TOWARD SUPPORTING FINE-GRAINED, STRUCTURED, MEANINGFUL AND ENGAGING FEEDBACK IN EDUCATIONAL APPLICATIONS

Florin Adrian Bulgarov

Dissertation Prepared for the Degree of

DOCTOR OF PHILOSOPHY

UNIVERSITY OF NORTH TEXAS

December 2018

APPROVED:

Rodney D. Nielsen, Major Professor
Alexis Palmer, Committee Member
Yan Huang, Committee Member
Eduardo Blanco, Committee Member
Barrett Bryant, Chair of the Department of Computer Science and Engineering
Yan Huang, Interim Dean of the College of Engineering
Victor Prybutok, Dean of the Toulouse Graduate School

Recent advancements in machine learning have started to put their mark on educational technology. Technology is evolving fast and, as people adopt it, schools and universities must also keep up (nearly 70% of primary and secondary schools in the UK are now using tablets for various purposes). As these numbers are likely going to follow the same increasing trend, it is imperative for schools to adapt and benefit from the advantages offered by technology: real-time processing of data, availability of different resources through connectivity, efficiency, and many others. To this end, this work contributes to the growth of educational technology by developing several algorithms and models that are meant to ease several tasks for the instructors, engage students in deep discussions and ultimately, increase their learning gains.
Copyright 2018
by
Florin Adrian Bulgarov
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>LIST OF TABLES</th>
<th>vi</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHAPTER 1 INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1. Motivation</td>
<td>1</td>
</tr>
<tr>
<td>1.2. Context</td>
<td>3</td>
</tr>
<tr>
<td>1.3. Overview</td>
<td>5</td>
</tr>
<tr>
<td>1.4. Contributions</td>
<td>6</td>
</tr>
<tr>
<td>CHAPTER 2 BACKGROUND AND RELATED WORK</td>
<td>8</td>
</tr>
<tr>
<td>2.1. Educational Applications</td>
<td>8</td>
</tr>
<tr>
<td>2.1.1. Classroom Response Systems</td>
<td>8</td>
</tr>
<tr>
<td>2.1.2. Assessment Systems</td>
<td>10</td>
</tr>
<tr>
<td>2.1.3. Intelligent Tutoring Systems</td>
<td>14</td>
</tr>
<tr>
<td>2.1.4. Analogy between Assessment Systems and Intelligent Tutoring Systems</td>
<td>17</td>
</tr>
<tr>
<td>2.2. Fine-Grained Semantic Units</td>
<td>19</td>
</tr>
<tr>
<td>2.3. Textual Alignment</td>
<td>21</td>
</tr>
<tr>
<td>2.4. Textual Clustering</td>
<td>22</td>
</tr>
<tr>
<td>2.5. Textual Similarity</td>
<td>25</td>
</tr>
<tr>
<td>2.6. Selecting Engaging Responses</td>
<td>27</td>
</tr>
<tr>
<td>CHAPTER 3 DATA OVERVIEW</td>
<td>29</td>
</tr>
<tr>
<td>3.1. Data</td>
<td>29</td>
</tr>
<tr>
<td>3.2. Preprocessing</td>
<td>31</td>
</tr>
<tr>
<td>CHAPTER 4 MINIMAL MEANINGFUL PROPOSITIONS$^1$</td>
<td>33</td>
</tr>
<tr>
<td>4.1. Data Description</td>
<td>34</td>
</tr>
<tr>
<td>4.2. MMP Extraction Algorithm</td>
<td>34</td>
</tr>
<tr>
<td>Chapter</td>
<td>Title</td>
</tr>
<tr>
<td>---------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>4.3</td>
<td>Results</td>
</tr>
<tr>
<td>4.4</td>
<td>Error Analysis</td>
</tr>
<tr>
<td>4.5</td>
<td>MMP Importance Classification</td>
</tr>
<tr>
<td></td>
<td><strong>CHAPTER 5 ALIGNMENT OF MINIMAL MEANINGFUL PROPOSITIONS</strong></td>
</tr>
<tr>
<td>5.1</td>
<td>Data Description</td>
</tr>
<tr>
<td>5.2</td>
<td>Classification</td>
</tr>
<tr>
<td>5.3</td>
<td>Results</td>
</tr>
<tr>
<td>5.4</td>
<td>Error Analysis</td>
</tr>
<tr>
<td>5.5</td>
<td>MMP Alignment Disadvantages</td>
</tr>
<tr>
<td></td>
<td><strong>CHAPTER 6 UNDERSTANDING OF MINIMAL MEANINGFUL PROPOSITIONS</strong></td>
</tr>
<tr>
<td>6.1</td>
<td>Data Description</td>
</tr>
<tr>
<td>6.2</td>
<td>Classification</td>
</tr>
<tr>
<td>6.3</td>
<td>Results</td>
</tr>
<tr>
<td>6.4</td>
<td>Error Analysis</td>
</tr>
<tr>
<td>6.5</td>
<td>Further Experimentation</td>
</tr>
<tr>
<td></td>
<td><strong>CHAPTER 7 CLUSTERING STUDENT RESPONSES</strong></td>
</tr>
<tr>
<td>7.1</td>
<td>Data Description</td>
</tr>
<tr>
<td>7.2</td>
<td>Clustering Responses</td>
</tr>
<tr>
<td></td>
<td>7.2.1. Student Response Representation</td>
</tr>
<tr>
<td></td>
<td>7.2.2. Algorithm</td>
</tr>
<tr>
<td>7.3</td>
<td>Results</td>
</tr>
<tr>
<td>7.4</td>
<td>Error Analysis</td>
</tr>
<tr>
<td></td>
<td><strong>CHAPTER 8 SELECTION OF ENGAGING RESPONSES</strong></td>
</tr>
<tr>
<td>8.1</td>
<td>Data Description</td>
</tr>
<tr>
<td>8.2</td>
<td>Classification</td>
</tr>
<tr>
<td>8.3</td>
<td>Results</td>
</tr>
<tr>
<td>8.4</td>
<td>Error Analysis</td>
</tr>
<tr>
<td>Table</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>--------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>4.1</td>
<td>MMP counts and the average length of the reference answers in words.</td>
</tr>
<tr>
<td>4.2</td>
<td>Most Frequent MMP Structures</td>
</tr>
<tr>
<td>4.3</td>
<td>MMP Extraction Results</td>
</tr>
<tr>
<td>4.4</td>
<td>MMP Importance Labels.</td>
</tr>
<tr>
<td>4.5</td>
<td>Training Set 10xCV Results</td>
</tr>
<tr>
<td>5.1</td>
<td>MMP Alignment Dataset Statistics</td>
</tr>
<tr>
<td>5.2</td>
<td>Description of features. <em>Gov</em> and <em>mod</em> refer to governor and modifier.</td>
</tr>
<tr>
<td>5.3</td>
<td>MMP Alignment Results</td>
</tr>
<tr>
<td>6.1</td>
<td>MMP Understanding Dataset (the number of understood pairs for a given question equals its number of reference answer MMPs multiplied by the number of student responses to the question)</td>
</tr>
<tr>
<td>6.2</td>
<td>MMP Understanding Results (NN = Neural Network; WE = Word Embeddings;)</td>
</tr>
<tr>
<td>6.3</td>
<td>MMP Understanding Experiments.</td>
</tr>
<tr>
<td>7.1</td>
<td>Clustering Dataset Statistical Distribution (annotator 1 (A1) had 28 questions with 3 clusters, which had an average of 8.3 student responses in the correct answer cluster and an average of 3.3 responses in the grab bag, outside the 3 annotator-created clusters).</td>
</tr>
<tr>
<td>7.2</td>
<td>Clustering Results</td>
</tr>
<tr>
<td>8.1</td>
<td>Engagingness Annotations – number of responses annotated with each grade</td>
</tr>
<tr>
<td>8.2</td>
<td>Engagingness Features</td>
</tr>
<tr>
<td>8.3</td>
<td>Engagingness Results</td>
</tr>
<tr>
<td>8.4</td>
<td>Engagingness Results - Deep Neural Network (DNN) vs. Human</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
</tr>
<tr>
<td>---------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>8.5</td>
<td>Engagingness Results - Precision@1 and Session Success Rate ($SSR$)</td>
</tr>
<tr>
<td>8.6</td>
<td>Engagingness Results - Self-Improving Deep Neural Network ($SSR =$</td>
</tr>
<tr>
<td></td>
<td>Session Success Rate)</td>
</tr>
</tbody>
</table>
CHAPTER 1

INTRODUCTION

The short term goal of the research presented here is to develop a framework of intelligent algorithms that can analyze classroom questions and associated student responses in real-time to provide analytics expected to support instructors in recognizing and interpreting classroom comprehension. The models will account for a fine-grained analysis of open student responses, consider semantics, predict detailed understanding relationships between student responses and the teacher’s reference answer, cluster responses based on students’ understanding of the most important concepts and finally, choose representative engagement responses from the cluster, that have the potential of starting insightful classroom discussions. The long term goal associated with this research is to facilitate an increase in classroom engagement, instructor interpretation of classroom comprehension, and ultimately, student learning gains.

1.1. Motivation

In the past few decades, technological advancements have completely and irreversibly changed most aspects of our life. From altering the way we talk to people, to the way we go to work, computers have improved our efficiency significantly. Relative to other major changes throughout history, this was a fast process which started from the desire to make different practices faster and at larger scales. However, one of the few areas that has not seen these dramatic changes is education. The way we acquire knowledge in schools across the world can definitely benefit from the current developments in areas like machine learning and natural language processing. Nevertheless, if the recent tech history has told us anything, it is that computers, and artificial intelligence especially, can disrupt entire domains in a short period of time. To me, this raises a few questions. Is it possible in the future for machines to engage individual students and make learning a more effective and less stressful experience? Might they even be capable of offering a personalized experience based on the child’s learning style, background and even interests? Can algorithms completely replace the
human instructors while also maximizing the learning outcomes, or they will only be able to assist them in this process? While these questions have yet to be answered, this work takes important steps towards filling the gap in educational technology by developing intelligent algorithms to assist the teacher in the instructional process and ultimately increase student learning gains.

After teaching the material, instructors usually assess students’ knowledge by asking related questions. With no technology, only one student can be selected to respond. Even if the response is wrong, and the process is repeated, the vast majority of students will not be engaged in the class. Days, weeks, and even months can pass by until all students have been engaged in at least one discussion.

With current technology, developing a system in which students can enter their response is not a hard task. This is also supported by the availability of digital devices to children. In 2016, in the United Kingdom, more than 44% of 5 to 15-year-olds owned a tablet or phone [81]. In Germany, 51% of children aged 6 to 13 owned a phone. This data generalizes well across Europe, where about 46% of children aged 9 to 16 own a smartphone [72]. In South Korea, about 72% of children owned a smartphone by the ages of 11 to 12 [52]. In the United States, children aged 10 to 12 not only held a smartphone, but they had their own service plan, as opposed to only using Wi-Fi. This data supports the current digitalization trend seen in most domains. Technology is evolving fast and, as people adopt it, schools and universities must also keep up (nearly 70% of primary and secondary schools in the UK are now using tablets for various purposes [20]). One reason for this shift is clear: children are learning more easily in a familiar environment. Other reasons lie in the advantages that technology has brought to many other fields: real-time processing of data, availability of different resources through connectivity, efficiency, and many others.

By answering a question using a digital device, students can remain engaged and detect early misconceptions. This activity, which is a form of self-explanation, often contains

1https://www.mpfs.de/studien/kim-studie/2016/
elaborations and ideas that were not explicitly stated in the text. Thus, students are going beyond the provided information and, by articulating responses, the learners are also actively involved in the classroom. Such constructive activities have been shown to increase student learning gains in numerous studies [17].

While simply responding to questions in classrooms can have significant benefits for students, an educational application can take advantage of this data to further increase the learning gains. Having the question, the teacher-input reference answer and the set of student responses in a digital format, various machine learning techniques can be used to offer formative feedback to the teacher and students, automatically assess student responses for increased accuracy, significantly minimize repeated workload for the teacher, and increase student engagement by providing the tools needed for the teacher to start insightful classroom discussions. To reach these goals, this work analyzes the teacher-provided information and the student responses on different levels. This analysis includes: extracting smaller, more manageable propositions from answers, aligning the propositions, predicting their importance relative to the question, establishing an understanding relation between each reference answer proposition and each student response, clustering the student responses by taking advantage of all previous information, and finally, providing the instructor with representative, engaging responses, that are likely to generate insightful classroom discussions.

1.2. Context

In most classrooms, instructors periodically assess the students’ level of knowledge by asking questions either to quickly grasp their understanding or to evaluate them as part of a test. While this can be a reasonable way to engage individual students and start classroom discussions, it is ineffective for the classroom as a whole due to the teacher’s capability to engage only one student at a time. Moreover, each student’s personality will have a say whether s/he will want to respond, and how often. Most of the time, only a handful of students are constantly engaged in the classroom. To this end, Comprehension SEEDING [83] is an educational application developed to allow all students to respond to questions in the classroom. Using a web-enabled device, i.e., mobile phone, tablet, PC, etc., students
can write responses to questions asked by the human instructor in the classroom. This way, all students are engaged and they can self-assess their acquired knowledge without feeling nervous about what others think in regard to what they know.

Allowing students to answer questions is not enough for such a system to be successful. Comprehension SEEDING uses the responses given by the students and organizes them to offer teachers real-time and formative feedback. It does so by clustering student responses and identifying the representative ones. Thus, the teacher is provided with a representative response for each cluster, displayed with the percentage of students which answered similarly. By only reading a handful of responses, the human instructor can quickly comprehend the current classroom level of knowledge and can address common misunderstandings or initiate classroom discussions based on the clusters or based on individual student responses. A word cloud of the most frequent words used by the students is also provided in real-time, while students are typing their responses.

Comprehension SEEDING is a new type of educational system referred to as a Classroom Engagement System (CES). Its main purpose is to support instructors with the goal of increasing student engagement in classrooms. Nevertheless, to increase student participation, the application needs to provide a summative evaluation of student responses and to structure the information in order to offer insightful feedback to instructors and students.

The Comprehension SEEDING system as described by Paiva et al. (2014) [83] relies entirely on lexical information – word overlap. The responses were received by the system and then compared at a lexical level to form clusters. Even though it is relatively simple, the system received great feedback from the teachers and students who used it. Nevertheless, such an educational application can greatly benefit from more complex analysis of student responses, further enhancing the feedback received by the teacher and finally, increasing the classroom engagement even more. The goal associated with this work is to develop new algorithms for the SEEDING system that will account for a fine-grained analysis of student responses, consider semantics, predict detailed understanding relationships between student responses and the teacher’s reference answer, cluster responses based on the stu-
dents’ understanding of the most important concepts in their response and finally, choose representative engagement responses from the clusters, that have the potential of starting insightful classroom discussions in which many students will participate.

1.3. Overview

The final goal of this work is to develop intelligent algorithms that will lead to increased classroom engagement and ultimately increase student learning gains. One way to increase the engagement is by giving all students the chance to answer the question. This is already being achieved and does not require any smart component, except a platform with an easy to follow user-interface in which students can enter their responses.

A second way to increase the classroom engagement is by providing teachers with organized and insightful feedback on the students’ responses, so they can initiate interesting classroom discussions. To do this, simply deciding whether a student response understood a reference answer is not enough. The student might address only part of the reference answer while contradicting or forgetting about another. I also need to understand how important is what the student did not address, with respect to the question. To this end, a new representation of the reference answer is introduced: Minimal Meaningful Propositions (MMPs). An MMP is defined as a proposition that cannot be broken down into finer-grained propositions (it is minimal) and still be interpretable without further context (it is meaningful on its own). The second contribution of this work is the development of an algorithm that automatically extracts Minimal Meaningful Propositions from the reference answer. The need for MMPs, the extraction algorithm and its performance will be discussed in Chapter 4.

Once I have a fine-grained representation of the reference answer, I then move on to create an MMP understanding algorithm to decide which of the MMPs from the reference answer were understood by the student. Doing so, in conjunction with a classifier that predicts the importance of each MMP with respect to the question, will help our end goal – showing the teacher representative engaging responses that can lead to insightful classroom discussions. The MMP understanding algorithm will be discussed in Chapter 6.
I now reached the point where I have the importance of each MMP and an understanding relation between each student’s response and each reference answer MMP. Considering this information, I can cluster the student responses based on what propositions each understood and, perhaps more importantly, I can create clusters of students that didn’t address important concepts from the reference answer. This is also the point where I can show the instructor real-time statistics, such as which are the most common MMPs expressed by the students, which reference answer MMPs have been addressed and which haven’t, and by how many students. The clustering algorithm is presented in Chapter 7.

The final step of this work is to select, for each group, a response that is likely to lead to interesting, engaging classroom discussion and that balances within-cluster representativeness with across-cluster uniqueness; henceforth, referred to as the cluster’s engagement response. The model doing this is described in Chapter 8.

1.4. Contributions

The goal associated with this research is to develop computational methods that could facilitate classroom engagement, instructor interpretation of classroom comprehension, and student learning. The individual contributions made toward the end goal are described below.

1. A novel knowledge representation called Minimal Meaningful Proposition (MMP) – a decomposition of text, such as a question’s answer, into the set of propositions that individually represent single minimal claims or arguments that cannot be further decomposed without losing contextual meaning, and taken as a whole represent the entire meaning of the text (Chapter 4);

2. The corresponding first automatic MMP extraction algorithm based on structural templates learned computationally from dependency parses of the MMP (Chapter 4);

3. (a) Two computational methods to recognize whether a student response implies that they understand a given reference answer MMP – one based on hand-crafted features and an SVM and the other using word embeddings and a neural network
(Chapter 6); (b) Experimental evidence that the MMPs extracted in (2) can be used to determine fine-grained student understanding;

(4) (a) A student response clustering framework that not only takes into consideration the similarity between student responses, similar to prior work on student answer clustering, but also for the first time considers (i) the fine-grained student understanding of MMPs in the teacher reference answer, as determined in (3), and (ii) the importance of each of those MMPs; (b) A cluster selection algorithm is also implemented that iteratively re-starts the process to select the best groups of responses from the hierarchy output by the agglomerative clustering algorithm, while filtering out responses with anomalous or low frequency semantics (Chapter 7), reaching a final performance that is 86.3% of the performance obtained by human annotators;

(5) (a) The first computational model to predict how engaging student responses will be in classroom discussion, which among other features relies on MMP understanding, reaching a performance that is 86.8% of the performance obtained by human annotators; and (b) A relatively simple online learning technique that reduces the average relative error by up to 32% through minimal feedback from the teacher.

(6) Tested (a) methods, (b) guidelines, and (c) annotated corpora for each task: extracting MMPs (see 2), classifying students’ MMP understanding (see 3), clustering student responses (see 4), and selection of engaging student responses (see 5); thus enabling further scientific research in the broader community.
CHAPTER 2

BACKGROUND AND RELATED WORK

2.1. Educational Applications

Educational applications have not been around for a long time, with Classroom Response Systems (or Clicker Systems) opening the way for active and engaging classrooms, followed by Assessment Systems – focused on teacher feedback and Intelligent Tutoring Systems (ITSs) – focused on student feedback. Numerous studies have found these systems to be useful for students, teachers, or both [54, 58, 105].

All systems focus on increasing the learning gains of students through the enhancement of different learning processes. As Internet and media use overwhelms children and adolescents [25, 30, 89], it has become increasingly harder for teachers to keep them focused in classrooms [7]. Thus, a new type of educational application is needed. The particularity of the work done in this dissertation is that it enables the enhancement of the learning process by focusing on increasing classroom engagement among students [7, 100]. This new type of system, called Classroom Engagement System (CES) integrates elements from all presented educational applications while also adding new ones, such as predicting whether or not student responses are engaging, or likely to lead to interesting classroom discussion if debated.

2.1.1. Classroom Response Systems

Classroom Response Systems, or clicker systems, became very popular around a decade ago. They were among the first means of transforming traditional passive lectures into active, engaging ones. Clicker systems give instructors the power to ask questions in the classroom in strategic moments and allow each student to answer them using a relatively small piece of personal wireless hardware. Students are pressing a number on the hardware associated with an answer to the question.

Various Classroom Response Systems allow for various processing of the answers. Most of them make the answers available to the classroom by showing a graphical represen-
Some are adding value by showing answers in real-time, allowing for comparisons (e.g., before a lecture and after it), and some allow instructors to make more detailed analysis of the students’ responses after the lecture was finished (e.g., generating reports, showing demographics, etc.).

Clicker systems add most of the value by allowing instructors to assess student understanding of course material by asking quick questions. By doing this, students are able to self-assess their comprehension, while instructors are being almost instantly provided with feedback on the student’s understanding of concepts and content. Therefore, in cases of misconceptions, students can ask clarifying questions early on. This will facilitate their comprehension on ulterior, related content. Moreover, instructors can use the system’s feedback to make sure everybody is on track with the presented material.

Although clicker systems are very helpful to evaluate student knowledge, the major advantage is the stimulation of classroom engagement among students. Teachers can also choose to ask questions querying for opinions on controversial issues. Given the responses, teachers can either group students to discuss their arguments or they can start a general classroom discussion. Therefore, a simple system can help generate complex discussions and significantly contribute to improve the critical thinking skills of the students.

A 2007 study on clicker systems was carried by R. Kaleta and T. Joosten [54]. It involved 28 faculty members and 3500 students. The survey was designed to evaluate the perceived impact of clickers on classroom engagement and student learning and to evaluate faculty and student satisfaction with the use of clickers for teaching and learning. The results of the study showed that both students and instructors who used clicker systems were left with a very good impression over their impact on classroom engagement and learning. An overwhelming majority of the instructors agreed that clicker systems improve student engagement (94%) and participation (87%). The students also agreed on this aspect, although with a smaller majority (69% and 70%, respectively). When it comes to student learning, 58% of the faculty believed that clickers helped increase student performance in class. Also, 53% of the students believed that clickers have been beneficial to their learning.
Clicker systems proved to be very effective for both students and instructors in enhancing student learning and engagement, but they were only concentrating on evaluating simple conceptual knowledge [29, 33, 57]. By using them, teachers are limited to ask multiple choice questions and their added value is limited to the quality of those choices. To make the whole learning process more efficient, more complicated systems are needed to automatically evaluate open responses, offer complex, real-time and customized feedback to both teachers and students and to increase the engagement more directly.

The similarity between Clicker Systems and the work presented here resides in the engagement aspect. Both systems start from the hypothesis that all students should respond to classroom questions in order to self-assess their level of knowledge. However, the significant difference is made by the fact that Clicker Systems require students to pick a variant of those input by the teacher, while in the proposed framework, students are required to articulate their own open-responses leading to increased benefits [17, 18]. Analyzing open-responses requires a significantly larger amount of work, generating the need to incorporate elements from Assessment Systems and Intelligent Tutoring Systems as well as introducing new ones.

2.1.2. Assessment Systems

Assessment systems have been developed with the purpose of liberating the human instructor from doing repetitive tasks such as grading responses or essays. Most of the assessment systems developed so far tackle only short responses, i.e. at most a paragraph. Also, many of the early systems required a lot of human-intensive labor, not only for new domains, but also for every new question added to the system [88]. Classifiers were being built for every question and thus, many training responses were needed for an accurate entailment score prediction.

Hybrid algorithms such as the one described by Rosé et al. (2003) [92] are combining syntactic features with text classifications. In this particular example, the authors were trying to analyze student essay answers to qualitative physics questions. Even though they show that the hybrid approach outperforms two LSA and Naïve Bayes bag-of-words approaches, their method requires 100 to 500 annotated student responses in order to train
the classifiers for one single question. This makes the task almost impossible to scale and restricts the instructors to only a handful of questions which they can ask.

Concept Rater (or c-rater) is an automated scoring engine developed by the Educational Testing Service (ETS) [63]. The purpose of c-rater is to score short responses (up to 100 words) to open questions. In order to determine the paraphrase relation or the similarity, the system requires a set of reference answers as a model, input by an expert. There are four main steps in c-rater. First, in the Model Building phase, a set of correct answers are generated. This step is the most time consuming and labor intensive part of the process. The instructors have to enter separate sentences for each concept they use in their answer. Then, multiple paraphrases of the same sentence are being created manually, as well as synonyms of the concepts. To shorten the human effort of this approach the authors came up with a method that only requires manual concept-based scoring, generating the lexicon automatically [104]. The results indicate that the unweighted kappa values for the two approaches (manual and automatic) are “comparable” in 11 out of 12 scenarios, with the remaining scenario having the highest number of concepts, i.e., seven. Second, model answers and students’ responses are processed using Natural Language Processing techniques and linguistic features are extracted. Third, using the features previously extracted, the Goldmap matching algorithm is used to automatically determine whether a student’s response entails the model answer. Finally, scoring rules are applied to produce a score and feedback for the student.

Nielsen et al., (2009) [80] were among the first to actually try to accurately show which part of the teacher’s reference answer the student failed to address or misunderstood. They approach this task by coming up with a new representation of reference answers, called facets – fine-grained semantic components roughly derived from typed dependencies. Instead of predicting whether a student understood a reference answer as a whole, they predict entailment relations between the student response and individual facets from the reference answer. By doing so, they can pin-point the exact concept or piece of information the student failed to address or contradicted. This information can be used extensively
by educational applications to enhance their feedback to both students and teachers and to grade answers more accurately. The paper also shows that this approach achieves good accuracy for both domain (75.5%) and out-of-domain (68%) questions.

Horbach et al., (2013) [44] describe an approach for assessing student responses to reading comprehension tasks for foreign language learning. Instead of only comparing the student’s response with a correct response, they also consider the text from where the question was derived. This is the first use of reading texts for automatic short answer scoring in the context of foreign language learning. They show that, for German, simply using text-based features improves classification over models that only consider teacher authored responses.

SemEval-2014 included an entailment competition under the Evaluation of Compositional Distributional Semantic Models on Full Sentences through Semantic Relatedness and Entailment task [70]. Here, systems were presented with pairs of sentences and were evaluated on their ability to predict human judgments on (1) semantic relatedness and (2) entailment. The task attracted 21 teams, most of which participated in both subtasks. One team notes that by combining word overlap and antonyms one can detect 83.6% of neutral pairs and 82.6% of entailment pairs. The top-ranking systems in both tasks used compositional features and most of them also used external resources, especially WordNet [74]. Almost all the participating systems outperformed the proposed baselines in both tasks.

Even though neural networks have been successful in detecting paraphrase relations [48, 101, 113], neural architectures often fail to obtain acceptable performance scores in Recognizing Textual Entailment (RTE) tasks due to the lack of large high-quality datasets. However, recently, Bowman et al. (2015) [11] published the Stanford Natural Language Inference (SNLI) dataset, which due to its large size and high quality, allowed researchers to achieve high accuracy without hand-crafted features. Using neural networks with long short-term memory units (LSTM), they reached an accuracy of 77.6% on this dataset. This was the first generic neural model without hand-crafted features to achieve performance close to that of a traditional classifier using manually-engineered features for RTE. They accomplished
this by encoding both sentences as fixed-length vectors and used their concatenation in a
multi-layer perceptron for classification. Shortly after, Rocktaschel et al. (2015) [90] altered
Bowman et al.’s method by proposing an attentive neural network capable of reasoning over
entailments of pairs of words and phrases by processing the hypothesis conditioned on the
premise. By doing so, they achieved a higher accuracy on the SNLI dataset of 83.5%. They
presented three major contributions: (1) they built a neural model based on LSTMs that
reads the two sentences at once to determine entailment; (2) they extended the first model
with a neural word-by-word attention mechanism to encourage reasoning over entailments
of pairs of words and phrases; and (3) they provided a detailed qualitative analysis of neural
attention for RTE.

Chen et al. (2016) [16], adopt an enhanced sequential inference model, which outper-
formed previous, more complicated network architectures. Their model uses bidirectional
LSTMs (BiLSTM) for both local inference modeling and inference composition. Additional
improvement is achieved by incorporating syntactic parsing information. This model sets the
current highest performance on this dataset – 88.6%. The same performance is also reached
by Wang et al. (2017) [107], who used a bilateral multi-perspective matching (BiMPM)
model. Given two sentences, their model first encodes them with a BiLSTM encoder and
then matches them in both directions. Another BiLSTM layer is used to aggregate the
matching results into a fixed-length vector. Based on it, a decision is made through a fully
connected layer. Although reaching high accuracies on the SNLI dataset, these approaches
are not appropriate for our data due to the small dataset size (20,000 vs 570,000 instances)
and the nature of student responses (which are often ungrammatical).

Han et al. 2017 [39] propose a different approach to the task of recognizing textual
entailment by mapping phrases to their visual denotations and compare their meaning in
terms of their images. This approach shows improvement when combined with specific
linguistic and logic features.

In 2017, Williams et al. [109] introduced another large RTE corpora made of 433k ex-
amples called Multi-Genre Natural Language Inference (MultiNLI). The dataset is designed
for use in developing and evaluating machine learning models for sentence understanding. This corpora differs from the SNLI dataset in two ways. First, its sentences are derived from ten distinct genres of written and spoken English and second, it is improving the difficulty level allowing for fine-grained comparisons between strong models.

Conneau et al. 2017 [22] leverages both SNLI and MultiNLI datasets to learning universal representations of sentences as opposed to words. A sentence encoder model is trained first on the SNLI dataset, then on both to show a significant boost in performance compared to the model trained only on SNLI.

A light-weight neural net, Directional Self-Attention Network (DiSAN), is proposed by Sehn et al. 2017 [96] to learn sentence embeddings. Despite its simple form, DiSAN out-performs complicated Recurrent Neural Network (RNN) models on both prediction quality and time efficiency. At the time of its publication, DiSAN obtained the best results on both SNLI [11] and MultiNLP [108] datasets as well as on Stanford Sentiment Treebank (SST) [102], Sentences Involving Compositional Knowledge (SICK) [71], and others.

Assessment systems and those that recognize textual entailment can bring great benefits in classrooms especially for teachers, by minimizing the time spent by them grading responses or essays. A similar task is carried-out as part of this work, in order to decide if students understand or not specific aspects of the teacher-input reference answer. This helps the analysis of the answers articulated by students in response to the open-ended classroom questions. However, the work proposed here does not score responses and it does not detect contradictions.

2.1.3. Intelligent Tutoring Systems

Intelligent Tutoring Systems (ITSs) were created with the goal of improving learning through real-time and personalized feedback for students [37, 40, 92]. ITSs need to be able to interpret complex student responses and improve their feedback as they process more questions and responses, but essentially, most existing systems require skilled developers, or at least a content matter expert, to write rules or train classifiers for each additional question, or a new domain. Generally, an ITS only provides feedback to students, and when
they do provide feedback to instructors, it is typically just high-level information regarding the correctness of each student response.

AutoTutor, introduced by Graesser et al. 2001 [37], is a computer program that simulates patterns and strategies of a typical human instructor. It was originally developed to help college students learn the fundamentals of computer hardware, operating systems, and the Internet. As discussed in Section 2.1.2, AutoTutor used Latent Semantic Analysis (LSA) to evaluate student answers. Data obtained using AutoTutor was recently successfully used in a survey by Shi et al. 2018 [99] to assess reading comprehension ability in 52 low-literacy adults who interacted with the system.

Gaze tutor [28] is an Intelligent Tutoring System which aims to increase student engagement by detecting and responding to their boredom and disengagement. The system uses a commercial eye tracker and attempts to reengage the students when it detects a state of boredom or zoning out. Gaze tutor was tested on 48 students on 4 biology topics with both versions of gaze-reactive and non-gaze-reactive tutors. The results suggest that the system was successful in reorienting students to important areas of the interface and increasing learning gains for questions that required deep reasoning. However, the students’ motivation and self-reported engagement increased only by a minimal margin.

ASSISTments [40] is one of the largest tutoring systems currently deployed in the United States. During the school year of 2013-2014, 50,000 students used it, together with a few hundred teachers. What started out initially as a mathematics platform, is now covering many other subjects, i.e., Science, English, Statistics, etc. ASSISTments is a simple quizzing system allowing teachers to write their own questions while students get instant feedback on their responses. If their response is incorrect, they can try again. The teachers also know what questions saw the most mistakes so they can address the least understood concepts. However, while this may work well for exact-sciences such as mathematics, it is not very successful with free-response questions. In such cases, the platform only shows the human instructor a list with every single student’s response. Another advantage of the ASSISTments platform is the amount of data it has gathered. The authors plan to use this data to mine
information in regards to most common math mistakes in order to prepare instructors ahead of teaching of what needs to be addressed in particular. Because teachers also have the possibility to add hints to each question, the authors also plan to crowd-source the hints and feedback messages from students, so they can help each other out.

The Assessment and LEarning in Knowledge Spaces (ALEKS) is a Web-based, artificially intelligent learning system that uses adaptive questioning to quickly and accurately determine exactly what a student knows and does not know in a course [31]. ALEKS was recently used in a study by Huang et al. 2016 [50] that found the ITS to reduce learning gaps between advantaged and disadvantaged groups. The 6th grade student volunteers that used ALEKS versus comparable teacher-led mathematics teaching, performed more similarly on the state test.

VanLehn, (2011) [105] studied the relative effectiveness of human tutoring versus Intelligent Educational Systems. Even though the common belief is that a human tutor is more effective than an intelligent system, the report concludes with an interesting statement: *ITSs are, within the limitations of the article, just as effective as adult, one-on-one human tutoring for increasing learning gains in STEM topics.* Moreover, another notable contribution of VanLehn’s review is that the effect of human tutoring compared against *no tutoring* is significantly smaller than what was previously reported by Bloom in 1984 [10]. Even though none of the studied systems attempted to replace an actual classroom teacher with an ITS, VanLehn argues that ITSs should be used to replace homework, seatwork and other activities. Nevertheless, an Intelligent Tutoring System can’t yet replace a whole classroom experience.

A large-scale review of the effectiveness of Intelligent Educational Systems was also carried out by Kulik and Fletcher in 2016 [58]. Fifty controlled evaluations were considered and the meta-analysis showed that ITSs are very effective tools for enhancing the learning process. The evaluations were all independent from one another, taking place at different times (over 3 decades), in different places (9 countries and 4 continents) and in different educational settings. Students who used such a system outperformed students from regular
classes in 92% of the 50 controlled evaluations. Moreover, the improvement was great enough to be considered of substantive importance in 39 of the 50 studies [58].

These extensive reviews support the idea that Intelligent Tutoring Systems aid students and increase their learning gains in the same way a human tutor does. ITS elements are integrated into this work in order to let students know, with the help of the professor, if they responded correctly, and more importantly, what specific aspects are missing from their responses.

2.1.4. Analogy between Assessment Systems and Intelligent Tutoring Systems

The main difference between the two types of systems is their goal. Assessment systems have been created with the purpose of grading or studying the entailment relationships between a student’s response to a question and a correct answer, usually input by the teacher. The feedback is usually given only to the teachers, so they can take action when multiple students encounter difficulties to the same question. When they first appeared, such systems were using simple word overlap to determine the similarity between the two responses. This didn’t make them very reliable, as students tend to use a non-formal language and thus, different words than the instructor. As the research advanced, Latent Semantic Analysis was adopted to evaluate the quality of the student’s understanding [37]. This made the systems more flexible, but still not very reliable. Hybrid machine learning algorithms [92] followed by facet representations of reference answers [80] and reading comprehension approaches, which were also looking at the input text that generated the question [44], have advanced the research during the past recent years. With the development of the Stanford Natural Language Inference dataset, researchers finally had a dataset large enough to apply deep neural networks with word embeddings to detect textual entailment [11, 16, 90, 107]. Although a very difficult task to solve, student response assessment algorithms don’t offer much if they are not incorporated into a larger educational application. Starting from their output, smart algorithms should optimize and structure information in order to present the human instructor, or the students, enhanced feedback that should make the learning process more efficient.
Intelligent Tutoring Systems (ITSs) were initially developed to improve the feedback offered to the students. While clicker systems were offering feedback only to the instructor, ITSs aimed at enhancing the classroom experience by giving students real-time and personalized feedback. For this, ITSs had to bring a user interface (UI) to the assessment systems, optimize the algorithms for running in real-time and structure the information in a way that would be easy to comprehend. One major disadvantage of existing systems is their lack of adaptability to new domains. Most of the ITSs developed so far are either dependent on intensive human labor when adding new questions, or to the type of question being asked, i.e., free-response vs. exact response.

In the future, complete educational applications should be able to offer meaningful feedback to both students and teachers. They should be able to adapt to individual instructor needs and take into consideration preferences of how the lecture should go. By quickly looking at large amounts of external data, it should also be able to make accurate recommendations based on the students’ history and knowledge of what questions should be asked next or what concepts need to be insisted on, in order to have a greater impact on the students. When it comes to the students, the future educational application should be able to take into account the student’s personality, interests, and learning capabilities. Whether some students may be more interested in the subject than others, or this was the first mistake a student made in a long time, the system should be able to act differently. Hints may be offered quicker to struggling students if they are not making adequate progress. Such hints can come from external data or by crowd-sourcing questions.

Adapting to different domains without much human intervention is equally important for such systems to scale properly. The students cannot use different educational applications for each subject as this will confuse them. Moreover, an intelligent system should be able to handle any type of question, whether it requires a free-response or an exact one. In the future, such systems will also evaluate students without having the teacher manually input a reference response. A solution to this will either be crowd-source based, where answers to similar questions will be found and compared against the students’, or based on external
data, such as Wikipedia.

2.2. Fine-Grained Semantic Units

Minimal Meaningful Propositions (MMPs) were introduced in Godea, Bulgarov and Nielsen (2016) [36]. An MMP is defined as being a proposition that cannot be broken down into finer-grained propositions (it is minimal) and still be interpretable without further context (it is meaningful on its own). I make use of MMPs in order to break complex answers into simpler units that are easier to analyze. Breaking the teacher input reference answer into MMPs is the first step in the application and it is the foundation of offering teachers qualitative, structured and real-time feedback. Once the reference answer MMPs are available, I predict their importance relative to the question and understanding relations between them and student responses. Using this information I then create groups of similar responses and select engagement responses to be shown to the instructor. Although the MMP concept is novel, other concepts have been researched in the past with similar goals – breaking down complex semantic structures into fine-grained units that are easier to analyze.

Nielsen et al. (2009) [80] introduced the concept of semantic facets. Rather than strictly checking whether the student’s response is a paraphrase of or entails the reference answer as a whole, the authors are breaking the teacher’s reference answer into facets. Facets are roughly derived from the typed dependencies in a parse and they allow pinpointing the exact concept which the student understood, contradicted or did not address. However, the facets are often too fine grained to be meaningful on their own, out of the context of the proposition of which they are a part. Thus, entailing a semantic facet can be misleading with regard to the context of the question and the student’s understanding.

Elementary Discourse Units (EDUs) were developed for discourse segmentation and are defined as minimal non-overlapping textual spans representing units of discourse. EDUs are generally used as a precursor to identifying relationships between discourse segments, with various applications, such as text summarization, question-answering and dialogue generation. Previous works extracting EDUs have proposed rule-based approaches [87], classification of discourse boundaries [3] and sequence labeling [42]. However, in contrast with
MMPs, EDUs are not necessarily either minimal or meaningful – for example, conditionals required for *meaningful* interpretation of a proposition are not included in the same EDU as their consequent, and a “minimal” discourse unit text span can often be broken into multiple finer-grained *minimal* propositions.

Nenkova and Passonneau (2004) [78] introduced the *Summary Content Unit* (SCU) as a key component of the Pyramid evaluation method for multi-document summarization. SCUs are defined as semantically-motivated, sub-sentential units of variable length and emerge from the annotation of multiple human summaries for the same input. Previous works extract SCUs manually [78, 79] or use topic modeling to match topics with manually-extracted SCUs [41]. To the best of my knowledge, there is no model for automatically constructing SCUs. Another important difference is the input data being used. SCUs are extracted from multiple, well-structured human summaries; whereas, MMPs emerge from a single version of a potentially poorly-structured answer. A review of SCUs also finds that, while they are *meaningful*, they are not necessarily *minimal* – many SCUs are syntactically complex and would be divided into multiple MMPs.

*Discourse commitments* were introduced by Andrew Hickl (2008) [43] in order to facilitate recognizing textual entailment from longer text-hypothesis pairs. This framework depends on the extraction of a subset of the publicly-held beliefs (i.e., discourse commitments) available from the linguistic meaning of a text or hypothesis. They are extracted by first preprocessing the text (part-of-speech tagging, named entity recognition, syntactic and semantic dependency parsing, normalized temporal expressions and coreference resolution) and second, decomposing complex sentences into well-formed simple ones. This is done by using sets of heuristics to transform sentences that contain subordinations, relative clauses, lists or coordination. Nevertheless, the heuristics of discourse commitments are not public and their applicability to general text is unproven. Since they go beyond just extracting the stated semantics to implicit semantics, in my case, the student would have to entail each implicature, which would not be of great values.
2.3. Textual Alignment

Outside of the education field, sentence alignment has been studied in the literature with important implications in fields such as machine translation [110, 111, 115], rewriting systems and paraphrasing [61, 77] or learning text simplification rules [32, 46, 82]. In machine translation, sentence aligned bilingual corpora are a crucial resource [103, 110, 111]. However, aligned bilingual corpora are generally close translations; whereas in monolingual corpora, sentences encounter a much lower level of alignment, with similar content being expressed using very different words, grammatical form or sentence ordering. As a consequence, many of the simple methods that prove to work well in bilingual datasets, such as sentence length correlations, lexical similarity and word overlap [56, 60], are less likely to be effective on monolingual sentence pairs.

Wolk and Marasek, 2014 [110] propose a language independent sentence alignment approach based on Polish to English experiments. The authors used TED Talks as their data but they claim that the model can be used for any text domain or language pair. The method is also fully automatic, and relies on various heuristics for sentence recognition such as synonyms and semantic text structure.

Another important use of sentence alignment in machine translation is described by Xu et al. 2015 [111] in a large scale study where the problem of alignment between textual units for literary text is tackled. The purpose of this task is to help readers better appreciate the content and spirit of a book by referring to the translation of an expression in a more familiar language. The authors use as corpora publicly available novels translated in English, French and Spanish. They use a two-pass alignment process borrowed from Yu et al. 2012 [114], where, as a first pass, computes only high-confidence parallel sentences leaving residual gaps for another, more computationally costly process.

Although alignment in text summarization is fundamentally similar to aligning MMPs, there are important differences. Summarization is generally based on well structured formal sentences; whereas, even our teacher responses are frequently ungrammatical and middle school student responses are extremely colloquial. This changes the dependencies between
the words and makes it difficult to detect and compare important semantic information. Furthermore, the unaligned MMP pairs are still likely to contain overlapping words or information from the question. For example, even though the students may not address the required concepts, they will often talk about very related topics using words taken directly from the question, making the task more difficult. Perhaps, more importantly in this context, not all aligned MMPs have a paraphrase relation – an aligned student response MMP might contradict the reference answer MMP or express a related misconception.

2.4. Textual Clustering

Clustering is the unsupervised classification of patterns (observations, data items, or feature vectors) into groups (clusters) [51]. Clustering has provided solutions to many problems in many different disciplines, and in spite the fact that, at its core, clustering is a very basic task, it has been proven to be extremely useful in numerous textual tasks: document summarization [117], keyphrase extraction [116], sentiment classification [65, 66], document clustering [47] and many others.

The aim of clustering is to discover meaningful patterns in datasets. However, the significance of “meaningful” can differ greatly depending on the nature of the data and the scope of its use. This is one of the reasons why the theoretical foundations of clustering are very scant. Ackerman and Ben-David, (2009) [2] propose a theoretical study on what they call “clusterability”, which is defined as a measure of clustered structure in a data set. The new notion of clusterability is aiming to capture clustering robustness to center perturbations, meaning that, if the data is well-clusterable, the optimal clustering should be robust to (small) center perturbations. One of the study’s conclusions after surveying several notions of clusterability proposed in the literature is that, when a data set is well-clusterable, it is computationally easy to find a near optimal clustering of that data set. The authors also explore the difficulty of computing the degree of clusterability given a data set, and show that, for the notions that recognize a wide range of well-clustered data, this is an NP-hard problem.

While there are many clustering algorithms which have been proposed over the years,
some of them stand out as they have been used extensively for a large variety of tasks. Distance-based clustering algorithms are designed around a similarity measure used to determine the distance between texts. The most common similarity measures used are the euclidean distance and cosine similarity functions.

Hierarchical clustering is of great interest for a huge number of applications. A hierarchical clustering model groups the data into binary-like trees, also called dendrograms. These trees provide a view of the data at different levels of abstraction [118]. Hierarchical clusters can be built in two ways: agglomerative algorithms start clustering from the leaves, grouping the data points two-by-two repeatedly until the whole tree is formed [38, 55]. On the other hand, partitional algorithms start with a cluster of all data points and then repeatedly bisect them until each instance is contained in a cluster by itself [26].

Agglomerative clustering methods are joining data points successively into clusters based on their similarity with one another. Most often, euclidean or cosine similarities are used to determine the closeness of the data points, with three main methods to determine which distance or distances have to be considered: (1) single-linkage – iteratively combines the pair of clusters that contain the closest pair of elements; (2) complete-linkage – combines the two clusters with the worst-case similarity between any pair of data points is considered; and (3) average-linkage – combines the two groups whose average similarity between all pairs of data points is closest.

Another widely adopted category of clustering algorithms is taking as an input a certain number of clusters that needs to be outputted. For example, $k$-means starts off with $k$ random cluster centers and assigns data points to these clusters based on the closest similarity. The cluster centers are then adapted iteratively until convergence or until a certain number of iterations has been reached. Although, generally, $k$-means uses a small number of iterations to converge, it does have some disadvantages. For example, it is sensitive to the initial random cluster centers that it picks. It is also not very suited for some tasks which do not require a certain number of clusters beforehand, in which case, some other measurements have to be made to estimate the number of clusters in the dataset.
The use of clustering in the context of educational applications is limited. In [5], the authors build user models for exploratory learning environments and use clustering to group students who interact with a learning environment in similar ways. The features used to represent the students include frequency of actions, mean and standard deviation of the latency between the actions and eye-tracking data. The generated clusters are then analyzed to decide which groups of students show effective versus ineffective interaction behaviors.

Romero et al. 2013 [91] use clustering to improve the prediction of students’ final performance using data from social network forums. Using data gathered from 114 university students during a computer science course the authors compare various classification algorithms with clustering models to predict if students will fail or pass the course. The results showed that the methods can obtain a final prediction at the end of the course as well as an early prediction before the end of the course, but with an accuracy tradeoff.

Clustering can also be a way to structure feedback and give it meaning by grouping student responses based on their similarity. Horbach et al. (2014) [45] are minimizing teachers’ grading time by exploring the tradeoffs between grading accuracy and reduction of teacher workload. In the context of foreign language learning, the authors focus on automatically scoring similar short responses (usually one or two sentences long) to listening comprehension exercises. They start from the assumption that highly similar student responses are likely to receive the same grade from a teacher and thus, they cluster similar responses and show them to a teacher as a single unit.

Similarly, Basu et al. (2013) [8] target the grading task for short responses by creating clusters and evaluating the number of human actions needed to correct a set of responses. They do so by modeling a distance function by training a classifier that predicts whether two answers should be grouped together.

Whereas these related works cluster student response in order to ease the task of grading responses for the teacher, I use clustering in order to organize the feedback to make it easier for the teacher to understand the classroom situation. Moreover, in my case, the clustering process is also an intermediate step to increasing classroom engagement.
While numerous other clustering algorithms exist, I presented here the most widely adopted and most relevant to this work. In my case, I experimented with both $k$-means and hierarchical models to group student responses. I settled for the hierarchical model since it offers more flexibility, allowing me to select the final number of clusters based on both pre-defined and dynamic parameters. Also, there is no need to re-cluster when a different number of clusters is desired. Instead, I can simply move up or down the dendrogram.

2.5. Textual Similarity

In educational applications, textual similarity is an important step to detect how correct a response is. Usually, a student response is compared against a reference answer and, using textual similarity, one can assess how likely it is that the student responses entails the information present in the reference answer. Textual similarity can also be used to assess how similar are two or more student responses.

SEMILAR is a tool introduced by Rus et al. (2013) [94] which implements a number of word-to-word, sentence-to-sentence and document-to-document semantic similarity measures. The authors implemented both bidirectional measures (paraphrase relations), as well as unidirectional similarity measures (a text $T$ logically entails a hypothesis text $H$ but $H$ does not entail $T$). Some of the measures implemented are described in this section. SEMILAR can provide 3,456 variants of lexical similarity. They are based on different options and parameters, such as: preprocessing options (collocation detection, punctuation, stop word removal, etc.), filtering options (all words, content words, etc.), weighting schemes (global versus local weighting, binary weighting, etc.), and normalization factors (largest text, weighted average, etc.).

Corley and Mihalcea (2005) [23] proposed a greedy word-to-word similarity measure where, for each word uses the maximum similarity score to any word in the other text. The score is then normalized by an IDF-weighted average. Any word similarity measure can be used in their similarity equation, e.g., any WordNet similarity or Latent Semantic Analysis (LSA). SEMILAR also uses word sense disambiguation (WSD) to compute WordNet similarity in two ways: (1) selecting the most frequent sense of a word; and (2) using all
senses of a word and selecting the average relatedness score for each pair of word senses.

Lintean et al. (2010) [68] studied different semantic similarity measures using Latent Semantic Analysis (LSA) [62]. They combine individual word vectors through weighted sums. They experimented with 3 local weights and 3 global ones.

Rus and Lintean (2012) [93] proposed a solution for text-to-text similarity based on word-to-word similarity measures. The lexical matching is based on the optimal assignment problem, a fundamental combinatorial optimization problem which consists of finding a maximum weight matching in a weighted bipartite graph.

Yang et al. (2018) [112] present a novel approach that learns representations for sentence-level semantic similarity using conversational data. Starting from the hypothesis that semantically similar input sentences have a similar distribution of response sentences, the authors test the approach on the semantic textual similarity (STS) benchmark [15] and a question-question similarity subtask from SemEval 2017’s Community Question Answering (CQA) evaluation [76]. The basic model that learns from a Reddit conversation dataset [4] is on par with existing sentence-level encoders on the STS task. However, another multitask model is trained on Reddit and SNLI data [11] to achieve the state-of-the-art for sentence encoding based models on the STS benchmark.

While these text-to-text similarity measure have a broader scope, education-specific systems are often called assessment systems (as described in Section 1.2.2). Text comparisons in educational applications are often made between student responses and correct answers (which are constructed by the teacher or a content matter expert). Thus, most relations between texts are unidirectional (did the student’s response entail the reference answer?), whereas general textual similarity used in information extraction, document summarization, document classification, etc., is looking for paraphrases, or bidirectional relations. Nevertheless, assessment systems often use features and formulas derived from paraphrase detection algorithms.
2.6. Selecting Engaging Responses

In the process of offering teachers real-time and structured feedback, student responses are grouped into a maximum of four clusters. To provide the instructor qualitative insight into student conceptions in each cluster and increase active student participation, I identify and provide teachers an engagement response for each cluster, together with the associated percentage of responses that are also in that cluster. While representativeness is a factor, the engagement response is not necessarily the one that is the closest in meaning to all other responses. Instead, I would like to choose the response that is most likely to lead to interesting classroom discussion, thus encouraging students to participate.

While I am not aware of any other task that chooses an engagement response from a group of responses, some similarities can be identified between this current task and document summarization. Some document summarization approaches identify certain sentences or phrases from the document which are most informative and thus, can be used in a shorter representation of that text. This method, also known as extractive summarization, usually ranks the sentences in the documents according to a variation of scores computed based on a predefined feature set [106].

Extractive summarization methods can also be split into 3 categories [73]: (1) summarization through identification of key terms; (2) scoring sentences with respect to a specific aspect such as topicality, quality or diversity; and (3) by selecting sentences from a ranked list.

Celikyilmaz and Hakkani-Tur, (2010) [14] present an approach that formulates multiple document summarization as a prediction problem. Their hybrid approach first discovers the topic structures of all sentences by constructing a hierarchical probabilistic model using Latent Dirichlet Allocation [9]. Sentences are then represented by meta-features to determine their candidacy for inclusion in the summary. They include nGram meta-features, where the probability of each frequent word (non stop word) is computed and document word frequency meta-features, which identify whether the words in a sentence are specific in a document or are commonly used in the document cluster. In the third step, support
vector regression is used to predict sentence scores.

Li at al., (2013) [64] use the integer linear programming framework (ILP), introduced by Gillick and Favre (2009) [35], in order to rank sentences that maximize the sum of the weights of the language concepts that appear in the summary. They use bigrams with a frequency of at least 3 as their language concepts. The goal is to make the bigram frequency in the system summary as close as possible to that in the reference, optimizing coverage and diversity.
CHAPTER 3

DATA OVERVIEW

3.1. Data

The dataset used for the experiments consists of 315 STEM questions asked in real, middle-school classrooms. Each question comes with an associated reference answer input by the teacher and an average of 22 student responses. The reference answers and the student responses were manually annotated in different ways in order to train and evaluate the quality of the proposed models. Annotations were done by two graduate students from the Education and Linguistics Department, with a third annotator acting as an adjudicator.

First, reference answers were split into Minimal Meaningful Propositions (MMPs; Chapter 4). Then, each of the 1,212 resulting MMPs was annotated with one out of four degrees of importance relative to the question (primary, secondary, extraneous and redundant).

The following example shows a question (Q), the teacher input reference answer (RA) and the annotated MMPs, together with each one’s importance label.

Example (A)

Q: What is an organism?

RA: A complete living thing composed of one or more cells.

MMPs:

(1) An organism is a complete living thing → primary

(2) An organism must be composed of at least one cell → secondary

(3) An organism may be composed of more than one cell → secondary

Understanding labels were also added to student responses relative to each reference answer MMP (understood, misunderstood, not understood). For understood and misunderstood relations, annotating evidence from the student response was mandatory – the piece of text that motivated their decision. A total of 20,815 pairs of reference answer MMPs
and student responses were obtained for this task. Based on the evidence, alignment labels were automatically created for 92,378 pairs of student response MMPs and reference answer MMPs. If an automatically extracted student response MMP contained the evidence, it was marked as aligned with the reference answer MMP.

For example, in response to the question in Example A, one student response (SR) is:

SR: An organism is anything that is living.

Thus, annotators have labeled MMP 1 as being understood by the student, while MMPs 2 and 3 are labeled as not understood.

A variety of these STEM questions were manually grouped together into an average between the two annotators of of 220 clusters. Moreover, each of the 2,000 student responses was also marked with an engagingness score on a scale of 1 to 4, 4 being the highest, to create the first dataset for enabling the computational selection of engagement responses in a classroom.

Once again, starting from the question in Example A, here are some other student responses, how they were clustered and what engagement label each has:

**Cluster 1**

SR: An organism is a living thing that needs nutrients. → 2

SR: Something living that needs nutrients to survive. → 2

SR: An organism is any living thing that needs nutrients that feed its living cells. Also it moves and grows. → 4

**Cluster 2**

SR: An organism is a living thing. → 1

SR: An organism is a living thing that breathes and has DNA. → 2

SR: An organism is any living thing from one cell to one thousand cells. → 4

As can be seen, all students understood the primary MMP that an organism is a living thing. However, more students mentioned that an organism needs nutrients, thus forming a separate cluster of responses (i.e., Cluster 1). In this case, the most engaging response is the one giving more information worthy of discussion: Also it moves and grows. The second
cluster is not focused on *nutrients* but only on the primary MMP that *an organism is a living thing*.

Statistics, examples and further explanations are provided where individual experiments are discussed.

3.2. Preprocessing

One problem with middle-school students is that their responses are often incorrect grammatically and contain misspelled words. Since this work focuses on the analysis of student responses, every task discussed next has to deal with this problem. Thus, as a preprocessing phase, I try to limit the consequences of ungrammaticality and misspellings before every classification problem. Moreover, mistakes are encountered even in teacher input reference answers. For example, consider the following question (Q) and student response (SR):

**Q:** “What are fluids and how they affect motion?”

**SR:** “*i down kow, but snow it makes you go slower than you normally go, because snow is more compact. fluids also effect motion because its harder talk through. fluids are like waetur, snow, and rain*”

While some spelling mistakes can easily be fixed with an automatic spell checker, others depend on the context and they are harder to correct automatically: “*effect*” instead of “affect”, “*talk*” instead of “*to walk*” and “*down*” instead of “*don’t*”. This changes the word embeddings as well as the relations between the words and their sense as perceived by the classifiers. Such mistakes are often encountered as students are either unaware of the correct spelling or not paying enough attention:

**SR:** “*yes because the giviel is weaker than oaw panteit*”

**SR:** “*the elecricoll field around chaed the magnet when a negitiv charge meets with a positive*”.

To account for such errors I incorporated an automatic spell checker in the preprocessing phase. However, instead of simply correcting the misspelled words, I first take into
account all responses from the students, as well as words from the question and reference answer. Before correcting a misspelled word, I look to see which of its five most probable corrected forms were used by other students or the teacher. The stemmed version of words and the Levenshtein distance are also taken into account when comparing words. For example, to avoid replacing a misspelled word with a corrected form that has nothing to do with the current context, I set a Levenshtein distance limit of 2 between the two words (at most 2 substitutions are sufficient to transform the initial word into its correct form). By doing so, I mitigate against replacing misspelled words with real words that have nothing to do with the context of the question.
CHAPTER 4

MINIMAL MEANINGFUL PROPOSITIONS

Consider a sentence to be comprised of a set of related propositions. I define a Minimal Meaningful Proposition (MMP) as a proposition that cannot be broken down into finer-grained propositions (it is minimal) and still be interpretable without further context (it is meaningful on its own). A sentence usually contains more than one MMP. The goal of the work presented in this chapter is to automatically extract MMPs from a question’s reference answer and classify them according to their importance. Thus, this chapter introduces a new knowledge representation that is both minimal and meaningful as well as a recursive MMP extraction algorithm based on structural patterns of a syntactic parse that does not require any human labor after it is trained.

The example below consists of a real question (Q) and reference answer (RA) asked in a classroom, and its human-extracted MMPs.

**Q:** How did Rutherford figure out that atoms are mostly empty space, and that the nucleus is positive?

**RA:** He used gold foil hammered about an atom thick, and placed radium in a lead lined box that emitted positive alpha particles towards the gold foil.

**MMPs:**

1. Rutherford used gold foil with the thickness of an atom.
2. Rutherford placed radium in a lead lined box.
3. The lead lined box emitted positive alpha particles towards the gold foil.

Extracted MMPs can contain information that was initially spread out over the sentence. These fine-grained propositions allow the separation of the different pieces of information expected from a student, the classification of their importance and the ability to infer a relatively fine-grained representation of a student’s understanding. An educational

---

1Parts of this chapter have been previously published, either in part or in full, from Andreea Godea, Florin Adrian Bulgarov, and Rodney Nielsen, Automatic Generation and Classification of Minimal Meaningful Propositions in Educational Systems, COLING, 2016, pp. 3226–3236 [36]. Original article licensed under a Creative Commons Attribution 4.0 International License; authors retain copyright.
application that successfully uses MMPs will be able to tell the teacher which concepts were understood, contradicted, or omitted by the students.

4.1. Data Description

The first stage of the annotation process was identifying the MMPs in the instructors’ reference answers. Annotators were provided guidelines for restating a reference answer as its corresponding set of minimal meaningful propositions, or distinct stand-alone claims, and given several guiding examples.

As shown in Table 4.1, the 315 reference answers were split into 1,212 MMPs with about 3.8 MMPs per question. The average length of a reference answer was about 24.3 words. I further divided the questions into train (208 questions or 66%) and test (107 questions or 33%) sets.

4.2. MMP Extraction Algorithm

A high level summary of the MMP extraction process is as follows. In the first phase, the unique set of syntactic patterns covering all of the gold-standard human generated MMPs in the training data are learned. Then, in the application or testing phase, I process test set sentences recursively, on each call extracting the MMP associated with the longest matching pattern learned from the training set and recursively processing the remainder of the sentence. If part of the sentence remains and no further patterns match, the remainder forms the final MMP. The details of this process follow.

<table>
<thead>
<tr>
<th>Data</th>
<th>#Questions</th>
<th>#Avg. Ans. Length</th>
<th>#MMP</th>
<th>MMP/Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>208</td>
<td>25.5</td>
<td>829</td>
<td>4.0</td>
</tr>
<tr>
<td>Test</td>
<td>107</td>
<td>22.1</td>
<td>383</td>
<td>3.6</td>
</tr>
<tr>
<td>Total</td>
<td>315</td>
<td>24.3</td>
<td>1,212</td>
<td>3.8</td>
</tr>
</tbody>
</table>

Table 4.1. MMP counts and the average length of the reference answers in words.
In the learning phase, the algorithm learns structural templates from a shallow parse (i.e., chunks) of the gold-standard MMPs in the training data. These templates will be used to extract MMPs from test set answers. Figure 4.1 shows an MMP and the structure extracted. Table 4.2 shows the most frequent structures in the dataset. The frequencies follow a Zipfian distribution, with the five most common structures covering almost 50% of the MMPs.

**Figure 4.1. MMP Structure**

<table>
<thead>
<tr>
<th>Rank</th>
<th>Structure</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><em>NP, VP, NP</em></td>
<td>19.5</td>
</tr>
<tr>
<td>2</td>
<td><em>NP, VP, NP, PP, NP</em></td>
<td>11.3</td>
</tr>
<tr>
<td>3</td>
<td><em>NP, VP, PP, NP</em></td>
<td>8.5</td>
</tr>
<tr>
<td>4</td>
<td><em>NP, VP</em></td>
<td>7.6</td>
</tr>
<tr>
<td>5</td>
<td><em>NP, PP, NP, VP, NP</em></td>
<td>2.6</td>
</tr>
</tbody>
</table>

**Table 4.2. Most Frequent MMP Structures**

In the test phase, the algorithm first splits the answer into sentences, which it parses using Stanford CoreNLP [69]. Then, conjunctions are automatically preprocessed to: replace enumerations with a single base phrase type, and split conjoined SVO structures into separate sentences, replicating the subject, verb and or object as appropriate.

Then, for each sentence, the algorithm finds the longest structure matching a pattern learned during training. If the matching pattern only covers a portion of the sentence, the algorithm extracts that portion as an MMP and recursively processes the unmatched portion of the sentence. If any base phrases remain unmatched when the recursion bottoms out, they become the final MMP. Due to the nature of the algorithm, the method is generalizable to different question types or domains.
Table 4.3. MMP Extraction Results

Given the example question in Section 4, the automatically extracted MMPs are:

(1) He used gold foil hammered about an atom thick.
(2) Placed radium in a lead lined box.
(3) Emitted positive alpha particles towards the gold foil.

As can be seen, the automatically extracted MMPs are very similar to the human-generated examples. The major difference being the missing subject in the last 2 MMPs.

Human extracted MMPs:

(1) Rutherford used gold foil with the thickness of an atom.
(2) Rutherford placed radium in a lead lined box.
(3) The lead lined box emitted positive alpha particles towards the gold foil.

4.3. Results

To evaluate the performance, I compared the set of system-generated MMPs with the gold standard for each question. I preprocessed both system-generated and gold-standard MMPs to remove stop words and stemmed the remaining words using the Porter Stemmer. I report the Precision, Recall and $F_1$-score as well as the BLEU score. Precision is computed as the number of matching words divided by the total number of words in the system-generated MMP. Recall is the number of matching words divided by the total number of words in the gold-standard MMP. The BLEU score, introduced by Papineni et al. (2002) [85], is a highly-adopted method for automatic evaluation of machine translation systems. BLEU is based on a modified computation of precision, using the number of matching $n$-grams. It ranges
from 0 to 1, with values closer to 1 representing more similar texts. Using different values of \( n \), different aspects of the evaluation can be measured, adequacy: \( n = 1-2 \), and fluency: \( n = 3-4 \).

MMP-level metric values are based on a greedy iterative alignment of system-generated and gold-standard MMPs, where on a given iteration the algorithm aligns, processes, and then removes the pair with the highest \( F_1 \)-score (or BLEU score). MMP-level values are averaged to get a question-level value, and finally, Table 4.3 presents the average over all questions.

For comparison, I also computed two baselines. The Sentence Baseline is a method in which every sentence of the reference answer is treated as an MMP. In the Predicates Baseline, I build an MMP for every verb predicate in a sentence. Using SENNA [21], I then identify the predicate’s arguments and attach them to the MMP. As can be seen, the proposed method, MMP Extraction, significantly outperforms the two baselines, achieving an \( F_1 \)-score of 0.627 and showing that the pattern matching approach generalizes well on new, unseen questions. The precision is higher than the recall, meaning that the system generated MMPs are shorter than the human-generated ones.

The BLEU score achieved for different \( n \)-gram sizes are also considerably higher than the baselines’. The adequacy scores (unigrams and bigrams) are fairly high for this new task. As I increase the number of consecutive words to be scored, the score drops. This is a normal behavior for tasks where different solutions to a problem exist without any compromises. When using the BLEU metric to score the fluency (trigrams and 4-grams), it is strongly recommended that you have more than one reference solution or, in this case, human-annotated MMPs. When a single reference solution exists, substantially lower values are expected [85]. For comparison, in a translation task, on a test corpus of about 500 sentences, a human translator scored 0.346 against four references and scored 0.257 against two references, when \( n = 4 \) [85].
4.4. Error Analysis

Some of the most common errors occurring in the extraction phase are associated with the subjectivity of the task. Consider the following reference answer and its system-generated MMPs:

**RA:** Once light reaches our eyes, signals are sent to our brain and our brain deciphers the information in order to detect the appearance, location and movement of objects.

(1) Light reaches our eyes.
(2) Signals are sent to our brain.
(3) Brain deciphers the information.
(4) In order to detect the appearance, location and movement of objects.

First, in the human-annotated set of MMPs, 1 and 2 are joined into a single proposition:

*When light reaches our eyes, signals are sent to our brain.*

The annotators believed that the two pieces of information were too dependent to be separated. On the other hand, the system found two different claims and therefore, extracted two propositions. In addition, the fourth MMP generated by the system is dependent on the context and thus, not very good. The annotators broke this piece of text into three different MMPs, one for each element detected by the brain. This issue can be addressed using the semantic roles of constituents. Also, the detection of agents can be improved using coreference resolution.

Other errors made by the system can be fixed by including more syntactic and semantic information. For example, in the MMP *“John realized the mistake”* the *object* has a crucial role but the algorithm does not include it. On the other hand, an MMP such as *“John died”* makes sense and does not require an *object*. The validity of verb usages can be verified using an external resource such as VerbNet [95].
4.5. MMP Importance Classification\(^2\)

As the data shows, not all parts of a teacher’s reference answer are equally important. A student’s understanding cannot be assumed as being wrong just because it contains a different example of the situation than the teacher’s reference answer. Thus, a classification of the importance of each MMP, relative to the question, is introduced. Given a question, its reference answer and its MMPs, the goal is to identify the importance of each MMP with respect to the question.

**Data Description.** In the second stage of the annotation process, MMPs were manually assigned into one of the following classes:

1. **primary:** fundamental to answering question
2. **secondary:** relevant but not integral to the answer – often clarify or qualify a primary MMP
3. **extraneous:** unnecessary or minimally relevant to answering the question
4. **redundant:** contain information directly or indirectly provided by the question

As can be seen in Table 4.4, of the 1,212 total MMPs, 999 were annotated as primary, 145 as secondary, 53 as extraneous and 15 as redundant.

<table>
<thead>
<tr>
<th>Data</th>
<th>Primary</th>
<th>Secondary</th>
<th>Extraneous</th>
<th>Redundant</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>676 (81.5%)</td>
<td>100 (12%)</td>
<td>41 (4.9%)</td>
<td>12 (1.4%)</td>
<td>829</td>
</tr>
<tr>
<td>Test</td>
<td>323 (84.3%)</td>
<td>45 (11.7%)</td>
<td>12 (3.1%)</td>
<td>3 (0.7%)</td>
<td>383</td>
</tr>
<tr>
<td>Total</td>
<td>999 (82.4%)</td>
<td>145 (11.9%)</td>
<td>53 (4.3%)</td>
<td>15 (1.2%)</td>
<td>1,212</td>
</tr>
</tbody>
</table>

| Table 4.4. MMP Importance Labels. |

\(^2\)This section was primarily the work of the first author in Andreea Godea, Florin Adrian Bulgarov, and Rodney Nielsen, Automatic Generation and Classification of Minimal Meaningful Propositions in Educational Systems, COLING, 2016, pp. 3226–3236 [36].
<table>
<thead>
<tr>
<th>Classes</th>
<th>Precision</th>
<th>Recall</th>
<th>F₁-score</th>
<th>Precision</th>
<th>Recall</th>
<th>F₁-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary</td>
<td>0.840</td>
<td>0.970</td>
<td>0.900</td>
<td>0.884</td>
<td>0.976</td>
<td>0.927</td>
</tr>
<tr>
<td>Secondary</td>
<td>0.675</td>
<td>0.250</td>
<td>0.365</td>
<td>0.835</td>
<td>0.303</td>
<td>0.429</td>
</tr>
<tr>
<td>Redundant</td>
<td>0.875</td>
<td>0.662</td>
<td>0.753</td>
<td>0.856</td>
<td>0.797</td>
<td>0.810</td>
</tr>
<tr>
<td>Extraneous</td>
<td>0.882</td>
<td>0.365</td>
<td>0.517</td>
<td>0.950</td>
<td>0.514</td>
<td>0.588</td>
</tr>
</tbody>
</table>

Table 4.5. Training Set 10xCV Results

In the question preprocessing phase some information is filtered out: (1) hints – if it was generic (i.e., the PMI value between any of its words and the content words of the question was below a threshold, \( t = 0.28 \), that was empirically determined based strictly on the training data); (2) content independent punctuated instructions; (3) parentheticals – again, if they contained content independent instructions.

In the second phase of the process key-concepts are identified in the questions that will be used in the feature extractions process. These key-concepts are identified using part-of-speech tags and grammatical dependency relations. For example, given the question: “What evidence can indicate if a change is physical?”, the system identifies the key concepts: evidence and physical change.

The 27 features used by the supervised model were designed based on the question, the MMPs, the reference answer as a whole and the relations between them. Two versions of the question were considered in the process: the original question and a version based strictly on its interrogative and imperative sentences. Some of the features that were explored are: (1) features derived from the question representation; (2) semantic features; (3) syntactic features; (4) features focused on mismatches between the MMP and question; and (5) features focused on overlapping information between the MMP and question.

The results in Table 4.5 show that primary and redundant are the best performing classes, with F₁-scores of 0.9 and 0.753, as they follow more recognizable patterns. While the extraneous class achieves an F₁-score of 0.517, secondary is the worst performing class with
an $F_1$-score of 0.365. Choosing between primary and secondary was hard for both the system and human annotators, suggesting that there is more of a continuum between primary and secondary rather than a sharp decision boundary.
CHAPTER 5

ALIGNMENT OF MINIMAL MEANINGFUL PROPOSITIONS\textsuperscript{1}

Given a question, a reference answer and a student response, the goal is to align each student response MMP with the appropriate reference answer MMP, if such a valid alignment exists. This task is important not only for aligning pairs of MMPs that address the same concept or idea, but also to identify cases where there is one or more reference answer MMPs with no corresponding MMP in the student’s response. In such cases, I can draw the conclusion that the student did not address that specific aspect of the reference answer. Thus, this chapter introduces a computational model for textual alignment that relies on small semantic units, helping it to deal with ungrammatical text while outperforming several alternative systems.

Figure 5.1 illustrates the alignment of MMPs from a student response with MMPs from the corresponding teacher reference answer. Both answers have been broken down into MMPs. The goal is to align the two sets of MMPs so that pairs can be formed addressing the same concept or idea. As can be seen in the figure, the student addresses two of the three MMPs expected by the teacher and the algorithm successfully aligns the right pairs.

5.1. Data Description

Two graduate students from the Education and Linguistics Department established the proper understanding relations between each pair of reference answer MMP and student response – understood, misunderstood or not understood, with a third annotator acting as an adjudicator. For the first two labels, evidence from the student response was mandatory – the piece of text that motivated their decision.

The alignment labels were automatically created by splitting the student response into MMPs using the algorithm described in Chapter 4.2, and using the evidence provided

\textsuperscript{1}Parts of this chapter have been previously published, either in part or in full, from Florin Bulgarov and Rodney Nielsen, Minimal Meaningful Propositions Alignment in Student Response Comparisons, International Conference on Artificial Intelligence in Education, Springer, 2017, pp. 472–475 [12], with permission from Springer.
**Q:** What protection measures do we take when we're doing experiments in the lab?

<table>
<thead>
<tr>
<th>Reference Answer</th>
<th>Student Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>We wear gloves when handling substances and we put on goggles because eyes need protection.</td>
<td>Stuff could get in your eyes so you need to wear glasses at all times when dealing with substances.</td>
</tr>
</tbody>
</table>

**Figure 5.1.** MMP Alignment. The curved arrows signify the alignment of the MMPs.

<table>
<thead>
<tr>
<th>Classes</th>
<th>Questions</th>
<th>Aligned</th>
<th>Not Aligned</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>157</td>
<td>2,632 (4.7%)</td>
<td>52,794 (95.3%)</td>
<td>55,426</td>
</tr>
<tr>
<td>Development</td>
<td>54</td>
<td>977 (5.1%)</td>
<td>17,881 (94.9%)</td>
<td>18,858</td>
</tr>
<tr>
<td>Test</td>
<td>55</td>
<td>778 (4.2%)</td>
<td>17,316 (95.8%)</td>
<td>18,094</td>
</tr>
<tr>
<td>Total</td>
<td>266</td>
<td>4,387 (4.7%)</td>
<td>87,991 (95.3%)</td>
<td>92,378</td>
</tr>
</tbody>
</table>

**Table 5.1.** MMP Alignment Dataset Statistics

by the annotators at the previous step. If the student MMP contained the evidence, it was marked as aligned.

From the total of 315 questions, 266 were annotated for this task. They were split into training, development and test sets following the percentages: 60% training, 20% development and 20% test. Table 5.1 presents the number of *aligned* and *not aligned* instances in each set. A total of 92,378 instances were built, out of which only 4.7%, or 4,387, were in the *aligned* class.

5.2. Classification

An instance in the model is represented by two MMPs, one extracted from the reference answer and the other from the student’s response. I follow a supervised approach to classify the MMP pairs as either *aligned* or *not-aligned* by employing a Random Forest model.
trained on 254 features, briefly described in Table 5.2. The first 48 features describe general relations between the two MMPs, such as the overall similarities between words, relations, Pointwise Mutual Information (PMI) scores, etc. The other subset of features is extracted based on the most likely aligned facet [80] between the two MMPs. The decision to use facet information was motivated by their granularity level, allowing me to pinpoint the most likely understood relation between two MMPs.

To choose the most likely understood facet I first automatically extract all facets from the reference answer MMP. Then, I iterate over them and, for each, I compute the PMI score between the governors and modifiers for each facet in the student response. I select the facet with the highest PMI as being the most relevant, and I use it in the feature extraction

<table>
<thead>
<tr>
<th>General Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 10 Lengths, number of words, number of overlapping words and BLEU score for n = 1, 2, 3 and 4</td>
</tr>
<tr>
<td>11 - 13 Avg, max and min PMI score between RA MMP facets and Student MMP facets</td>
</tr>
<tr>
<td>14, 15 Student MMP contains animate / inanimate pronoun</td>
</tr>
<tr>
<td>16 - 37 Relatedness - indicates the overall similarity between words in the RA MMP and the Student MMP</td>
</tr>
<tr>
<td>38 - 47 Facet Similarity - indicates the overall similarity of facets in the RA MMP and the Student MMP</td>
</tr>
<tr>
<td>48 The facet found to be the most similar</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Facet Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>49 - 56 Governor, modifier and relation individual features (i.e., word, part of speech, category, etc.)</td>
</tr>
<tr>
<td>57 - 78 Governor detailed features (i.e., similarity features between the gov and its best match in the Student MMP)</td>
</tr>
<tr>
<td>79 - 100 Modifier detailed features (i.e., similarity features between the mod and its best match in the Student MMP)</td>
</tr>
<tr>
<td>101, 102 Product between the similarity of the gov / mod and their best matching nodes</td>
</tr>
<tr>
<td>103 - 110 Boolean features indicating whether the Student MMP had an exact match in the facet</td>
</tr>
<tr>
<td>111 - 115 Processed facet features (the relation between the gov and mod, markers for negation, prepositions, etc.)</td>
</tr>
<tr>
<td>116 - 145 Features of the path from the modifier to the governor and between their best matches, such as: direction, length, negations, comparisons between the paths, etc.</td>
</tr>
<tr>
<td>146 - 165 Features of all facets where the modifier is either governor or modifier</td>
</tr>
<tr>
<td>166 - 185 Features of all facets where the governor is either governor or modifier</td>
</tr>
<tr>
<td>186 - 195 Combined scores of all facets in which the gov or the mod occur</td>
</tr>
<tr>
<td>196 - 205 Combined scores of all facets on the path between the mod and the gov</td>
</tr>
<tr>
<td>206 - 214 Features of the best facet match in the Student MMP</td>
</tr>
<tr>
<td>215 - 241 Pronoun coreference features</td>
</tr>
<tr>
<td>242 - 254 Relatedness scores of facets that express relations between higher-level propositions</td>
</tr>
</tbody>
</table>

Table 5.2. Description of features. Gov and mod refer to governor and modifier.
5.3. Results

Because only about 5% of the instances are in the positive (aligned) class, I balanced the training set by weighting the classes. Hence, in the training set, the total weight of the negative instances is equal to the total weight of the positive instances. The test set was maintained at its original unweighted class distribution.

Two baselines were computed to validate the method. The first one is the Word Overlap Baseline in Table 5.3. Here, I remove stop words from each MMP and stem the remaining words using the Porter Stemmer. I then compare the number of overlapping words between the student MMP and the reference answer MMP and, if the overlap is above a threshold $t$, I mark the pair as aligned. The threshold was tuned on the development set by computing the average number of overlapping words for the positive class ($t = 1.5$). The low value of $t$, shows the difficulty of the task, illustrating that students and teachers use different words when expressing the same idea. A second baseline (i.e., Baseline Features) contains 10 features, such as: length of each MMP in words and in characters, number of overlapping words (original and stemmed), and the BLEU scores for $n = 1, 2, 3, \text{ and } 4$ [85]. A Naïve Bayes model is then learned from the training data and applied to unseen instances in the test set. I experimented with various classifiers, but Naïve Bayes was superior.

Two alternative approaches have also been tested to provide a meaningful comparison. For this, I used the SEMILAR toolkit [94] and its implementation of Latent Semantic Analysis (LSA) [62] and Meteor [24]. While LSA has been a known method for years, Meteor is a state-of-the-art algorithm that evaluates machine translation hypotheses by aligning them to the reference translation and then calculating sentence-level similarity scores. Similar to the Word Overlap baseline, a threshold $t$ was estimated on the development set (LSA: $t = 0.3$; Meteor: $t = 0.05$).

The results for the positive class (i.e., aligned) are shown in Table 5.3. As can be seen, the baselines achieve relatively low $F_1$-scores of 0.146 and 0.191, respectively. Latent Semantic Analysis shows an $F_1$-score of 0.303 while the Meteor method achieves an $F_1$-score.
of 0.421. The proposed MMP Alignment algorithm manages to outperform all methods, achieving an $F_1$-score of 0.520, 23% higher than Meteor. While the proposed method has a high precision, the recall is significantly lower. This is due to the low percentage of aligned instances in the dataset, which leads the classifier to over-predict instances in the negative class. However, out of the instances predicted as being aligned, most of the predictions were correct, hence the high precision. For all other algorithms, the recall is higher than precision, meaning that the models return many false positives. This is understandable since, although a reference answer MMP may not be addressed, the student will likely discuss related topics using close words, confusing the classifiers. This issue does not occur in the proposed method, since the features do a much better job of differentiating between aligned and not aligned instances, hence the much higher $F_1$-score.

Additionally, I show results when using the two feature categories individually. The Facet Features (features 49 to 254 in Table 5.2) also perform better than the alternative methods, but they are further helped by the General Features (features 1 to 48), to reach an overall higher performance.
5.4. Error Analysis

The difficulty of the task comes not only from the different styles of language used by the teacher and the student, but also because there are two automatic models working together. Specifically, the gold standard was built utilizing the word sequence from the student answer that the annotators believed justified their annotation indicating the student’s understanding of a reference answer MMP. In the first phase of the preprocessing, when MMPs are extracted, the evidence can be split into more than one MMP. While I experimented with various ways to overcome this, such multiple alignments decrease the model’s performance. Allowing many-to-many alignments between the MMPs is likely to solve this issue.

Another aspect that harms the performance is the high number of related words in the two pairs. Generally, all words in an MMP pair are somehow related, addressing the same general concepts or ideas. Again, this can be confusing for the classifier and thus, it will generate more false positives. Since not all of these words aid the task, I plan to filter out words that do not add any new information such as words contained in the question, which are frequently repeated by students.

5.5. MMP Alignment Disadvantages

In theory, aligning reference answer MMPs with student response MMPs before determining understanding relations should increase the precision of the feedback offered to the teacher. However, I saw that in practice, the alignment information was a step that did not bring in much information to the MMP understanding algorithm. Several uses of the alignment information have been tried – determining understanding relations between aligned MMPs, using the alignment features in the MMP understanding model or simply enhancing the MMP understanding algorithm with alignment predictions.

None of these uses aided the MMP understanding task or added any significant value to the alternative of determining understanding relations between reference answer MMPs and whole student responses. Several other enhancements to the alignment algorithm have also been tried – new features, ablation studies, using word embeddings with deep neu-
ral networks, combining word embeddings and hand-crafted features. Furthermore, several improvements can be made to consider many-to-many relations between the MMPs. For example, a reference answer MMP can be aligned with more than one student response MMP (the vice versa is also true).

After several experiments, comparisons and error analysis, I decided against using the alignment information in the following models.
CHAPTER 6

UNDERSTANDING OF MINIMAL MEANINGFUL PROPOSITIONS\textsuperscript{1}

Rather than strictly checking whether the student’s response is a paraphrase of, or understands the teacher’s reference answer as a whole, I break the target conceptual knowledge into smaller propositions, or clauses (MMPs). This helps separate complex structures in the reference answer and identify which specific propositions or clauses the student understood. The goal of the work carried out in this chapter is to use the student responses to assess whether or not they entail an understanding of a given reference answer MMP – I refer to this task throughout the chapter as MMP Understanding. In the following sections, I report experiments, show results and discuss the main errors that occurred when predicting understanding relations between student responses and reference answer MMPs.

Figure 6.1 exemplifies the approach by showing a real classroom question, the reference answer input by the teacher and a student response. The proposed method first breaks down the reference answer into three MMPs, and then predicts understanding relations between the learner’s response and each of the reference answer MMPs. As can be seen, part of the student’s response is correct, showing that s/he understood that the eyes need protection and goggles are a way to do this. However, s/he does not address the fact that wearing gloves is also a good protection measure while conducting experiments in the lab.

This example shows that instead of having to output a compromised understanding label for the whole reference answer, individual claims can be treated separately. If this information is combined with all the other students’ responses, the instructor can use it to generate insightful classroom discussions and correct most common misconceptions. Nevertheless, this work does not identify misconceptions or score the student responses. Instead, this chapter focuses only on the detection of understood versus not understood MMPs from the reference answer, for each individual student (as shown in Figure 6.1).

\textsuperscript{1}Parts of this chapter have been previously published, either in part or in full, from Florin Adrian Bulgarov and Rodney Nielsen, Proposition Entailment in Educational Applications using Deep Neural Networks, AAAI, 2018 [13], with permission from the Association for the Advancement of Artificial Intelligence.
The work carried out in this chapter consists of two computational approaches for predicting MMP understanding labels – one based on word embeddings input into a two-hidden layer neural network and another based on manually engineered features input into an SVM. Both approaches outperform two alternative variants, a method based on a recent short answer scoring model and another based on a pre-trained, state-of-the-art recognized textual entailment (RTE) system (DiSAN; [96]). Predicting understanding labels at the MMP level will enable us to provide detailed information about the percentage of students who understood each core concept expected by the instructor.

6.1. Data Description

The data used is made up of the same 266 questions used for the alignment task. The same split was done, at the question level (60% training, 20% development and 20% test), which is also why there are a different number of understanding instances for the development and test splits (i.e., the number of responses and MMPs varies for each question).

The two graduate students from the Education and Linguistics Department established the proper understanding relations between each pair of reference answer MMP and student response – understood, misunderstood or not understood, with a third annotator acting as an adjudicator. I treated the misunderstood instances as not understood due to the
Table 6.1. MMP Understanding Dataset (the number of understood pairs for a given question equals its number of reference answer MMPs multiplied by the number of student responses to the question)

<table>
<thead>
<tr>
<th></th>
<th>Questions</th>
<th>Total</th>
<th>Avg.</th>
<th>Understood</th>
<th>Not Understood</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>157</td>
<td>536</td>
<td>3.6</td>
<td>3,380</td>
<td>8,305</td>
<td>11,685</td>
</tr>
<tr>
<td>Development</td>
<td>54</td>
<td>215</td>
<td>4</td>
<td>1,418</td>
<td>3,204</td>
<td>4,622</td>
</tr>
<tr>
<td>Test</td>
<td>55</td>
<td>214</td>
<td>3.9</td>
<td>1,573</td>
<td>2,935</td>
<td>4,508</td>
</tr>
<tr>
<td>Total</td>
<td>266</td>
<td>992</td>
<td>3.7</td>
<td>6,371</td>
<td>14,444</td>
<td>20,815</td>
</tr>
</tbody>
</table>

extremely low number of instances in this class (around 3%). As shown in Table 6.1, there are a total of 20,815 instances for this task, with about 30% of them being in the understood class and the remaining 70% being in the not understood class.

6.2. Classification

A first approach to this task is to use hand-crafted features. The features that I used are similar to those described in Table 5.2. The two main differences are: (1) – following a thorough analysis of the features I discovered that only the numeric and boolean features are helping in this task and thus, I removed nominal features; (2) – facet features are used 3 times – for the least and most likely understood facet and as an average over all facets. The general features describe general relations between the reference answer MMP and the student response, such as the overall similarities and relations between words, Pointwise Mutual Information (PMI) scores, overlapping content, BLEU score [85], etc. The least and most likely understood facets are chosen based on the following algorithm: I automatically extract all facets from the reference answer MMP and compute the PMI similarity between its governors and modifiers and the governors and modifiers from all facets in the student’s response. The facets with the lowest and highest average PMI score are chosen as the least and the most likely understood and are used in the feature extraction process.

The second approach to this task is using word embeddings with a two layer neural...
<table>
<thead>
<tr>
<th>Model</th>
<th>Prec.</th>
<th>Recall</th>
<th>$F_1$-score</th>
<th>Prec.</th>
<th>Recall</th>
<th>$F_1$-score</th>
<th>$F_1$-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Majority Baseline</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.70</td>
<td>1</td>
<td>0.82</td>
<td>0.58</td>
</tr>
<tr>
<td>Latent Semantic Analysis</td>
<td>0.48</td>
<td>0.40</td>
<td>0.44</td>
<td>0.72</td>
<td>0.78</td>
<td>0.75</td>
<td>0.66</td>
</tr>
<tr>
<td>Corley and Mihalcea, 2005</td>
<td>0.50</td>
<td>0.37</td>
<td>0.43</td>
<td>0.72</td>
<td>0.81</td>
<td>0.76</td>
<td>0.66</td>
</tr>
<tr>
<td>Horbach et al., 2013</td>
<td>0.61</td>
<td>0.29</td>
<td>0.39</td>
<td>0.75</td>
<td>0.92</td>
<td>0.83</td>
<td>0.67</td>
</tr>
<tr>
<td>DiSAN, 2017</td>
<td>0.42</td>
<td>0.68</td>
<td>0.52</td>
<td>0.74</td>
<td>0.50</td>
<td>0.60</td>
<td>0.57</td>
</tr>
<tr>
<td>SVM – manual ftrs.</td>
<td>0.73</td>
<td>0.41</td>
<td>0.53</td>
<td>0.76</td>
<td>0.92</td>
<td>0.83</td>
<td>0.73</td>
</tr>
<tr>
<td>NN w/ 50 dim. WE</td>
<td>0.69</td>
<td>0.57</td>
<td>0.63</td>
<td>0.79</td>
<td>0.86</td>
<td>0.83</td>
<td>0.76</td>
</tr>
<tr>
<td>NN w/ 100 dim. WE</td>
<td>0.63</td>
<td>0.58</td>
<td>0.60</td>
<td>0.78</td>
<td>0.82</td>
<td>0.80</td>
<td>0.73</td>
</tr>
<tr>
<td>NN w/ 200 dim. WE</td>
<td>0.63</td>
<td>0.64</td>
<td>0.63</td>
<td>0.81</td>
<td>0.80</td>
<td>0.80</td>
<td>0.74</td>
</tr>
</tbody>
</table>

Table 6.2. MMP Understanding Results (NN = Neural Network; WE = Word Embeddings.)

network. I used GloVe word embeddings with 50, 100 and 200 dimensions pre-trained on six billion tokens from Wikipedia 2014 and Gigaword 5 [86]. Specifically, I computed the average embedding vector for each text (reference answer MMP and student response), and combined them into a single vector by concatenating the element-by-element product vector and absolute difference vector (thus, experiments with 200-dimensional word embeddings resulted in a 400-dimensional input to the neural network).

6.3. Results

In Table 6.2 I report the precision, recall and $F_1$-score for each class (understood and not understood), as well as the weighted average $F_1$-score. While there is no other approach that predicts understanding labels between student responses and reference answer MMPs, I also include the results obtained by a majority baseline, Latent Semantic Analysis (LSA), and Corley and Mihalcea’s [23] unsupervised system for measuring the semantic similarity of texts. The two latter approaches were computed using the SEMILAR toolkit [94]. A score was obtained for each pairing of a reference answer MMP and a student response for the
associated question. All pairs which obtained a score higher than a threshold \( t \), were marked as understood. I used the development set to estimate the best value for \( t \) (LSA: \( t = 0.5 \); Corley and Mihalcea: \( t = 0.6 \)).

A recent method for scoring short answers, proposed by Horbach et al. (2013) [44], was also adapted to this task for a more task-oriented comparison. While their system was altered slightly, both approaches were developed essentially to assess the quality of the students’ responses and the main features remained unchanged:

- **Lemma Overlap**: two lemma overlap features. One normalized by the number of learner answer tokens, the other by the number of tokens in the MMP
- **Dependency Triple Overlap**: four features. Full match between dependency triples (modifier, dependency relation, governor) or a match between the two lemmatized words, both being normalized by either number of tokens (in the MMP or in the student response)
- **WordNet Similarity**, using the aggregation methods proposed by Mohler and Mihalcea [75] and Jiang and Conrath [53]
- **String Similarity**: Levenshtein Distance
- **Number of MMPs in the reference answer**

DiSAN (Directional Self-Attention Network) [96] is a light-weight neural net that learns sentence embeddings based solely on the proposed attention without any RNN/CNN structure. Despite its simple form, DiSAN outperformed complicated Recurrent Neural Network (RNN) models on both prediction quality and time efficiency. At the time of its publication (November 2017), DiSAN obtained the best results on two well-known language inference datasets: Stanford Natural Language Inference (SNLI) [11] (accuracy = 85.62%) and Multi-Genre Natural Language Inference (MultiNLP) [108] (accuracy = 71.1%). Moreover, it also became the state-of-the-art system on other NLP tasks and corpora: Stanford Sentiment Treebank (SST) [102], Sentences Involving Compositional Knowledge (SICK) [71], Customer Review [49], Subjectivity Dataset (SUBJ) [84], TREC question-type classification dataset [67]. Given the fact that the train set used for this task contains about 11,000 ex-
amples, which is significantly less than what it was created for, DiSAN was trained on the SNLI dataset which contains 570,000 human-written English pairs supporting the task of natural language inference or recognizing textual entailment.

The reported pre-trained word embedding results were obtained using Keras [19] with TensorFlow [1]. All word embeddings experiments were enhanced with two additional features, computed by normalizing the number of identical lemmatized words by the number of words in the reference answer MMP and by the number of words in the student response, respectively. The best results on the development set were obtained using a feed forward neural network with two hidden layers, each having 64 hidden nodes. The dropout rate was set to 0.5, for both layers. Only a small number of iterations was needed to reach the results (between 10 and 20). A binary cross entropy loss function and a RMSprop optimizer were used to train the model. All parameters were tuned on the development set, while the reported results were obtained on the test set. A second approach uses the aforementioned manual features, which are fed to an SVM classifier.

As can be seen in Table 6.2, the approach using word embeddings with 50 dimensions achieves the highest weighted average $F_1$-score of 0.76, performing about 13% better than the short answer scoring method and the two alternative approaches, and 33% better than DiSAN. The SVM model using hand crafted features obtained a close $F_1$-score, of 0.73. However, important differences can be observed on the understood class where word embeddings models achieve a significantly higher $F_1$-score of 0.63. This is a notable improvement of about 21% over using DiSAN, which obtained an $F_1$-score of 0.52. In comparison with Horbach et al.’s method, the proposed approach is seeing an increase of 61% on the understood class. On the not understood class, the difference in results between the alternative approaches and the proposed methods is significantly lower, or none in the case of Horbach et al.’s approach. This is mainly due to the effectiveness of classifying instances in this class utilizing only the word overlap, which is generally very low for the not understood class.

In regards to DiSAN, one possible reason for it reaching the lowest weighted $F_1$-score, is that the dataset on which DiSAN was trained (SNLI) and the datasets on which
it obtained state-of-the-art performance contains formal texts, whereas student responses in middle-school classrooms are often informal, containing spelling and grammatical mistakes. It is possible, however, that DiSAN would obtain higher performance if it was trained on the MMP understanding dataset.

It is worth mentioning that during the experimentation phase of predicting MMP understanding labels, other approaches have been tried in the attempt to increase the $F_1$-score on the *understood* class. Some will be briefly summarized here.

One inconvenience with the *understood* class was its lower number of examples (only about 30%). To mitigate against this, one approach that has been tried was by using transductive learning. Here, I trained a classifier on a given training set then made predictions on questions individually. While making these predictions, I moved instances on which the classifier was confident into the training set after which I retrained. To make sure that the confidence is high, I looked at probabilities scores and used ensembles of classifiers. Nevertheless, the performance improvement was not significant using this approach.

Besides the neural network presented in the results sections, other, more complex, neural network architectures were experimented on the development set including Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks. Another experiment included training 2 separate deep neural networks (DNNs) and then combining their output into a third DNN. The 2 initial networks were trained separately, one on manual features and the other on word embeddings. For all experiments, many parameters were tried. These experiments did not achieve an $F_1$-score higher than the one reported, even though some architectures came close. I believe that the reason that a simpler neural network achieves higher results is simply due to the number of instances in the dataset. Training a complex neural network on around 11,000 examples is not enough to yield great performance on this task.

6.4. Error Analysis

A challenge for the proposed approach is that many of the questions in the dataset have multiple valid answers:
“Tell me what you know about acids, bases, salts, reactants, products and the neutralization process!”

Questions like this can be answered correctly in more than one way without making any compromises. One possible approach would be for teachers to supply multiple correct responses, similar to the process followed in evaluating machine translation systems.

The error analysis suggests that first classifying questions according to their expected answer type could substantially improve the ability to determine whether student understanding is entailed. Expecting a certain type of student responses, such as a short response (a noun phrase), free-response (a paragraph), an opinion, an enumeration, etc., could potentially increase the chances of success significantly. In such cases, a modified version of the MMP understanding algorithm can be applied, and educational applications can treat questions differently, adjusting their feedback accordingly.

Another interesting observation made when looking at the data was regarding the word overlap between the reference answer MMPs and the student response. Not understood instances have a very low word overlap, making them easier to classify by straightforward baselines. On the other hand, word overlap for the understood instances varies greatly, reaching an average of only 1.8, when stemmed. While some understood pairs have a high overlap of over four identical content words, others may not have any content words in common. Furthermore, even when the word overlap is low, both understood and not understood pairs will contain related words, because generally, students will answer questions by talking about related concepts and ideas. This makes differentiating between the classes harder.

6.5. Further Experimentation

In the error analysis process I also checked whether the learning algorithm or the features is the cause of the $F_1$-score increase. Moreover, since the manual features and word embeddings are fairly independent of each other, combining them would, in theory, improve the results even more. In addition, I also experimented with adding LSA and Corley & Mihalcea’s scores to the existing feature sets. These results are shown in Table 6.3. As can be seen in rows 1 and 2, SVM achieves a higher $F_1$-score on the understood class when
Table 6.3. MMP Understanding Experiments.

<table>
<thead>
<tr>
<th>No.</th>
<th>Model</th>
<th>Prec.</th>
<th>Rec.</th>
<th>F1-score</th>
<th>Prec.</th>
<th>Rec.</th>
<th>F1-score</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SVM w/ WE</td>
<td>0.74</td>
<td>0.5</td>
<td>0.6</td>
<td>0.77</td>
<td>0.9</td>
<td>0.83</td>
<td>0.75</td>
</tr>
<tr>
<td>2</td>
<td>SVM w/ WE + man.</td>
<td>0.73</td>
<td>0.54</td>
<td>0.61</td>
<td>0.78</td>
<td>0.89</td>
<td>0.83</td>
<td>0.76</td>
</tr>
<tr>
<td>3</td>
<td>NN w/ man.</td>
<td>0.69</td>
<td>0.40</td>
<td>0.50</td>
<td>0.75</td>
<td>0.91</td>
<td>0.82</td>
<td>0.72</td>
</tr>
<tr>
<td>4</td>
<td>NN w/ WE + man.</td>
<td>0.71</td>
<td>0.45</td>
<td>0.55</td>
<td>0.75</td>
<td>0.90</td>
<td>0.82</td>
<td>0.73</td>
</tr>
<tr>
<td>5</td>
<td>SVM w/ man. + LSA</td>
<td>0.73</td>
<td>0.42</td>
<td>0.53</td>
<td>0.76</td>
<td>0.92</td>
<td>0.83</td>
<td>0.73</td>
</tr>
<tr>
<td>6</td>
<td>SVM w/ man. + C&amp;M</td>
<td>0.73</td>
<td>0.42</td>
<td>0.53</td>
<td>0.76</td>
<td>0.92</td>
<td>0.83</td>
<td>0.73</td>
</tr>
<tr>
<td>7</td>
<td>SVM w/ man. + LSA + C&amp;M</td>
<td>0.73</td>
<td>0.42</td>
<td>0.53</td>
<td>0.76</td>
<td>0.92</td>
<td>0.83</td>
<td>0.73</td>
</tr>
<tr>
<td>8</td>
<td>NN w/ WE + LSA</td>
<td>0.61</td>
<td>0.32</td>
<td>0.42</td>
<td>0.71</td>
<td>0.89</td>
<td>0.79</td>
<td>0.66</td>
</tr>
<tr>
<td>9</td>
<td>NN w/ WE + C&amp;M</td>
<td>0.66</td>
<td>0.25</td>
<td>0.36</td>
<td>0.7</td>
<td>0.93</td>
<td>0.8</td>
<td>0.65</td>
</tr>
<tr>
<td>10</td>
<td>NN w/ WE + LSA + C&amp;M</td>
<td>0.6</td>
<td>0.32</td>
<td>0.42</td>
<td>0.71</td>
<td>0.88</td>
<td>0.79</td>
<td>0.66</td>
</tr>
<tr>
<td>11</td>
<td>SVM w/ man.</td>
<td>0.73</td>
<td>0.41</td>
<td>0.53</td>
<td>0.76</td>
<td>0.92</td>
<td>0.83</td>
<td>0.73</td>
</tr>
<tr>
<td>12</td>
<td>NN w/ WE</td>
<td>0.69</td>
<td>0.57</td>
<td>0.63</td>
<td>0.79</td>
<td>0.86</td>
<td>0.83</td>
<td>0.76</td>
</tr>
</tbody>
</table>

WE - Word Embeddings 50 dim.; man. = manual features; NN = Neural Network; C&M = Corley & Mihalcea

fed with word embeddings (WE) as opposed to manual features. In fact, adding the word embeddings to the best SVM approach (row 11) will improve the results to the point where the weighted average $F_1$-score reaches 0.76 (row 2). In contrast, using the manual features with a neural network (NN) (rows 3 and 4), is decreasing the results on the understood class. Adding the WE features on top of manual features is offering a slight improvement. A conclusion that can be drawn from these experiments is that WE features are slightly more helpful than manual features, as we initially saw in Table 6.2. Moreover, even though they are independent of each other, combining them does not offer a significant improvement.

Rows 5 through 10 show experiments when either LSA, Corley & Mihalcea (C&M), or both scores are added to the best approaches (rows 11 and 12). It can be seen that for SVM, adding these measures offers no change in the final $F_1$-scores. On the other hand, the results drop significantly when using NNs. This may happen because the NN is putting too
much weight of the added scores and no longer considers WE features as much as it should.
CHAPTER 7

CLUSTERING STUDENT RESPONSES

The importance of open-ended or constructed student responses has become increasingly clear in helping students acquire knowledge through the means of self-explanation [17, 18]. However, one problem encountered by most instructors when asking such open questions is being able to provide high quality feedback to the students, or even grading the responses. Automatic systems have been able to grade student responses by comparing them with teacher provided reference answers [104] and even by looking at external sources, such as the text from where the question was derived [44]. Intelligent Tutoring Systems (ITSs) have been shown to be very effective in supplying the student with real-time feedback [58, 105]. On the teacher’s end, assessment systems have been effective in grading student responses and minimizing the teacher’s workload [80, 104].

The aforementioned systems aid the learning process by offering feedback to either students or teachers. Although grading systems have made impressive progress in the last years, they are usually not accurate enough that teachers are confident in relying solely on their output. Another issue is their scalability. More and more students have started to enroll in MOOCs (Massive Open Online Courses) which means that instructors cannot check response grades individually in order to give personalized feedback. More importantly, even in small classrooms, it can be difficult and time consuming for teachers to identify patterns of common misunderstandings among students.

I posit that scoring responses is not all that is needed for an educational application to be effective. Feedback information needs to cover more aspects while being communicated in an effective and qualitative manner (rather than by simple grading), and be helpful for both teachers and students. I propose to address the problem of formative assessment by clustering the student responses into groups that suggest similar conceptual beliefs. To group student responses, I am not only using lexical information (word overlap and word similarity) but rather, I exploit the propositional information that resides in the MMPs extracted
from the reference answer, their importance and the MMP understanding relations between individual reference answer MMPs and student responses. The end result is a grouping of student responses that is based on their conceptual understanding of the reference answer, the importance of the concepts understood (or not) and the similarity of their responses. On average, the proposed model reaches a performance of 86.3% of the performance achieved by humans on the same task and dataset.

7.1. Data Description

Even though the grouping of student responses is an unsupervised task, human annotators labeled data for development or tuning in order to improve the clustering algorithm and a held out test dataset to measure the system’s final performance.

After creating specific guidelines and examples, two graduate students performed clustering of more than 2,000 student responses from 100 questions. The annotated question types varied: some required short answers (e.g.: “How many protons are in the carbon atom?”) but most required deeper responses (e.g.: “What is an organism?”). Moreover, some questions may have multiple correct answers (e.g.: “Name one important lab safety procedure and explain why it is important.”).

First, responses that do not directly address the question or that do not make sense were put aside: “i don’t know”, “asdkmdkmdka”, “aaaaaaa”. I refer to these responses as the grab bag and the annotators were instructed not to use them in the clustering. Besides these responses that do not address the question, annotators were also instructed to put the low frequency responses (those that appear only one time and are significantly different from the others) in the grab bag. Second, the annotators were instructed to place all student responses which entail an understanding of the teacher input reference answer in full in a single cluster, to which I will be referring as the reference answer cluster.

The remaining responses were grouped based on the understanding of the different reference answer concepts, or based on their semantic similarity. A few more rules were provided: (1) a maximum of four clusters per question could be made including the reference answer cluster (based on feedback from teachers in user-centered design); (2) a cluster must
have at least two responses, but (3) it could have all of the responses, though that should be rare and would indicate a trivial question.

The three subtables in Table 7.1 show distributional statistics regarding the annotation. As can be seen, the two annotators had very similar distributions in the number of clusters they created.

The average number of responses in the reference answer cluster column shows the average number of responses the annotator placed in the reference answer cluster when they created the specified total number of clusters indicated in column 1. The average responses in grab bag column shows the average number of responses, for questions with the number of clusters indicated in column 1, that either don’t make sense, don’t address the question, or are significantly different from all other responses. When compared one against the other, the two annotators obtained an averaged BCubed score of 0.785 per Amigó et al., [6]. The inter-annotator agreement was substantial, with a weighted kappa of 0.68.

<table>
<thead>
<tr>
<th>#Clusters</th>
<th>#Questions</th>
<th>Avg. Resps. in Ref. Cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A1</td>
<td>A2</td>
</tr>
<tr>
<td>1</td>
<td>18.0</td>
<td>17.1</td>
</tr>
<tr>
<td>2</td>
<td>12.6</td>
<td>9.5</td>
</tr>
<tr>
<td>3</td>
<td>8.3</td>
<td>7.6</td>
</tr>
<tr>
<td>4</td>
<td>6.3</td>
<td>7.4</td>
</tr>
</tbody>
</table>

Table 7.1. Clustering Dataset Statistical Distribution (annotator 1 (A1) had 28 questions with 3 clusters, which had an average of 8.3 student responses in the correct answer cluster and an average of 3.3 responses in the grab bag, outside the 3 annotator-created clusters).

7.2. Clustering Responses

At a high level, the algorithm first takes student responses and represents them as feature vectors which include both proposition understanding features and response similarity features. Then, the feature vectors are input to an agglomerative clustering algorithm which,
starting from each response being assigned to its own unique cluster, groups together pairs of clusters until it ends up with a single cluster at the topmost level of the hierarchy. Once I have the hierarchy (or the dendrogram), I choose the best clusters using a recursive selection algorithm. The final selected clusters are then compared to human annotated clusters and the results are reported and discussed.

7.2.1. Student Response Representation

By splitting the instructor’s reference answer into MMPs, I can make a more thorough analysis between the learner’s answer and the individual claims expressed by the instructor. Moreover, this simplifies the task of predicting MMP understanding relations and creates the ability to cluster student responses based on propositional information. To illustrate, the example below consists of a question (Q), a reference answer provided by an instructor (RA) and its human-extracted MMPs.

**Q:** “What makes popcorn pop?”

**RA:** “Water inside heats up and expands into a gas which causes the kernel to explode.”

**MMPs:**

1. The water inside heats up.
2. The water inside expands into a gas.
3. The expansion of gas causes the kernel to explode.

In this example, the popcorn pops because of the expansion of gas, making the third MMP more important to correctly respond to this question. Hence, the clustering algorithm will favor grouping together responses that contain this information. The system can also be modified in the future to adapt to individual teacher preferences, adjusting the weights based on real-time teacher feedback.

To represent student responses for the clustering algorithm, I use feature vectors with 2 components: (1) reference answer MMP understanding and importance features; and (2) features indicating similarity between student responses.
Reference Answer MMP Understanding and Importance Features. As I already mentioned, each reference answer MMP is classified based on two criteria. First, each MMP is associated with a predicted importance label relative to the question: primary, secondary, redundant and extraneous. Each importance level is associated with a different weight: primary MMPs are weighted 2.5 times more than secondary MMPs and 5 times more than redundant MMPs, while extraneous MMPs are not considered. Second, each reference answer MMP has a predicted understood probability relative to each student response (indicating how likely it is that the student understood that MMP). I combine these two pieces of information by multiplying the weight of the MMP’s importance with the probability of the MMP being understood by the student’s response (as predicted by the classifier). By doing so, I capture the information needed to cluster responses based on the students’ understanding of the concepts expressed by the teacher.

Response Similarity Features. Adding similarity features as a second component in the vector, not only increases the performance but it also provides some important advantages: the clustering succeeds even when the reference answer doesn’t exist, all students completely understood it, or all students completely misunderstood it. In the future, this can also help by dynamically changing the clustering algorithm based on the teacher’s preferences. For example, the teacher can choose to put more weight on the similarity of student responses rather than on their correctness. To capture the similarity of student responses I experiment with two different approaches:

- **Word Embeddings** – each content word in a student’s response is represented by an $n$ dimensional feature vector using pre-trained word embeddings. To create a vector representation for the whole response, I average the word embedding values for all content words. Multiple experiments were performed with a different number of dimensions. In the end, the best results on the development set were obtained with 50 dimensional GloVe word embedding vectors that were pretrained on 6 billion tokens from Wikipedia 2014 and Gigaword 5 [86]. In theory, it should be possible to gain better results by training my own word embeddings. However, there is no
Figure 7.1. Example of a Clustering Dendrogram. The letters represent student responses. The hierarchical agglomerative model eventually clusters all student responses into a single cluster at the topmost level. A recursive selection algorithm then looks for the best possible clusters.

A 6-billion word corpus of middle school science text, let alone middle school student writing about science, and building even the former would be a very time-consuming process that is unlikely to result in substantially better results than those attain from the Wikipedia-based word embeddings.

- **Lexical Weighting** – a second approach was tested by first creating a vocabulary of unique stemmed words from the question, reference answer and student responses. A vector is then created for each response based on the words it contains. Words are weighted differently according to four different scenarios: (1) words that are found only in the reference answer (total weight = 0.45); (2) words found only in the question (total weight = 0.1); (3) words found in both reference answer and question (total weight = 0.225); and (4) words found only in student responses (total weight = 0.225). For each category, the weight for each word is set equal to the total category weight divided by the number of words in that category.
7.2.2. Algorithm

I propose an agglomerative approach that starts with each response in an individual cluster and ends with a single cluster containing all responses. Again, different clustering techniques were tested on the development set\textsuperscript{1}. I experimented with *cosine* and *euclidean* similarity functions and different linkage options (*single, complete, average, mean, centroid*, etc.). In my case, a combination of the *cosine* similarity function and the *average* linkage provided the best results on the development dataset.

The result of the clustering algorithm is a tree diagram (dendrogram). An example of a dendrogram is shown in Figure 7.1\textsuperscript{2}.

Once I end up with a hierarchy of clusters, I implement a recursive selection algorithm that moves down the hierarchy, checking for the best clusters as follows. The recursive selection algorithm starts from the cluster containing all responses and moves down the binary dendrogram to identify groups of responses by looking at several parameters. The main parameter is the one that decides the maximum distance between two responses in order to be considered part of the same cluster. Other constraints include: (1) the maximum number of clusters, which was set to be four based on the feedback received from the instructors; (2) the number of responses that should not be included in any cluster (grab bag).

The selection algorithm is shown in Algorithm 1. The function *getClusters* takes as input the resulting cluster hierarchy output by the agglomerative algorithm. The cluster is split into its *left* and *right* clusters and the distance between them is checked. If the distance is larger than a threshold $t$, the *selectClusters* function is called to consider further dividing these into subclusters; otherwise, I consider the parent cluster final. The value of $t$ was determined empirically based on performance on the development dataset. The *selectClusters* function first checks the size of the cluster: if there is only one student response in it, it treats it as different from all other responses and adds it to the *grabBag*. If there is more than one student response in the cluster, the algorithm checks the distance between

\textsuperscript{1}Other experiments included variants of hierarchical clustering and k-means.

\textsuperscript{2}Source: http://datavizproject.com/data-type/dendrogram/
Algorithm 1 Cluster Selection

Require: $C$ (hierarchy output by agglomerative clustering), $t$ (division threshold)

function getClusters($C$)
    $C_L, C_R ← C$
    if dist($C_L, C_R$) > $t$ then
        selectClusters($C_L$)
        selectClusters($C_R$)
    else
        finalClusters ← $C$
    end
end function

function selectClusters($C$)
    if size($C$) > 1 then
        $C_L, C_R ← C$
        if dist($C_L, C_R$) > $t$ then
            getClusters($C$)
        else
            finalClusters ← $C$
        end
    else
        grabBag ← $C$
    end
end function

Ensure: finalClusters, grabBag

while size(finalClusters) > 4 or size(grabBag) > (size(responses) × 0.25) do
    $t ← t + t × 0.15$
    getClusters($C$)
end while

left and right clusters that are contained in it. If the distance is higher than the threshold $t$, the algorithm recursively calls the getCluster function; otherwise, it considers the cluster final.
<table>
<thead>
<tr>
<th>Method</th>
<th>BCubed relative to:</th>
<th>Average</th>
<th>% of human perf.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Annotator 1</td>
<td>Annotator 2</td>
<td></td>
</tr>
<tr>
<td>Baseline</td>
<td>0.621</td>
<td>0.613</td>
<td>0.617</td>
</tr>
<tr>
<td>Word Embeddings</td>
<td>0.634</td>
<td>0.635</td>
<td>0.634</td>
</tr>
<tr>
<td>Lexical Weighting</td>
<td>0.673</td>
<td>0.646</td>
<td>0.659</td>
</tr>
<tr>
<td>MMP Understanding + Word Embeddings</td>
<td>0.653</td>
<td>0.665</td>
<td>0.659</td>
</tr>
<tr>
<td>MMP Understanding + Lexical Weighting</td>
<td><strong>0.682</strong></td>
<td><strong>0.674</strong></td>
<td><strong>0.678</strong></td>
</tr>
<tr>
<td>Annotator 1</td>
<td>n/a</td>
<td>0.784</td>
<td></td>
</tr>
<tr>
<td>Annotator 2</td>
<td>0.786</td>
<td>n/a</td>
<td>0.785</td>
</tr>
</tbody>
</table>

**Table 7.2. Clustering Results**

Although all parameters were estimated on the development set, some are adjusted automatically. For example, the threshold $t$ is initially set to 0.24. However, if the selection algorithm outputs more than four clusters, the threshold is increased by 15% and the selection is repeated. Also, for a usual question, less than 25% of the responses are allowed to be in the grab bag. If more are found, the threshold is also increased. However, if after the first selection more than 80% of the responses have been found to be in the grab bag (80% of the responses have a distance between them larger than the initial empirically derived threshold), the question will be marked as *creative* and more responses will be allowed to be put in the grab bag.

A unique aspect of the algorithm relies in its representation of student responses. Unlike other student response clustering models which rely on the responses’ similarity, this method also uses MMP understanding features which, as we will see in the next section, help improve the results. This style of feature, which has not been used before, makes this clustering framework unique.

7.3. Results

To validate the approach I compare the resulting clusters with the gold-standard clusters as annotated by two graduate students. For this comparison, I am using the BCubed
measure [6]. The BCubed metric decomposes the evaluation process estimating the precision and recall associated with each instance being clustered. The instance’s precision is computed as the percentage of instances in the same system cluster that were also in the same gold-standard cluster as it. Symmetrically, the recall associated with an instance represents the percentage of instances from its gold-standard cluster that appear in its system cluster. BCubed precision and recall are then combined using the following formula (\(\alpha = \beta = 0.5\)):

\[
BCubed = \frac{1}{\alpha \ast \frac{1}{\text{prec.}} + \beta \ast \frac{1}{\text{recall}}}
\]

For each question, I compute the BCubed score relative to each of the two human annotations, while also reporting the average.

Table 7.2 reports the results. For comparison, I also include the results of a baseline. Here, a vocabulary is created comprised of unique words from the reference answer, question and all student responses. Then, each student response is represented as a binary vector of a length equal to the vocabulary size. I also present a high-bar inter-annotator agreement, computed as the average BCubed measure between annotators. The final column presents the performance of each system method as a percentage of the inter-annotator agreement.

The baseline reaches an average BCubed score of 0.617, or 78.5% of the human performance. The 50 dimensional word embeddings and lexical weighting approaches show increased results, reaching average BCubed scores of 0.634 and 0.659, respectively. When MMP understanding features are added, both approaches increase their results. The MMP Understanding with Word Embeddings method reaches an average BCubed score of 0.659. Combining MMP understanding features and the lexical weighting scheme of words, I reach the best performance: a BCubed score of 0.678 or 86.3% of the human performance. I also experimented with more complicated weighting schemes for the word embeddings approach based on each word’s relevance, applying a higher weight to words found in the reference answer and a lower weight to words found only in the question. However, these variations
7.4. Error Analysis

The aim of clustering is to discover meaningful patterns or instance similarities within datasets. However, the significance of “meaningful” can differ greatly depending on the nature of the data and the scope of its use. This is one of the reasons why the theoretical foundations of clustering are very scant and is also why clustering tasks, in general, are more subjective tasks. The following example illustrates the aspect of clustering subjectivity in the task.

**Q:** What is an atom?

**RA:** A fundamental piece of matter.

**Student Responses:**

(1) A atom is a small thing.

(2) It is a small molecule.

(3) A microscopic piece of something.

(4) An atom is a small particle.

(5) An atom is a small thing that is made up of protons and neutrons and makes up other objects.

At the first glance, some people might put all responses in the same cluster since all refer to something very small and none understands the reference answer. However, the full set of responses is larger and contains responses that are extremely similar to each of the responses presented here. One of the annotators thought that responses 1, 3 and 5 should be in the same cluster, while the other separated them in three different clusters. The annotators also disagreed on responses 2 and 4. The system puts responses 2 and 4 in the same cluster and 1, 3 and 5 in another cluster. All three approaches to cluster this set of responses differ.

The analysis showed that while the clustering performance on the responses of a question can be low when compared against human annotations, the groups created still have
patterns in common. While evaluating and comparing against a gold standard is necessary, the clustering algorithm manages to capture conceptually coherent clusters most of the time. I found that even if the system does not perform as well as humans on a particular question, the system clusters will still provide interesting insights to teachers. This aspect is especially useful in Massive Open Online Courses (MOOCs) where hundreds or thousands of students respond to the same questions.

There are a few measures I took to tackle the aspect of subjectivity. First, specific guidelines were given to the annotators in order to minimize the impact of subjectivity by separating responses that were not meaningful and responses that were both correct and complete (they understood the teacher’s reference answer in full). Only the remaining responses were supposed to be grouped based on the underlying concepts understood by students and based on the semantic similarity of student responses. Second, two different sets of annotations from two different people were obtained. In cases where the algorithm was underperforming only against one annotation, but not on the other, I could attribute the error to subjectivity. The subjectivity is also why I compared the results to each annotator’s clusters separately, rather than build a single adjudicated corpus.

Another issue that arises is that different models are pipelined and thus, the decision of higher level models will depend on the decision of lower level classifiers. In this case, I make clustering decisions partially by counting on the information that comes from the MMP understanding classifier, which in turn relies on the information given to it by the MMP extraction algorithm. In cases where the MMP extraction algorithm fails to identify correct MMPs, the error will be propagated to the resulting clustering and the relationships between the responses will fail to make sense for the teacher.

This is a known issue encountered when multiple models depend on one another. The first and obvious way to handle situations like this is to minimize the errors produced by individual models, something that I did when they were built.

A second way is to decrease as much as possible the dependence between the models. Thus, besides MMP understanding features, the clustering algorithm also uses features to
find independent similarities between responses. As it was already mentioned, two different similarity feature sets are evaluated in the experiments: word embeddings, and lexically weighted features. By not relying entirely on the MMP understanding information I achieve two things. First, I decrease the reliance on the MMP understanding information. By also taking into account independent similarity features, I make sure that any classification errors produced by lower level models will not influence the final results in a significant manner. Second, there are cases where the majority (if not all) students will not address any of the MMPs in the teacher’s reference answer. Since no students will have any MMP understanding information associated with their responses, the clustering will shift its focus to the similarity of student responses regardless of the reference answer. Similarly, if the teacher does not input a reference answer (or it is not meaningful), the same scenario takes place.
The benefits of educational processes are rarely obvious to students. One reason for this is that the rewards are projected far into the future. This is, among others, an important reason for students losing their motivation in school. Often, coming up with new ways to keep students engaged throughout the class is a very important part of a teacher’s job, especially when there is rising research showing that it has become increasingly difficult for teachers to keep students engaged [7]. The reasons for this are vast but strongly correlated to the recent developments outside the classroom: increasing Internet and media use [25]. Moreover, over the last decades, the excessive use of media has also led to a decrease in IQ in children and adolescents [30, 89]. From a research perspective, it has been concluded that with so many factors drawing students’ attention in several different directions, new ways of student engagement practices are needed [7, 100].

Engagement often entails more than motivation. Students may be academically motivated to succeed in school in a general way but they can lack interest in certain tasks or subjects. Engagement calls for special attention to the social context that helps activate the underlaying motivation of students or even generate new motivation [59]. Thus, one of the primary components of effective teaching is student engagement [27]. Newmann, Wehlage and Lamborn [59] defined engagement as active involvement, commitment and concentrated attention as opposed to superficial participation, apathy or lack of interest.

Nevertheless, the level of engagement among children depends on many aspects such as: classroom, teacher, task, exercises, previous knowledge or interest regarding the subject, and many other things. For the unmotivated children, things do not go bad suddenly. Dropping out of school is a gradual process that is caused by disengagement, alienation, tardiness, absenteeism, failing classes and transitions between schools [34]. At the other end of the spectrum, high engagement during tasks in high school has been a significant predictor of motivation and commitment as well as overall performance in college [97, 98].
The goal of the work presented in this chapter is to aid teachers in the process of effectively increasing classroom participation among all students. The first step in reaching this goal was the creation of an online application allowing all students in a classroom to respond to a question. With current technology, such a system that engages all students can easily be developed. Moreover, its usefulness is supported by the familiarity of students with digital devices.

The second step in reaching this goal was being able to provide teachers (and students) with high quality feedback based on the student responses in the classroom and their correctness. Chapters 4, 5 and 6 introduce means of predicting the understood relations between student responses and a teacher input reference answer at a propositional level. Chapter 7 takes all this information (understood relations at a propositional level for each student), adds the similarity of student responses with other student responses and structures it in a way that is easily readable for teachers (creating groups of student responses that convey the same underlying information).

The third step in the task of increasing student participation in the classroom is described in this section. The groups of responses presented to the teacher are enhanced by choosing and presenting a single, representative response that is also engaging. While representativeness is a factor, the engagement response is not necessarily the one that is the closest in meaning to all other responses. Instead, I would like to choose the response that is most likely to lead to interesting classroom discussion, thus encouraging students to participate.

In order to accomplish this, annotations were added to the dataset such that each student response is graded relative to its *engagingness* on a scale of 1 to 4, where 1 is a poor to mediocre choice, 2 is okay, and 3 and 4 are essentially equally very good choices, with 4 being assigned to the single response form the annotator felt would be best to use.

This is the first approach ever to define and create a model for predicting if a student response is engaging or not in the context of classroom discussion. The model described to predict engagement obtains a performance that is 86.8% of the performance achieved by
8.1. Data Description

Once again, guidelines and examples were provided to the two annotators, including: (1) when all clusters are considered, their selected engaging responses need to show diversity – overlapping information among different clusters’ engaging responses should be kept at a minimum; (2) grammaticality should be considered; (3) only one grade of 4 per cluster can be assigned (except when there are multiple identical responses). For the cluster that conveys the same information as the reference answer, high grades should be assigned to responses that are the most similar to the reference answer. The two annotators obtained moderate inter-annotator agreement (weighted Kappa = 0.5), with Table 8.1 showing the number of responses annotated with each grade. The 100 questions were randomly split into train (70%) and test (30%). The parameters for the classifier were estimated based on the average five fold cross-validation score on the training set.

8.2. Classification

In order to show engaging responses to teachers, each student response is classified as engaging or not engaging. In the feature engineering process, several aspects have to be taken into account that the selected response needs to meet. At a high level, features

<table>
<thead>
<tr>
<th>Grade</th>
<th>Annotator 1</th>
<th>Annotator 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>246</td>
<td>283</td>
</tr>
<tr>
<td>2</td>
<td>435</td>
<td>473</td>
</tr>
<tr>
<td>3</td>
<td>653</td>
<td>721</td>
</tr>
<tr>
<td>4</td>
<td>366</td>
<td>328</td>
</tr>
</tbody>
</table>

Table 8.1. Engagingness Annotations – number of responses annotated with each grade
<table>
<thead>
<tr>
<th>No.</th>
<th>Feature Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>no-words-Q</td>
<td>The number of words in the question (Q).</td>
</tr>
<tr>
<td>2</td>
<td>no-words-RA</td>
<td>The number of words in the reference answer (RA).</td>
</tr>
<tr>
<td>3</td>
<td>no-words-SR</td>
<td>The number of words in the student response (SR).</td>
</tr>
<tr>
<td>4</td>
<td>no-words-overlap-SR-Q</td>
<td>The number of words in both student response and question.</td>
</tr>
<tr>
<td>5</td>
<td>no-words-overlap-SR-RA</td>
<td>The number of words in both student response and the reference answer.</td>
</tr>
<tr>
<td>6</td>
<td>primary-average-understood-percentage</td>
<td>The percentage of primary MMPs understood by the student.</td>
</tr>
<tr>
<td>7</td>
<td>primary-average-understood-prob</td>
<td>The average understood probability for primary MMPs.</td>
</tr>
<tr>
<td>8</td>
<td>primary-max-understood-prob</td>
<td>The maximum understood probability for primary MMPs.</td>
</tr>
<tr>
<td>9</td>
<td>primary-min-understood-prob</td>
<td>The minimum understood probability for primary MMPs.</td>
</tr>
<tr>
<td>10-13</td>
<td>secondary</td>
<td>Features 6 through 9 for secondary MMPs.</td>
</tr>
<tr>
<td>14-17</td>
<td>optional</td>
<td>Features 6 through 9 for both extraneous and redundant MMPs.</td>
</tr>
<tr>
<td>18</td>
<td>embedding-distance</td>
<td>Average embedding distance between the student response and the reference answer.</td>
</tr>
<tr>
<td>19</td>
<td>diversity</td>
<td>Boolean feature. A separate algorithm chooses engaging responses based only on diversity.</td>
</tr>
<tr>
<td>20</td>
<td>response-in-correct-cluster</td>
<td>Boolean feature. True if the response is in the correct cluster (the one which understood the reference answer).</td>
</tr>
<tr>
<td>21</td>
<td>grammaticality</td>
<td>Percentage of wrongly written words divided by the number of words in the student response.</td>
</tr>
<tr>
<td>22</td>
<td>distance-from-cluster-average</td>
<td>Cosine distance between the average word embeddings of the student response and the average word embeddings of the cluster.</td>
</tr>
</tbody>
</table>

Table 8.2. Engagingness Features

Features engineered for this task are inspired by the underlying information in the guidelines created for the annotators and are thus meant to cover particularities such as: length of the response, grammaticality, correctness at a propositional level when compared against the teacher input reference answer and diversity – if the responses of a question are grouped into more than one cluster, the selected engaging responses need to be diverse.

Table 8.2 shows a brief description of the features used for this task. Features 1 through 5 consist of word counts and word overlap counts between the student response and
the reference answer or question. Features 6 through 17 cover the aspect of correctness. A total of 12 features are equally split based on the importance of MMPs. For three importance levels: primary, secondary and optional (which includes extraneous and redundant MMPs), four features are selected: the understood percentage, the average understood probability as well as the maximum and minimum understood probabilities. Feature 18, embedding distance, is the average distance between each student response and the reference answer. It is computed based on 50 dimensional GloVe word embeddings trained on Wikipedia 2014 and Gigaword 5 [86]. This feature emphasizes how far away are the responses from the reference answer.

Feature 19, diversity, is obtained based on a separate algorithm that selects cluster representative responses based only on diversity. The algorithm starts by identifying the response that is closest in meaning with the reference answer. This is done by looking at the distances between the vector representations and by heavily considering the number of understood MMPs and their importance, for each student. The cluster of the most accurate response will be marked as correct and the response will be marked as the representative response for the cluster. The selection algorithm uses this response as its starting point. If a correct response is not found (for example, there is no response that understood the most important MMPs from the reference answer), the average vector representation of all student responses will be used as the starting point. Once I have the starting vector (either the most accurate response or the average vector representation), I move on to the next cluster and compute the distance between the starting vector and each response in the cluster. The higher the distance, the more different are the responses and thus, they will be more likely to generate diverse discussion. To choose a diverse engagement response for the \( n \)th cluster, I select the response that has the maximum distance between it and all \( n - 1 \) representative responses that were already selected. The diversity feature is binary based on whether the response was selected as a diverse representative for the cluster or not.

Feature 20 is true if the response is in the correct cluster, which was defined in the previous paragraph. Feature 21 is concerned with the grammaticality of the student
responses. To obtain this value, stop words are removed from the student response and, using a spell checker, the percentage of words that are written correctly are checked. Feature 22 is meant to capture the distance of a student response from the cluster average. This distance is computed using the cosine similarity between the average word embeddings of the student response and the average word embeddings of all responses in the respective cluster.

8.3. Results

Each response in the dataset was annotated with a score ranging from 1 to 4, depending on its engagingness potential in the context of its cluster. Since the difference between responses with scores of 3 and 4 is negligible, both are considered as engaging.

The $F_1$-score is used to determine the performance on the engaging class. For each human-annotated cluster in each question, the $F_1$-score between the system’s predictions and the gold-standard is computed. The score is then averaged for all clusters in the question and for all questions in the test set.

The results of two baselines are also reported. First, the positive-class baseline is computed where all responses are predicted as being engaging in the test set. The reason why the engaging class has more responses is due to the fact that for simpler questions, many of the responses are correct and very similar. The Length & Overlap Baseline uses the size of the responses and word overlap features (features 1 through 5 in Table 8.2) which are input into a Random Forest classifier. The Random Forest model obtained the best

<table>
<thead>
<tr>
<th></th>
<th>vs. Annotator 1</th>
<th></th>
<th>vs. Annotator 2</th>
<th></th>
<th>Average</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Prec. Rec. F$_1$-score</td>
<td></td>
<td>Prec. Rec. F$_1$-score</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Positive-Class Baseline</td>
<td>0.542 1 0.702</td>
<td></td>
<td>0.553 1 0.712</td>
<td></td>
<td>0.707</td>
<td></td>
</tr>
<tr>
<td>Length &amp; Overlap Baseline</td>
<td>0.638 0.73 0.681</td>
<td></td>
<td>0.646 0.758 0.697</td>
<td></td>
<td>0.689</td>
<td></td>
</tr>
<tr>
<td>Deep Neural Network</td>
<td>0.672 0.833 0.743</td>
<td></td>
<td>0.682 0.864 0.762</td>
<td></td>
<td>0.752</td>
<td></td>
</tr>
</tbody>
</table>

Table 8.3. Engagingness Results
cross-validation results on the training set out of several other classifiers that were tried.

In the proposed approach, features are input into a feed forward deep neural network with 3 hidden layers, having 8, 4 and 32 hidden nodes. Rectified Linear Unit (ReLU) was used as the activation function while Adam was chosen as the optimizer. A small learning rate of 0.0005 was used and the neural network finished learning after about 5000 iterations (there was no more improvement in the training cross-validation $F_1$-score).

The results are reported in Table 8.3. As can be seen, the proposed deep neural network obtains, on average, an $F_1$-score of 0.752 on the *engaging* class as opposed to the *Positive-Class Baseline* which obtains an average $F_1$-score of 0.707. The *Length & Overlap Baseline* obtains the lowest average $F_1$-score of 0.689\(^1\).

In a separate experiment, I compare the results obtained by the proposed approach with the results humans would obtain when their annotations are compared using the $F_1$-score. For this experiment to make sense, I test only on the questions from the test set where the BCubed score between the human annotation clusterings is 1 (i.e., the human annotators clustered the student responses in the same way). From the 510 responses that were in the test set in the previous experiment, 124 remain.

The results are reported in Table 8.4. When the human annotations are compared, they obtain an average $F_1$-score of 0.906. The proposed automated method reaches an average $F_1$-score of 0.780 and 0.795. Averaging the results obtained versus the two annotators, we see that the automated method reaches 86.8% of the performance obtained by the annotators.

Since the teacher will see only one engaging response for each cluster, in another experiment I compute the percentage of times the neural network predicts an engaging response as its top pick (*precision@1*). To do so, for each cluster, I select only one response as being the one predicted by the classifier to be the engagement response shown to the teacher. The response is chosen based on the probability estimates returned by the neural network.

\(^1\)A different approach with word-embeddings input into a deep neural network was also tried. The $F_1$-score was lower than the reported results.
Table 8.4. Engagingness Results - Deep Neural Network (DNN) vs. Human Annotators

<table>
<thead>
<tr>
<th>Method</th>
<th>vs. Annotator 1</th>
<th>vs. Annotator 2</th>
<th>Average</th>
<th>% of human perf.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Prec.</td>
<td>Rec.</td>
<td>F1-score</td>
<td>Prec.</td>
</tr>
<tr>
<td>DNN</td>
<td>0.663</td>
<td>0.946</td>
<td>0.780</td>
<td>0.699</td>
</tr>
<tr>
<td>Annotator 1</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>0.872</td>
</tr>
<tr>
<td>Annotator 2</td>
<td>0.944</td>
<td>0.872</td>
<td>0.906</td>
<td>n/a</td>
</tr>
</tbody>
</table>

Table 8.5. Engagingness Results - Precision@1 and Session Success Rate (SSR)

<table>
<thead>
<tr>
<th>Method</th>
<th>Annotator 1</th>
<th>Annotator 2</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision@1</td>
<td>SSR</td>
<td>Precision@1</td>
</tr>
<tr>
<td>Deep Neural Network</td>
<td>0.710</td>
<td>0.876</td>
<td>0.663</td>
</tr>
</tbody>
</table>

In the future, the plan is to allow teachers to click a dropdown box in the application and select another engaging response. In the dropdown menu, the instructor will have access to the next three most probable engaging responses. Thus, in the same experiment, I compute the session success rate. This measure considers it a success each time the classifier predicts an engaging response in its top four picks (again, based on the probability estimates).

The results, reported in Table 8.5, show that, on average, the neural network successfully predicts an engaging response as its top pick 68% of the time. Moreover, if we consider the top four picks (i.e., the session success rate), an engaging response will be found 87% of the time.

Since the instructors will be allowed to replace the engagement response, their input can thus be considered feedback for the neural network. Using this feedback, the neural network can be retrained and, in theory, improved. To test this self-improving neural network, I resample the training and testing sets such that the training set does not contain any duplicated questions. Moreover, from the remaining questions in the training set, I start
with only 33% percent of the questions, while the rest are also moved in the test set.

For each cluster in each question I check the four most confident examples. For each response that was incorrectly labeled by the system as engaging, up to the first example that had a gold standard label of engaging, I replace the system-generated label with the gold-standard label of not engaging. I also make sure that the first example with a gold standard label of engaging is labeled likewise. The data for each question is then added to the training set and the neural network is retrained. The process is repeated until all questions from the test set go through the classifier.

The results, which were averaged over 30 trials, are reported in Table 8.6. There is a statistically significant ($p < 0.05$) improvement when the neural network is retrained. When precision@1 is used, there is an average relative reduction in error of 32%. When session success rate is used, the average relative reduction in error is 27%.

This self-improvement alteration of the system is especially useful due to its minimal overhead. This approach is different from active learning systems which check all instances to find the ones that are best fit to be annotated and moved into training. Instead, I provide the teacher three additional responses ordered by the engagement probability as predicted by the classifier. Thus, it is possible to significantly improve the performance of the engagement classifier by using minimal feedback from the teacher. I believe that in the context of educational applications this method could reasonably be used in classrooms, in real-time.
8.4. Error Analysis

The most common errors (76%) are represented by over-predicting engaging responses (false-positives). From these errors, 87% have an annotated engaging score of 2 and the rest of 13% are have a score of 1. The rest of 24% of errors are false-negatives, out of which 79% were annotated with a score of 3.

To explain why most false-positive errors take place, take a look at the following example:

**Q:** What is the independent variable in an experiment?

**RA:** The independent variable is the variable that is changed or manipulated in an experiment.

**SR 1:** The independent variable is the variable you change in an experiment.

**SR 2:** The independent variable is what you change in an experiment.

**SR 3:** It’s the one you change.

**SR 4:** the variable that u change.

Student responses 1 and 2 are both engaging, but since the first responses is more complete, it is selected as the most fit responses to be engaging. Thus, response 1 is annotated with an engagingness score of 4 and response 2 is annotated with a score of 3. Responses 3 and 4 are both annotated with an engaging score of 2, since they are shorter versions of the first two responses. For this question, more complete variants of the last two responses exists and thus, they are marked as *not engaging*. However, since the underlying information is essentially the same, the algorithm classifies them both as *engaging*, generating false-positive errors.

Moreover, similar to clustering, the selection of engaging responses is a subjective task where the definition of *engaging* can vary greatly depending on the people judging it. The measures took to decrease the subjectivity of the analysis start from the annotation guidelines. Annotators were instructed to label responses on a scale of 1 through 4 instead of binary. Machine translation algorithms are evaluated by comparing the system translation with multiple, correct human translations. Similarly, the aim was to follow this approach.
When I underperform on one of the annotations, I can check the performance on the other. If only one annotator’s solution results in subpar performance, I can assume it is due to subjectivity. I also provide results on questions with high agreement between human annotators in order to set an upper limit for the performance.
CHAPTER 9

CONCLUSIONS

The longer term vision of this work consists of improving the learning experience in classrooms by taking advantage of current technological advancements in areas like Natural Language Processing. The main proposed strategy of achieving this is by increasing the engagement of students in classrooms. Their active participation in deep classroom discussions will help them self assess their current knowledge and freely exchange ideas with other students.

To achieve this, I proposed an approach that analyzes student responses, evaluates their correctness and structures them in a way that will allow the teacher to quickly grasp the level of knowledge of the students throughout the class and take action. One possibility for organizing student responses is to cluster them based on their correctness relative to the reference answer, the importance of the propositions understood and the similarity with other student responses. For this, a recursive cluster selection algorithm was created to choose the best groups of responses from the resulted cluster hierarchy. Overall, the computational model implemented for this task achieves 86.3% of the performance achieved by humans. Once the responses are grouped, the goal is to show the teacher an engaging response for each cluster with the purpose of generating deeply engaging classroom discussions. This is the first computational model developed to predict if a student response is engaging or not, reaching a performance 86.8% of the performance achieved by humans. A self-improving alteration of the system was able to reduce the relative error by up to 32%.

In order to group the responses in a relevant manner and to select a potential engaging response, another major task has to be solved: determining the correctness of a student response relative to the reference answer input by the teacher. To accomplish this, I have developed two computational models, one that uses word embeddings input into a two-hidden layer neural network and another that inputs hand-crafted features into an SVM. Both approaches outperform two alternative variants, a method based on a recent short
answer scoring model and another based on a pre-trained, state-of-the-art recognized textual entailment (RTE) system (DiSAN; [96]). The created model enables us to provide detailed information about the percentage of students who understood each core concept expected by the instructor.

However, a student can understand the teacher reference answer at different levels. The more complex the question, the larger the areas that need to be covered by a correct answer. Hence, the reference answer is split into smaller, meaningful propositions which are analyzed individually. I call this new knowledge representations as MMPs (Minimal Meaningful Propositions) defining them as being propositions that cannot be broken down into finer-grained propositions (they are minimal) and still be interpretable without further context (they are meaningful on their own). Unlike other semantic units, MMPs are extracted automatically based on structural patterns of a syntactic parse.

The attempt to improve the accuracy of the MMP understanding model led me to build another layer of classification. First aligning reference answer MMPs with student response MMPs before determining understanding relations should, in theory, increase the performance and the precision of the feedback offered to the teacher. Hence, I constructed a Random Forest alignment model trained on 254 features that overcame all systems and baselines tested. However, in practice, the extra step of aligning MMPs was not bringing in much information. Although several variants were experimented with, none aided the MMP understanding task in a significant way. Consequently, I decided to perform the MMP understanding task without aligning MMPs, thus predicting understanding relations between reference answer MMPs and whole student responses. Nevertheless, the alignment computational model still achieved a 23% higher $F_1$-score when compared against two alternative systems.

Another contribution brought by this work is the creation of the different datasets used throughout the experimentations. Methodologies, guidelines and annotations were created for four different tasks, having the potential to open new classroom feedback opportunities, thus improving the learning experience for both students and teachers.

84
The longer term goal to increase student engagement through formative feedback should be achieved by offering discussion-relevant feedback to teachers, minimizing their evaluation time and increasing the chances of engaging students in deep discussions. I hope that the end result of this work will also stimulate additional research into real-time, domain independent and personalized educational technology.
REFERENCES


at school and at home, (2015).


[30] Edward Dutton and Richard Lynn, A negative flynn effect in france, 1999 to 2008–9,
Intelligence 51 (2015), 67–70.


[40] Neil T Heffernan and Cristina Lindquist Heffernan, *The assistments ecosystem: build-


Fethi Lamraoui and Philippe Langlais, *Yet another fast, robust and open source sen-
tence aligner. time to reconsider sentence alignment, XIV Machine Translation Summit (2013).


[64] Chen Li, Xian Qian, and Yang Liu, Using supervised bigram-based ilp for extractive summarization., ACL (1), 2013, pp. 1004–1013.


[70] Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli, Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual
entailment., SemEval@ COLING, 2014, pp. 1–8.


Transactions on Speech and Language Processing (TSLP) 4 (2007), no. 2, 4.


[98] David J Shernoff and Lisa Hoogstra, *Continuing motivation beyond the high school*


[100] Erin Sibley, Maria Theodorakakis, Mary E Walsh, Claire Foley, Jessica Petrie, and Anastasia Raczek, The impact of comprehensive student support on teachers: Knowledge of the whole child, classroom practice, and teacher support, Teaching and Teacher Education 65 (2017), 145–156.


[102] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts, Recursive deep models for semantic compositionality over a sentiment treebank, Proceedings of the 2013 conference on empirical methods in natural language processing, 2013, pp. 1631–1642.


[106] Dingding Wang, Tao Li, Shenghuo Zhu, and Chris Ding, Multi-document summarization via sentence-level semantic analysis and symmetric matrix factorization, Proceedings of the 31st annual international ACM SIGIR conference on Research and


[116] Hongyuan Zha, Generic summarization and keyphrase extraction using mutual reinforcement principle and sentence clustering, Proceedings of the 25th annual interna-