2023

Automated short-answer grading and misconception detection using large language models

Nazmul H. Kazi
University of North Florida, nazmulkazi@oxiago.com

AUTOMATED SHORT-ANSWER GRADING AND MISCONCEPTION DETECTION USING LARGE LANGUAGE MODELS

by

Nazmul Hasan Kazi

A thesis submitted to the School of Computing
in partial fulfillment of the requirements for the degree

of

Master of Science

in

Computer and Information Sciences

SCHOOL OF COMPUTING
UNIVERSITY OF NORTH FLORIDA
Jacksonville, Florida

December 2023

Certificate of Approval

The thesis "Automated short-answer grading and misconception detection using large language models" submitted by Nazmul Hasan Kazi in partial fulfillment of the requirements for the degree of Master of Science in Computer and Information Sciences has been approved by the thesis committee:

Dr. Indika Kahanda Committee Chair	Date
Dr. Xudong Liu Committee Member	Date
Dr. Zornitza Prodanoff Committee Member	Date

DEDICATION

This thesis is dedicated to the memory of Alan Turing, the father of theoretical computer science and artificial intelligence. A pioneering mind in the realm of computing, his visionary insights and groundbreaking contributions have shaped the fabric of our technological landscape. His enduring legacy continues to inspire generations of researchers and thinkers, reminding us that the pursuit of knowledge knows no bounds. In his own words, *"This is only a foretaste of what is to come, and only the shadow of what is going to be."* May his spirit of innovation and resilience guide our endeavors and illuminate the path toward the future.

"For myself, I like a universe that includes much that is unknown and, at the same time, much that is knowable. A universe in which everything is known would be static and dull, as boring as the heaven of some weak-minded theologians. A universe that is unknowable is no fit place for a thinking being." — Carl Sagan

ACKNOWLEDGMENT

I express my deepest appreciation to the thesis committee, comprised of Dr. Indika Kahanda, Dr. Xudong Liu, and Dr. Zornitza Prodanoff, for their time, unwavering support, valuable advice, and constructive feedback. I extend my special gratitude to Dr. Kahanda, whose invaluable guidance, financial support through Graduate Research Assistantships, and dedicated supervision have been instrumental to the success of both research endeavors. The generosity of Dr. Kahanda's time, resources, and expertise has significantly contributed to the accomplishment of these research projects. I would like to express my gratitude to our collaborator, Dr. James Becker from Montana State University, for generously providing us with data and annotation, thereby enhancing the depth and quality of our research. Additionally, heartfelt thanks go to Dr. Asai Asaithambi for his invaluable support, time, advice, and mentorship. Their collective contributions have been instrumental in the successful completion of this thesis. I am also grateful to my family, friends, and colleagues for their support, company, and feedback.

We express our gratitude to the National Research Platform for providing access to the Nautilus HyperCluster, which served as the computational backbone for the analyses presented in this thesis.

TABLES OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS & ACRONYMS

2NLI	MNLI along with SNLI
ACC	Accuracy
ALBERT	A Lite BERT
AMD	Automated Misconception Detection
ANN	Artificial Neural Network
API	Application Programming Interface
ASAG	Automated Short-Answer Grading
BERT	Bidirectional Encoder Representations from Transformers
BIN	Binary
CEE	Conservation of Energy Errors
CN	Contradictory
CR	Correct
CVE	Constant Voltage Errors
DNN	Deep Neural Networks
DT	Decision Tree
ET	Extended Tokenizer
F1	F1-score
GPT	Generative Pre-trained Transformer
IIVSM	Ideal Independent Voltage Source Misconception
IR	Irrelevant
ITS	Intelligent Tutoring System
JSON	JavaScript Object Notation
LC	Limited Context
LLMs	Large Language Models
LM	Localized Misconception

M-F1	Macro-averaged F1-score
ML	Machine Learning
MLP	Multi-layer Perceptron
MNLI	Multi-Genre Natural Language Inference
ND	Non-domain
NLI	Natural Language Inference
NLP	Natural Language Processing
P	Precision
PC	Partially correct but incomplete
PCM	Precedence of Current Misconception
R	Recall
RCE	Resistor Combination Errors
ReLU	Rectified Linear Unit
RES	Response-level
RF	Random Forest
RoBERTa	Robustly Optimized BERT Pretraining Approach
RTE	Recognizing Textual Entailment
SENT	Sentence-level
SM	Sequential Misconception
SNLI	Stanford Natural Language Inference
SRA	Student Response Analysis
ST	Stock Tokenizer
STEM	Science, Technology, Engineering, and Math
SVM	Support Vector Machines
TF-IDF	Term Frequency - Inverse Document Frequency

UA	Unseen Answers
UD	Unseen Domains
UQ	Unseen Questions
W-F1	Weighted-averaged F1-score
XLM	Cross-lingual Language Models

ABSTRACT

As education technology continues to evolve, the domains of Automatic Short-Answer Grading (ASAG) and Automated Misconception Detection (AMD) stand at the forefront of innovative approaches to educational assessment. We explore the transformative potential of Large Language Models (LLMs) in revolutionizing these critical areas. Leveraging the remarkable capabilities of LLMs in semantic inference, contextual understanding, and transfer learning, we embark on a comprehensive journey to enhance both ASAG and AMD. On ASAG, we illuminate the efficacy of transfer learning by fine-tuning RoBERTa Large, a state-of-the-art LLM, on task-related corpora, e.g. the Multi-Genre Natural Language Inference (MNLI) corpus. The model's adaptability across unseen questions and domains on the minority class, coupled with its narrowed performance gap from unseen answers, highlights the profound impact of transfer learning on grading diverse student responses. In the emerging realm of AMD, we pioneer a dataset and methodology that inaugurates a new era in misconception detection. Framing the task as Recognizing Textual Entailment (RTE), our approach with RoBERTa Large MNLI captures nuanced misconceptions, unveiling the untapped potential of LLMs in unraveling the intricate landscape of automated misconception detection. The synergy between these endeavors presents a holistic view of the transformative role anticipated for LLMs in automated educational assessment. Our research, spanning adaptability in short-answer grading and groundbreaking advancements in misconception detection, establishes a foundation for a future where LLMs excel not only in understanding nuanced student responses but also in pinpointing and rectifying misconceptions with unparalleled precision. These insights contribute significantly to the dynamic field of educational technology, heralding a new era wherein the full potential of LLMs is utilized to shape the trajectory of educational assessment.

Chapter 1

Introduction

In the landscape of modern education, the quest for personalized and efficient learning experiences has never been more imperative [30]. Instructors strive not only to impart knowledge but also to assess the depth of their students' comprehension to cultivate a deeper understanding of the subject matter among their students. Typically, standardized multiple-choice (i.e., closed-ended) questions are employed in adaptive assessments due to their ease of implementation and evaluation. However, multiple-choice questions can only access a limited scope of knowledge, primarily focusing on factual recall rather than profound understanding and application of concepts [18]. Students might guess correct answers randomly without a full grasp of the materials, leading to an inaccurate assessment of their cognitive level [30].

Besides, the format of multiple-choice questions often provides a binary marking of "correct" or "incorrect" as feedback, potentially constraining students' opportunities to learn from their errors and enhance their understanding of the material. In contrast, open-ended questions frequently compel students to engage in critical and creative thinking, thereby fostering the development of metacognitive skills. This approach permits a more comprehensive evaluation of a student's comprehension and application of concepts [5]. When responding to open-ended questions, students are tasked with generating their own answers and elucidating their thought processes and reasoning. This process facilitates instructors in gaining invaluable insights into their student's learning progress and areas where improvement may be needed.

Moreover, written assignments, in which students express their comprehension of educational materials in their own words, yield invaluable information regarding students' understanding and perspectives on the subject matter [23]. These assignments essentially serve as records of students' cognitive processes, as they chronicle their mental engagement through writing. Employing such writing exercises as a means of assessing students not only verifies their knowledge but also contributes to more favorable learning outcomes.

The education system is meticulously crafted to furnish students with the essential knowledge and skills required for success in life. Nevertheless, students frequently confront obstacles in their educational journey that impede their advancement. Among these challenges, the prevalence of misconceptions—a condition wherein a student holds an erroneous understanding of a specific concept—stands out. These misconceptions, if left unaddressed, can prove detrimental to a student's academic progression and, when left unchecked, may persist and cast a shadow over their future performance.

While the identification and correction of misconceptions are crucial components of effective teaching and learning, it's important to note that detecting misconceptions is distinct from the process of grading student responses. Grading primarily focuses on assessing the correctness or completeness of students' answers to questions and assignments. It provides a quantitative measure of their performance, often in the form of scores or grades. In contrast, detecting misconceptions delves deeper into the qualitative aspects of a student's understanding. It involves identifying not only whether an answer is right or wrong but also why a student might hold a particular belief or misunderstanding. This nuanced approach requires a careful examination of the thought processes and reasoning behind a student's response. It's a formative as-

sessment process aimed at improving a student's conceptual grasp rather than merely assigning a grade.

Nonetheless, manually grading or detecting misconceptions in open-ended responses can prove to be a time-consuming and laborious task [17]. This can divert instructors from their primary objectives of teaching and offering personalized feedback to students, hindering the students' ability to learn from their errors and enhance their understanding. Grading requires meticulous assessment, considering not only correctness but also the depth of understanding, critical thinking, and communication skills displayed by students. Detecting misconceptions adds an additional layer of complexity. It necessitates a deep understanding of both the subject matter and common student misconceptions within that domain. In the context of open-ended questions, where student responses can vary widely in terms of content, quality, and clarity, educators often face a dual challenge. They must efficiently evaluate student responses for correctness while also providing constructive feedback to address misconceptions. This challenge becomes especially pronounced when dealing with a large volume of responses, potentially compromising the quality of feedback and the overall learning experience for students.

Recognizing these challenges, automatic short-answer grading (ASAG), a field of study with over a decade of dedicated research and development, emerges as a pivotal component within intelligent tutoring systems [30]. ASAG involves the automated assessment of the correctness of short answers, allowing for real-time feedback to students and streamlining the evaluation process for instructors. By automating grading, ASAG enables educators to allocate more time to teaching and assisting students while ensuring a consistent and efficient assessment of student responses. ASAG is a challenging task, as it demands proficiency in both semantic inference and textual

entailment recognition. Moreover, ASAG introduces an additional layer of complexity by necessitating transfer learning to ensure cross-domain compatibility, rendering it inherently as a data-driven problem.

There is a burgeoning interest in the development of automated misconception detection (AMD) within student responses [4]. Much like ASAG, AMD necessitates expertise in semantic inference, textual entailment recognition, and transfer learning. However, AMD presents an even greater challenge compared to ASAG. While ASAG aims to determine whether a response is factually accurate or aligns with the expected answer, AMD goes beyond mere correctness and aims to uncover any underlying errors in a student's understanding of the subject matter. AMD is inherently more complex as it requires identifying nuanced misconceptions or errors in a student's reasoning. This entails a deeper understanding of the subject matter and common misconceptions within that domain. Generally, ASAG pursues to render the level of correctness and/or completeness of an answer via a score or grade (interchangeable), such as 8 out of 10, 80%, A, F, pass, etc. This grading framework remains consistent across different domains. In contrast, misconceptions can vary widely within a single domain, shifting from topic to topic (see Appendix A for more details). Consequently, AMD demands models to be pretrained on an extensive range of domain-specific text and relies heavily on transfer learning to encompass domain-specific misconceptions.

Leveraging Large Language Models (LLMs), that have revolutionized the field of Natural Language Processing, presents a remarkable opportunity for both ASAG and AMD. These sophisticated models have demonstrated their prowess in understanding and generating human-like text across a wide array of domains and topics. Their ability to grasp nuances in language, recognize contextual cues, and perform complex semantic inference makes them invaluable assets in educational assessment. Their

capacity to process vast amounts of domain-specific text through transfer learning is particularly advantageous. In essence, LLMs offer a promising avenue for advancing both ASAG and AMD, heralding a new era of sophisticated and efficient educational assessment methods.

In our research, we embark on a comprehensive exploration of both ASAG and AMD. Our investigation into ASAG centers on enhancing the transfer learning capabilities of LLMs through fine-tuning on task-related corpora, a subject that we delve into in Chapter 2. On the expedition of AMD, we harness the power of LLMs to detect prevalent misconceptions among students in an introductory circuit analysis course, a critical endeavor discussed in detail in Chapter 3. These two chapters elucidate our methodologies, findings, and future research directions in these vital areas of educational assessment.

Chapter 2

Enhancing Transfer Learning of LLMs through Fine-Tuning on Task-Related Corpora for Automated Short-Answer Grading

A portion of the research presented in this chapter has been accepted for publication at the International Conference on Machine Learning and Applications (ICMLA) 2023 [15]. This accepted publication encompasses research on classical machine learning models (Table 2.3 & 2.4), fine-tuning a RoBERTa Large model using 3-way labels (Table 2.5), and an exploration of the potential benefits of the task-related MNLI corpus in enhancing the performance of RoBERTa Large in the context of 3-way labeling (Table 2.6).

2.1 Introduction

Adaptive testing and assessment capture the cognitive level of students which is essential for formulating personalized learning routes [30]. Typically, standardized multiple-choice (i.e., closed-ended) questions are used in adaptive assessments due to their simplicity of implementation and assessment. However, multiple-choice questions can only test a limited range of knowledge, mainly focusing on factual recall rather than a deeper understanding or application of concepts [18]. In some cases, students might randomly guess correct answers without a full grasp of the material, leading to an inaccurate reflection of their cognitive proficiency in the assessment [30].

Besides, multiple-choice questions commonly provide binary feedback (correct or incorrect), which might limit the student's ability to learn from their mistakes and improve their understanding of the material. On the contrary, open-ended questions often require students to think critically and creatively. This promotes meta-cognitive skills and allows for a more comprehensive evaluation of a student's understanding and application of concepts [5]. In open-ended questions, students are asked to generate their own answers and explain their thought processes and reasoning. This can help instructors gain valuable insight into their student's learning progress and identify areas for improvement.

However, grading open-ended questions manually can be time-consuming and tedious [17]. This process can detract instructors from their primary objectives of teaching and providing personalized feedback to help students improve upon their mistakes. Therefore, automated short-answer grading (ASAG) is a crucial component of any intelligent tutoring system [30]. ASAG automatically assesses the correctness of short answers, providing real-time feedback to students and helping instructors evaluate students more efficiently. This would allow the instructors to devote more time to teaching and supporting students.

Therefore, there has been a growing interest in developing automated systems that can evaluate student responses to open-ended questions across various domains and topics [30]. ASAG is a challenging task, as it requires both semantic inference and recognizing textual entailment (RTE). Besides, ASAG presents an additional layer of complexity by requiring transfer learning to ensure cross-domain compatibility, making it inherently a data-driven problem. The SemEval-2013 Task 7 [12] and the SciEntsBank dataset included in this challenge are widely used benchmarks for ASAG research.

Various approaches have been proposed but only a handful of researchers have explored the potential of LLMs for this task and the SciEntsBank dataset. LLMs have shown great promise in various natural language processing tasks, including language modeling, machine translation, and sentiment analysis due to their ability to capture complex contextual information and relationships between words and phrases [10, 19]. Compared to classical ML approaches, LLMs have the potential to improve the accuracy and efficiency of ASAG. Dzikovska et al. [12] reported 0.63 as the best weighted-average F1 for 3-way labeling (i.e., classification into three distinct classes), and all the reported models use some form of classical ML. Recent studies demonstrate the superior performance of LLMs in ASAG. Sung et al. [27] achieved a weighted-average F1 of 0.68 using BERT-base [10], while Zhu et al. [30] reported a weighted-average F1 of 0.67 using BERT-base and 0.69 using a BERT-based DNN network. Camus and Filighera [8] fine-tune and compare the model performance of various LLMs including BERT, RoBERTa, AlBERT, XLM, and XLMRoBERTa; RoBERTa Large fine-tuned over the MNLI corpus performed the best with a weighted-average F1 of 0.72.

In this project, we investigate two main research questions related to automated short-answer grading using LLMs. Firstly, we assess the potential of various well-known classical machine learning algorithms to surpass the performance of the lexical baseline established in SemEval-2013 Task 7. The goal is to establish a new baseline for LLMs that underlines the highest potential of popular classical ML algorithms. We test several algorithms, including Decision Tree, Random Forest, Support Vector Machines (SVM), and Multi-layer Perceptron (MLP) with various feature extraction techniques, such as bag-of-words and TF-IDF. Secondly, we investigate whether fine-tuning RoBERTa-Large on a more extensive and diverse corpus, such as the Multi-Genre Natural Language Inference (MNLI) corpus [28], could help the model with semantic inference and transfer learning to boost the model performance. By

addressing these research questions, we aim to contribute to the ongoing efforts to improve the accuracy and efficiency of ASAG using LLMs.

2.2 Dataset

2.2.1 SciEntsBank

We utilized a portion of the Student Response Analysis (SRA) corpus [11], known as the SciEntsBank dataset, for this experiment. The dataset has been annotated with SRA labels by human annotators [12]. The dataset was released with three distinct labeling versions: 5-way, 3-way, and 2-way. We centered our experiments around the 3-way labeling scheme. However, we also employed the 5-way labeling for one of the experiments. In the 5-way labeling scheme, each sample is assigned one

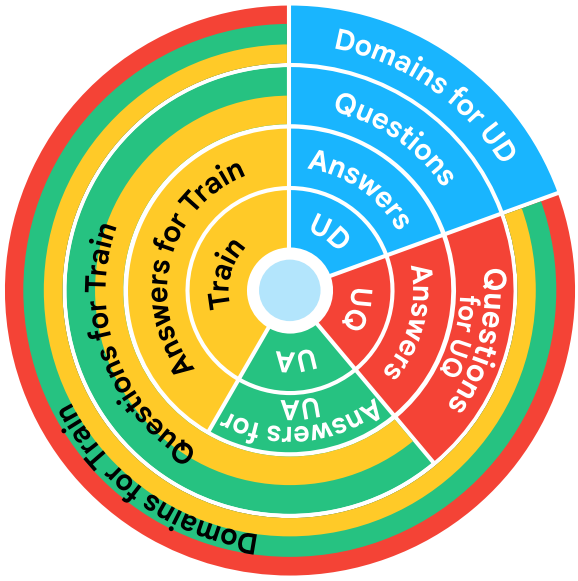


Figure 2.1: An overview of the data distribution adapted to generate three test sets and, subsequently, a training set by sequentially excluding samples based on domain, question, and answer. Each set is represented by a distinct color corresponding to its scope. The outermost ring signifies the domains, followed by the adjacent inner ring representing questions, then the subsequent ring indicating answers, and finally, the innermost ring illustrates the division of the four sets.

of the following labels: “correct”, “partially correct but incomplete”, “contradictory”, “irrelevant”, and “non-domain”. The 5-way labeling scheme is reduced down to a 3-way labeling structure by consolidating “partially correct but incomplete“, “irrelevant”, and “non-domain” into a unified category labeled as “incorrect”.

The SciEntsBank dataset comprises four distinct sets: a training set and three test sets. The sample distribution process among these four sets is illustrated in Figure 2.1. The test sets are tailored to assess the adaptability of a model across various problems and domains. The details regarding the creation and purpose of these test sets are outlined below:

1. **Unseen domains (UD):** The authors set aside the complete set of questions and answers of three science domains from training to create this set. The purpose of this set is to evaluate a model’s versatility and adaptability across diverse knowledge domains.

2. **Unseen questions (UQ):** Within the 12 domains selected for training, the authors randomly selected a subset of questions and held out all responses to these selected questions to create this set. This set evaluates a model’s capacity to accommodate novel questions within familiar domains.

3. **Unseen answer (UA):** From the questions selected for the training set, the authors withheld a subset of randomly selected responses from the training set to create this set. This set of unseen answers is the most typical approach to model evaluation. Its purpose is to assess the model’s proficiency in grading responses it has not previously encountered.

The training set consists of samples that are not present in any of the three test sets.

Labels	Train	Test		
		Unseen Answers	Unseen Questions	Unseen Domains
Correct	2,008	233	301	1,917
Contradictory	499	58	64	417
Partially correct but incomplete	1,324	113	175	986
Irrelevant	1,115	133	193	1,222
Non-domain	23	3	-	20
Total	4,969	540	733	4,562
		5,835		
	10,804			

Table 2.1: Sample distribution of SciEntsBank dataset across a train and three test sets in the 5-way labeling scheme.

Table 2.1 displays the distribution of the 5-way labels in the dataset. Please note that the dataset is imbalanced. In the 3-way labeling scheme, the contradictory class has a significantly low number of samples (only 10% of the dataset) compared to the other two classes: correct and incorrect.

2.2.2 Natural Language Inference (NLI) Corpora

We have utilized two NLI corpora for this research: a) Stanford Natural Language Inference (SNLI) corpus [6], and 2) Multi-Genre Natural Language Inference (MNLI) corpus [28]. These meticulously curated corpora stand as pivotal resources in the field, serving as well-established and widely adopted benchmark datasets for NLI tasks, particularly in recognizing textual entailment (RTE). The SNLI corpus comprises a diverse range of sentence pairs labeled for entailment, contradiction, or neutrality. On the other hand, the MNLI corpus is modeled after the SNLI corpus and extends the scope by including a variety of genres, ensuring a more comprehensive evaluation

Corpus	Train	Validation	Test	Total
SNLI	550,152	10,000	10,000	570,152
MNLI	392,702	20,000	20,000	432,702

Table 2.2: Sample distribution of SNLI and MNLI corpora across train, validation (i.e., eval), and test sets.

of models across different linguistic contexts. It is important to note that there is no sample overlap between the MNLI and SNLI corpora. The sample distribution of these corpora is depicted in Table 2.2.

2.3 Models

2.3.1 Classical ML Models (Baseline)

In this study, we establish a baseline for our deep learning models using four popular classical ML models. These models require feature engineering, selection, and extraction. We use the TfidfVectorizer with the word analyzer to generate features. In this context, features refer to TF-IDF (term frequency-inverse document frequency) values of words. To reduce the dimensionality of the feature space (i.e., the feature matrix where each row corresponds to a sample and each column corresponds to a feature), we remove all English stop words[1] and lemmatize the remaining words. We also include uni-grams and bi-grams in our feature space but limit it to the top (i.e., most relevant) 10,000 features for training.

We experiment with two tree-based models: Decision Tree (DT) and Random Forest (RF). DT is a non-parametric model that generates parameters from the provided

[1]https://ir.dcs.gla.ac.uk/resources/linguistic_utils/stop_words

	Macro			Weighted		
Value of C	0.1	0.5	1.0	0.1	0.5	1.0
Unseen Answers (UA)	0.43	0.48	**0.51**	0.54	0.59	**0.61**
Unseen Questions (UQ)	0.32	0.34	**0.35**	0.42	0.45	**0.46**
Unseen Domains (UD)	0.34	0.36	**0.39**	0.37	0.45	**0.47**

Table 2.3: Performance of Linear SVM in F1 for different C values.

features and builds tree structures for the decision process. On the other hand, RF fits multiple decision trees on different subsets of the dataset and decides on a label by averaging over all decision trees. We use an RF model consisting of one hundred decision trees to estimate the labels.

Another model we explore is the Support Vector Machines (SVM), a supervised learning model that tries to define a hyperplane on the feature space that distinctly separates the data points into distinct classes. SVM has demonstrated promising results in various NLP tasks. We use the linear kernel SVM classifier (as linear kernel works well with TF-IDF features, and is less prone to overfitting on large feature space compared to non-linear kernels) from Sci-Kit Learn, and we experiment with three C values, finding the best performance for $C = 1$ (see Table 2.3).

Additionally, we investigate the performance of an artificial neural network (ANN) model, specifically, the Multilayer Perceptron (MLP) model, a fully connected feed-forward neural network. We use ReLU as the activation function and the *Adam* optimizer with two hidden layers with one thousand and one hundred neurons, respectively, considering the complexity of the problem.

2.3.2 Large Language Models (LLMs)

Large Language Models (LLMs) such as RoBERTa Large have shown remarkable success in natural language processing tasks. RoBERTa Large is a pre-trained LLM that has been trained on a massive amount of unlabelled text data. Fine-tuning a pre-trained RoBERTa Large model on a specific task can lead to significant improvements in performance [19].

Each instance in the SciEntsBank dataset consists of a question, a reference answer, a student answer, and an associated label. The goal is to classify the student answer within the context of the provided question and reference answer. In this project, we frame ASAG as an RTE problem. To achieve this, we construct the premise by concatenating the question with the reference answer. The student answer then serves as the hypothesis in our modeling approach.

We crafted three models, leveraging the pre-trained RoBERTa Large Language Model (LLM) as the foundational base. The objectives and the configuration of these three models are elaborated below:

1. **RoBERTa Large:** We fine-tuned this model solely on the SciEntsBank dataset. The primary purpose was to assess the inherent capabilities of RoBERTa Large on the SciEntsBank dataset.

2. **RoBERTa Large MNLI:** We fine-tuned a RoBERTa Large model on the MNLI corpus and subsequently on the SciEntsBank dataset. this model aimed to investigate the potential performance enhancement achieved by fine-tuning a pretrained model on a task-related corpus.

3. **RoBERTa Large 2NLI:** Impressively, the RoBERTa Large MNLI model exhibited significantly higher performance compared to the RoBERTa Large model. Observing the benefits of fine-tuning on a task-related corpus, we delved deeper into exploring whether fine-tuning on multiple corpora could lead to further improvements in model performance. To this end, we developed this model that underwent fine-tuning on the MNLI corpus, then on the SNLI corpus, and finally on the SciEntsBank dataset. We named this model 2NLI due to its fine-tuning on two NLI datasets.

We defined two distinct sets of hyper-parameters to fine-tune the models, one for fine-tuning on the NLI corpus and another to fine-tune on the SciEntsBank dataset. Both hyper-parameter sets are detailed below:

1. **MNLI & SNLI:** These hyperparameters were adapted from the configuration provided by the authors of the RoBERTa Large model [19]. Our adaptation includes utilizing the Adam optimizer with $\epsilon = 1e-6$, $\beta_1 = 0.9$, $\beta_1 = 0.98$, a learning rate of $2e-5$, and a weight decay of 0.1. The learning rate scheduler type was defined as linear, and a warm-up ratio of 6% facilitated a gradual transition into the main training phase. We fine-tune for a maximum of 10 epochs with early stopping. For MNLI, we used a batch size of 32. The batch size of 64 returned better results (data not shown) for SNLI. Model performance was recorded after each epoch, with the best model chosen based on the epoch displaying the highest macro F1 score.

2. **SciEntsBank:** These hyperparameters were tailored to suit the specificities of the dataset. We employed the Adam optimizer with $\epsilon = 1e-8$, a learning rate of $2e-5$, and a weight decay of 0.01. We establish a warm-up step of 500 to

facilitate a controlled start of the training. We fine-tuned for a maximum of 20 epochs with early stopping and a batch size of 5. We recorded the model performance after each epoch and selected the best model based on the epoch exhibiting the highest macro F1 score.

2.3.3 Experimental Setup

To evaluate the performance of our models, we use the same three metrics as used by [12]: a) accuracy, b) macro-average F1, and c) weighted-average F1. Macro-average F1 calculates the average F1 score for all classes without considering the size of each class. Please note that the dataset is imbalanced as shown in Table 2.1, with the contradictory class having a significantly smaller number of samples compared to the other two classes. Weighted-average F1 considers the size of each class in the calculation and returns a balanced score over imbalanced datasets.

For classical ML models and evaluation metrics, we used the Sci-Kit Learn[2] library. We implemented the LLMs using PyTorch[3] and the Hugging Face Transformers[4] library.

2.4 Results and Discussion

Table 2.4 presents the performances of four classical ML models: Decision Tree (DT), Random Forest (RF), Support Vector Machine (SVM), and Multi-Layer Perceptron

[2]https://scikit-learn.org/
[3]https://pytorch.org/
[4]https://huggingface.co/

Classifier	Unseen Answers			Unseen Questions			Unseen Domains		
	Acc	M-F1	W-F1	Acc	M-F1	W-F1	Acc	M-F1	W-F1
Lexical Baseline [12]	0.56	0.41	0.52	0.54	**0.39**	0.52	**0.58**	**0.42**	**0.55**
Decision Tree (DT)	0.65	0.57	0.64	0.48	0.33	0.44	0.48	0.34	0.43
Random Forest (RF)	0.63	0.54	0.62	**0.56**	**0.39**	**0.53**	0.55	0.37	0.52
Support Vector Machines (SVM)	0.63	0.52	0.61	0.47	0.34	0.46	0.50	0.38	0.47
Multilayer Perceptron (MLP)	**0.67**	**0.63**	**0.67**	0.38	0.33	0.38	0.47	0.39	0.47

Table 2.4: Performance of classical machine learning models on SciEntsBank dataset. Acc: Accuracy, M-F1: Macro-Average F1, W-F1: Weighted-Average F1.

(MLP) on three test sets for 3-way labeling. The table also includes the lexical baseline from SemEval-2013 Task 7 [12]. MLP outperforms all other classifiers on the Unseen Answers test set in all metrics. On the Unseen Questions test set, RF achieves the highest scores in all metrics while achieving the same macro-averaged F1 as the lexical baseline. On the Unseen Domains test set, RF yields the highest accuracy and weighted-average F1 while MLP returns the best macro-averaged F1 among our models but none of the models outperform the lexical baseline. Overall, MLP shows superior performance over other ML models for intra-domain classification whereas RF shows a high potential for transfer learning. We establish a new baseline for the LLMs by taking the maximum score per test set and metric—a method chosen to highlight the best potential performance of classical ML across different aspects—which is denoted as "Baseline" in Table 2.5.

Table 2.5 presents the performance of LLMs on all test sets. The baseline indicates that the LLMs have performed significantly better. The SemEval 2013 Task 7 results are also presented in the table, which shows the maximum scores achieved by any model/algorithm in that competition. These SemEval Best scores serve as another

Classifier	Unseen Answers			Unseen Questions			Unseen Domains		
	Acc	M-F1	W-F1	Acc	M-F1	W-F1	Acc	M-F1	W-F1
Baseline	0.67	0.63	0.67	0.56	0.39	0.53	0.58	0.42	0.55
SemEval 2013 Task 7 (the best score per set and metric) [12]	0.72	0.65	0.71	0.66	0.47	0.63	0.64	0.49	0.62
BERT Base by Sung et al., 2019 [27]	0.76	0.72	0.76	0.65	0.58	0.65	0.64	0.58	0.63
BERT Base by Zhu et al., 2022 [30]	0.74	0.69	0.73	0.66	0.55	0.65	0.66	0.56	0.64
BERT-based DNN by Zhu et al., 2022 [30]	0.77	0.71	0.76	0.69	0.58	0.67	0.66	0.56	0.65
RoBERTa (Large + MNLI) by Camus and Filighera, 2020 [8]	0.79	0.78	0.79	0.66	0.66	0.66	0.72	0.71	0.72
RoBERTa Large	0.76	0.71	0.75	0.65	0.54	0.66	0.67	0.60	0.67
RoBERTa Large MNLI	0.77	0.74	0.77	0.72	0.67	0.72	0.72	0.70	0.72
RoBERTa Large 2NLI	0.78	0.75	0.78	0.72	0.68	0.72	0.72	0.67	0.72

Table 2.5: Performance of large language models (LLMs) on SciEntsBank dataset. Acc: Accuracy, M-F1: Macro-Average F1, W-F1: Weighted-Average F1.

point of comparison for the LLMs. The BERT-base fine-tuned by Sung et al. [27] outperforms the SemEval Best in all test sets and metrics except UQ and UD on the accuracy, whereas the BERT-base fine-tuned by Zhu et al. [30] outperforms the SemEval Best in each test set and metric. However, the BERT-base fine-tuned by Sung et al. [27] performs slightly better than the BERT-base fine-tuned by Zhu et al. [30] based on weighted-average F1 (0.68 vs 0.67). BERT-based DNN performs equally or better than both BERT-base models except for shows under-performance for UA and UD on macro-averaged F1 compared to the BERT-base fine-tuned by Sung et al. [27] (0.71 vs 0.72 and 0.56 vs 0.58, respectively). With a weighted average F1 of 0.69, BERT-based DNN outperforms both BERT-base models. RoBERTa Large underperforms compared to BERT-based DNN on UA and UQ but outperforms the

model on UD in all metrics. RoBERTa Large MNLI outperforms all models on all test sets and all metrics. RoBERTa Large MNLI significantly improves upon the UQ and UD sets compared to any other models on any metrics. Notably, RoBERTa Large MNLI shows closely similar performance on the UQ and UD sets portraying little to no difference between unseen questions and domains while narrowing down the performance gap with UA.

The RoBERTa Large model fine-tuned by Camus and Filighera [8] on the MNLI corpus demonstrates similar performance to our own. While their model achieved higher performance for the UA set on all metrics, our model excels on the UQ set. Notably, both models exhibit almost identical performance on the UD set. We strongly attribute the performance differences to our choice of hyperparameters, emphasizing that batch size can significantly impact the scores as well. Unfortunately, we are unable to compare and highlight the differences in our hyper-parameters since the authors only mentioned the value of the learning rate (same as ours) and the batch size of 16. Importantly, our results not only validate their model performance but also reciprocally affirm our findings. It is crucial to emphasize that the objective of this experiment is not solely to validate their model's performance but rather to investigate the effect of fine-tuning on the MNLI corpus, shedding light on its influence on model performance and its utility in transfer learning—an aspect unexplored by Camus and Filighera [8].

Table 2.6 presents a performance comparison between RoBERTa Large and RoBERTa Large MNLI based on F1 scores. It is evident that fine-tuning the model on the MNLI corpus has significantly enhanced its capacity for semantic inference, resulting in a notable boost in performance, especially within the contradictory class. This improvement is most pronounced in the UQ and UD sets, with increases of 0.31 and

Classifier	Unseen Answers			Unseen Questions			Unseen Domains		
	CR	CN	IN	CR	CN	IN	CR	CN	IN
RoBERTa Large	0.77	0.58	0.78	0.68	0.24	0.70	0.67	0.40	0.72
RoBERTa Large MNLI	0.78	0.65	0.78	0.70	0.55	0.76	0.71	0.64	0.75
Improvement	0.01	**0.07**	0.00	0.02	**0.31**	0.06	0.04	**0.24**	0.03

Table 2.6: Improvement in F1 scores illustrating the impact of MNLI corpus and transfer learning in model performance on 3-way labeling scheme. CR: correct, CN: contradictory, and IN: incorrect.

0.24, respectively. Additionally, the UA set demonstrates a respectable increase of 0.07.

Notably, the SciEntsBank dataset comprises a considerably smaller number of training samples within the contradictory class, accounting for only 10% of the dataset (see Table 2.1). As a consequence, RoBERTa Large initially exhibits comparatively lower performance for contradictory when contrasted with both correct and incorrect across all test sets. However, the fine-tuning of the model on the MNLI dataset has played a pivotal role in enhancing its performance within this minority class through effective transfer learning.

Table 2.7 illustrates the F1 scores of RoBERTa Large and RoBERTa Large MNLI, along with a comparison in the context of the 5-way labeling scheme. When evaluating the average of the weighted-average F1 scores across the test sets, RoBERTa Large demonstrates superior performance in the 3-way labeling scheme compared to the 5-way labeling scheme (0.69 vs 0.56). While maintaining similar performance for the correct class in UA and the contradictory class in UQ, there is an evident increase in performance for the contradictory class in both UA (0.58 vs 0.69) and UD (0.40 vs 0.46), accompanied by a notable decrease in performance for the correct class in both UQ (0.68 vs 0.61) and UD (0.67 vs 0.60), as the dataset transitions from the

Test Set	Model	CR	PC	CN	IR	ND
Unseen Answers	RoBERTa Large	0.78	0.52	0.69	0.72	0.67
	RoBERTa Large MNLI	0.81	0.57	0.73	0.77	0.00
	Improvement	0.03	0.05	0.04	0.05	**-0.67**
Unseen Questions	RoBERTa Large	0.61	0.33	0.25	0.49	-
	RoBERTa Large MNLI	0.60	0.42	0.54	0.59	-
	Improvement	-0.01	**0.09**	**0.29**	**0.10**	-
Unseen Domains	RoBERTa Large	0.60	0.27	0.46	0.51	0.61
	RoBERTa Large MNLI	0.68	0.38	0.64	0.64	0.74
	Improvement	**0.08**	**0.11**	**0.18**	**0.13**	**0.13**

Table 2.7: Changes in F1 scores illustrating the impact of MNLI corpus and transfer learning in model performance on 5-way labeling scheme. CR: correct, PC: Partially correct but incomplete, CN: contradictory, IR: irrelevant, and ND: non-domain.

3-way to the 5-way labeling scheme (see Table 2.6 and 2.7). Intriguingly, the model correctly identifies two out of three non-domain samples, mislabeling only one sample from other classes.

RoBERTa Large MNLI exhibits a marginal performance improvement (0.03-0.05) across all classes, except for the non-domain class in UA, where the model labels no samples, resulting in a score of zero. In UQ, the model demonstrates similar performance for the correct class, while showing a significant increase for other classes, notably a boost of 0.29 for the contradictory class. In UD, there is a consistent performance increase across all classes, with a minimum improvement of 0.08, and the contradictory class experiences the most substantial improvement (0.18). Despite the introduction of additional class labels and potential sample imbalances, the contradictory class shows the most pronounced improvement in both labeling schemes and across all test sets.

Recognizing the potential advantages of the MNLI corpus for enhancing the RoBERTa

Classifier	Unseen Answers			Unseen Questions			Unseen Domains		
	CR	CN	IN	CR	CN	IN	CR	CN	IN
RoBERTa Large MNLI	0.78	0.65	0.78	0.70	0.55	0.76	0.71	0.64	0.75
RoBERTa Large 2NLI	0.79	0.66	0.79	0.72	0.59	0.75	0.72	0.64	0.73
Improvement	0.01	0.01	0.01	0.02	0.04	-0.01	0.01	0.00	-0.02

Table 2.8: Changes in F1 scores illustrating the impact of 2NLI corpus in model performance on 3-way labeling scheme. CR: correct, CN: contradictory, and IN: incorrect.

Large model, we pursued fine-tuning with RoBERTa Large 2NLI to explore the effects of fine-tuning the model across multiple NLI corpora. LLMs are known to be data-hungry, and leveraging additional large corpora for fine-tuning can contribute to enhanced model performance. In this context, the MNLI corpus stands out as one of the largest corpora for RTE, boasting over 392K training samples. Given its substantial size, the MNLI corpus might already provide sufficient training data, potentially limiting the additional benefits gained from incorporating the SNLI corpus. The comparative performance of RoBERTa Large MNLI and RoBERTa Large 2NLI is detailed and analyzed in Table 2.8. RoBERTa Large 2NLI exhibits little to no improvement over RoBERTa Large MNLI across all classes and test sets. The model shows the highest performance increase for the contradictory class (0.04) in UQ and a decrease for the incorrect class (-0.02) in UD. The extensive computing power required for fine-tuning over the SNLI corpus underscores a crucial observation: when the initial corpus, such as MNLI, is substantial in size, the subsequent inclusion of another large corpus, like SNLI, might not contribute to further improvements in model performance. In essence, the findings highlight the diminishing returns of additional corpora, emphasizing the need for judicious consideration of computational resources in the pursuit of model training and optimization.

2.5 Conclusions and Future Work

The ability to comprehend and accurately categorize student answers is a pivotal aspect of automated scoring systems, and it holds great potential to assist students in their educational pursuits. However, this task presents challenges, given the diverse and intricate nature of student responses. Traditional rule-based approaches often struggle to capture the subtleties and nuances of language usage. In contrast, Large Language Models (LLMs) have exhibited exceptional performance in this domain compared to classical machine learning methods.

Through our experimentation, we have uncovered a significant enhancement in model performance by fine-tuning LLMs on the MNLI corpus. This fine-tuning equips the model with a profound understanding of semantic inference, thereby contributing to its improved performance. Our findings underscore the effectiveness of transfer learning, a critical component for tasks such as Automated Short Answer Grading, particularly for ensuring cross-domain adaptability and compliance.

In future research, this experiment can be expanded by including the SRA corpus, which incorporates the Beetle dataset alongside the SciEntsBank dataset. Moreover, given the limited scope for hyper-parameter tuning in the current study, a more thorough investigation into optimal hyper-parameter configurations could be a valuable avenue for future work. In addition, considering the continuous advancements in language models, assessing the performance of more recent models such as Megatron-Turing NLG or PaLM 2 on the SRA corpus presents another promising avenue for future exploration.

The F1 score is a harmonic mean between precision and recall. In the context of

ASAG, one can argue that recall takes precedence over precision and it is more important to minimize false negatives [4]. As a potential enhancement for future assessments, the consideration of the F-beta score, which allows the adjustment of the balance between precision and recall, could provide a more nuanced evaluation tailored to the specific needs. However, it is important to note that in multi-class classification, an increase in recall for one label at the expense of an increase in false positives will lower the recall of another label(s).

Chapter 3

Automated Misconception Detection using LLMs

The educational framework is intricately designed to equip students with the fundamental knowledge and skills crucial for navigating a successful career and a life journey. However, within this carefully crafted system, the presence of misconceptions can pose a significant impediment to a student's educational journey [2]. Misconceptions, whether arising from misunderstood concepts or incomplete information, can create cognitive roadblocks that hinder the acquisition of accurate knowledge [25]. When left unaddressed, these misconceptions tend to perpetuate, leading to a cascade of aftereffects that extend beyond the classroom. Students may find themselves grappling with distorted understandings of foundational concepts, impacting not only their academic performance but also their ability to apply acquired knowledge in real-world scenarios [25]. Therefore, addressing and rectifying misconceptions become crucial steps in fostering a learning environment that nurtures genuine comprehension and paves the way for sustained academic success and practical application of skills in life [3].

Writing exercises play a pivotal role in evaluating students by fostering critical thinking, creativity, and metacognitive skills. The open-ended nature of these questions compels students to delve deep into their understanding of concepts, encouraging them to generate thoughtful and reasoned responses. Through articulating their answers in writing, students not only demonstrate comprehension but also provide insights into their unique thought processes, individual perspectives, and learning progress. Therefore, writing exercises stand as an invaluable pedagogical tool for

academic assessment, enhancing the overall educational experience and preparing students for both academic and real-world challenges.

In addition to their evaluative role, writing exercises serve as an effective means of identifying misconceptions among students [4]. The process of translating knowledge into written form requires a deeper engagement with the subject matter, increasing the likelihood for students to recognize and address incomplete knowledge and simple misunderstandings. Naturally, individuals interpret new information in the context of their existing knowledge [25]. As part of the learning process, there is a tendency to validate the new information against what is already known. However, if the new information seamlessly aligns with existing knowledge without triggering conflicts, it integrates into the individual's knowledge and becomes a reference point for future information [25]. Commonly, individuals do not doubt, question, or re-evaluate the validity of their existing knowledge independently or when it is in contrast to new information. Consequently, misconceptions go unnoticed and unaddressed by the students. Educators who hold accurate knowledge can identify these misconceptions by carefully examining student's reflections of comprehension within products such as writing exercises [20].

Detecting misconceptions in writing exercises can be a labor-intensive and time-consuming task for educators [26]. Analyzing written responses requires a meticulous examination of each student's work, necessitating careful attention to detail and a nuanced understanding of the subject matter [4]. Educators must navigate through diverse perspectives and thought processes articulated in the student's writings, identifying subtle nuances that may indicate misconceptions. Furthermore, providing constructive feedback to address these misconceptions requires additional time and effort [24]. The time commitment for detecting misconceptions demands the class to

be reasonably small. As class sizes increase, this task becomes increasingly challenging, compromising the availability of educators to provide timely and comprehensive feedback [22]. It not only stretches educators thin but also limits the scope of their engagement, making it difficult to address misconceptions effectively in larger classes.

There is a growing interest in advancing the field of automated misconception detection (AMD) systems [4]. The development of AMD involves the integration of semantic inference, recognition of textual entailment, and the utilization of transfer learning techniques. The inherent complexity of AMD lies in its ability to identify subtle misconceptions or errors in a student's reasoning, requiring a profound understanding of the subject matter and the prevalent misconceptions within that particular domain. To effectively fulfill this role, AMD heavily relies on transfer learning to encapsulate domain-specific misconceptions, enabling the system to discern nuanced misunderstandings that may arise in diverse educational contexts.

Several pioneering studies have endeavored to address or automate this intricate educational challenge through a variety of methodologies. Schmidt [25] identified several misconceptions held by students related to chemical terms. They developed test questions that can be easily used by teachers to find out whether these misconceptions appear in a learning group. They found that discussions among students have been a good way of dealing with these problems. Danielsiek et al. [9] identified several core topics that are prone to errors in algorithms and data structures. They verified misconceptions known from the literature and identified previously unknown misconceptions related to algorithms and data structures. They reported on methodological issues and pointed out the importance of a two-pronged approach to data collection. Britos et al. [7] presented a work in progress that focuses on data mining tools used for discovering programming misunderstandings of students. They used decision trees to

discover knowledge in rule format that constitutes a model for detecting misconceptions in learning processes. Michalenko et al. [21] introduced a novel NLP framework, employing probabilistic classical machine learning models to detect misconceptions in students' textual responses. Yang et al. [29] took a unique approach by employing concept maps and lists to diagnose misconceptions in circuit courses, encouraging students to actively identify and rectify their misconceptions. Arbogast and Montfort [1] delved into linguistic indicators derived from transcripts and computational grammar to correlate linguistic changes with conceptual understanding in engineering topics. Elmadani et al. [13] explored data-driven techniques to uncover students' misconceptions within interactions with Intelligent Tutoring Systems (ITSs), highlighting the potential of ITS-based AMD. Goncher et al. [14] ventured into automated text analysis, focusing on thematic analysis and high-level concept extraction from open-ended student responses to assess conceptual understanding.

Despite these valuable efforts, AMD has received relatively little attention compared to other educational assessment fields. The major drawback of previous methods lies in their limited capacity to address nuanced misconceptions and the need for manual rule-based coding, which is time-intensive and may lack scalability. However, LLMs have the potential to revolutionize AMD. With their pre-trained knowledge and contextual understanding, these models can efficiently capture complex linguistic patterns indicative of misconceptions. Surprisingly, to date, no researchers have harnessed the capabilities of LLMs for AMD, presenting a significant opportunity to explore uncharted territory in the quest for advancing misconception detection in education.

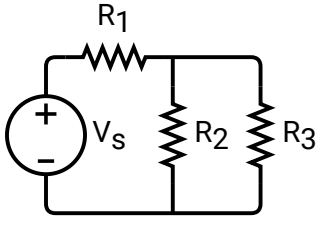
In this project, we aim to develop a framework that leverages the capabilities of LLMs to detect common misconceptions prevalent among students enrolled in introductory

STEM courses. We are collaborating with Montana State University to obtain data for this endeavor, focusing on an introductory circuit analysis course titled *EELE 201: Circuits I for Engineering* [4]. Our overarching goal is to design and develop an AMD system adaptable across a wide spectrum of STEM courses. By achieving this, we intend to empower students to gain a deeper and clearer understanding of the subject matter.

Our research and the outcomes of this project are structured into three phases. As we progressed through each phase, we aimed to address the limitations identified in the preceding stage, leveraging the lessons learned to enhance our methodology. Each phase is allocated a dedicated section, where we thoroughly examine the associated data, methodology, and results specific to that phase. Section 3.1 details the quiz question used across all three phases to collect the student responses.

3.1 Writing Exercise

Figure 3.1 illustrates the quiz question employed to gather student responses for this project. Becker and Plumb [2] conducted a comprehensive study spanning seven semesters, examining the performance of students who failed the course. The findings



Consider the four-element circuit depicted below and argue what will happen to the power (increase, decrease, or remain the same) of each of the circuit's four elements when the resistance of resistor R2 decreases. Treat all elements as ideal including the independent voltage source and thoroughly justify your response.

Figure 3.1: The question of the writing quiz employed to collect student responses. [4]

revealed a direct correlation between the final grade and their performance on Exam 1. Becker et al. [3] developed this quiz with the dual purpose of identifying at-risk students and detecting common misconceptions. Notably, this quiz is administered on the fifth day of the class.

The instructions provided along with the quiz question have undergone evolution over time. For instance, in the Fall of 2017, students heavily relied on formulas in their responses. Consequently, future students were explicitly instructed to use as few formulas as possible when answering the question. Despite these adjustments in instructions, the core question has remained consistent throughout the entire project.

An example student response is presented below. In this instance, the underlined sentence highlights a *sequential misconception*. The sequential misconception in electric circuits is the belief that elements located further downstream from a source, such as R2 and R3 in Figure 3.1, receive current after elements positioned closer to the source, like R1.

> Vs is an ideal component, so changing R2 will not affect it. R1 is in series with iR23 so changing R2 will also not affect the power associated with it. R3 will slowly have a decrease in power as R2's resistance goes to 0, with R2 at zero coinciding with no power in R3. R2 will increase in power, because the current is going to increase with a decrease in resistance. So Vs is the same, R1 is the same, R2 is larger and R3 is smaller.

3.2 Phase 1: Prelude

We initiated our work on this project in early 2021 upon receiving the initial dataset from our collaborator. In this phase, our focus centered on the exploration and assessment of the usability and potential of LLMs for AMD.

Semesters	F17	F18	S19	F20	Total
Number of responses	56	61	23	45	185
Shows misconception	11	9	9	0	29
No misconception	45	52	14	45	156

Table 3.1: Sample and label distribution of Dataset 101 across four semesters.

3.2.1 Datasets

3.2.1.1 Dataset 101

This inaugural dataset in the project comprises 185 student responses collected over four semesters. Annotated with a single misconception known as the *sequential misconception*, labels are assigned at the response level. The distribution of samples across semesters and labels is detailed in Table 3.1. For model training, 60% of the dataset (111 responses) is utilized, and the remaining 40% (74 responses) for testing.

3.2.1.2 Dataset 102

Upon analyzing the outcomes of the model fine-tuned on Dataset 101, a hypothesis emerges that fine-tuning the model at the sentence level may enhance performance. A notable limitation of Dataset 101 lies in labeling responses at the response level. Consequently, Dataset 102 is created by reannotating responses presenting misconceptions at the sentence level. Out of 119 sentences across 29 responses featuring misconceptions in Dataset 101, only 34 sentences are annotated with the misconception. The training and testing distribution of this dataset mirrors Dataset 101.

3.2.2 Model

We fine-tune BERT Base uncased [10] modeling the task as a binary classification problem. The optimization process employs the Adam optimizer with $\epsilon = 1e-8$, a learning rate set at $2e-5$, and a weight decay of 0.01. For model training, we utilize the Binary Cross-Entropy loss function with a sigmoid layer. The fine-tuning process is capped at a maximum of 50 epochs with an early stopping mechanism.

3.2.3 Results and Discussion

The performance of the BERT model is outlined in Table 3.2. Initially, we conduct fine-tuning on Dataset 101, resulting in a model with a notable recall of 0.92 but a comparatively lower precision of 0.31. Concurrently, we implement a rule-based model employing keyword matching techniques within spaCy [4], which attains perfect recall (1.0) coupled with a high precision of 0.63. In the context of misconception detection, where identifying misconceptions is crucial for effective learning, recall takes precedence over precision [4]. It is vital to avoid missing genuine misconceptions, as these can significantly impact students' understanding and progress. However, achieving a moderate precision with the LLM remains essential for the feasibility of our approach.

Model	Dataset	Precision	Recall	F1
Rule-based	101	0.63	1.00	0.77
BERT Base	101	0.31	0.92	0.47
BERT Base	102	0.61	0.92	0.73

Table 3.2: Performance of the rule-based and BERT Base models for detecting a single misconception in student responses of Dataset 101 & 102.

Considering the performance of the rule-based model, we hypothesize the necessity of altering our fine-tuning approach. A meticulous examination of the responses reveals that the majority of sentences are correct within a response presenting a misconception. This observation prompts the hypothesis that fine-tuning the model at the sentence level might enhance model performance. Consequently, we introduce Dataset 102 and fine-tune the model at the sentence level by providing the remaining sentences in the same response as context. To predict the labels at the response level for comparison with the rule-based model, we utilize the model to predict labels at the sentence level and assign the misconception label to the response if any sentences within it are labeled as presenting the misconception. This approach significantly elevates precision while maintaining the same recall, resulting in a notable increase in F1 from 0.47 to 0.73.

In summary, the BERT model falls short of outperforming the rule-based model. However, it is crucial to note that the rule-based method necessitates manual analysis of responses and the creation of keyword-matching rules for detecting misconceptions, presenting a significant obstacle to scalability. Some misconceptions even require sentence-pair analysis [4], further complicating the process. LLMs automate the feature extraction process and alleviate the challenges associated with the rule-based approach. Despite achieving lower performance compared to the rule-based model, LLMs successfully showcase their potential and utility in the task of detecting misconceptions.

3.3 Phase 2: Foundation

Detecting misconception is inherently a data-driven problem. In early 2022, our primary focus revolves around expanding the dataset to enhance model performance.

Despite the course (our data source) being offered nearly every semester, the quantity of responses is constrained by student enrollment. As we actively accumulate more data, new challenges surface concerning data collection and processing. On one front, we are engaged in the creation of an in-house data annotation web app, while on the other front, we engage in enhancing the model performance and scrutinizing the drawbacks of our approach. Subsequently, we formulate annotation guidelines for a consistent and standardized annotation process.

3.3.1 Glechon: Annotation Web App

During the development of Dataset 101, student responses were collected on paper and handwritten by the students during an in-class quiz. The digitalization of these responses was carried out by a typist, who assigned a continuous numerical identifier to each response. Annotations were then added using a text editor, with labels typed next to the respective identifiers. For Dataset 102, a PDF file was generated, containing responses presenting misconceptions, and sentences were labeled by highlighting them. With no standardized structure, annotations in both datasets were manually organized into the required format for model training.

As we commenced the annotation of new data, the need for an annotation tool to standardize the process became evident. Unfortunately, no readily usable annotation tool was found, leading us to use Microsoft Word for annotating the responses. Soon after transitioning to multiple labels, inconsistencies in labeling and formatting challenges emerged. Word processors lacked the capability to enforce a strict structure and automatically extract responses along with labels. The current state necessitates a time-consuming manual post-processing to format the data for model training. The realization of the need for an annotation tool became evident to facilitate system-

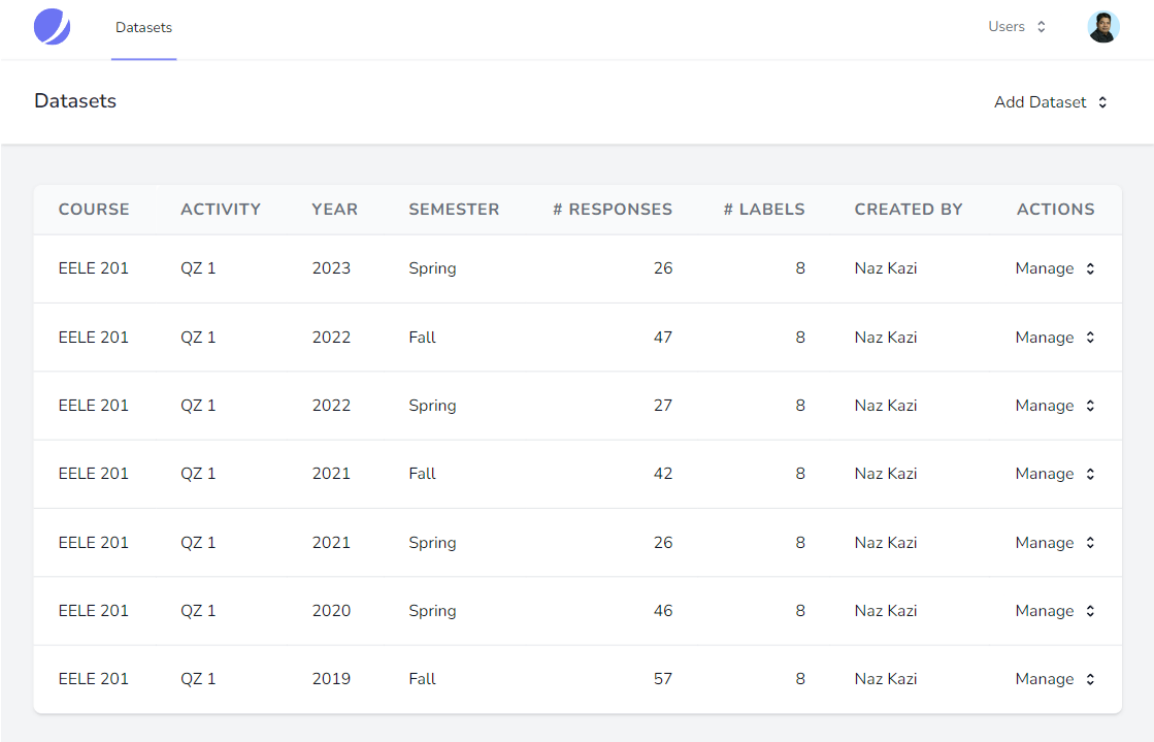


Figure 3.2: The dashboard of the Glechon web app, developed in-house to streamline data annotation and processing. The dashboard presents a list of available datasets (i.e., responses grouped semester-wise) along with their metadata and an action menu.

atic, efficient, and large-scale data collection. With no other viable options available through the community, we dedicated our efforts to developing an in-house annotation web app called Glechon[1].

We take advantage of the Laravel framework[2] to develop the web app. We prioritized versatility in its design, ensuring its adaptability to various courses. The dashboard (see Figure 3.2), presented in a tabular format, offers an overview of datasets, with each dataset row containing essential metadata such as the course name, activity type, year, semester, response count, and an accessible dropdown menu. The dropdown

[1]https://glechon.oxiago.com
[2]https://laravel.com

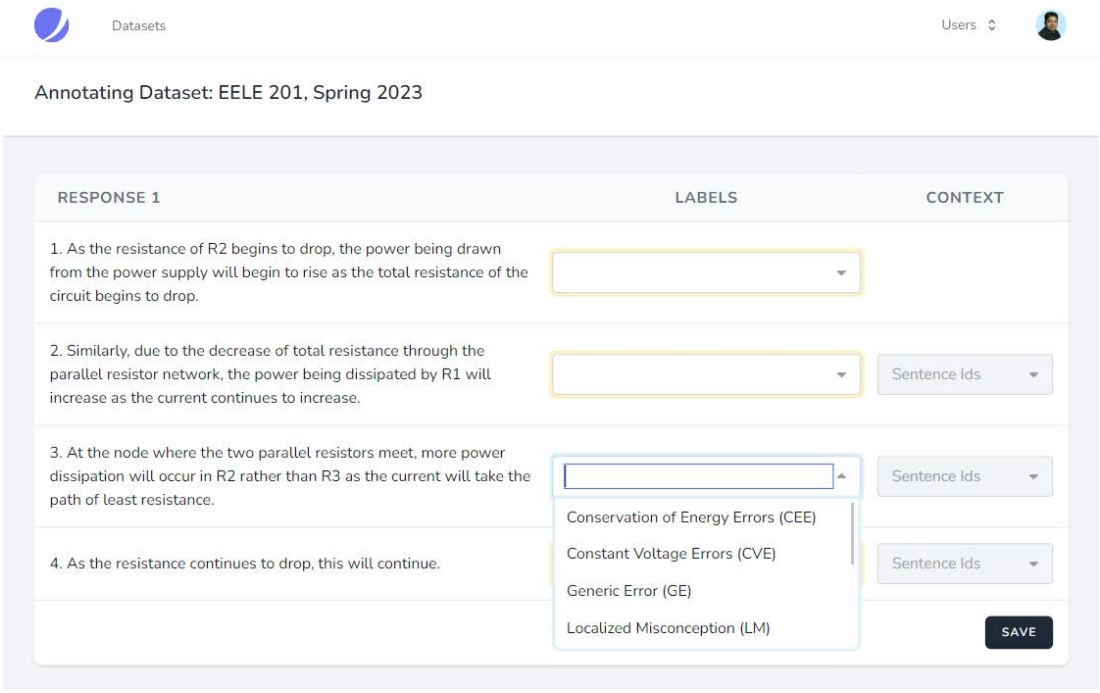


Figure 3.3: A overview of the annotation page featuring a response with four sentences. The label menu for the third sentence is open, displaying the available labels.

menu, tailored to user roles, offers a range of actions, including viewing, editing, annotating, exporting, and deleting datasets. Additionally, users have the option to import datasets from files or URLs. An independent web app is currently under development to collect student responses digitally. Importing datasets via URLs will allow us to import student responses directly from another server in the future.

Within the annotation interface (see Figure 3.3), all responses from a dataset are displayed, each presented in an individual table complete with a unique identifier. Responses may consist of multiple sentences, each enumerated and displayed in separate rows. Each row features two multi-tag boxes—one for labels and another for selecting sentences providing context for the chosen label. In anticipation of numerous labels, both multi-tag boxes incorporate search functionality. Annotators can apply multiple labels to a sentence and specify context sentences per label (see Figure 3.4). For convenience, each response includes a dedicated save button, allowing annotators

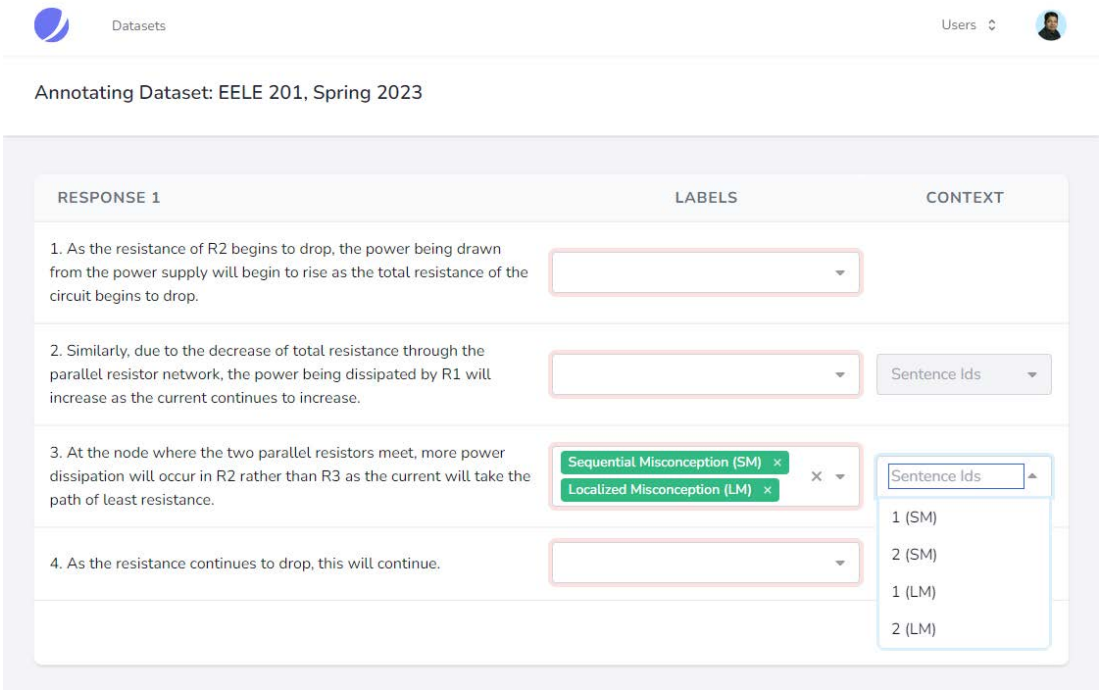


Figure 3.4: A glimpse of the annotation page featuring a sentence annotated (unsaved) with two labels and the respective context menu populated with options based on the selected labels.

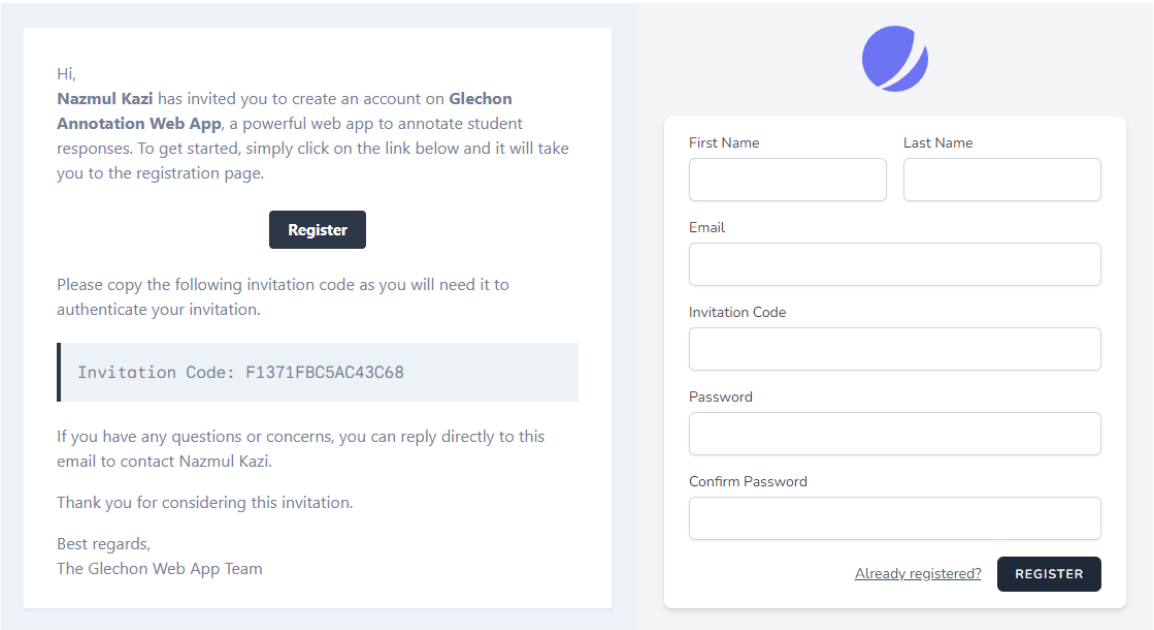


Figure 3.5: A snapshot showcasing a user invitation email with a unique code on the left and the user registration form on the right.

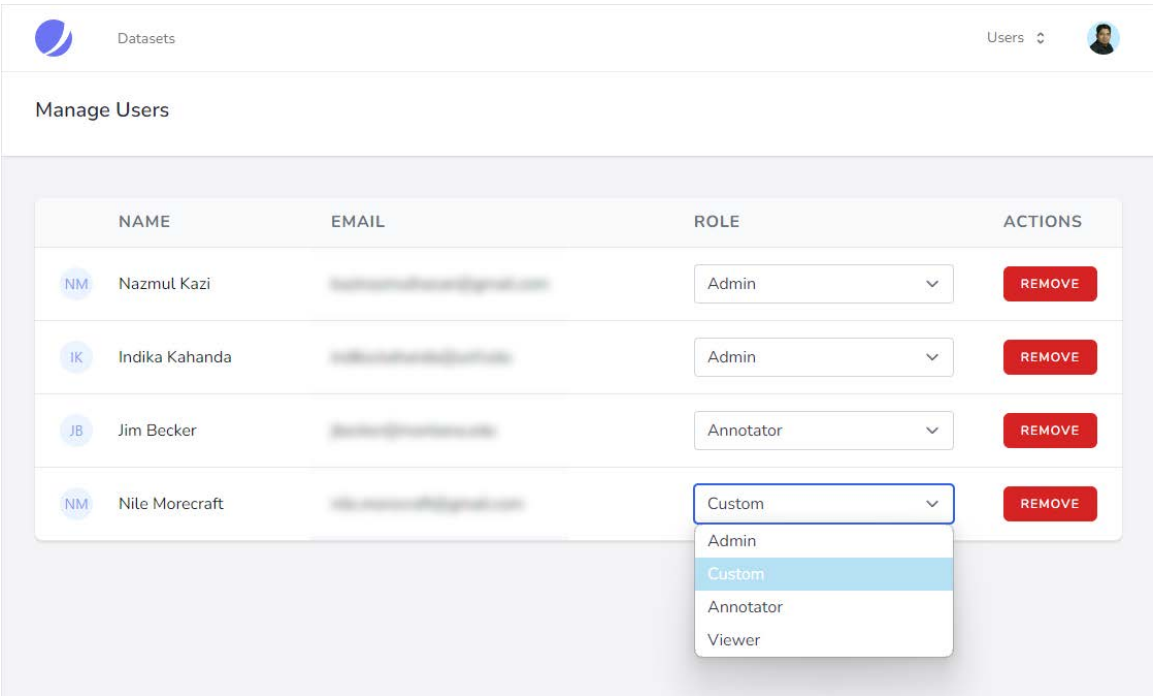


Figure 3.6: A screenshot of the user management page illustrating the current role of various users and providing options to assign roles or remove users from the system.

to pause and easily track their progress. The input boxes are color-coded, with amber indicating *never annotated*, red for *unsaved*, and green for *successfully saved*.

To ensure security, the application is safeguarded behind a login page, with all communication encrypted. Only administrators have the ability to invite new users by providing their email addresses in the system. An automated email containing a unique code (see Figure 3.5) is sent to the provided address to prevent the creation of malicious accounts. After a user signs up, administrators have the authority to assign roles and customize permissions (see Figure 3.6). User permissions can be tailored to include specific actions within the app. Furthermore, we have implemented role-based user permissions, simplifying the assignment of permissions to common user roles, such as annotators, for efficient and streamlined access control.

Users with appropriate permissions have the capability to export any dataset in JSON

Source	Dataset 102	F21	S22	Total
Number of responses	185	42	27	254
Shows misconception	29	9	5	43
No misconception	156	33	22	211

Table 3.3: Sample and label distribution of Dataset 201 across Dataset 102 and two new semesters.

format, a widely recognized and supported file format. The JSON file schema detailing the organization and structure of a dataset is depicted in Appendix B. The exported file encompasses dataset metadata, responses segmented into sentences, and annotations for each sentence from every annotator, all organized in a standardized format. The application also incorporates API support, providing users with tokens. This allows users to employ the API through programming scripts, enabling direct data retrieval from the server and seamless integration into their model training.

3.3.2 Enhancing Model Performance

3.3.2.1 Dataset 201

We gather responses from two additional semesters (Fall 2021 and Spring 2022) and annotate the datasets at the sentence level for sequential misconceptions. This new dataset is created by combining the annotated data from these two semesters with our existing Dataset 102. The sample and label distribution of this dataset is illustrated in Table 3.3. Among the 14 new responses presenting misconceptions, 20 sentences are annotated with the label. For model training, 60% of the dataset (152 responses) is utilized, while the remaining 40% (102 responses) is allocated for testing.

3.3.2.2 Model

We employ the same approach and hyperparameters used for fine-tuning our model on Dataset 102, with a couple of modifications, to fine-tune a BERT Base uncased model on Dataset 201. A notable change is the warming up of the learning rate over the initial 500 steps. Additionally, we fine-tune the model using only the preceding sentences, if any, as context. When evaluating a response, the identification of a misconception in a given sentence is solely based on the information from the preceding sentences. If a misconception is identified in light of a following sentence, the label should be assigned to the latter sentence. This is because the information necessary to detect the misconception was not available until the latter sentence was reached.

3.3.2.3 Results and Discussion

The performance of the model is presented in Table 3.4. The model's precision has slightly increased (0.63 vs. 0.61), while the recall has notably decreased (0.88 vs. 0.92). Despite the changes in precision and recall, the F1 score remains consistent (0.73). These scores indicate that the additional data did not contribute to improving the model's performance as anticipated.

We experimented with various sets of hyperparameters, but the model's performance

Dataset	Precision	Recall	F1
102	0.61	0.92	0.73
201	0.63	0.88	0.73

Table 3.4: Performance of BERT Base model for detecting a single misconception in student responses of Dataset 201.

did not show any improvement. Subsequently, we shifted our focus to identifying potential reasons behind the challenge in enhancing the model's performance. Upon a meticulous examination of the data, we discovered many responses, primarily from Dataset 101/102, to be highly noisy. Numerous responses contain misspellings and incomplete sentences, while some sentences are challenging to comprehend even for humans. Additionally, certain students have utilized diagrams or computations to answer questions, which were omitted during the transcription process. Unfortunately, the original copies of the responses from Dataset 101 were unavailable, preventing us from curating the data. However, unused responses from several other semesters were unearthed, prompting us to redirect our efforts toward creating a new dataset with seven misconceptions addressing the issues observed in our past datasets.

3.3.3 Annotation Guidelines

As we commence our work on developing the new dataset across multiple semesters and seven misconceptions, it becomes imperative to construct annotation guidelines to ensure consistent policies and procedures are followed throughout the entire process. These guidelines provide the annotation team with a set of rules and instructions for the annotation process and equip the NLP team with a profound understanding of the process of detecting misconceptions in responses. These insights provide us with a means of designing a new approach and making design decisions for model training in the next phase. We utilize these guidelines to annotate responses from eight semesters, spanning from the Fall of 2019 to the Fall of 2023. The annotation guidelines are listed below:

1. A response contains one or more sentences. When tagging the sentences, predefined misconception labels should be used where applicable.

2. If a response indicates a misconception, at least one sentence should be tagged with that misconception. However, a sentence can be incorrect and yet express no misconceptions. In such cases, the sentence should not be tagged with any misconceptions.

3. There are two scenarios in which a sentence may exhibit a misconception:

   (a) A sentence displays the misconception by itself and does not require any context from other sentences in the response. In this case, the sentence should be tagged with the misconception, and no sentences should be selected as context.

   (b) A sentence requires context from one or more preceding sentences to exhibit misconceptions. In this case, the last sentence in the group that exhibits the misconception should be tagged with the misconception, and the other sentences in the group should be mentioned as context. Only the last sentence in the group should be tagged with the misconception because the misconception is detected only after reading the last sentence.

4. There may be cases where two or more groups of sentences express the same misconception independently in the same response. In such cases, the last sentence in each group must be different, but the two groups may contain common context sentences. The last sentence of each group should be tagged with the misconception type, and the other sentences within the groups should be marked as context, including common sentences. In a rare case where the last sentence from the former group provides context to the latter group, the last sentence from the former group should be tagged with the misconception as part of the former group and also mentioned as context as part of the latter group.

5. It is important to note that other sentences should not be mentioned as context for a sentence if the sentence is not tagged with any misconceptions.

6. In rare cases of a sentence containing multiple misconceptions, it should be tagged with each applicable label. Then, context sentences should be added for each misconception.

3.4 Phase 3: Transformation

The emergence of a new dataset, comprising over 300 responses annotated with seven misconceptions, not only revitalized our research efforts but also unlocked doors to new possibilities. These seven misconception label types are described elsewhere [4]. Leveraging this dataset, we fine-tune one of the largest and most powerful LLMs in the BERT series. We meticulously analyze and prepare the dataset for fine-tuning, and employ various strategies to enhance the model performance.

3.4.1 Dataset 301

To address the issues encountered with Dataset 201, the research team ensured to receive unchanged digital copies of the responses for each semester. We utilized a sentence tokenizer from spaCy[3] for sentence segmentation, complemented by regular expressions to identify undetected sentence boundaries (only a few cases) by matching terminal punctuations between words. A Python script was employed to organize the responses into the JSON structure (see Appendix B) defined for Glechon. Each response underwent a meticulous review, addressing mis-spellings and potential typos, as mis-spelled words can introduce noise for most LLMs in the BERT series, impacting tokenization. Grammatical errors, if present, were retained, as altering them

[3]https://spacy.io

might compromise the originality of the responses. During this review, we semantically segmented sentences that remained undetected due to a lack of punctuation or complex sentence structure. This detailed formatting process was necessary, considering that most, if not all, of these limitations is expected to be resolved in the future as responses are collected through a web-based app with native browser-based spell-checking enabled.

The student composition in a course changes every semester, and it is not uncommon to have multiple instructors teaching the same course concurrently in sections or interchangeably from one semester to another. Several factors can influence the student's knowledge base, leading to variations in their performance and the quality of their responses across semesters. For instance, Becker and Plumb [2] established a strong correlation between students' performance on Exam 1 of EELE 201 (our data source) and their performance in a prerequisite course, Calculus II. Additionally, as mentioned earlier, the instructions provided with the quiz questions have evolved over time. Instructors continually refine their teaching methods to enhance the learning experience for their students. Consequently, training a model with data from one semester and testing it on responses from different semesters may result in unreliable outcomes. To address this, we store and annotate responses separately for each semester. This organizational strategy later facilitates the development of training, validation, and test sets by incorporating responses from various semesters, ensuring diversity in the dataset splits.

Table 3.6 illustrates the number of responses per label and their distribution across semesters. The full title of the abbreviated labels are listed in Table 3.5. We have annotated the responses at the sentence level, and the distribution of sentences across labels and semesters is presented in Table 3.7. It is important to acknowledge that

Abbreviation	Full Name
CEE	Conservation of Energy Errors
CVE	Constant Voltage Errors
IIVSM	Ideal Independent Voltage Source Misconception
LM	Localized Misconception
PCM	Precedence of Current Misconception
RCE	Resistor Combination Errors
SM	Sequential Misconception

Table 3.5: Abbreviations and full titles of the misconception labels in Dataset 301. The full definitions of these seven misconception label types are provided elsewhere [4]

Misconceptions	F19	S20	S21	F21	S22	F22	S23	F23	Total
Number of responses	57	46	26	42	27	47	26	42	313
Shows misconception	31	19	10	28	17	23	13	35	176
No misconception	26	27	16	14	10	24	13	7	137
CEE	0	0	2	2	2	2	0	3	11
CVE	9	8	4	8	7	6	3	7	52
IIVSM	8	6	1	6	1	6	3	5	36
LM	1	0	0	1	2	2	0	2	8
PCM	1	0	2	0	2	4	1	0	10
RCE	14	5	0	10	5	2	5	10	51
SM	7	4	4	9	4	11	5	14	58

Table 3.6: Distribution of collected responses across eight semesters and seven misconceptions. Please note that the three rows (e.g., *Number of responses*) in the top section of the table are not sums of any numbers in that column since the responses are annotated at the sentence level with multiple labels.

certain sentences carry multiple labels, with the count reflected under each applicable label. Notably, in both tables, the *Number of responses* and *Shows misconception* rows do not represent any sums derived from the lower part of the table. The *Shows misconception* row indicates the count of samples presenting at least one misconception, while the *No misconception* row signifies samples devoid of any misconceptions.

Misconceptions	F19	S20	S21	F21	S22	F22	S23	F23	Total
Number of sentences	317	212	143	257	186	284	165	227	1,791
Shows misconception	44	29	16	41	28	37	19	76	290
No misconception	273	183	127	216	158	247	146	151	1501
CEE	0	0	2	2	2	2	0	3	11
CVE	10	9	5	11	10	6	3	12	66
IIVSM	8	6	1	7	1	7	3	6	39
LM	1	0	0	1	2	2	0	3	9
PCM	1	0	2	0	3	4	1	0	11
RCE	17	9	0	11	5	2	6	11	61
SM	8	5	7	10	5	14	7	24	80

Table 3.7: Distribution of sentences, in the collected responses, across eight semesters and seven misconceptions.

3.4.1.1 Filtering Training Data

Sentences from merely six responses across five semesters have been annotated with multiple labels. While detecting misconceptions inherently involves a multi-label classification, our dataset lacks the comprehensive information needed to address it as such. Given this limitation, our primary emphasis lies in modeling the data for multi-class classification. Recognizing the presence of a few responses with multiple labels, we have opted to exclude these six responses entirely from our training data.

In our previous datasets, we encountered responses that consisted of only 1-2 sentences. Some of these responses were deemed too brief for effective misconception detection, as comprehensive details are crucial for accurate analysis. Figure 3.7 illustrates a histogram of response distribution based on the number of sentences, with the majority containing 3-5 sentences. Notably, 20 responses consist of only 1-2 sentences. However, sentence length varies, and a response with two lengthy sentences can provide sufficient information to identify misconceptions. Therefore, solely re-

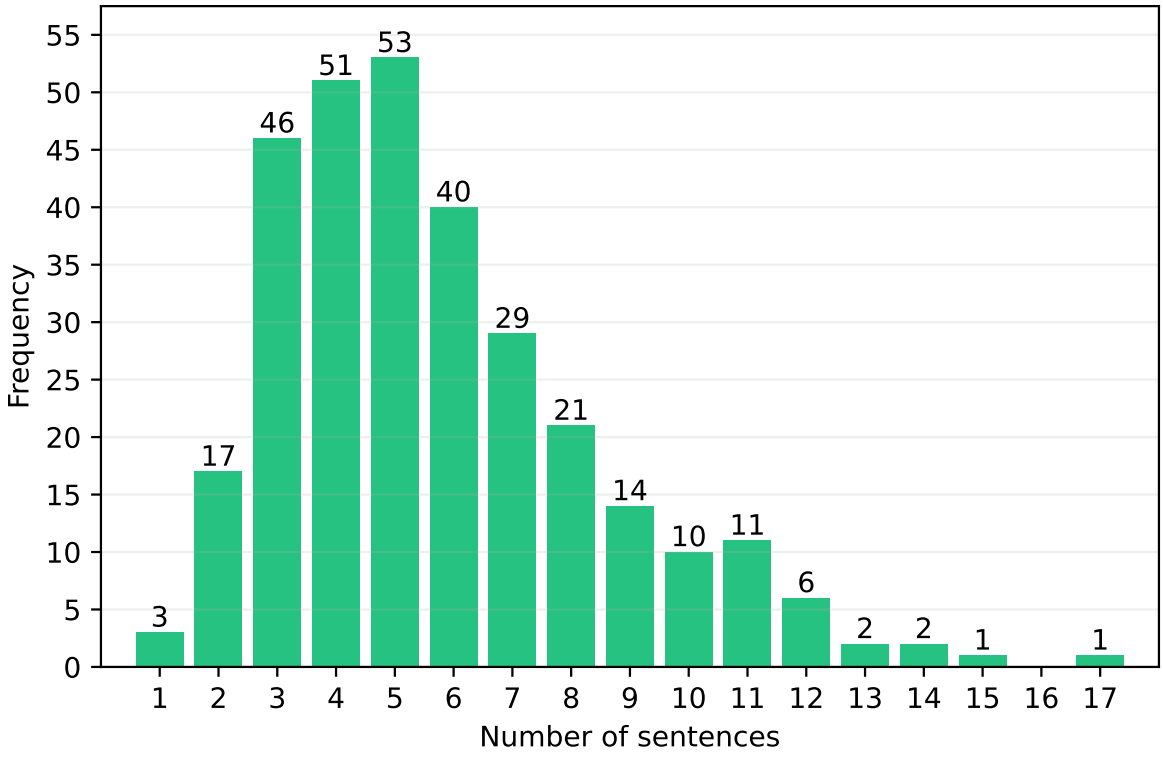


Figure 3.7: Distribution of responses based on the number of sentences.

lying on the number of sentences as a criterion for removal is not deemed optimal. Consequently, we extended our analysis to consider response length in terms of tokens. Figure 3.8 presents a histogram based on the number of tokens, revealing that the majority of responses contain around 105 tokens. There are 39 responses with less than 70 tokens and 19 responses with less than 50 tokens. To address the challenge of short responses while minimizing data loss, we set the token threshold at 60 and remove 25 responses. A few responses with less than three sentences met the token threshold and remain in the training data.

LLMs have a token limit for input data, and truncation is a common strategy to manage lengthy inputs. However, due to our specific input design for the model (see section 3.4.2), truncating inputs was not a viable option. Upon finalizing the structure of the inputs (see section 3.4.2), we were restricted to only 349 tokens for

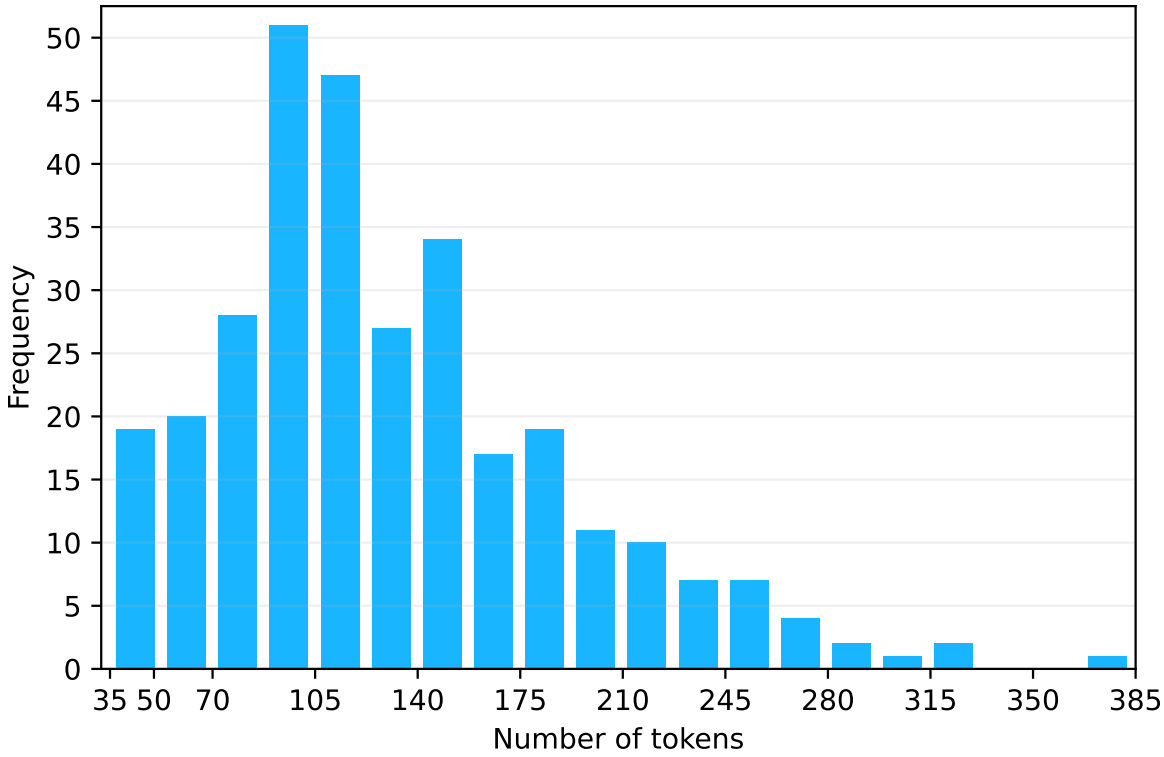


Figure 3.8: Distribution of responses based on the number of tokens.

each response. Fortunately, only one response exceeded this limit with 384 tokens. Given the absence of options to increase the token limit or accommodate this response within the derived limit, we decide to exclude it from our training data.

In previous phases, we evaluated our models at the response level to compare their performance with the rule-based model. However, in this phase, developing a rule-based model became impractical. Consequently, we decided to assess our models at the sentence level, aligning more realistically with the task of misconception detection. While some misconceptions are common among students, others are very rare. Out of the seven misconceptions used to annotate the dataset, there are only a handful of responses under CEE, LM, and PCM (refer to Table 3.6). These labels lack sufficient responses to be distributed across train, validation, and test sets. Therefore, we opt to exclude these labels from fine-tuning, retaining the responses without these labels.

Misconceptions	F19	S20	S21	F21	S22	F22	S23	F23	Total
Number of responses	52	41	22	37	26	46	21	36	281
Shows misconception	27	19	6	21	13	18	10	22	136
No misconception	25	22	16	16	13	28	11	14	145
CVE	8	8	4	7	7	6	3	7	50
IIVSM	8	6	0	4	1	6	2	4	31
RCE	12	5	0	7	5	2	4	8	43
SM	7	4	3	9	4	11	3	12	53

Table 3.8: Distribution of responses across eight semesters and four misconceptions in Dataset 301.

Misconceptions	F19	S20	S21	F21	S22	F22	S23	F23	Total
CVE	9	9	5	9	10	6	3	12	63
IIVSM	8	6	0	5	1	7	2	4	33
RCE	15	9	0	8	5	2	4	8	51
SM	8	5	6	10	5	14	4	20	72
Shows misconception	40	29	11	32	21	29	13	44	219
No misconception	264	170	117	211	162	252	125	167	1468
Number of sentences	304	199	128	243	183	281	138	211	1687

Table 3.9: Distribution of sentences across eight semesters and four misconceptions in Dataset 301.

The distribution of responses and sentences after all filtering is shown in Table 3.8 and Table 3.9, respectively.

3.4.1.2 Training Splits

We divide the training data into three sets: train, validation, and test. Although we are fine-tuning at the sentence level, we cannot split the data at the sentence level due to the design of the input to the models. A response should be entirely assigned to at most one set.

Considering the diversity of responses, it is essential to construct each set with responses balanced across all semesters. However, some semesters have too few responses to consider for the test or validation set, such as IIVSM in S22 or S23. To simplify the process, we focus on assembling the test and validation set first and then constructing the train set from the remaining responses. We set a simple rule to decide whether to use responses from a given semester for a given label. We calculate the median number of responses across semesters for each label. Then, we designate a semester as a potential source for the test or validation set if the number of responses from that semester is greater than or equal to the median number of responses for that label.

Since responses are annotated with multiple labels, we also need to consider the label distribution while splitting the data. Otherwise, a split may result in no responses

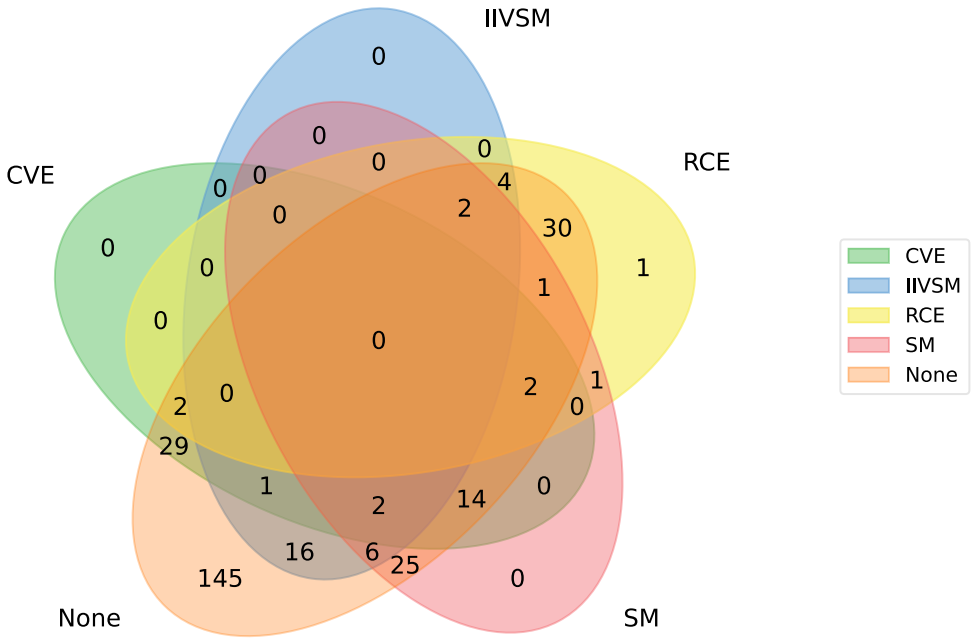


Figure 3.9: A Venn diagram illustrating the distribution of responses among five distinct labels, demonstrating the uniqueness and overlap in occurrences of misconceptions within Dataset 301. The *None* label represents no misconception.

for a particular label. To gain insights into the overlap of labels, we plot a Venn diagram with the number of responses per label, as shown in Figure 3.9. The label *none* in the Venn diagram represents the absence of misconceptions in responses in the non-overlapping region and in sentences within regions that overlap with other labels. Figure 3.10 illustrates the number of responses per number of labels, where zero denotes no labels. For the test and validation sets, we aim to maintain this distribution as closely as possible.

We require a substantial portion of the data for fine-tuning, especially given the complexity of the problem. However, it is equally crucial to have sufficient samples in the test and validation sets to effectively evaluate the models. Considering the dataset size, distribution of responses, and the distribution of the number of misconceptions per response, we set the test and validation sizes to each be 35 responses. To eliminate bias, we randomly select the responses from a uniform distribution. The distribution of the number of misconceptions per response in the three sets is shown in Figure 3.11. The distribution of responses in train, validation, and test sets is depicted in Table 3.10, Table 3.11, and Table 3.12, respectively. The distribution of sentences in train, validation, and test sets is depicted in Table 3.13, Table 3.14, and Table 3.15, respectively.

3.4.2 Fine-tuning Approach and Design Decisions

Misconception detection demands domain and subject knowledge. Large Language Models (LLMs) are pretrained on extensive corpora for general language comprehension. While LLMs can be fine-tuned for specific tasks, it is crucial to note that discriminative LLMs do not inherently acquire domain knowledge through this process. Pretraining LLMs specifically for domain knowledge is a resource-intensive and

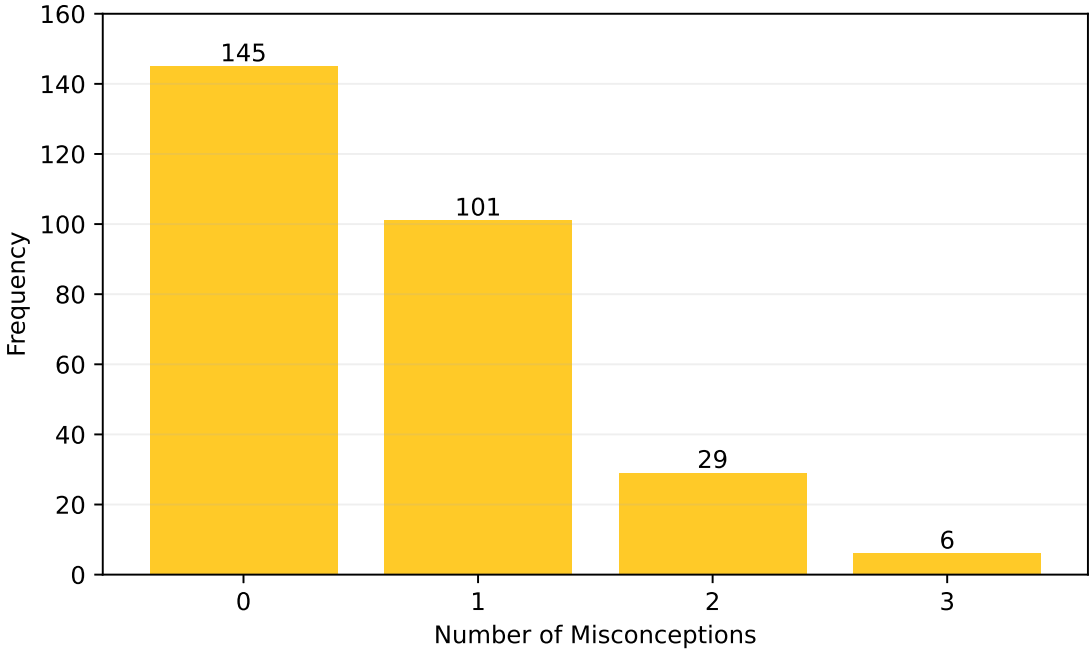
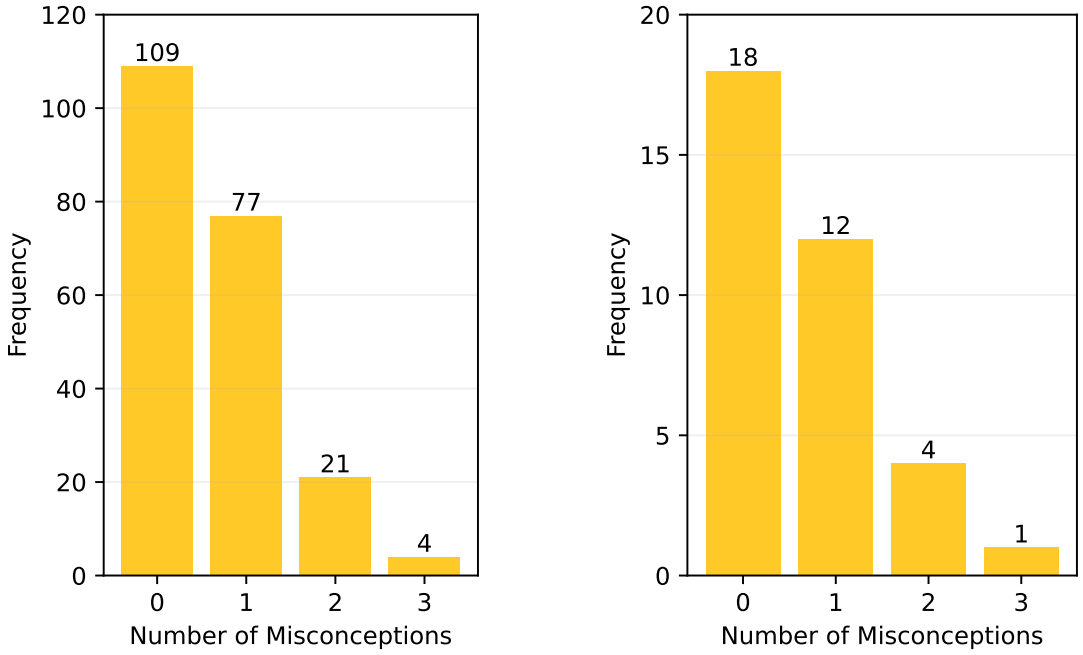


Figure 3.10: Distribution of responses based on the number of misconceptions per response in Dataset 301. The first bar signifies responses with no misconceptions.



(a) Train

(b) Validation / Test

Figure 3.11: Distribution of responses based on the number of misconceptions per response in train, validation, and test sets. The first bar in each subplot signifies responses with no misconceptions.

Misconceptions	F19	S20	S21	F21	S22	F22	S23	F23	Total
Number of responses	36	31	22	24	26	29	21	22	211
Shows misconception	18	14	6	14	13	14	10	13	102
No misconception	18	17	16	10	13	15	11	9	109
CVE	6	6	4	5	7	6	3	2	39
IIVSM	5	4	0	3	1	3	2	4	22
RCE	7	3	0	4	5	2	4	6	31
SM	4	4	3	6	4	9	3	6	39

Table 3.10: Distribution of responses across labels and semesters in the train set.

Misconceptions	F19	S20	F21	F22	F23	Total
Number of responses	6	5	6	11	7	35
Shows misconception	4	2	3	4	4	17
No misconception	2	3	3	7	3	18
CVE	1	1	1	0	3	6
IIVSM	1	1	0	3	0	5
RCE	2	1	2	0	1	6
SM	0	0	1	2	3	6

Table 3.11: Distribution of responses across labels and semesters in the validation set.

Misconceptions	F19	S20	F21	F22	F23	Total
Number of responses	10	5	7	6	7	35
Shows misconception	5	3	4	0	5	17
No misconception	5	2	3	6	2	18
CVE	1	1	1	0	2	5
IIVSM	2	1	1	0	0	4
RCE	3	1	1	0	1	6
SM	3	0	2	0	3	8

Table 3.12: Distribution of responses across labels and semesters in the test set.

Misconceptions	F19	S20	S21	F21	S22	F22	S23	F23	Total
CVE	7	7	5	7	10	6	3	3	48
IIVSM	5	4	0	4	1	4	2	4	24
RCE	9	6	0	4	5	2	4	6	36
SM	4	5	6	7	5	12	4	8	51
Shows misconception	25	22	11	22	21	24	13	21	159
No misconception	189	128	117	133	162	165	125	97	1116
Number of sentences	214	150	128	155	183	189	138	118	1275

Table 3.13: Distribution of sentences across labels and semesters in the train set.

Misconceptions	F19	S20	F21	F22	F23	Total
CVE	1	1	1	0	5	8
IIVSM	1	1	0	3	0	5
RCE	2	2	2	0	1	7
SM	0	0	1	2	7	10
Shows misconception	4	4	4	5	13	30
No misconception	26	22	37	52	37	174
Number of sentences	30	26	41	57	50	204

Table 3.14: Distribution of sentences across labels and semesters in the validation set.

Misconceptions	F19	S20	F21	F22	F23	Total
CVE	1	1	1	0	4	7
IIVSM	2	1	1	0	0	4
RCE	4	1	2	0	1	8
SM	4	0	2	0	5	11
Shows misconception	11	3	6	0	10	30
No misconception	49	20	41	35	33	178
Number of sentences	60	23	47	35	43	208

Table 3.15: Distribution of sentences across labels and semesters in the test set.

costly endeavor, and it falls outside the scope of our project. Consequently, we must impart subject knowledge to LLMs through the input. Drawing from our experience with the ASAG project (Chapter 2), we conceptualize misconception detection as a Recognizing Textual Entailment (RTE) problem. In this framework, we map the response as a hypothesis and provide the requisite knowledge through the premise.

The selection of the premise is crucial in RTE tasks, as it involves determining if the hypothesis entails the premise. In our case, the quiz question contains valuable information for the model, given the models' lack of background or subject knowledge. Omitting the question would be akin to expecting the model to extract an answer from a passage without providing a question. Additionally, based on our previous study [16], we have observed that questions offer valuable context to the models. However, the quiz question includes a circuit diagram, and we cannot input images into a text model. To address this issue, we extend the quiz question by describing the circuit diagram using words.

The question alone is insufficient to equip the model with the necessary knowledge since it provides only context and lacks a reference point for entailment. To address this, we augment the premise with a reference answer, following our established practice from the ASAG project. We concatenate the quiz question and the reference answer with a single space. The same premise is used for all inputs to the model.

The input size of LLMs is restricted by a predetermined number of tokens, established during pre-training. For this task, we utilize a RoBERTa Large LLMs [19], which has a token limit of 512. Consequently, the combined length of the premise and hypothesis cannot exceed 509 tokens. This poses a challenge since the reference answer provided by the instructor is 352 tokens long. To overcome this limitation and ensure there

is sufficient space for the hypothesis, we focus on condensing the token count in the premise. We address this challenge by manually summarizing both the question and the reference answer, resulting in 62 tokens (43 words) for the question and 98 tokens (76 words) for the reference answer, while preserving essential information. In summarizing the reference answer, we aimed to succinctly convey the changes in power and the reasons behind them using as few words as possible. This yields a premise of 160 tokens, leaving 349 tokens for the hypothesis. The summarized question and reference answer are provided below:

> **Question:** Resistors R1 and R2 are connected in series with a voltage source Vs and resistor R3 is connected in parallel to R2. When the resistance of R2 decreases, will the power on R1, R2, R3, and Vs increase, decrease, or remain the same?

> **Reference Answer:** As the resistance of R2 decreases, the power of R1 will increase since P=I^2R and the current flow increases. The power consumed by the Vs will increase, as it carries more current through the decreasing equivalent resistance. The power of R3 will decrease because less voltage drops across the parallel combination of R2 and R3. As R2 approaches zero resistance, the power of R2 will eventually decrease to zero, as it will sustain no voltage drop.

Our data is annotated at the sentence level. Some misconceptions exhibit inter-sentence dependencies, making them challenging to detect when analyzing sentences independently, even for instructors. Furthermore, we instructed annotators to list context sentences, if any, along with the labels. Upon reviewing the annotations, we

observed that in many cases, a sentence presents a misconception only in the context of a preceding sentence. Therefore, it is crucial to include the preceding sentences in the input.

The inclusion of preceding sentences raises a dilemma of how to incorporate these sentences into the inputs. On one hand, a common practice is to use the premise to input all context to the model. However, the preceding sentence may contain misconceptions or incorrect information, introducing noise or contradiction in the premise and potentially harming the model's performance. Therefore, incorporating the preceding sentences in the premise is not an option. On the other hand, adding the preceding sentences can divert the model's attention from the sentence we are asking the model to classify. To keep the preceding sentences separate from the classifying sentence, we concatenate them with a newline character. Since all the sentences in a response are collapsed into a single paragraph and have been segmented into sentences, a response cannot contain a newline character. Thus, we chose the newline character as glue in the hypothesis and hypothesized (due to the black-box nature of the models) that it would be recognized by the model as a separator. In the case of the first sentence of a response, we employed an empty string in place of the preceding sentences. In other words, the hypothesis begins with a newline character.

Due to the design of our input structure, any form of truncation would lead to the loss of information. Truncating the premise would jeopardize the reference answer. Placing the classifying sentence at the beginning would compromise context and dependencies from the preceding sentences. Positioning the classifying sentence at the end would compromise the sentence itself. Consequently, we have disabled truncation and ensured the hypothesis adheres to the token limit.

3.4.3 Model

We fine-tune RoBERTa Large MNLI using the Adam optimizer with $\epsilon = 1e-8$, a learning rate of $1e-5$, a weight decay of 0.01, and Cross Entropy Loss function. The training is warmed up linearly over the first 10% of the data. The fine-tuning process lasts for a maximum of 20 epochs with early stopping. A batch size of 5 is utilized. Model performance is recorded after each epoch, and the best model is selected based on the epoch displaying the highest macro F1 score on the validation set.

The responses contain domain-specific words that may be unfamiliar to the model's tokenizer. Such words are segmented into smaller subword units recognizable by the tokenizer. However, when segmentation is not possible, the tokenizer marks these words as *unknown* and they are subsequently ignored by the model. Given the substantial weight and semantic context carried by domain-specific words in this task, it is crucial to incorporate them into the tokenizer's vocabulary and the model's embeddings. To achieve this, we employ spaCy for tokenization and generate a list of all unique words present in the dataset. Subsequently, we compare this list with the tokenizer's vocabulary (case-sensitive) to identify words unknown to the model. We have identified and added the following 27 words to the model's vocabulary:

1. G1
2. G2
3. G3
4. I1
5. I2
6. I3
7. Ieq
8. KCL
9. KVL
10. Kirchhoff
11. P1
12. P2
13. P3
14. R1
15. R2
16. R3
17. Reff
18. Req
19. Rn
20. Rtot
21. Rtotal
22. V1
23. V2
24. V3
25. Vout
26. amperage
27. wattage

Remarkably, we discovered that many domain-specific words are already present in the model's vocabulary. Examples include R, Vs, W, Amps, Ohm, and current. R denotes resistor and Vs refers to a voltage source (i.e., battery) in the electrical engineering domain. Although these terms exist in the model's vocabulary, the specific meanings of words like R and Vs to the model remain unknown to us.

3.4.4 Results and Discussion

The comprehensive performance of our models is illustrated in Table 3.16. Initially, we fine-tune a RoBERTa Large MNLI model with input tokenized using the stock tokenizer (i.e., the tokenizer with the original vocabulary), codenamed ST (Stock Tokenizer). Subsequently, we fine-tune another RoBERTa Large MNLI model using our tokenizer with an extended vocabulary, referred to as ET (Extended Tokenizer). The ET model significantly outperforms the ST model in terms of recall (0.60 vs. 0.55) and F1 score (0.60 vs. 0.54). However, the ST model exhibits marginally better precision (0.64 vs. 0.63) than the ET model. To validate the importance of preceding sentences as context, we fine-tune a RoBERTa Large MNLI model, codenamed LC (Limited Context), at the sentence level without incorporating the preceding sentences in the inputs. The ET model surpasses the LC model across all metrics, showcasing the significance of preceding sentences in providing valuable context and improving both precision (0.58 vs. 0.63) and recall (0.55 vs. 0.60). The epoch column denotes the training epoch at which the model achieves the highest F1 on the validation set. Notably, the ET model has a lower epoch than both ST and LC models and verifies the effectiveness of our design.

The performance of the ET model for each label is detailed in Table 3.17. The ET model excels in detecting SM, boasting perfect precision (1.00) and a recall of 0.64.

Model	Epoch	Valid			Test		
		P	R	F1	P	R	F1
ST (Stock Tokenizer)	15	0.45	0.42	0.42	0.64	0.55	0.54
LC (Limited Context)	19	0.60	0.56	0.55	0.58	0.55	0.56
ET (Extended Tokenizer)	12	0.61	0.56	0.57	0.63	0.60	0.60

Table 3.16: Performance of RoBERTa Large MNLI multi-class models fine-tuned on Dataset 301 with three distinct configurations.

Label	Valid			Test		
	P	R	F1	P	R	F1
CVE	0.19	0.38	0.25	0.21	0.43	0.29
IIVSM	0.40	0.40	0.40	0.33	0.25	0.29
RCE	0.83	0.71	0.77	0.67	0.75	0.71
SM	0.67	0.40	0.50	1.00	0.64	0.78
No misconception	0.94	0.93	0.93	0.95	0.93	0.94
Macro Average	0.61	0.56	0.57	0.63	0.60	0.60
Weighted Average	0.88	0.86	0.87	0.90	0.88	0.89

Table 3.17: Performance of the RoBERTa Large MNLI model with extended tokenizer (ET), the best multi-class model, with scores detailed per misconception label.

Model	Epoch	Valid			Test		
		P	R	F1	P	R	F1
BIN RES	3	0.86	0.71	0.78	0.67	0.71	0.69
BIN SENT	5	0.65	0.73	0.69	0.67	0.67	0.67
BIN CVE	7	0.43	0.38	0.40	0.50	0.57	0.53
BIN IIVSM	10	0.00	0.00	0.00	0.00	0.00	0.00
BIN RCE	5	1.00	0.71	0.83	0.86	0.75	0.80
BIN SM	4	0.73	0.80	0.76	0.73	0.73	0.73

Table 3.18: Performance of six RoBERTa Large MNLI binary models fine-tuned on Dataset 301 at the response and sentence levels, and one for each misconception label.

It achieves the highest recall on RCE (0.75) with the second-best precision (0.67) and F1 score (0.71). However, the model faces challenges in detecting CVE and IIVSM, attaining the same F1 score (0.29) for these labels with a higher recall on CVE (0.43

vs. 0.23). A notable observation is that the model, despite having twice the sample size for CVE (48) compared to IIVSM (24) and only a few samples less than SM (51), achieves the lowest F1 on CVE. This suggests that detecting CVE in responses is inherently more complex than the other labels. Remarkably, the model achieves an F1 of 0.94 in detecting sentences with no misconceptions.

The ET model demonstrates superior performance compared to the BERT model fine-tuned on Dataset 201. While predicting three additional labels, it also outperforms the BERT model on SM achieving a higher F1 score (0.78 vs. 0.73) and an F1 score of 0.71 for another label, RCE. Although the ET model exhibits a lower recall (0.64 vs. 0.88) than the BERT model on SM, it significantly enhances precision (1.00 vs. 0.63), which may serve as a foundation for further improvements in F1. This improved performance underscores the effectiveness of our model design.

To gain valuable insights into the challenges and complexities of detecting misconceptions in general and specific instances, we fine-tune six binary models using RoBERTa Large MNLI, prefixed with BIN for binary, and the scores are presented in Table 3.18. First, we fine-tune a binary model, codenamed BIN RES, to detect the presence of any misconceptions at the response level. The BIN RES model exhibits higher scores on each metric, notably a significantly higher recall (0.71 vs. 0.60) than ET. This suggests that detecting misconceptions, in general, might share common patterns and supports our choice of approaching misconception detection as an RTE problem. Second, we fine-tune another binary model, codenamed BIN SENT, to investigate how the challenge of detecting the presence of misconceptions changes when transitioning from the response to the sentence level. The BIN SENT model achieves a lower recall (0.67 vs. 0.71) compared to the BIN RES model while maintaining the same precision. This indicates that detecting misconceptions at the sentence level

is harder than at the response level. However, the BIN SENT model outperforms the ET model in all metrics, particularly in F1 (0.67 vs. 0.60). This result further supports our hypothesis that common patterns across different misconception types make it easier to detect misconceptions in general.

Third, we fine-tune a binary model for each misconception label at the sentence level. These misconception-specific models are codenamed with the prefix BIN, followed by their corresponding label name. The BIN CVE model significantly outperforms the ET model in all metrics, particularly in precision (0.50 vs. 0.21) and F1 (0.53 vs. 0.29), for detecting CVE. Similarly, the BIN RCE model exceeds the ET model in precision (0.86 vs. 0.67) and F1 (0.80 vs. 0.71), while maintaining the same recall. However, the ET model surpasses the binary models in F1 for detecting IIVSM (0.29 vs. 0.00) and SM (0.78 vs. 0.73). Notably, the BIN IIVSM model labels all samples as misconception-free even after 10 epochs. Although extensive hyperparameter tuning may enhance the performance of BIN IIVSM, it remains unexplored given the low performance of the ET model on IIVSM. Interestingly, the binary models excel for two labels while underperforming for the other two, and this performance trend does not correlate with the sample size. The fact that the epoch of BIN CVE is higher than all other binary models and nearly double that of the BIN SM model aligns with the earlier observation suggesting that detecting CVE is inherently more challenging, pointing to its complexity as a misconception.

Recognizing the strengths and weaknesses of the models, we develop an ensemble model by combining the ET model with multiple binary models. BIN CVE, BIN RCE, and BIN SM outperform the ET model in F1 on the validation set. Consequently, we use these three binary models to predict their corresponding labels, while the ET model for IIVSM. We apply a misconception label to a sample if the sample

Label	Valid			Test		
	P	R	F1	P	R	F1
CVE	0.50 **0.31** ↑	0.12 **0.26** ↓	0.20 **0.05** ↓	0.60 **0.39** ↑	0.43	0.50 **0.21** ↑
IIVSM	0.40	0.40	0.40	0.50 **0.17** ↑	0.25	0.33 **0.04** ↑
RCE	1.00 **0.17** ↑	0.71	0.83 **0.06** ↑	1.00 **0.33** ↑	0.62 **0.13** ↓	0.77 **0.06** ↑
SM	0.86 **0.19** ↑	0.60 **0.20** ↑	0.71 **0.21** ↑	0.89 **0.11** ↓	0.73 **0.09** ↑	0.80 **0.02** ↑
No misconception	0.92 **0.02** ↓	0.98 **0.05** ↑	0.95 **0.02** ↑	0.94 **0.01** ↓	0.98 **0.05** ↑	0.96 **0.02** ↑
Macro Average	0.74 **0.13** ↑	0.56	0.62 **0.05** ↑	0.78 **0.15** ↑	0.60	0.67 **0.07** ↑
Weighted Average	0.89 **0.01** ↑	0.91 **0.05** ↑	0.89 **0.02** ↑	0.92 **0.02** ↑	0.92 **0.04** ↑	0.92 **0.03** ↑

Table 3.19: Performance of RoBERTa Large MNLI ensemble model with scores outlined per misconception label. The second row within each label indicates the difference in performance compared to the ET model. The blue color with up arrows signifies an increase in performance while the amber color with down arrows indicates a decrease in performance.

is predicted for misconception by the BIN SENT model. The performance of the ensemble model is outlined in Table 3.19. The ensemble model achieves higher F1 scores for all labels, with a notable increase for CVE (0.50 vs. 0.29). The precision for SM decreased (0.89 vs. 1.00), whereas the recall has increased (0.73 vs. 0.64). Notably, the precision for RCE (1.00 vs. 0.67) and IIVSM (0.50 vs. 0.33) have also increased significantly. Overall, the recall remains unchanged while the precision increases significantly (0.78 vs. 0.63) contributing to a higher F1 score (0.67 vs. 0.60).

Prior to adopting RoBERTa Large MNLI as our base model, we fine-tune a RoBERTa Large pretrained model on Dataset 301. Surprisingly, the model does not predict any samples with misconceptions. Despite conducting extensive hyperparameter tuning,

limiting the training to a maximum of 10 epochs, no changes are observed. This observation suggests that modeling the task as recognizing textual entailment may necessitate fine-tuning the model on a task-related corpus before fine-tuning it on the target dataset. Alternatively, the issue could be attributed to the relatively small dataset size.

3.5 Conclusions and Future Work

Our study delves into the nuanced task of misconceptions detection in student responses, employing state-of-the-art language models like RoBERTa Large MNLI. Formulating the task as recognizing textual entailment by incorporating the question along with a reference answer as premise and the student response as hypothesis, demonstrates notable advancements in the detection of specific misconceptions. While the ensemble model exhibits moderate precision and F1 scores with considerable recall, indicating its proficiency in capturing misconceptions, it is crucial to acknowledge the challenges posed by certain complex misconceptions.

Our exploration extends beyond the binary classification approach used in previous models, embracing a multi-class classification framework. The results underscore the intricacies involved in developing the dataset and fine-tuning models for diverse misconceptions, with varying levels of complexity and prevalence.

Our study lays the groundwork for investigating the impact of sample size on model performance. Future work could delve deeper into understanding how sample distribution and size influence the efficacy of misconception detection, potentially leading to insights that inform data collection strategies.

As we navigate the evolving landscape of misconception detection, several other avenues for future research emerge. First, to adhere to the token limitation of the model, we had to minimize the size of both the quiz question and the reference answer, sacrificing valuable information. Exploring LLMs, such as T5, PaLM, and Megatron-Turing NLG, that have a higher token limit and surpass RoBERTa on RTE could be a promising avenue for future research. T5 boasts a token limit of 4096 tokens, while both PaLM and Megatron-Turing NLG further extend the limit to 8000 tokens, providing an excellent opportunity to enrich the context through premise. Second, the observed challenges in detecting complex misconceptions highlight the need for more sophisticated strategies and scrupulous model design. Third, we perform shallow hyperparameter tuning on RoBERTa Large MNLI and extensive hyperparameter tuning may further improve the overall performance or on specific misconceptions. Fourth, misconception detection is a multi-label classification problem that is not explored due to the lack of sufficient data. Fifth, we have excluded three misconceptions from our training data owing to their low sample counts, which can be a part of future research. In essence, our study opens the door to a realm of possibilities in the pursuit of enhancing misconceptions detection in student responses.

3.6 Ethical Considerations and IRB Approval

Ethical considerations for this research project were addressed in accordance with the established guidelines of the Institutional Review Board (IRB). The study received approval under exempt protocol number JB061421-EX.

3.7 Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant IUSE-2120466. Additionally, this research is partially funded by Auld & White.

Chapter 4

Conclusion

In the landscape of automated educational assessment, Large Language Models (LLMs) emerge as transformative catalysts, reshaping our understanding and approach. LLMs, with their extensive knowledge and contextual understanding, stand as beacons of innovation in the realm of natural language processing. Their ability to capture intricate linguistic patterns, coupled with adaptability across diverse tasks, positions them as powerful tools for unraveling the underlying nuanced expression of student responses. As we embark on this exploration, the potential of LLMs becomes not only a technological advancement but also a paradigm shift in perception and engagement with the multifaceted nature of education assessment. Our journey through automated short-answer grading and misconception detection unveils the immense possibilities of LLMs for the future of automated educational assessment.

In our relentless pursuit to propel automated short-answer grading and revolutionize automated misconception detection, our exploration unfolds across two pivotal chapters. On automated short-answer grading, we illuminate the transformative potential of transfer learning on a state-of-the-art LLM, RoBERTa Large. The model's remarkable adaptability across unseen answers, questions, and domains, showcases the effectiveness of task-related corpora in discerning contradiction in diverse student responses.

As we transition to automated misconception detection, we delve into an unexplored territory of educational assessment. We meticulously craft a misconception detection

dataset poised to serve as a guiding light for future research in this domain. Framing the task as recognizing textual entailment, our innovative approach with RoBERTa Large MNLI heralds a breakthrough in misconception detection. The model's prowess in capturing nuanced misconceptions shines a spotlight on the untapped potential of LLMs in unraveling the underlying intricate expression of student responses and misunderstandings.

The harmonious interplay between these two projects paints a comprehensive picture of the transformative role we envision for LLMs in automated educational assessment. From the showcased adaptability in short-answer grading to the pioneering strides in misconception detection, our findings lay the groundwork for a future where assessment tools will not only decipher the subtleties of student responses but also detect and address misconceptions with unparalleled accuracy. We find ourselves standing at the crossroads of groundbreaking research, ushering in a new era of educational assessment—one where the capabilities of LLMs are recognized and fully harnessed to their utmost potential.

Chapter 5

Contributions

5.1 Automated Short Answer Grading (ASAG)

1. We formulate ASAG as a Recognizing Text Entailment (RTE) problem, where the question description concatenated with the reference answer becomes the *premise*, and the student answer becomes the *hypothesis*.

2. We demonstrate that fine-tuning models on task-related corpus (prior to fine-tuning on task-specific data) holds promise for cross-domain generalization, which effectively enhances the model performance by leveraging successful transfer learning. However, the model does not benefit from multiple corpora.

3. We published these findings in Kazi and Kahanda (2023) [15].

5.2 Automated Misconception Detection (AMD)

1. In collaboration with Montana State University, we developed a versatile annotation app and guidelines applicable to any domain or subject and utilized them to annotate an extended misconception dataset containing over 300 student answers and seven misconceptions types related to Electric Circuits.

2. We detail recommendations for novel techniques on data curation, data cleaning, filtering outliers, train/test splitting, and formulating AMD as an RTE problem where the question description concatenated with the reference answer becomes

the *premise*, and each sentence prepended with previous "context" sentences in a student response become the *hypothesis*.

3. We uncover the importance of domain-specific terminology and sentence dependency in detecting misconceptions. We ensemble multi-class with binary models to strike a balance and maximize overall performance to demonstrate the utility of our approach.

REFERENCES

[1] Christian Anderson Arbogast and Devlin Montfort. Applying natural language processing techniques to an assessment of student conceptual understanding. In *2016 ASEE Annual Conference & Exposition*, 2016.

[2] James P Becker and Carolyn Plumb. Identifying at-risk students in a basic electric circuits course using instruments to probe students' conceptual understanding. In *2018 ASEE Annual Conference Proceedings*, 2018.

[3] James P Becker, Emily Sior, Jerad Hoy, and Indika Kahanda. Board 11: Predicting at-risk students in a circuit analysis course using supervised machine learning. In *2019 ASEE Annual Conference & Exposition*, 2019.

[4] James P Becker, Indika Kahanda, and Nazmul H Kazi. WIP: Detection of Student Misconceptions of Electrical Circuit Concepts in a Short Answer Question Using NLP. In *2021 ASEE Virtual Annual Conference Content Access*, 2021.

[5] Sridevi Bonthu, S Rama Sree, and MHM Krishna Prasad. Automated short answer grading using deep learning: A survey. In *Machine Learning and Knowledge Extraction: 5th IFIP TC 5, TC 12, WG 8.4, WG 8.9, WG 12.9 International Cross-Domain Conference, CD-MAKE 2021, Virtual Event, August 17–20, 2021, Proceedings 5*, pages 61–78. Springer, 2021.

[6] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In Lluís Màrquez, Chris Callison-Burch, and Jian Su, editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1075. URL `https://aclanthology.org/D15-1075`.

[7] Paola Britos, Elizabeth Jiménez Rey, Darío Rodríguez, and Ramón García-Martínez. Work in progress-programming misunderstandings discovering process based on intelligent data mining tools. In *2008 38th Annual Frontiers in Education Conference*, pages F4H–1. IEEE, 2008.

[8] Leon Camus and Anna Filighera. Investigating transformers for automatic short answer grading. In *Artificial Intelligence in Education: 21st International Conference, AIED 2020, Ifrane, Morocco, July 6–10, 2020, Proceedings, Part II 21*, pages 43–48. Springer, 2020.

[9] Holger Danielsiek, Wolfgang Paul, and Jan Vahrenhold. Detecting and understanding students' misconceptions related to algorithms and data structures. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education*, pages 21–26, 2012.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423.

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

[12] Myroslava O Dzikovska, Rodney D Nielsen, Chris Brew, Claudia Leacock, Danilo Giampiccolo, Luisa Bentivogli, Peter Clark, Ido Dagan, and Hoa T Dang. Semeval-2013 task 7: The joint student response analysis and 8th recognizing textual entailment challenge. Technical report, University of North Texas, 2013.

[13] Myse Elmadani, Moffat Mathews, and Antonija Mitrovic. Data-driven misconception discovery in constraint-based intelligent tutoring systems. *20th International Conference on Computers in Education (ICCE)*, 2012.

[14] Andrea Goncher, Wageeh Boles, and Dhammika Jayalath. Using automated text analysis to evaluate students' conceptual understanding. In *Proceedings of the 2014 Annual Conference of the Australasian Association for Engineering Education (AAEE2014)*, pages 1–9. Massey University, 2014.

[15] Nazmul Kazi and Indika Kahanda. Enhancing transfer learning of llms through fine-tuning on task-related corpora for automated short-answer grading. In *22nd IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1677–1681. IEEE, 2023. doi: 10.1109/ICMLA58977.2023.00255.

[16] Nazmul Kazi, Indika Kahanda, S. Indu Rupassara, and John W. Kindt Jr. Zero-shot information extraction with community-fine-tuned large language models from open-ended interview transcripts. In *22nd IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 919–924. IEEE, 2023. doi: 10.1109/ICMLA58977.2023.00138.

[17] Sachin Kumar, Soumen Chakrabarti, and Shourya Roy. Earth mover's distance pooling over siamese lstms for automatic short answer grading. In *IJCAI*, pages 2046–2052, 2017.

[18] Ou Lydia Liu, Hee-Sun Lee, Carolyn Hofstetter, and Marcia C Linn. Assessing knowledge integration in science: Construct, measures, and evidence. *Educational Assessment*, 13(1):33–55, 2008.

[19] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv*, 2019. doi: 1907.11692.

[20] Joel Michael. Misconceptions—what students think they know. *Advances in Physiology Education*, 26(1):5–6, 2002.

[21] Joshua J. Michalenko, Andrew S. Lan, and Richard G. Baraniuk. Data-mining textual responses to uncover misconception patterns. In *Proceedings of the Fourth (2017) ACM Conference on Learning @ Scale*, L@S '17, page 245–248, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450344500. doi: 10.1145/3051457.3053996. URL `https://doi.org/10.1145/3051457.3053996`.

[22] Catherine Mulryan-Kyne. Teaching large classes at college and university level: Challenges and opportunities. *Teaching in higher Education*, 15(2):175–185, 2010.

[23] David K Pugalee. *Writing to develop mathematical understanding*. Christopher-Gordon, 2005.

[24] Allan J Rossman, B Chance, and CPSL Obispo. Anticipating and addressing student misconceptions. In *ARTIST Conference on assessment in Statistics, Lawrence University*, pages 1–4, 2004.

[25] Hans-Jürgen Schmidt. Students' misconceptions—looking for a pattern. *Science education*, 81(2):123–135, 1997.

[26] Albert Stevens, Allan Collins, and Sarah E Goldin. Misconceptions in student's understanding. *International Journal of Man-Machine Studies*, 11(1):145–156, 1979.

[27] Chul Sung, Tejas Indulal Dhamecha, and Nirmal Mukhi. Improving short answer grading using Transformer-based pre-training. In *Artificial Intelligence in Education: 20th International Conference, AIED 2019, Chicago, IL, USA, June 25-29, 2019, Proceedings, Part I 20*, pages 469–481. Springer, 2019.

[28] Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics, 2018. URL `http://aclweb.org/anthology/N18-1101`.

[29] Yuanwang Yang, Hong Ye, Jianhua Deng, Changjiang You, Weijian Chen, Yiru Zhang, and Xiaoli Xiong. Diagnosis of misconception in electronic circuits engineering courses: A case study at uestc, china. In *2019 IEEE International Conference on Engineering, Technology and Education (TALE)*, pages 1–10. IEEE, 2019.

[30] Xinhua Zhu, Han Wu, and Lanfang Zhang. Automatic short-answer grading via bert-based deep neural networks. *IEEE Transactions on Learning Technologies*, 15(3):364–375, 2022.

Appendices

Appendix A

Variation of Misconceptions

Let's consider a scenario where responding to a question involves writing a for loop with five iterations. The presence of an incorrect number of iterations in a Java course may result from an operator error ($i \leq 5$), as exemplified below:

```
for (int i = 0; i <= 5; i++) {}
```

Following the standard practice, this particular operator error or misconception is not possible in a Python course. Conversely, in a Python course, the potential for an erroneous iteration count arises from a misunderstanding of the exclusivity of the *to* argument in the range method, illustrated by the following code:

```
for i in range(4):
```

It is noteworthy that Java lacks a built-in range method similar to Python, eliminating the possibility of encountering the same misconception in a Java course. This highlights that the same question or topic can elicit two distinct sets of misconceptions, with some commonalities, by simply altering the programming language.

A curriculum encompasses diverse types of courses, with undergraduate computer science (CS) students being obligated to enroll in classes covering programming languages, data structures, algorithms, and other related subjects. Within a programming course, misconceptions frequently revolve around the intricacies of language syntax, as students grapple with coding rules and conventions. In contrast, a data structure class, being more generalized and not confined to a specific programming language, exposes students to a broader array of concepts related to various data structures. Misconceptions in a data structure course are more likely to center around the understanding and application of diverse data structure concepts. Moreover, in an algorithm course, misconceptions predominantly pertain to algorithms, although some may extend to programming languages and data structures. This highlights the nuanced nature of misconceptions, demonstrating how they can vary from course to course based on the specific focus and content.

Misconceptions within the domain of engineering exhibit a considerable breadth, given the diverse nature of engineering disciplines such as computer science, electrical engineering, mechanical engineering, and more. Each of these fields embodies distinct

subject matter, topics, and concepts, leading to a wide array of misconceptions unique to each discipline. In computer science, for instance, misconceptions may revolve around programming paradigms, algorithms, or software design principles. On the other hand, electrical engineering may involve misconceptions related to circuitry, signal processing, or electromagnetic phenomena. Similarly, mechanical engineering misconceptions could center around principles of thermodynamics, mechanics, or materials science.

It is evident that misconceptions show considerable variation not only between different domains but also within a single domain or subject. Furthermore, misconceptions related to a single topic may also differ.

Appendix B

Structure and Organization of Dataset in JSON

```
{
    "course": "EELE 201",
    "activity": "QZ 1",
    "semester": "Fall",
    "year": 2023,
    "labels": {
        "lbl1": "Label 1",
        "lbl2": "Label 2"
    },
    "responses": [
        {
            "id": 1,
            "text": [
                "sentence 1",
                "sentence 2",
                ...
            ],
            "annotators": [7, 9],
            "labels": {
                "7": {
                    "0": ["lbl1"],
                    "1": ["lbl1", "lbl2"]
                },
                "9": {
                    "1": ["lbl2"]
                }
            },
            "context":
            {
                "7": {
                    "1": {
                        "lbl2": [0]
                    }
                },
                "9": {}
            }
```

```
        },
        {
            "id": 2,
            "text": [
                "sentence a",
                "sentence b",
                ...
            ],
            "labels": null
        },
        ...
    ]
}
```