Chapter 15
Triangulating Computational and Qualitative Methods to Measure Scientific Uncertainty (Joshua M. Rosenberg, Hadi Bhidya, and Cody Pritchard)

Joshua M. Rosenberg, Hadi Bhidya, and Cody Pritchard, University of Tennessee, Knoxville

Abstract This chapter outlines steps to analyze a complex construct of interest to science education researchers in a very commonly used digital media platform, YouTube, particularly popular science education-related videos. The construct of interest is uncertainty—established as important but challenging for teachers and researchers alike to recognize and understand as many definitions and operationalizations of uncertainty as it relates to learning science exist. To study uncertainty, transcripts of videos are created using Python and the Python packages pytube and Whisper, and a two-step triangulation approach that combines a computational (a dictionary-based text analysis) and qualitative approach. In the text analysis step, transcripts of videos are searched for key uncertainty-related terms using the statistical software R. Next, qualitative coding of the transcripts is carried out, with the output from the first step as a support for the task of developing an initial set of codes for the types of uncertainty present in the science education videos.The proposed chapter contributes to the book by providing a practical guide for researchers interested in studying complex constructs using an approach that merges some of the benefits of quantitative and qualitative approaches. Python and R code are provided to support researchers to replicate and draw on the analysis carried out.

15.1 Introduction

Educational stakeholders are often most interested in supporting students in developing competencies that are complex, like an identity as someone who can do science (Brickhouse and Potter, 2001) or students' conceptual understanding of scientific mechanisms or processes (Krist et al., 2019). A key challenge pertinent to supporting students to develop such competencies is measuring or assessing them, which is

concomitantly complex (Wilson, 2023). One way to approach the complexity of measuring complex educational outcomes is to triangulate methodological approaches. In this chapter, we pursue such a triangulation approach.

Our triangulation strategy for measuring complex constructs differs from more conventional approaches in that it involves the combination not of quantitative and qualitative methods, but instead from combining *computational* and qualitative methods in a manner inspired by (Nelson, 2020) in the *computational grounded theory* three-step approach. Such an approach has been used in some prior science education research (Rosenberg and Krist, 2021). Here, we loosely follow this approach's first two steps of *computationally exploring* a large and complex data source (the first of the three computational grounded theory steps) and then *qualitatively analyzing* the data in light of the output of the first step (the second computational grounded theory step). We do so with a focus on the construct of uncertainty, a central element of science teaching and learning (Rosenberg et al., 2022; Manz and Suárez, 2018).

The goal of this chapter is to demonstrate how we employ a triangulation approach. Our approach loosely follows Nelson (2020) computational grounded theory approach to specifically measure a complex construct (i.e., the use of uncertainty within an informal science learning environment (Youtube)). While this chapter specifically measures how uncertainty is used with science learning videos on YouTube, a similar approach can be applied to measuring other scientifically-related complex constructs in other online educational spaces (e.g., Khan Academy and Coursera), with the exception of YouTube-specific data mining techniques discussed within the chapter.

In addition to showing how we can carry out a triangulation approach to measure a complex construct, we have a secondary aim: showing how a relevant data source can serve as the basis for a new kind of learning analytics- or educational data science-inspired investigation of uncertainty as it pertains to science teaching and learning. Namely, we show how we can efficiently collect data from a large corpus of science education videos, using the speech (and transcripts of speech that we can create) to play out and test the ideas about the nature of uncertainty that have been explored primarily in classroom contexts, but not in the context of educational media.

Before describing our data collection and data analysis method, we first briefly review some of the prior research on uncertainty and science education in the next section.

15.2 Prior Research on Uncertainty and Science Education

Uncertainty is ubiquitous in conversations among scientists Kirch (2008, 2010) and is a fundamental driver for continued research and the refinement of knowledge Allchin (2012). The *Framework for K-12 Science Standards* states that "Scientific knowledge is a particular kind of knowledge with its own sources, justifications, ways of dealing with uncertainties, and agreed-on levels of certainty" (**?**, p. 251).

Despite its presence and importance in scientific inquiry and learning, there is little consensus on how uncertainty is conceived in specific domains.

In the scientific community, uncertainty is communicated alongside its relationship with probability and often refers to the level of variability that exists in relationship to a scientific model or hypothesis. However, uncertainty is not a concept exclusive to the scientific community, but rather is a fact and condition present in our daily lives Pollack (2003). **?** define uncertainty as a cognitive feeling or "an individual's subjective experience of doubting, being unsure, or wondering about how the future will unfold, what the present means, or how to interpret the past" (p. 492). (Kahneman and Tversky, 1982) discuss how variants of psychological uncertainty, which may not follow rules of formal scientific inquiry, may be correlated with expressions within our natural language. The discourses surrounding our understanding of uncertainty are complex enough that some researchers have suggested discipline-specific typologies of uncertainty because data types and scientific models vary so widely based on one's domain Bateman et al. (2022).

Different domains have defined uncertainty differently. In risk analyses, for example, Rowe (1993) proposed defining "uncertainty" in terms of four dimensions or classes, including uncertainty related to anticipating changes over time and measurement-related uncertainty. Lord (2022) suggests using Rowe's (1993) dimensions of uncertainty to bolster students' scientific reasoning in order "to move students from explanations of personal uncertainty ("I just know it's going to happen!") toward explanations of uncertainty that exhibit scientific reasoning" (Lord, 2022, para. 5). While such dimensions may prove useful for pedagogical purposes, our analysis requires determining how "uncertainty" is conveyed through language.

For these purposes, we employ (Kirch, 2010) typology of uncertainty as a starting point for our exploration of uncertainty in educational videos. Kirch (2010) define uncertainty as both a psychological experience and a *mathematical object*. Kirsch writes, "uncertainty refers to a psychological condition of being in doubt (e.g., I am uncertain about something or someone. . .). It also refers to a statistical (or mathematical) object (e.g., a statistical estimation of uncertainty)" (Kirch, 2011, p. 57). As a psychological condition, uncertainty can be procedural, sociocultural, epistemological, and ontological. As a statistical object, it refers to measurement, sampling, repeatability, and predictive value Kirch (2011). We use this framework later in our analysis, focusing on psychological uncertainty as an initial step.

Thus, our aim in this analysis is to try to understand the nature of psychological uncertainty as it is expressed in educational videos in the science domain. To do so, we triangulate methodological approaches, using both an automated, computational approach, and a qualitative approach.

15.3 Data Analysis Overview

This is structured into four distinct sections. The first section – Data Analysis Step #1 – guides you through extracting audio streams from YouTube playlist URLs using

'Pytube'. In the second section – Data Analysis Step #2 – we focus on converting these audio files into transcriptions, which can be in the form of SRT or plain text files.

Each section is designed to be comprehensive, providing step-by-step instructions to ensure a smooth and effective learning experience.

We then proceed to two analytic steps, one computational (Data Analysis Step 3) and one qualitative (Data Analysis Step 4).

15.4 Data Analysis Step #1: Programmatically Accessing YouTube Video Data

The primary aim of this and the second step is to develop a method that converts YouTube playlists into transcriptions of all videos within those playlists. For these steps, we employ Python. If you haven't used python before, the easiest way to get started in our view is to download the Anaconda distribution, a particular version of Python combined with some already-included python libraries and tools for using Python: `https://www.anaconda.com/`

We will be using two Python libraries: 'pytube' for downloading videos and 'whisper' for converting audio to text.

To replicate this process, the prerequisites are minimal. You'll need a computer capable of running Python and some understanding of how Python works. For beginners, assistance from AI tools like ChatGPT can also be invaluable in navigating the learning curve.

15.4.1 Converting playlists to audio

Pytube is a highly capable Python library, freely available for interacting with YouTube via URL links. Its primary function is to download audio and video streams, captions, and search results. In this first part of our tutorial, we will focus on using Pytube to retrieve audio streams from YouTube playlists.

Before diving into the code, it's essential to install Pytube. This can be done by running the following command in your terminal or command prompt:

Python code snippet: installing pytube

```
pip install pytube
```

Additionally, familiarizing yourself with Pytube's documentation (available at https://pytube.io/en/latest/) is highly recommended. It provides a wealth of information and will enhance your understanding of the library's capabilities.

This segment of the tutorial will guide you through creating a program that downloads and saves the audio streams of a YouTube playlist's videos into a specified folder using Pytube.

Start by importing the Playlist class from Pytube and then create a Playlist object by passing the URL of the playlist - here, for a playlist from the YouTube channel Veritasium on the physics concept of intertia - as a string:

Python code snippet: importing the playlist class and reading in a playlist URL

```
from pytube import Playlist
playlist_url = 'https://www.youtube.com/playlist?list=PLAB27A3C12C31E663'
playlist = Playlist(playlist_url)
```

Iterate through each video in the playlist using a for loop. The loop will handle two main tasks: extracting the highest resolution audio stream and downloading it to a specified path. This ensures all videos in a playlist are stored in a single folder, aiding in organization and future analysis. We did this for around one dozen playlists from two channels: Crash Course (`https://www.youtube.com/user/crashcourse`) and Veritasium (`https://www.youtube.com/channel/UCHnyfMqiRRG1u-2MsSQLbXA`).

Python code snippet: iterating through each video

```
for video in playlist.videos:
    try:
        # Get the highest resolution audio stream, the first

        audio_stream = video.streams.filter(only_audio=True).first()

        # Download the audio stream and save it to the path below
        # For each playlist downloaded, change the folder it goes to

        print(f'Downloading: {video.title}')
        audio_stream.download(output_path=
        '/Users/actualuser/Desktop/path/to/save/veritasium/Inertia')

    except Exception as e:
        print(f"Error downloading {video.title}: {str(e)}")
```

By following these steps, you will be able to download audio streams from YouTube playlists, setting the stage for the next part of our tutorial where these audio files will be transcribed. We did this for a total of 282 videos, 146 from Crash Course and 126 from Veritasium.

15.5 Data Analysis Step #2: Creating Transcripts Using an Automatic Speech Recognition Tool

Whisper, developed by OpenAI (the same organization behind ChatGPT), is an open-source speech recognition model designed to handle a variety of tasks involving audio files, including automatic speech recognition. Some early work suggests that it may be better than other automatic speech recognition tools, especially when the audio is not wholly clear (Palaguachi et al., 2023).

In this part of our project, we'll use Whisper to convert the audio files we've gathered into subtitle—transcript—files.

To work with Whisper, you need to install its Python library. This can be done using the command:

Python code snippet: installing the Python package Whisper

```
pip install -U openai-whisper
```

Additionally, you'll need the Command Line tool 'ffmpeg', essential for handling multimedia files. Detailed instructions and additional information about Whisper, including its GitHub repository, can be found at https://github.com/openai/whisper. Note that the other required libraries for this part of the project are already included in the standard Python installation, so there's no need for additional downloads.

Whisper, while robust, does have some known limitations. Specifically, it may struggle with transcribing videos longer than 10 minutes and could inaccurately insert words like "oks" and "yeahs" in the transcript. If you encounter these issues or want to learn more about potential solutions and workarounds, visit the discussion at https://github.com/openai/whisper/discussions/679. This page offers valuable insights and community-driven advice on addressing these challenges.

This program utilizes Whisper to transcribe audio streams into subtitle files from the folder created using Pytube.

Begin by importing the necessary libraries. Whisper for transcribing audio, os and pathlib for handling file paths, and get_writer from Whisper utils for creating subtitle files.

Python code snippet: importing Whisper and other libraries

```
import whisper
from whisper.utils import get_writer
import os
import pathlib
```

Define the folder containing your audio files and convert it into a path object for easier handling. This should be the same path the files in the above step were saved to. Then, generate a list of all files in the specified directory, filtering only audio files (in this case, .mp4 files).

Python code snippet: installing the Python package Whisper and listing the audio files

```
directory_path = r"path_to_files"
directory_path = pathlib.Path(directory_path)

all_files = os.listdir(directory_path);
all_mp4 = [audio for audio in all_files if audio.endswith(".mp4")]
```

Next, load the Whisper model with a suitable model size. The choice of model ('small', 'medium', and 'large') impacts the speed and accuracy of transcription, with larger models potentially being more accurate, but also taking considerably longer. For this tutorial, we use the 'small' model, though for research uses, the additional time may be worth the investment. A counter is used to track progress.

Python code snippet: loading the Whisper model

```
model = whisper.load_model("small")
```

Python code snippet: iterating over each audio file

```
i = 0
for mp4s in all_mp4:
```

```
# Convert the current .mp4 file to a string as a parameter

mp4s_path = directory_path / mp4s
result = model.transcribe((str(mp4s_path)), fp16=False)

# whisper.utils get_writer will output the text with timestamps
srt_writer = get_writer("srt", directory_path)
srt_writer(result, (str(mp4s_path)))

# printing out the name of the file just written
name = str(mp4s)
print(name)
i += 1
print(i)
```

The transcript files should be saved to the same folder as the audio files. These end in the extension `.srt`.

The result was transcripts for each of the 272 videos we accessed in the last stpe. On average, each video has 8 minutes, 12 seconds of speech, for a total of 36,780 utterances, or around 1.54 days worth of speech (36.96 hours, or 2,217 minutes). In other words, a fairly large collection of transcribed speech data, motivating the computational approach described next.

15.6 Data Analysis Step #3: Computational Analysis

Let's pick up where we left off from the first two steps, with one big difference – we'll be using R, a statistical software and programming language used in other chapters in this book – instead of Python.

You can find instructions on downloading R and RStudio here, in a chapter in the book *Data Science in Education Using R* (Estrellado et al., 2020): `https://datascienceineducation.com/c05`

We'll assume some basic knowledge of R here.

First, let's load the tidyverse library—a set of R packages that work together for common analytic tasks.

```
library(tidyverse)
```

Then, let's find the `.srt` files we created in the last step and then read them in, saving them to the object l. We have now read in the transcripts! They should look like this (here is the first one, accessed by indexing the first list item — the first ten rows):

R code snippet and output: Reading in the transcripts

```
file_paths <- list.files(path = ".",
                         pattern = "\\.srt$",
                         recursive = TRUE,
                         full.names = TRUE)

l <- map(file_paths, read_lines)

l[[1]] %>%
  head(10)

##  [1] "1"
##  [2] "00:00:00,000 --> 00:00:07,160"
##  [3] "Hey folks, Phil Plait here, and for the past few episodes"

##  [4] ""
##  [5] "2"
##  [6] "00:00:07,160 --> 00:00:11,880"
##  [7] "know about the structure, history, and evolution of the universe"
##  [8] ""
##  [9] "3"
## [10] "00:00:11,880 --> 00:00:14,160"
```

Now, we have our transcripts loaded, and our data ready. The next step is a big one that we'll introduce primarily through comments - this is code to create a manual *function* that we will use to read each of the transcript files:

R code snippet: A function to process transcripts

```
process_transcripts <- function(d) {

  my_nrow <- length(d) # this is to find out how long each transcript file is

  d %>%
    as_tibble() %>%
    rename(X1 = 1) %>% # to make this easier to type
    # create different values for each of the rows of the transcript
    mutate(id = rep(c("i", "time", "transcript", "blank"), my_nrow/4),
           index = rep(1:(my_nrow/4), times = 4) %>% sort()) %>%
    spread(id, X1) %>% # change the data from long to wide format
    select(-i) %>%
```

```
    # process the time stamps
    separate(time, into = c("start", "end"), sep = "-->") %>%
    # trim the time stamps so they are easier to read and use
    mutate(start = str_trim(start),
           end = str_trim(end))
}
```

Whereas with Python we used a "for loop", in R, for loops are less common than *apply* functions. These two approaches share a commonality: they are both used for iteration. Given the kinds of data R is chiefly intended to work with and how R as a programming language most efficiently works, apply functions are generally the better way to go. Here, we will use the `map()` function that is a part of the tidyverse package (specifically, the purrr package).

R code snippet: Iterating to create a single data frame with processed transcripts

```
ll <- l %>%
  # here, we use the apply function; possibly is used to handle errors
  map(possibly(process_transcripts, NULL))

# this removes any NULL list items that resulted from errors
ll <- compact(ll)

# this adds an index for the rows associated with each transcript

ll <- imap(ll, ~ mutate(.x, group = .y))

bound_rows <- ll %>%
  map_df(~.) # this changes the list of data frames into a single data frame
```

We are getting close to ready for analyses. Next, we process the transcript to create several variables that will be useful for our analysis. Most important among these is two variables we have created:

For these purposes, we employ Kirch's (2011) typology of uncertainty as both a **psychological experience** and a **mathematical object**. Kirsch writes, "uncertainty refers to a psychological condition of being in doubt (e.g., I am uncertain about something or someone. . .). It also refers to a statistical (or mathematical) object (e.g., a statistical estimation of uncertainty)" (Kirch, 2011, p. 57). As a psychological condition, uncertainty can be procedural, sociocultural, epistemological, and

ontological. As noted earlier, we focus on psychological uncertainty as a starting point and illustration.

Our computational approach is a dictionary-based approach (see Nelson et al., 2021 for a definition). This approach is a relatively straightforward text-analysis technique - it involves searching for key words in text. The dictionary is provided by the analayst, but we can use R to conduct the search automatically. We acknowledge that more complex approaches could be helpful, but we chose this approach given our aim of triangulating evidence—qualitative analyses can complement this approach by providing context and depth to what the computational approach reveals.

Our dictionary corresponding to our conception of psychological uncertainty follows.

R code snippet: Defining dictionaries

```
psychological_uncertainty <- c(
  "unsure", "not sure", "maybe", "kind of", "sort of", "don't know",
  "doubt", "doubtful", "no clue", "unclear", "confused", "confusing",
  "hesitant", "don't get", "don't understand", "ambivalent",
  "can't decide", "questioning", "question", "wondering", "wonder",
  "weird", "strange", "odd", "weirded out", "puzzled", "puzzling",
  "don't get it", "weird feeling", "weirdly", "skeptical", "skeptic",
  "guessing", "guess", "vague", "ambiguous", "indefinite",
  "uncertain", "iffy", "on the fence", "mixed up", "unsure what to do"
)
```

First, let's process the transcripts a bit further to create some useful variables and select columns.

R code snippet and output:

```
out <- bound_rows %>%
  mutate(start = str_sub(start, 1, 8),
         end = str_sub(end, 1, 8)) %>%
  mutate(start = chron::chron(times = start),
         end = chron::chron(times = end)) %>%
  mutate(duration = end - start) %>%
  select(index, start, end, duration, everything())
```

Next, we can apply these lists.

R code snippet: Conducting the dictionary-based analysis

```
# function to count words from a dictionary in a text
count_words <- function(text, dictionary) {
  sum(str_count(text, paste0("\\b", dictionary, "\\b")))
}

# apply the function to each row of your dataframe
out <- out %>%
  mutate(transcript = tolower(transcript)) %>%
  mutate(
    count_psychological_uncertainty = map_dbl(transcript,
    ~count_words(.x, psychological_uncertainty))
  )
```

The result of this step is the following data frame (represented through the use of an R function that summarizes data frames) below. We can see that the data frame contains over 36,000 rows, one for each utterance in the video. We can also see counts of psychological uncertainty for each utterance.

R code snippet: Resulting transcript

```
> d %>% glimpse()
Rows: 36,780
Columns: 12
$ group                           <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
$ channel                         <chr> "crash course", "crash cours...
$ playlist                        <chr> "astronomy", "astronomy", "a...
$ video                           <chr> "audio_A Brief History of th...
$ start                           <time> 00:00:00, 00:00:07, 00:00:1...
$ end                             <time> 00:00:07, 00:00:11, 00:00:1...
$ duration                        <time> 00:00:07, 00:00:04, 00:00:0...
$ blank                           <lgl> NA, NA, NA, NA, NA, NA, NA, ...
$ transcript                      <chr> "hey folks, phil plait here,...
$ count_psychological_uncertainty <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
$ path                            <chr> "./crash course/astronomy/au...
```

We can briefly explore the prevalence of psychological uncertainty with the following R code, which shows us that 747 utterances contain one word from our psychological uncertainty dictionary, and 36 utterances contain two.

R code snippet: Exploring the frequency of words associated with psychological uncertainty

```
> d %>% count(count_psychological_uncertainty)
# A tibble: 3 × 2
  count_psychological_uncertainty     n
                            <dbl> <int>
1                               0 35997
2                               1   747
3                               2    36
```

We are now ready to proceed to the qualitative analysis phase.

15.7 Data Analysis Step #4: Qualitative Analysis

In this step, we conduct a qualitative analysis in two phases.

15.7.1 Inspecting the Utterances With the Most Uncertainty Detected

First, we inspect the utterances with the most uncertainty detected for psychological uncertainty. We do so by arranging the above data frame in descending order based simply on the *count* (or frequency) of the number of uncertainty-related words in our dictionary detected.

For psychological uncertainty, we examined the 50 utterances with the most uncertainty-related words by reading the utterances and considering them in light of our definition of psychological uncertainty: "a psychological condition of being in doubt" (Kirch, 2011, p. 57), which includes procedural, sociocultural, epistemological, and ontological elements.

The most common forms of psychological uncertainty were procedural and epistemological.

Epistemological uncertainty was fairly common, evidenced by around one-third of the utterances with the greatest amounts of psychological uncertainty detected. Examples are as follows.

- ”We don’t know what kind of atmospheres these planets will have or what they’re composed.” (Crash Course - Astronomy)
- ”like maybe that animal is hard to find in the wild, or maybe it can’t be kept in captivity.” (Crash Course - Zoology)

Procedural uncertainty was also fairly common, present in around one-half of the utterances.

- ”Well, we don’t know what we don’t know.” (Crash Course - Biology)
- ”i guess maybe about that far?” (Veritasium - Misconceptions)

We also saw a degree of measurement or statistical uncertainty, even in these utterances that the computational analysis suggested were psychological in nature; these were relative uncommon:

- ”maybe the observation is wrong, or maybe we’re misinterpreting it.” (Crash Course - Astronomy)
- ”although the numbers are a little bit uncertain, something like a third to half of all stars”

As this is a tutorial, let us consider that a more systematic qualitative analysis indeed suggested that there are three forms of uncertainty that are common in science education videos: epistemological, psychological, and measurement. Were we to expand on this analysis, we would likely want to qualitatively investigate other videos to understand whether forms of uncertainty — and to substantially deepen our analysis of the utterances above by understanding their context in the videos and how truly common (or not) they are across the entire set of transcript data. For now, our purpose was to demonstrate how a finer-grained qualitative approach could complement the automated computational approach we carried out in the last step.

15.8 Findings and Discussion

In this chapter, we sought to demonstrate how a triangulating approach that combines computational and qualitative methods could be used to measure uncertainty. We played out this approach at the same that that we developed a data set that could be suitable for answering it - a collection of 272 science education-related YouTube videos. We showed four data analysis steps:

1. **Downloading YouTube Videos:** The study utilized Python libraries, namely ’pytube’ and ’whisper’, to extract and transcribe audio streams from selected YouTube playlists. This process efficiently converted a substantial corpus of science education-related videos into a format suitable for text-based analysis.
2. **Transcription Using Whisper:** The Whisper tool was employed to transcribe the audio files into textual data. This transformation was crucial in standardizing diverse video content into a uniform textual format, primed for computational analysis.

3. **Computational Text Analysis:** Applying a computational approach with the use of R, the study focused on a dictionary-based text analysis to identify the presence and frequency of terms related to scientific uncertainty in the video transcripts. This quantitative analysis provided an overarching perspective of the manifestation of uncertainty in the video content.
4. **Qualitative Analysis:** Complementing the computational analysis, a qualitative examination of the context and nuances surrounding uncertainty-related terms in the videos was conducted. This approach offered a deeper and more nuanced understanding of the nature and presentation of scientific uncertainty in the educational content.

The findings from this tutorial suggest that the representation of scientific uncertainty in educational videos is predominantly characterized by procedural and epistemological uncertainties. We note that the intent of working through these four steps was to illustrate how to access and create transcripts of YouTube videos and to demonstrate a triangulation approach. Of course, more systematic inquiry would be necessary to substantiate this finding. Here, we showed the very first stages of doing so, setting the stage for the establishment of the reliability and validity of a measure that we could use to answer substantive questions about the nature of uncertainty in educational videos. Later, such an approach could help us to better understand the role of uncertainty in science teaching and learning within and beyond classroom settings. We also note that the methodology delineated here can be adapted for analyzing other complex constructs across various digital platforms, thereby expanding the research scope within the field of educational technology and digital media analysis. For further reading on the triangulation approach used, we recommend Nelson (2020), Rosenberg and Krist (2021), and Tschisgale et al. (2023).

Link to analytic code in the OSF: `https://osf.io/v2x7j/`

Part III
Future directions

Chapter 16
Risks and ethical considerations in the context of machine learning research in science education (Cynthia M. D'Angelo)

Abstract Besides the tremendous potentials of machine learning (ML) methods, many ethical challenges such as biased datasets with regard to gender or race have to be considered. In this chapter, a conclusive reflection on the particular challenges in science education research based on the case studies and prior research will be outlined. Paths to address these challenges will finish this chapter.

Author(s): Cynthia M. D'Angelo

16.1 Bias and ethics and equity

Hopefully you are committed to minimizing biases in your research and interrogating your process in order to help achieve this goal. It may not be possible to remove all biases, but the more that you can engage in the reflection and strategies necessary to minimize biases, the more your work will be able to address issues of equity and justice in science education.

There are lots of potential biases to consider: race/ethnicity, language/linguistics, gender, disability, and socio-economic status. While that is a long list of biases to consider and interrogate in your work, the more you can address these biases, the stronger your work will be and more able to address the true diversity of experiences that students and teachers bring into a science classroom or learning environment.

It's not magic. It's math. It's important to remember that as you use these advanced techniques. They do not magically get rid of the biases in our society by doing complicated math. The biases come from the humans that are creating these ML methods and models and from the data being fed into them, all of which reflect the biases of our society. These ML techniques have been created to do specific things by humans. The more you can understand these motivations and the designed use cases of these different approaches, the more you will be able to understand the inherent

trade-offs when making decisions about whether or not to use ML techniques and which one is most appropriate for your purposes and research situation.

Part of this equity-focused approach is the need to be intentional about what you mean by equity. What or who are you designing for? Who are you centering? Who is being marginalized in this process? What kinds of questions are your asking and how are these questions privileging certain ways of being in a science class or teaching science? What does it mean for a ML model to be "accurate"? Who is it accurate for? Under what circumstances and contexts is it accurate? Is your model only accurate for students who fall in the most common categories that you are looking at or is more inclusive of students that typically fall outside those majority categories? With ML techniques, you need to think carefully about low-occurrence categories or situations and the students/teachers that fall into them. It is much more difficult to accurately model these low-occurrence events, so if these kinds of events are something you are interested in, you might want to consider different methods or modifications that will allow more of a focus on these events. If you are working with text data and natural language processing types of approaches that might help with auto-grading or evaluating short answers in science there are many issues to be aware of. For short science answers specifically, it's really complicated to do well. If you just want to check for some keywords or simple constructions, that's not too bad, but it's a simple approach and will give you limited information about the science concepts. It also privileges native English speakers, especially those who are particularly good at school English. You need to also ensure that you're not just checking to see if someone knows science vocabulary, but actually understands the concepts behind those words. That is much harder. More recent advances in ML are improving this type of task, but it is always important to look into what kinds of science answers were being used to train these more advanced models and who is represented (and not) in those data sets.

If you are working with video or image data and are using vision-based ML approaches you have another set of challenges. For instance, if you're trying to extract human skeletons to look at, you need to think about students with disabilities and why those skeletons (and the resulting analysis) might not be accurate/fair/appropriate for certain students and contexts. This challenge is not just for visible physical disabilities, but also for students that exhibit neurodivergent behaviors and how that might show up when tracking a person's movement. Again, with vision-based ML, it is essential to understand about the images (and labels) that have been used to train the models, as these historically have been not representative of the diversity of our student populations.

16.2 Purposes and trade-offs

When considering whether or which ML techniques you should use with your research, you need to think about your research questions and what you are trying to achieve with your research and for whom. In order to address the challenges of

these kinds of approaches you will have to make decisions that involve trade-offs with different approaches. Part of this process is to think carefully about why you are using ML and what your goals with it are. Would other kinds of analyses or methods be better? Sometimes there is a tendency to want to use the more "advanced" or newer techniques just because they are more advanced or newer. But that doesn't mean that they are better for answering your particular research questions with your data set and context. What are the trade-offs with different kinds of uses of ML or algorithms? Is ML the right tool for accomplishing your goals or would another approach be more appropriate?

On potential pitfall is to choose ML techniques in order to be more efficient with your research. Prioritizing efficiency can lead to problems - you are always making a trade-off when choosing one approach over another. You might want to ask yourself why you are prioritizing efficiency. Why is it important that this process be efficient? Is it because you don't have the sufficient resources to do this a different way? Is that a good enough reason to risk the many potential issues with a ML approach?

If you are using ML to predict outcomes for students or teachers, there are additional questions to consider. What is the goal of prediction for your study? Will the predictions end up coming back to the students or the teachers? How might a prediction about their future behavior or learning affect them? Care needs to be given if you are going to reveal these predictions to students or teachers, making sure to message effectively about their ability to change. It also means that you need to be even more sure that there are not major biases or errors in your model and analysis. It raises the stakes considerably for your analysis and you should be even more intentional in your design and reflection on your data set and approach.

Are these data about people learning (aka something that is in progress) or is it assessment data to evaluate learning that has happened? There are different considerations to make about your models depending on the kind of data you have. The stakes are higher for assessment or evaluative models, so it's more important to consider the ways in which your model might be biased towards certain groups or certain kinds of outcomes. These types of data are also typically missing context to a larger extent than in-progress learning data are. The stakes are lower for learning data, but also it's important to consider the nature of learning (or teaching) and wanting to perhaps reach different kinds of conclusions or produce different kinds of models with this type of data.

16.3 Data characteristics

The plan for collecting the data (including the structure of it, the modality of it, the levels of it, the conditions under which it was collected) to be used in your machine learning approach is the most important part of the process. The more you know about your data and how and why and how it was collected in the way that it was, the better able you will be to make careful and considered decisions about the construction of your model (including important pre-processing steps).

The modality of the data can also prioritize certain people and/or certain ways of demonstrating knowledge and skills. So it's important to consider these questions at the beginning of your research, when you are planning your data collection strategies, not just toward the end when you're doing analysis.

Is this (the output) going to narrow avenues for students or expand them? There are lots of different ways for students can show up and demonstrate their knowledge. What do you do with any outliers? What do outliers even mean in this situation? Sometimes outliers are just ignored or even deleted. But, in a lot of science education research contexts, these outliers are students. Would it make sense to ignore a student? Trying to fit people into boxes or categories just to put them into those boxes or categories may not be a good use of this kind of technology/tool, even if that is what it is especially good at.

There is also the issue of context in your data set. How much information do you have about the contexts under which the data were collected? How much information should you have? How do different contextual factors change within your data set and how are you taking that into account in your model?

Missing data is another important aspect of your data set to consider when using ML techniques. There are different kinds of missing data. Are the data missing because a student was absent that day? Or is it incomplete because a student wasn't able to finish an assignment? Do you, as the researcher, know why a student didn't finish something? Could there have been a fire alarm in the building or a medical issue or did they run out of time or did they not know the answer? As much as possible it's important to know about why the data are missing - what that incompleteness means - as different kinds of missing data need to be handled differently in how you process and interpret your data.

16.4 Privacy, transparency, and agency

Privacy is an important issue to consider not just when reporting your results but throughout the whole process, including the plan for your data collection. One way to help protect your participants' privacy is to not collect more data than what you actually need to answer your research questions. That includes meta-data that could potentially identify or expose your participant if the data were stolen. Some advanced ML techniques are able to identify individuals, even if names aren't included, because of the amount and detail of the data collected. So, taking care to protect and anonymize data as early as possible can be crucial to protecting the privacy of your participants.

One way to mitigate some issues related to privacy is to allow your research participants more agency in the data collection process. For instance, building in ways for participants to opt-out temporarily of data collection when they don't want certain things about them being collected. Give them more agency by choosing how and when their data is being collected and providing them more context and information about how their data is going to be used by you and your team.

You also need to think about what kinds of services you are accessing when using different ML approaches. For instance, some software might require you to upload your data to their servers in order to use the algorithms. This type of access is typically disallowed by the guidelines of most Institutional Review Boards - mostly due to the risk with putting your potentially sensitive education-related data on the servers of a company. Local solutions (that is, solutions that reside on your local computer or on a server that you or your institution control) are occasionally more difficult to implement, but are much safer for your participants.

16.5 Paths to address these challenges

There is no one central path to address these challenges. The challenges themselves, as outlined above, are myriad and depend on many factors unique to your data set, research questions, and science education setting. But a set of questions can help with finding the right path for you to wrangle the challenges that you face. The sections above contain many questions that you can ask of yourself as a researcher, your dataset, and your models in order to help minimize the challenges of using these kinds of techniques.

Data and algorithm auditing can be an important strategy to help mitigate some of the risks of using these techniques. This involves scheduling time as part of your project to intentionally investigate your data and the algorithms used and models produced for biases. You can proactively look for instances of different kinds of bias in your data. If you find them, you can then make changes to whichever part of the process you find the bias. Additionally, being transparent about this process and reporting it along with your more typical results can help others see the limitations and caveats with your results (that are true with all findings, but are not always disclosed). This can also help you and others be more intentional in your next data collection plan to help minimize these biases in the future.

Part of the solution to address these challenges with the risks of ML approaches is to use these tools conscientiously, understanding the risks, and only when willing to mitigate the challenges and be responsible. Educate students and research partners on how these algorithms work and how they could impact their lives (or learning or teaching).