteor. Assume that the meteor is far enough away from the sun and the planets to neglect gravity.

Question: How will the meteor move after the rocket stops pushing?

Student response (DTSU040): it will move at a constant speed

Reference answer: When the rocket stops pushing, no forces are acting on the meteor anymore and therefore will move with constant velocity in a straight line

Concepts in student answer:

- a1. [VP moves/move/VBZ] – q2, r7
- a2. [NP constant/constant/JJ speed/speed/NN] – r8

Concepts in Reference answer:

- r1. [NP rocket/rocket/NN] – q3
- r2. [VP stops/stop/VBZ] – q4
- r3. [VP pushing/push/VBG] – q5
- r4. [NP zero/zero/CD forces/force/NNS]
- r5. [VP acting/act/VBG]
- r6. [NP meteor/meteor/NN] – q1
- r7. [VP move/move/VB] - a1, q2
- r8. [NP constant/constant/JJ velocity/velocity/NN] – a2
- r9. [NP straight/straight/JJ line/line/NN]

Concepts in question:

- q1. [NP meteor/meteor/NN] – r6
- q2. [VP move/move/VB] – r7
- q3. [NP rocket/rocket/NN] – r1
- q4. [VP stops/stop/VBZ] – r2
- q5. [VP pushing/push/VBG] – r3

Figure 3: Image showing the concepts mappings among student answer, question, and reference answer.

The process is as follows.

- a. Do not give any credit if the concept in student response matches (maps) with the concept in question.
- b. Do not give any weight to concept in reference answer if it is present in the question itself irrespective of whether the concept is present in the student answer. Even if the concept is present in the student response, no credit is given for that concept.

- c. From the remaining concepts in student response and reference answer, the expectation coverage score is calculated.

How does it address ellipsis handling and co-reference resolution?

Research has shown (Niraula, Rus, Banjade, Stefanescu, Baggett, & Morgan, 2013) that most of the time the student co-refers (if any) an entity in the answer itself (31.54% of the times) or an entity in the question (53.22% of the time). They analyzed pronominal referents only. Our hypothesis is that not only pronominal references but also other entities co-referred by the student are found most of the times in the answer itself or in the question. For example, students mention *them* to refer to *forces*, *laws* etc. If the entity is in the answer itself, there is possibility of mapping that entity with an entity in the reference answers. On the other hand, if the answer co-refers an entity in the question and that entity is found in the reference answer, the entity in the reference answer would map with the question. So, it effectively works as co-reference resolution.

Similarly, if the concept present in the question is present in expected answer also but it is missing in the student answer, then by aligning the concept in question with the concept in reference answer would effectively work as aligning the concept in answer and the reference answer. Thus it effectively makes the student utterance more complete.

5.2 Scoring Model Development

We built a two-way scoring model. In this two-way scoring model, the answer is either marked as correct (1) or incorrect (0).

5.2.1 Features

We selected the following features:

1. Expectation coverage score (ECS)

2. Presence of contradicting concept(s) in student response (PCC)
3. Uncovered concepts in the reference answer (UCRA):
4. Uncovered concepts in the student answer (UCSA)

Expectation Coverage Score (ECS)

The Expectation Coverage Score (ECS) is the normalized concepts coverage score for the most covered reference answer, explained shortly. It quantifies how much of the reference answer is covered by the student answer. Concept coverage score is calculated for each reference answer and the most covered reference answer is chosen. To calculate the concepts coverage score for a specific reference answer, concepts in the reference answer and the student answer were aligned optimally (with and without discarding concepts found in the question; we tried both cases and results are shown in Table 7) based on the concept to concept similarity. Similarly, words in the concepts were aligned optimally to calculate the concept-to-concept matching score.

Word to word similarity

The words are deemed similar when their similarity score is above a certain threshold. We experimented with many word-to-word similarity methods that are available in SEMILAR⁴ tool (Rus, Lintean, Banjade, Niraula, & Stefanescu, 2013). In addition, we applied word-to-word similarity models developed by others too. Since we need to find whether the answer matches with reference answer, the relatedness measures are not very helpful as they do not distinguish the highly similar and highly related words. For example, *velocity* and *acceleration* are highly related but they are not similar (i.e. one cannot be substituted by other). So, we experimented first with a similarity corpus

⁴ Available for download at <http://semanticsimilarity.org>

Simlex-999 (Hill, Reichart, & Korhonen, 2014) to identify the better word-to-word similarity method.

We tested following methods.

- WordNet based methods (Pedersen, Patwardhan, & Michelizzi, 2004): **WNLin**, **HSO**, **Lesk**, **WUP**, **Path**, and **Res**.
- **LSAWiki**, **LSATasa**: Similarities based on LSA model developed from the whole Wikipedia articles and TASA corpus (Rus et al., 2013; Ștefănescu, Banjade, & Rus, 2014).
- **CRDE**⁵: Similarity using vectors generated using Deep Learning technique proposed by Collobert and Weston (2008) and reproduced by Turian, Ratinov, and Bengio (2010).
- **UMBC**: Similarity calculated using UMBC system (Han, Kashyap, Finin, Mayfield, & Weese, 2013) WebService⁶.
- **WS**: Whether there is some synonymy relation in WordNet with each other.
- **WA**: Whether there is antonym relation in WordNet with each other.
- **MK-NLM**⁷: Neuro probabilistic language model based word representations developed by Mikolov, Chen, Corrado, and Dean (2013).
- **Glove-42B**: Word representation model proposed by Pennington, Socher, and Manning (2014) and trained on 42 billion words.

⁵ Downloaded from <http://metaoptimize.com/projects/wordreprs/>

⁶ <http://swoogle.umbc.edu/SimService/api.html>

⁷ The models were downloaded from <http://code.google.com/p/word2vec/>

Table 5: Correlation (measured as Pearson correlation) between similarity scores given by different methods and the human annotated scores in Simlex-999 corpus

SN	Method	Correlation (All/A/N/V)
	Hill et al. (2014)	0.414 (rho)
1	WNLesk	0.347/0.418/0.373/0.301
2	WNWup	NA/NA/0.471/0.246
3	WNRes	NA/NA/0.454/0.245
4	WNJcn	NA/NA/0.462/0.279
5	WNLin	NA/NA/0.462/0.289
6	WNPath	NA/NA/0.513/0.216
7	WNLch	NA/NA/0.534/0.109
8	WNHso	0.324/0.264/0.421/0.223
9	WN	0.362
10	LSATasa	0.251
11	LSAWiki	0.277
12	CRDE200	0.144
13	UMBC	0.557
14	GloVe-42B	0.400
15	Mk-NLM	0.453
16	Average1 (9-15)	0.520
17	Average2 (9, 13-15)	0.565

The correlations between human annotated scores and the similarity scores produced by various word-to-word similarity methods are presented in Table 5. The WordNet based methods start with *WN*. We used LSA models⁸ developed from TASA corpus and whole Wikipedia articles (Steafanescu et al., 2014). The Deep Learning based word embeddings (representations) published by Turian et al. (2010) were used. We used 200 dimension word representations available in their website⁹. From the available vectors, we calculated cosine similarity score for the given word pair. Please note that the similarity scores calculated from the word embeddings were poorly correlated with the

⁸ The models are available for download at SEMILAR website (<http://semanticsimilarity.org>)

⁹ Word vectors (embeddings) downloaded from <http://metaoptimize.com/projects/wordreps/>

human annotated scores. We also took the average of the scores generated from different methods.

We linearly combined these results obtained from various similarity methods by applying linear regression technique (R) and support vector machine regression in weka (S) and support vector regression with RBF kernel in LibSVM¹⁰. The 10-fold cross validation results (Pearson correlation with human judgment scores) using different combinations is shown in Table 6.

Table 6: Results (Pearson correlation and Root Mean Square Errors) after combining scores obtained from different methods. The effective features are enclosed in bracket.

Regression Method (features)	Correlation	RMSE
R1: WA,WS, 9-16 (9-15)	0.634	0.202
R2: WA, WS, 9-12,14-15 (WA,WS, 9, 14-15)	0.601	0.209
S1: WA, WS, 9-15	0.630	0.203
S2: WA, WS, 9-12,14-15	0.586	0.211
LS1: WA,WS, 9-15	0.657	0.197
LS2: WA, WS, 9-12, 14-15	0.621	0.205

Concept- to- Concept Similarity

The concept to concept similarity is calculated using the optimal word alignment technique. The optimal alignment aims at finding the best overall matching based on the similarity values of words using the efficient Hungarian algorithm. The assignment problem is one of the fundamental combinatorial optimization problems and consists of finding a maximum weight matching in a weighted bipartite graph. Rus et al. (2012) compared greedy and optimal word alignment based similarity methods. We use their technique implemented in the SEMILAR library to compute concept similarity scores. SEMILAR includes the Hungarian algorithm for optimal alignment.

¹⁰ <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

Calculating Expectation Coverage Score

Finally, the concepts in the reference answer were optimally matched with the concepts in the student answer (similar to the optimal word-to-word alignment in two concepts). The similarity score is normalized by the number of concepts in the reference answer. All the concepts are weighted equally (at this moment).

Presence of Contradicting Concept (PCC)

The presence of contradicting concept(s) or disjoint set concept(s) in the student answer is (are) a sign that the student answer contradicts (parts of) the expected answer. Even a single contradicting concept can nullify the concept coverage score. For instance, in Example 1.1, the concept *increasing* in the student answer is contradicting the expected concept *constant*. The answer is incorrect although the majority of the expected concepts are present in the student response. So, our method checks whether there is any uncovered concept in the expected answer which is contradicting to an uncovered concept in the student answer.

If any one of the words in the concept contradict with a word in the another concept, they are treated as contradicting. For example, *increasing velocity* and *decreasing velocity* are contradicting concepts.

The similarity methods that we have access to give similarity score in the range of 0 to 1. Sometimes they give high score for related but dissimilar words too. For this reason, we created a dissimilarity method based on WordNet. For the given two words, their morphologically varying words and synonyms are retrieved from WordNet. And then, for each word group, words that hold antonym relations with them are collected. If any word in the first word synonym group is found in the antonym group of second word

or vice versa, the dissimilarity is 1. Otherwise, the dissimilarity score is set to 0. We use JAWS tool to query WordNet (version 3.0). If any of the unaligned concepts in the student answer contradicts with an unaligned concept in the expected answer, the flag is turned on (i.e. the value of this feature is set to 1).

Uncovered Concepts in the Reference Answer (UCRA)

It is the number of uncovered concepts in the reference answer normalized by the total number of concepts in it.

Uncovered Concepts in the Student Response (UCSR)

It is the number of uncovered concepts in the student response normalized by the number of concepts present in the response itself. Only the domain specific concepts are considered to calculate this score.

Classification

Finally, a logistic model is trained and evaluated using 10-fold cross validation. The logistic function classifies the answer either as correct or incorrect (similar to the human judgment) based on the features. We used Weka¹¹ tool to fit the logistic function.

5.2.2 Learning Similarity Threshold Values

Ideally the word-to-word and concept-to-concept similarity values should be 1 to consider them perfectly matching. However, the similarity methods can assign a very small value even though words or concepts convey the same (or almost same) meaning. The similarity calculation methods we used do not look at the context and the range of scores also depends on the inherent method of calculating the similarity score. For this reason, we may need to align words or concepts whose similarity score is not 1. Since it is difficult to learn the threshold by manually assigning the similarity scores for the word

¹¹ <http://www.cs.waikato.ac.nz/ml/weka/>

pairs and comparing with the calculated scores, we learnt them indirectly. We tried a range of threshold values (0.4 to 1.0 incremented at the rate of 0.05) for word-to-word similarity and concept-to-concept similarity, and calculated the performance of the system for answer assessment. We applied ten-fold cross validation process to find out the threshold values.

5.3 Evaluation Metrics

Since we have human annotated data - 1 (correct) and 0 (incorrect), the output given by the system (the system gives similar format output) is evaluated by calculating the agreement between human and the system in terms of accuracy, precision, recall, F1-score, and kappa.

5.4 Results

The Table 7 presents the scoring results obtained by 10-fold cross validation and 70/30 split (using Weka tool) in terms of accuracy, weighted precision, recall, F1-measure, and kappa values. The baseline method is described in experiment design section (i.e. section 5.1.1). The results obtained by applying **DeS** word-to-word similarity method, with Implicit Co-reference resolution and Ellipsis handling (**ICE**; No-ICE for without it) and feature set **A** (All 4 features - ECS, PCC, UCRA, and UCSA) has been presented as DeS_ICE_A. Similarly, **B** represents a subset of features - ECS, and PCC, and **C** includes just ECS.

Table 7: Scoring results using DeS word metric (The results obtained by 10-fold cross validation (first) and 70/30 split are separated by /).

Experiment	Accuracy	Precision	Recall	F1-Score	Kappa
Baseline	67.31/66.48	66.90/67.40	67.30/66.50	66.80/65.00	31.32/30.31
No-ICE_A	66.34/63.24	66.00/64.10	66.30/63.20	64.80/61.00	27.61/23.09
No-ICE_B	65.04/61.62	64.40/62.00	65.00/61.60	63.90/59.40	25.41/19.78
No-ICE_C	61.97/61.62	61.20/62.60	62.00/61.60	61.10/58.70	19.61/19.27
ICE_A	70.87/70.27	70.60/70.30	70.90/70.30	70.50/69.90	39.07/39.24
ICE_B	70.87/69.18	70.60/69.20	70.09/69.20	70.60/68.80	39.27/37.03
CE_C	69.25/69.18	68.90/69.40	69.30/69.20	68.70/68.50	35.40/36.65

The results obtained by implicitly handling the Co-reference resolution and Ellipsis (ICE) with feature set B (ECS, and PCC) are better than the results obtained without ICE and the baseline. We did 10-fold cross validation because the annotated dataset is not huge.

The results presented in Table 7 were obtained by using strict similarity method (called DeS) which looks at the synonym relations in WordNet (see Section 5.2.1).

Similarly, we applied linear combination of similarity methods that performed very well in Simlex-999 (see Table 6). Due to simplicity and based on the availability of resources including similarity models, we applied linear regression that comprises of WN, MK, and Glove-42B (represented as R2 in Table 6). The grading results obtained by applying regression model for word to word similarity is presented in Table 8.

Table 8: Scoring results obtained by using linear combination of word metrics (The results obtained by 10-fold cross validation (first) and 70/30 split are separated by /).

Experiment	Accuracy	Precision	Recall	F1-Score	Kappa
No-ICE_A	67.63/63.78	67.30/65.20	67.60/63.80	66.50/61.20	30.94/23.98
No-ICE_B	64.88/61.08	64.30/61.30	64.90/61.10	63.80/58.90	25.27/18.73
No-ICE_C	63.75/61.62	63.00/62.40	63.80/61.60	62.40/58.90	22.39/19.44
ICE_A	70.87/70.81	70.60/70.80	70.90/70.80	70.60/70.50	39.40/40.41
ICE_B	70.55/69.19	70.30/69.10	70.60/69.20	70.40/68.90	38.99/37.16
ICE_C	69.74/70.27	69.40/70.40	69.70/70.30	69.40/69.80	36.87/39.12

There is no significant difference between the results obtained by applying strict synonym based method and results by applying the linear combination of methods (Table 8). It seems that it is helpful to align words only when their similarity is very high.

6 Discussions

In this section we discuss on the results we obtained and various factors affecting the performance of automatic assessment system based on the analysis of student responses and experimental results.

Our system (DeepEval) performed significantly better than the baseline system (see Table 7 and Table 8). We achieved an F1-score of 70.60 and Kappa of 39.27 which are both better compared to the baseline system (F-score 67.31, Kappa 31.32). To simply put, the output of our system is better correlated with human judgment (the correlation is fair; Viera, & Garrett, 2005). The difference between DeepEval and the baseline system is the concept-based representation which is at a more specific level compared to using words as the basic unit of meaning representation, and implicit co-reference resolution and ellipsis handling. All the preprocessing steps were common in both experiments. This shows that systematically addressing the various issues and linguistic phenomena improves the performance of assessment models.

However, for various reasons we have left many issues untouched or unaddressed partially but we analyzed some of them as it is important to find out the sources of errors to guide future developments. We have not quantified the effect of each factor but we discuss here some of them.

Requiring Inference

Sometimes students give very abstract or vague answer which is correct (or incorrect) but the system cannot judge whether the student answer implies the expected answer (or a misconception). For instance, students may give an abstract answer when the question is asking for a more concrete (or specific to the problem) answer. In cases

where a student is expected to articulate Newton's second law ("*net force equals mass times acceleration*") and the student articulates *acceleration equals net force divided by its mass*, then the system should infer that the second form is same (in meaning) reference statement of Newton's second law (*net force equals mass times acceleration*).

Referring Visual Items

In addition to textual descriptions of problems, the visual illustrations (images or videos) are very effective ways of presenting problems or observations. But the students might give answers based on those visual illustrations. For example, instead of saying *east* or *west*, students might say *towards the man in the picture*.

Contextual Importance of Words or Concepts

The importance of word changes from place to place. For example, the word 'only' in the answer '*gravitational force is the only force acting on the object*' specifically indicates that there are no other forces acting on the object. However, in many places the word 'only' is considered as stop word. Given the short context of the problem description, it is very hard to measure the importance of words or concepts. We assumed that the words or concepts present in the reference answers prepared by experts are equally important. Otherwise, the subject expert would not expect those answers. However, in the student response, if student says something more, we have to measure the importance of words or concepts. If something important is found in the student answer but is not in the expected answer, they should be handled properly. We can ignore the unimportant concepts.

Extraneous Information

Evaluating the extraneous content in student response is difficult as that may be the correct explanation of the answer (which is not expected), expressing misconception, etc. Suppose the correct answer for a question requires *normal force* and *gravitational force* but student writes *normal force, gravitational force, and friction force*. The system should know that the extra information indicates the student has not properly solved the problem, i.e. the answer is in this case correct and incorrect depending on how strict one wants to be.

Detecting Misconceptions

Detecting whether student has misconception or not is very important as tutor has to rectify those misconceptions. Ideally, a misconception is a conclusion that cannot be derived given the domain and the problem at hand. But if such a solution would require automated reasoning capabilities and a knowledge base with world and domain knowledge. In a semantic similarity approach, it is impossible to track all the possible misconceptions. However, detecting a misunderstanding of a student that is explicitly expressed in the form of contradiction with the reference answer was pursued in this thesis (Section 5.2). For example, when the correct answer is *increasing velocity* but student says *decreasing velocity*. Moreover, those concepts in the student answer may not directly contradict but be in a disjoint set. For instance, instead of saying *decreasing velocity*, student can say *constant velocity or same velocity as before*. The *constant velocity* is not completely opposite of *increasing velocity* but it is as incorrect as saying *decreasing velocity*. The current resources are not sufficient in finding the disjoint concepts. In fact, they are more domain specific issues.

Different Linguistic Phenomena

There are a myriad of ways students express their responses for the same question and different linguistic phenomena are present in their responses. A particular example where student used a symbol is: *NET FORCE=ZERO* (same as *net force is zero*). Since natural language processing is still a growing field of research, these all linguistic phenomena are not solved (or have not achieved significant success).

Contextual or Domain Dependent Interpretations

Many things are understood in context or based on that particular domain. For example, in general *speed* and *velocity* are used interchangeably. But in physics, they are two different things; *velocity* is a vector having both magnitude and direction whereas *speed* is a scalar. The prior knowledge and the assumptions are also important. Almost everywhere, the contextual and domain specific information is required to properly address various issues.

Error Propagation

Since various components are required to process the student response as a sequence of steps, the possibility of propagating errors introduced in different steps is also high. For example, the *meteor* is a noun but when a student writes *meteor motion*, the POS tagger tagged *meteor* as an adjective (similar to the *slow* in *slow motion*).

In addition to the problems discussed above, there are some issues where some improvement is required in order to improve the accuracy of system. The issues we discuss below may not be the issues in human tutoring or may not apply in other datasets. However, they are potential problems of any system and they also justify (partially) our results.

Multiple Questions in the Same Utterance

There are cases where two questions are asked together in the same utterance (for example, *which Newton's law is applicable and why*) but the expected answer is in the combined form. Student can give correct answer to any one of them or both but it hard to evaluate as the reference answer is in the merged form.

Missing a Correct Form of Answer in the Reference Answer Set

It is very difficult (or impossible) to prepare a comprehensive list of reference answers. However, the chances are that all the commonly written forms are not present in the reference answers list so that it is hard to evaluate them correctly. To alleviate this issue, distinct student answers should be evaluated by experts and added in the reference answer list (if needed).

Seeking Explanations

Sometimes the tutor expects explanations even though the question is not explicitly asking for. For example, in a case where tutor asks '*Which Newton's law is applicable in this situation?*' but only some of the students give the reason why while others just mention a Newton's law. On the other hand, the reference answers contain explanations too.

More Open Ended Questions

Questions such as asking about the motion of the ball (*what about the motion of the ball?*) are more open ended. There are multiple possible ways of thinking about the motion. One might think about velocity, speed (magnitude), acceleration, direction of movement, etc.

7 Conclusions and Future Work

As part of this thesis work, we thoroughly analyzed tutor-student interactions recorded during a DeepTutor experiment and built an end-to-end automatic short answer evaluation system (DeepEval) for intelligent tutoring systems. We implemented various important components where each of them handles a specific linguistic phenomenon. Our system performed significantly better compared to the baseline system which is a typical semantic similarity calculation method. The performance improvement is a result of contributions of each component. This shows that systematically addressing various issues and linguistic phenomena improves the performance of assessment models.

However, due to various reasons we did not address all of the issues we identified (discussed in Section 3.5 and Section 6). In the future, we intend to improve the framework by evaluating different approaches applicable in each level and integrating the improved components in the framework. Also, we would like to explore on diagnostic feedback generation and follow up questions generation.

References

- Aggarwal, V., Minds, A., Srikant, S., & Shashidhar, V. (2014). Principles for using Machine Learning in the Assessment of Open Response Items: Programming Assessment as a Case Study.
- Attali, Y., Powers, D., Freedman, M., Harrison, M., & Obetz, S. (2008). Automated scoring of short-answer open-ended GRE subject test items. Retrieved from <https://www.ets.org/Media/Research/pdf>.
- Bailey, S., & Meurers, D. (2008, June). Diagnosing meaning errors in short answers to reading comprehension questions. In *Proceedings of the Third Workshop on Innovative Use of NLP for Building Educational Applications* (pp. 107-115). Association for Computational Linguistics.
- Baker, R.S.J.d., D'Mello, S.K., Rodrigo, M.M.T., & Graesser, A.C. (2010). Better to Be Frustrated than Bored. *The International Journal of Human-Computer Studies*, 68(4), 223-241
- Baker, R.S.J.d., Goldstein, A.B., & Heffernan, N.T. (2011). Detecting Learning Moment-by-Moment. *International Journal of Artificial Intelligence in Education*, 21(1-2), 5-25.
- Basu, S., Jacobs, C., & Vanderwende, L. (2013). Powergrading: a Clustering Approach to Amplify Human Effort for Short Answer Grading. *TACL*, 1, 391-402.
- Beevers, C. E., & Paterson, J. S. (2003). Automatic assessment of problem-solving skills in Mathematics. *Active Learning in Higher Education*, 4(2), 127-144.
- Bethard, S., Hang, H., Okoye, I., Martin, J. H., Sultan, M. A., & Sumner, T. (2012, June). Identifying science concepts and student misconceptions in an interactive essay

- writing tutor. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP* (pp. 12-21). Association for Computational Linguistics.
- Bhagat, R., & Hovy, E. (2013). What Is a Paraphrase? *Computational Linguistics*, 39(3), 463-472.
- Burstein, J. (2003). The E-rater® scoring engine: Automated essay scoring with natural language processing.
- Butcher, P. G., & Jordan, S. E. (2010). A comparison of human and computer marking of short free-text student responses. *Computers & Education*, 55(2), 489-499.
- Chaudhri, V. K., Goldenkranz, A., Fikes, R., & Seyed, P. (2011, June). What is hard about representing biology textbook knowledge. In *Proceedings of the sixth international conference on Knowledge capture* (pp. 185-186). ACM.
- Choi, J. I., & Hannafin, M. (1995). Situated cognition and learning environments: Roles, structures, and implications for design. *Educational Technology Research and Development*, 43(2), 53-69.
- Clark, P., Harrison, P., Balasubramanian, N., & Etzioni, O. (2012, June). Constructing a textual KB from a biology TextBook. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction* (pp. 74-78). Association for Computational Linguistics.
- Collobert, R., & Weston, J. (2008, July). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning* (pp. 160-167). ACM.

- Conati, C., & Maclaren, H. (2009). Empirically building and evaluating a probabilistic model of user affect. *User Modeling and User-Adapted Interaction*, 19(3), 267-303.
- Corcoran, T., Mosher, F.A., & Rogat, A. (2009). Learning progressions in science: An evidencebased approach to reform. Consortium for Policy Research in Education Report #RR-63. Philadelphia, PA: Consortium for Policy Research in Education.
- Dale, R., Anisimoff, I., & Narroway, G. (2012, June). HOO 2012: A report on the preposition and determiner error correction shared task. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP* (pp. 54-62). Association for Computational Linguistics.
- Dale, R., & Kilgariff, A. (2011, September). Helping our own: The HOO 2011 pilot shared task. In *Proceedings of the 13th European Workshop on Natural Language Generation* (pp. 242-249). Association for Computational Linguistics.
- Dzikovska, M. O., Nielsen, R. D., Brew, C., Leacock, C., Giampiccolo, D., Bentivogli, L., ... Dang, H. T. (2013). *SemEval-2013 task 7: The joint student response analysis and 8th recognizing textual entailment challenge*.
- Evens M., & Michael J. (2006). *One-on-One Tutoring by Humans and Computers*. Lawrence Erlbaum Associates.
- Fader, A., Zettlemoyer, L., & Etzioni, O. (2014). Open Question Answering Over Curated and Extracted Knowledge Bases. *KDD workshop*.
- Fan, J., Kalyanpur, A., Gondek, D. C., & Ferrucci, D. A. (2012). Automatic knowledge extraction from documents. *IBM Journal of Research and Development*, 56(3.4), 5-1.

- Ferrucci, D., Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A. A., ... Welty, C. (2010). Building Watson: An overview of the DeepQA project. *AI Magazine*, 31(3), 59-79.
- Franzke, M., & Streeter, L. (2006). Building student summarization, writing and reading comprehension skills with guided practice and automated feedback. Highlights From Research at the University of Colorado, A white paper from Pearson Knowledge Technologies.
- Friedland, N. S., Allen, P. G., Matthews, G., Witbrock, M., Baxter, D., Curtis, J., ... Clark, P. (2004). Project halo: Towards a digital aristotle. *AI Magazine*, 25(4), 29.
- Gabrilovich, E., & Markovitch, S. (2007, January). Computing Semantic Relatedness Using Wikipedia-based Explicit Semantic Analysis. In *IJCAI* (Vol. 7, pp. 1606-1611).
- Golding, A. R., & Roth, D. (1999). A winnow-based approach to context-sensitive spelling correction. *Machine Learning*, 34(1-3), 107-130.
- Graesser, A. C., Lu, S., Jackson, G. T., Mitchell, H. H., Ventura, M., Olney, A., & Louwerse, M. M. (2004). AutoTutor: A tutor with dialogue in natural language. *Behavior Research Methods, Instruments, & Computers*, 36(2), 180-192.
- Graesser, A. C., McNamara, D. S., Louwerse, M. M., & Cai, Z. (2004). Coh-Metrix: Analysis of text on cohesion and language. *Behavior Research Methods, Instruments, & Computers*, 36(2), 193-202.
- Graesser, A. C., Wiemer-Hastings, P., Wiemer-Hastings, K., Harter, D., Tutoring Research Group, T. R. G., & Person, N. (2000). Using latent semantic analysis to

- evaluate the contributions of students in AutoTutor. *Interactive Learning Environments*, 8(2), 129-147
- Gunning, D., Chaudhri, V. K., Clark, P. E., Barker, K., Chaw, S. Y., Greaves, M., ... Tien, J. (2010). Project Halo update—progress toward digital Aristotle. *AI Magazine*, 31(3), 33-58.
- Han, L., Kashyap, A., Finin, T., Mayfield, J., & Weese, J. (2013). UMBC EBIQUITY-CORE: Semantic textual similarity systems. *Atlanta:GA*, (p.44).
- He, Y., Hui, S. C., & Quan, T. T. (2009). Automatic summary assessment for intelligent tutoring systems. *Computers & Education*, 53(3), 890-899.
- Hill, F., Reichart, R., & Korhonen, A. (2014). Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *arXiv preprint arXiv:1408.3456*.
- Jimenez, S., Becerra, C., Gelbukh, A., Bátiz, A. J. D., & Mendizábal, A. (2013, June). SOFTCARDINALITY: hierarchical text overlap for student response analysis. In *Second Joint Conference on Lexical and Computational Semantics (*SEM)* (Vol. 2, pp. 280-284).
- Jordan, P.W., Makatchev, M. & VanLehn, K. (2004).Combining competing language understanding approaches in an intelligent tutoring system. In *7th ITS*.
- Jordan, S., & Mitchell, T. (2009). e-Assessment for learning? The potential of short-answer free-text questions with tailored feedback. *British Journal of Educational Technology*, 40(2), 371-385.
- Konstantinova, N., de Sousa, S. C., Díaz, N. P. C., López, M. J. M., Taboada, M., & Mitkov, R. (2012, May). A review corpus annotated for negation, speculation and their scope. In *LREC* (pp. 3190-3195).

- Kulkarni, C. E., Socher, R., Bernstein, M. S., & Klemmer, S. R. (2014, March). Scaling short-answer grading by combining peer assessment with algorithmic scoring. In *Proceedings of the first ACM conference on Learning@ scale conference* (pp. 99-108). ACM.
- Kulkarni, C., Wei, K. P., Le, H., Chia, D., Papadopoulos, K., Cheng, J., Koller, D., & Klemmer, S. R. (2013). Peer and self-assessment in massive online classes. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 20(6), 33.
- Landauer, T. K. (2003). Automatic essay assessment. *Assessment in education: Principles, Policy & Practice*, 10(3), 295-308.
- Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. *Discourse Processes*, 25(2-3), 259-284.
- Leacock, C. (2004). Scoring free-responses automatically: A case study of a large-scale assessment. In *Examens*.
- Leeman-Munk, S. P., Shelton, A., Wiebe, E. N., & Lester, J. C. (2014). Towards Domain-Independent Assessment of Elementary Students' Science Competency using Soft Cardinality. *ACL 2014*, 61.
- Leeman-Munk, S. P., Wiebe, E. N., & Lester, J. C. (2014, March). Assessing elementary students' science competency with text analytics. In *Proceedings of the Fourth International Conference on Learning Analytics And Knowledge* (pp. 143-147). ACM.
- Lemaire, B., & Dessus, P. (2001). A system to assess the semantic content of student essays. *Journal of Educational Computing Research*, 24(3), 305-320.
- Lemann, N. (1995, September). The great sorting. *Atlantic Monthly*, 276(3), 84-100.

- Martin, J., & VanLehn, K. (1995). Student assessment using Bayesian nets. *International Journal of Human-Computer Studies*, 42(6), 575-591.
- Matthew, C. T., & Stemler, S. E. (2013). Assessing mental flexibility with a new word recognition test. *Personality and Individual Differences*, 55(8), 915-920.
- Mays, E., Damerau, F. J., & Mercer, R. L. (1991). Context based spelling correction. *Information Processing & Management*, 27(5), 517-522.
- Mihalcea, R., Corley, C., & Strapparava, C. (2006, July). Corpus-based and knowledge-based measures of text semantic similarity. In *AAAI* (Vol. 6, pp. 775-780).
- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Miller, W. L., Petsche, K., Baker, R. S., Labrum, M. J., & Wagner, A. Z. (2013). Boredom Across Activities, and Across the Year, within Reasoning Mind.
- Mitchell, T., Russell, T., Broomhead, P., & Aldridge, N. (2002). Towards robust computerised marking of free-text responses.
- Mohler, M., Bunescu, R., & Mihalcea, R. (2011, June). 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-Volume 1* (pp. 752-762). Association for Computational Linguistics.
- Mohler, M., & Mihalcea, R. (2009, March). Text-to-text semantic similarity for automatic short answer grading. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics* (pp. 567-575). Association for Computational Linguistics.

- Moldovan, C., Rus, V., & Graesser, A. C. (2011, April). Automated Speech Act Classification For Online Chat. In *MAICS* (pp. 23-29).
- Murrugarra, N., Lu, S., & Li, M. L. (2013). Automatic Grading Student Answers. Retrieved from <http://people.cs.pitt.edu/~zhumihua/projects>.
- Ng, H. T., Wu, S. M., Briscoe, T., Hadiwinoto, C., Susanto, R.H., Bryant, C. (2014). The conll-2014 shared task on grammatical error correction. In *Proceedings of CoNLL*.
- Ng, H. T., Wu, S. M., Wu, Y., Hadiwinoto, C., & Tetreault, J. (2013, June). The conll-2013 shared task on grammatical error correction. In *Proceedings of CoNLL*.
- Nidumolu, S. R., Subramani, M., & Aldrich, A. (2001). Situated learning and the situated knowledge web: Exploring the ground beneath knowledge management. *Journal of Management Information Systems*, 18(1), 115-150.
- Nielsen, R. D., Ward, W., Martin, J. H., & Palmer, M. (2008, May). Annotating Students' Understanding of Science Concepts. In *LREC*.
- Niraula, N. B., Rus, V., Banjade, R., Stefanescu, D., Baggett, W., & Morgan, B. (2014). The dare corpus: A resource for anaphora resolution in dialogue based intelligent tutoring systems. *Proceedings of Language Resources and Evaluation, LREC*.
- Page, E.B. (1966), The imminence of Grading Essays by Computer, *Phi Delta Kappan*, 47, 238-243.
- Page, E. B., Fisher, G. A., & Fisher, M. A. (1968). Project essay grade-A Fortran program for statistical analysis of prose. *British journal of mathematical and statistical psychology*, 21, 139.

- Pedersen, T., Patwardhan, S., & Michelizzi, J. (2004, May). WordNet:: Similarity: measuring the relatedness of concepts. In *Demonstration Papers at HLT-NAACL 2004* (pp. 38-41). Association for Computational Linguistics.
- Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP 2014)*.
- Perelman, D., Gulwani, S., & Grossman, D. (2014). Test-Driven Synthesis for Automated Feedback for Introductory Computer Science Assignments. Retrieved from <http://aspiringminds.com>.
- Perelman, D., Gulwani, S., Grossman, D., & Provost, P. (2014, June). Test-driven synthesis. In *Proceedings of the 35th ACM SIGPLAN Conference on Programming Language Design and Implementation* (p. 43). ACM.
- Pérez, D., Gliozzo, A. M., Strapparava, C., Alfonseca, E., Rodríguez, P., & Magnini, B. (2005, May). Automatic Assessment of Students' Free-Text Answers Underpinned by the Combination of a BLEU-Inspired Algorithm and Latent Semantic Analysis. In *FLAIRS Conference* (pp. 358-363).
- Perez-Marin, D., Alfonseca, E., Rodriguez, P., & Pascual-Nieto, I. (2006). Willow: Automatic and adaptive assessment of students free-text answers. *Revista de la Sociedad Española de Procesamiento de Lenguaje Natural (SEPLN)*, 37, 367-368.
- Perry, J., & Shan, C. C. (2010, June). Generating quantifiers and negation to explain homework testing. In *Proceedings of the NAACL HLT 2010 Fifth Workshop on*

- Innovative Use of NLP for Building Educational Applications* (pp. 57-65). Association for Computational Linguistics.
- Pulman, S. G., & Sukkarieh, J. Z. (2005, June). Automatic short answer marking. In *Proceedings of the second workshop on Building Educational Applications Using NLP* (pp. 9-16). Association for Computational Linguistics.
- Rasila, A., Harjula, M., & Zenger, K. (2007, December). Automatic assessment of mathematics exercises: Experiences and future prospects. In *ReflekTori 2007 Symposium of Engineering Education* (pp. 70-80).
- Rich, C. S., Harrington, H., Kim, J., & West, B. (2008, March). Automated essay scoring in state formative and summative writing assessment. In annual meeting of the American Educational Research Association.
- Rus, V., Banjade, R., & Lintean, M. (2014). On Paraphrase Identification Corpora. In *Proceeding on the International Conference on Language Resources and Evaluation (LREC 2014)*.
- Rus, V., D'Mello, S. K., Hu, X., & Graesser, A. C. (2013). Recent Advances in Conversational Intelligent Tutoring Systems. *AI Magazine*, 34(3), 42-54.
- Rus, V., & Lintean, M. (2012, June). A comparison of greedy and optimal assessment of natural language student input using word-to-word similarity metrics. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP* (pp. 157-162). Association for Computational Linguistics.
- Rus, V., Lintean, M. C., Banjade, R., Niraula, N. B., & Stefanescu, D. (2013, August). SEMILAR: The Semantic Similarity Toolkit. In *ACL (Conference System Demonstrations)* (pp. 163-168).

- Rus, V., Niraula, N., Lintean, M., Banjade, R., Stefanescu, D., & Baggett, W. (2013). Recommendations for the Generalized Intelligent Framework for Tutoring based on the Development of the DeepTutor Tutoring Service, *Workshop on Generalized Intelligent Framework for Tutoring (GIFT)*, The 16th International Conference on Artificial Intelligence in Education (AIED 2013), July 9-13, Memphis, TN.
- Sabourin, J., Mott, B., and Lester, J. C. (2011). Modeling learner affect with theoretically grounded dynamic Bayesian networks. In *Affective Computing and Intelligent Interaction*. Springer, Berlin Heidelberg.
- Sangwin, C. (2004). Assessing mathematics automatically using computer algebra and the internet. *Teaching Mathematics and its Applications*, 23(1), 1-14.
- Shah, N. B., Bradley, J., Balakrishnan, S., Parekh, A., Ramchandran, K., & Wainwright, M. J. (2014). Some Scaling Laws for MOOC Assessments. Retrieved from <http://www.stat.berkeley.edu/~sbalakri/Papers>.
- Shah, N. B., Bradley, J. K., Parekh, A., Wainwright, M., & Ramchandran, K. (2013, December). A case for ordinal peer-evaluation in MOOCs. In *NIPS Workshop on Data Driven Education*.
- Shermis, M. D., Burstein, J., Higgins, D., & Zechner, K. (2010). Automated essay scoring: Writing assessment and instruction. *International Encyclopedia of Education*, 4, 20-26.
- Sil, A., Ketelhut, D. J., Shelton, A., & Yates, A. (2012, June). Automatic grading of scientific inquiry. In *Proceedings of the Seventh Workshop on Building*

- Educational Applications Using NLP* (pp. 22-32). Association for Computational Linguistics.
- Ștefănescu, D., Banjade, R., & Rus, V. (2014). Latent Semantic Analysis Models on Wikipedia and TASA. In *LREC*.
- Sukkarieh, J. Z., & Blackmore, J. (2009, March). c-rater: Automatic Content Scoring for Short Constructed Responses. In *FLAIRS Conference*.
- Sukkarieh, J., & Bolge, E. (2008, January). Leveraging c-rater's automated scoring capability for providing instructional feedback for short constructed responses. In *Intelligent Tutoring Systems* (pp. 779-783). Springer Berlin Heidelberg.
- Sukkarieh, J. Z., & Bolge, E. (2010). Building a Textual Entailment Suite for the Evaluation of Automatic Content Scoring Technologies. In *LREC*.
- Sukkarieh, J. Z., & Pulman, S. G. (2005, May). Information extraction and machine learning: Auto-marking short free text responses to science questions. In *Proceedings of the 2005 Conference on Artificial Intelligence in Education: Supporting Learning through Intelligent and Socially Informed Technology* (pp. 629-637). IOS Press.
- Sukkarieh, J. Z., & Stoyanchev, S. (2009, August). Automating Model Building in c-rater. In *Proceedings of the 2009 Workshop on Applied Textual Inference* (pp. 61-69). Association for Computational Linguistics.
- Turian, J., Ratinov, L., & Bengio, Y. (2010, July). Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics* (pp. 384-394). Association for Computational Linguistics.

VanLehn, K., Graesser, A. C., Jackson, G. T., Jordan, P., Olney, A., & Rose, C. P. (2007)

When are tutorial dialogues more effective than reading? *Cognitive Science*, 31, 3-62

VanLehn, K., Jordan, P. W., Rosé, C. P., Bhembé, D., Böttner, M., Gaydos, A., ...

Srivastava, R. (2002, January). The architecture of Why2-Atlas: A coach for qualitative physics essay writing. In *Intelligent Tutoring Systems* (pp. 158-167). Springer Berlin Heidelberg.

Viera, A. J., & Garrett, J. M. (2005). Understanding inter-observer agreement: the kappa statistic. *Fam Med*, 37(5), 360-363.

Whittington, D. & Hunt, H. (1999). Approaches to the computerized assessment of free text responses. In Danson, M. (Ed.), *Proceedings of the Sixth International Computer Assisted Assessment Conference*, Loughborough, UK

Wild, F., Stahl, C., Stermsek, G., & Neumann, G. (2005). Parameters driving effectiveness of automated essay scoring with LSA. In *Proceedings of the 9th International Computer Assisted Assessment (CAA) Conference*, Loughborough, UK, July 5, 2005 (pp. 485–494).

Young, M. (1995). Assessment of situated learning using computer environments. *Journal of Science Education and Technology*, 4(1), 89-96.