Engineering Educators Bringing the World Together
2025 ASEE Annual Conference & Exposition
Palais des congrès de Montréal, Montréal, QC • June 22–25, 2025
Paper ID #45886

Automated Analysis of Knowledge Types in Computer Science Textbooks: A Natural Language Processing Approach to Understanding Epistemic Climate

Mitchell Gerhardt, Virginia Polytechnic Institute and State University

Mitchell Gerhardt is a Ph.D. student in Engineering Education and a M.S. student in Computer Science at Virginia Tech. He holds a B.S. in Electrical Engineering and worked as a software engineer for General Motors in Detroit, Michigan, before returning to graduate school. Mitchell's research focuses on learning in STEM graduate education; in particular, how graduate students recognize and learn the ways of knowing and doing typical of their disciplines. To this end, his research asks about the long-term implications of graduate student and faculty AI use for the nature of knowledge and knowing writ large.

Dr. Andrew Katz, Virginia Polytechnic Institute and State University

Andrew Katz is an assistant professor in the Department of Engineering Education at Virginia Tech. He leads the Improving Decisions in Engineering Education Agents and Systems (IDEEAS) Lab.

Automated Analysis of Knowledge Types in Computer Science Textbooks: A Natural Language Processing Approach to Understanding Epistemic Climate

Introduction

Curricular materials, such as textbooks, assignments, and lecture slides, convey messages about disciplinary values, assumptions, and beliefs [1]. They help students recognize and learn the ways of knowing and doing typical of their disciplines, promoting students' domain identification and knowledge construction processes. Textbooks have been used to examine the nature of knowledge presented across various fields, revealing the narratives, questions, and content they prioritize and value [2]. For example, Robinson's [3] analysis of introductory electrical engineering textbooks spanning roughly 80 years suggests that more recent versions prioritize fact-based content through rote procedure application than earlier, more theoretical versions. These findings align with other disciplinary perspectives that frequently find textbooks favor technical and procedural knowledge over alternative approaches, portray fields as fixed bodies of knowledge, and minimize the positionality of disciplinary experts in defining and shaping disciplinary knowledge [4], [5], [6], [7], [8]. When narratives remain unchanging and monolithic, they not only obscure the dynamic, inquiry-driven nature of disciplinary work but also risk marginalizing students whose experiences and identities are not reflected in dominant epistemic assumptions [9]. Further, by recognizing and addressing these limitations, educators and researchers can promote curricular materials that more accurately represent the evolving character of knowledge in engineering education and foster a learning environment that values diverse perspectives.

Support for this effort is limited, as few textbook analyses have been performed in engineering education; particularly those focused on how engineering knowledge is represented and conveyed to students. Small study samples reveal certain aspects of how engineering epistemology exists and is shown to students, but such investigations are limited in both their transferability to alternative contexts and in their accessibility to researchers looking to examine textbooks in their field. For example, contemporary engineering textbooks may include technical language alongside practical real-world examples to illustrate how theoretical concepts translate into tangible applications. Consequently, applying an analysis process from mathematics textbook education (e.g., [10]) may not capture these nuances in sufficient detail. Such approaches may also fail to highlight aspects considered valuable for engineering educators. Similarly, a coding scheme developed for electrical engineering textbook analyses may not have the same applicability for biomedical or civil contexts. It follows that these traditional techniques are also time-intensive and frequently require disciplinary expertise for their analysis and interpretation, limiting educators' ability to understand how students experience engineering epistemology as it is displayed to them by their programs. In this regard, engineering knowledge is not a static, unchanging entity but an evolving body of knowing and doing that is often "hidden" from students [11]. "Visibilizing," [12] or making visible its qualities and mechanisms thus requires techniques able to bridge disciplinary boundaries and their characteristic ways of knowing.

To address these methodological limitations in textbook analysis, we propose a framework that conceptualizes engineering knowledge as knowledge types: representations of distinct ways of knowing or doing engineering that each has a unique purpose, implications, and way it is expressed in text. Knowledge types can be observed across various disciplines and their curricular materials, allowing for greater transferability between analyses. Further, we propose a methodological framework leveraging natural language processing (NLP) techniques to analyze the presence of knowledge types in computer science (CS) textbooks. Accordingly, this study focuses on developing and validating this methodology through three main contributions: (1) establishing a taxonomy of textbook knowledge types in CS education, (2) developing a synthetic dataset generation approach to create training data for knowledge type classification, and (3) demonstrating the feasibility of using transformer-based models to distinguish between different forms of knowledge presentation. By focusing on an element of the epistemic climate students experience during their engineering education [1], we aim to validate methods that could eventually enable systematic analyses of its knowledge presentation and application; helping connect students' epistemic cognitions to their identity development [13], [14]. Specifically, the following research questions guide this study:

1) How can knowledge types in computer science textbooks be systematically categorized to reflect different forms of knowledge presentation and their epistemic implications?
2) How effectively can synthetic datasets be generated and validated for training NLP models to identify knowledge types in CS educational materials?
3) How well can transformer-based models distinguish between different forms of knowledge presentation in CS textbooks, and what patterns emerge in their classification performance across knowledge types?

Background

As students experience and engage in engineering programs, they are exposed to various messages about engineering, including norms and assumptions about engineering knowledge and knowing. Transmitted through pedagogical techniques, relationships, curricular materials, evaluations, and support systems, these messages comprise the epistemic climate, or the aspects of an environment that impact students' epistemic development. Curricular materials, in particular, play a unique role in the engineering epistemic climate given their role as authorities and conveyors of knowledge. A form of curricular materials, engineering textbooks, place a disproportionate emphasis on technical content; however, textbook analyses are limited in their scope and size due to their resource intensiveness. However, advancements in NLP have opened new avenues for broadening textbook analyses, which we look to leverage to examine the knowledge representations present in engineering textbooks.

Engineering Education Context and Epistemic Development

Engineering programs are incubators for students' identity and professional formation. Through norms, practices, content, and messages, engineering programs shape how students come to see themselves and their relationship with engineering. In classrooms, labs, departments, and institutions, students experiences shape their learning and cognitive development [15]. One aspect of this cognitive development considers students' beliefs about the nature of knowledge and knowing, referred to as their epistemic beliefs [16]. Students' epistemic beliefs represent how they understand and approach knowledge and knowing, including their beliefs and attitudes about what knowledge is, how it's acquired, and what makes something "valid." Such beliefs and attitudes about knowledge are central to engineering practice, where knowledge and practices are often borrowed from other domains [17], requiring a sophisticated ability to collect, evaluate, and apply information.

Engineering programs influence students' epistemic development through spaces and opportunities to learn and enact the knowledge and knowing processes associated with engineering practice [18]. Educational psychologists have referred to these environments as "epistemic climates," which can be understood as the overall environment or context in which ideas about knowledge and knowing are communicated, negotiated, and reinforced. More specifically, Muis et al. [1, p. 335] defines an epistemic climate as the "facets of knowledge and knowing that are salient in a learning or educational environment, that interact with and influence a learner's epistemic beliefs." From this conceptualization, they identify five central elements in educational epistemic climates: pedagogical approaches, authority structures, curricular materials, evaluation practices, and support mechanisms. These epistemic climate attributes interact and collectively shape how students perceive the nature of engineering knowledge; for instance, whether they see it as fixed, evolving, situated, interconnected, and how they learn to justify and validate what counts as that knowledge [1].

The epistemic climates in engineering programs provide opportunities for students' epistemic development [13]. Students receive cues about what counts as valid knowledge and processes of knowing, both from explicit content and from unspoken cultural and disciplinary norms. For example, Faber et al. [14] showed that participation in undergraduate research experiences (UREs) shapes engineering students' epistemic development through social interactions, legitimate participation in research practices, reflection, and identity formation. They noted that different research environments activated different epistemic beliefs and assumptions in students, contributing to their understanding of what research is and how engineering knowledge is generated. In a related study, Faber and Benson [19] similarly found that the epistemic climate significantly shapes students' problem-solving strategies by influencing their goals and motivation, depth of engagement, source selection, evaluation approaches, and willingness to consider alternative solutions. One student, Lily, exemplified this influence saying, "Yeah, if this [homework] was a huge part of my grade, or was a big test or big group project, I definitely would have looked further than on my online book, my textbook, and my slides. I would have gone to the library and found books on [the homework topic], and pulled my information to support my ideas, or talked about things that could oppose what I was trying to prove. Because it was just a homework assignment, I did not do that" [19, p. 692]. Montfort et al. [20] argue that engineering students' epistemic beliefs are likely outcomes of early engineering course structures, with their focus on simplified, well-defined problems that promote an epistemic climate portraying knowledge as certain and static. Students interpret instructor actions, like reassigning problems, as signals about the nature of knowledge, despite not being the instructor's intent. The authors reveal the complexity of civil engineering faculty's beliefs about engineering knowledge, but show faculty tended to evaluate knowledge claims differently in teaching versus research contexts. This suggests that faculty members' beliefs both shape and reflect an epistemic climate that can sometimes send mixed messages to students about the nature of engineering knowledge and problem-solving.

By illustrating the ways in which research experiences, course structures, and faculty practices convey messages about what counts as valid knowledge, these studies highlight the multifaceted nature of epistemic climates within engineering programs. Students' beliefs about engineering knowledge are shaped not only by explicit instruction but also by the subtle signals embedded in pedagogical choices, assessment formats, and disciplinary norms. Although existing work underscores the significance of these climates, further inquiry is needed to fully understand how various elements of an engineering curriculum operate in tandem to influence students' epistemic development. One such element, the curricular materials themselves—particularly textbooks—serve as a powerful yet often overlooked source of epistemic messages. The following section focuses on textbook analysis, examining how textbooks convey assumptions about engineering knowledge and the potential implications for students' epistemic beliefs and professional identity formation.

Curricular Materials and Textbooks as Conveyors of Epistemic Messages

Engineering programs traditionally rely on curricular materials, such as textbooks, lecture notes, and assignments, to complement time spent in classrooms [2], [21], [22]. These materials provide content and structure for students' learning, helping them relate to and understand engineering knowledge and practices. In traditional, lecture-based instruction that dominates engineering classrooms [23], curricular materials often function as instructional aides to reinforce what is delivered in class [2], [22]. They may also offer students structured continuity, additional depth, and varied methods of engagement to support their learning. Alternative forms of student-directed (or student-centered) learning approaches similarly emphasize curricular materials as entry points for classroom learning [22], [24], [25]. These pedagogical models often rely on students taking the initiative—discovering, exploring, and constructing knowledge on their own—making curricular materials function as both a primary source of information and a scaffold for learning. Across these and other pedagogical approaches common in engineering programs, curricular materials reflect priorities for student engagement and knowledge construction, doing more than merely transmitting information; they shape learning environments, support learning autonomy and collaboration, and influence how knowledge is constructed, used, and validated. Muis et al. (2016) highlight this central role of curricular materials in students' cognitive development, noting that students' critical thinking abilities depend on content that challenges their epistemic beliefs. That is, for students to become critical consumers of information, including their abilities to synthesize, evaluate, practice, and justify engineering knowledge, "content and domain-specific epistemology (i.e., methods used within a specific domain regarding the advancement of knowledge and processes of knowing) be embedded within the curriculum" [1, p. 352].

For these reasons, analyses of textbooks and other curricular materials can be directed to help researchers understand how engineering knowledge is described and conveyed to students. Many curricular materials currently used—not just textbooks—rely on predefined, decontextualized, closed-ended technical problem-solving, despite the complex, contextualized, open-ended sociotechnical problems practicing engineers face [26]. While some engineering textbooks (e.g., [27]) have sought to integrate social and technical content by incorporating social justice, ethics, problem definitions, and professional development considerations, many still rely on technically-focused a developed during the Cold War ([28], as cited in [24]). Robinson [3] analyzed engineering textbooks' approaches to teaching electrical circuits over about 80 years (1940-2017), focusing on how they present and understand engineering knowledge. Although more recent textbooks included brief "real-world" applications at the beginnings and ends of chapters, they primarily concentrated on mathematical analysis, problem-solving, and technical details, minimizing theoretical explanations. By contrast, earlier textbooks contained more detailed written explanations, emphasized theoretical understanding, and showed how fundamental principles explain diverse phenomena. These changes reflect disciplinary cultures and values, shaping how students understand engineering knowledge and knowing processes [1]. Each era's textbooks reflect and reinforce distinct professional cultures, hierarchies of knowledge, and assumptions about what constitutes "real" engineering practice. Engineering education research, therefore, must not only improve current textbooks and curricular materials to reflect engineering practice but critically evaluate them for the beliefs and attitudes they impart about engineering knowledge and knowing.

Textbook analyses conducted in other disciplines have similarly revealed their overly technical and procedural knowledge emphasis. For example, in science education, textbook analyses examining the representation of scientific knowledge find that it is often presented as bodies of facts rather than as processes of inquiry and experimentation [4]. Mathematical textbooks likewise tend to emphasize procedural knowledge over conceptual understanding in proofs, prioritize imitative reasoning, and portray mathematics as a fixed body of knowledge [5], [6], [7]. In history education, analyses have found variations in how textbooks handle conflicting historical accounts, presenting historical knowledge differently (e.g., as fixed narratives vs. interpretive processes) [8]. Notwithstanding, history education pedagogy frequently requires students to memorize facts, dates, and themes rather than engage in critical historical inquiry, decreasing students' interest in the subject [29], [30]. Common across these disciplinary findings is a tendency toward absolutist and static presentations of knowledge in curricular materials, which has consequences for students' critical thinking and engagement. Danielak et al. [9] underscore this concern, showing that rigid representations of engineering knowledge do not resonate with all learners, hindering their identity development and domain identification. To this end, Nilsson and colleagues [31], [32] exemplify the current challenges facing equitable and representative curricular materials, showing that chemistry textbooks commonly minimize the contributions of female scientists through sex-linked, subsidiary, communal, and doubt-laden language, lionize them as models of perseverance, and underrepresent females in images and other textbook components. Combined, textbook content informs students' conceptions of what information is considered valuable, which approaches to knowledge construction are legitimate, who has the authority to define success, and what practices are recognized as competent; that is, textbooks help students recognize and learn their epistemic climates [1], [12]. However, engineering and non-engineering textbooks frequently favor static and procedural representations of knowledge and knowing, reinforcing values and beliefs that neither represent actual practice nor prioritize diverse ways of knowing and learning in their fields.

Leveraging Natural Language Processing for Textbook Analysis

Despite being central components in many curricula, textbook analyses—particularly those examining the presentation of knowledge—are scarce in engineering education and are often limited in their number and scope. This may be a consequence of the variation in textbook content and use across engineering programs and disciplines, or because they require time-intensive manual analysis processes, namely close reading and coding. Additionally, there are challenges in maintaining consistency in qualitative coding across textbooks as researchers may not have the domain experience to recognize unique disciplinary language, particularly for complex argumentative or explanatory text [10]. These limitations hinder textbook comparisons across contexts and periods; the coding scheme from one textbook may not be applied to another or across time. By contrast, quantitative forms of textbook analysis have used frequency counts of certain terms or concepts to analyze content [33]. However, these approaches may rely on specific word sequences or language, omitting semantically similar language and reducing complex ideas to simple, countable units. These word sequences may cover broad patterns but miss deeper, more nuanced meanings and how ideas develop and connect throughout a text.

However, recent advances in generative artificial intelligence (gen-AI) have opened avenues to wider systematic textbook analyses, including those focused on knowledge representations. In particular, natural language processing (NLP) a subset of gen-AI, enables computers to quickly parse and understand text by identifying the meaningful parts of sentences [34]. Since the release of ChatGPT and similar chatbots, engineering education researchers have explored diverse use cases of NLP, including for analyzing student writing and assignments, examining curriculums, research data processing, student support, and assessment [35], [36], [37]. Recent work by our research group [38] has also demonstrated the potential for NLP to aid qualitative thematic analysis by expediting the codebook generation process. Importantly, these efforts take advantage of how NLP handles semantically and syntactically different text by identifying patterns between word embeddings. Models learn these patterns from massive amounts of text data through a process called "self-supervised learning" during which the model predicts missing words or next words in sequences [34]. This builds a base understanding of language by encoding the role and meaning of words into their embedding patterns and relationships. NLP researchers commonly refer to models like BERT, GPT, T5, RoBERTa, etc. with this base understanding as "foundational models" because they serve as a foundation that can be adapted for many downstream tasks [39], [40]. As an example, the GPT model from OpenAI refers to the core "Generative Pre-trained Transformer" model, which is used as the foundation for all GPT variants, including GPT-4o. These foundational models capture general language and transfer well across domains, enabling them to be tailored for different tasks through processes like fine-tuning. Fine-tuning augments a pre-trained model by re-training and optimizing it for a specific, smaller dataset. This process improves its performance in tasks that require an advanced understanding of that dataset [41]; for instance, fine-tuning a model using clinical notes to identify patients' demographic information [42].

Applied to textbook analyses, fine-tuning a model could theoretically be performant at determining the types of knowledge represented in textbooks. However, this presents two questions: What are "knowledge representations," and how can a model distinguish between their "types?" To illustrate, for a model to ascertain that a textbook predominately contains mathematical knowledge, it would need a functional concept of mathematical knowledge to guide its detection. The following section addresses these questions, which are further complicated by the nonexistence of a dataset containing engineering textbook passages labeled for the types and styles of knowledge they possess. In this investigation, we sought to address this need by exploring the potential of synthetic datasets for fine-tuning. As a preliminary step toward this goal, we leveraged large-language models (LLMs)—massively-sized (billions to trillions of parameters) pre-trained models focused on language tasks [37]—to generate a synthetic dataset of textbook passages labeled according to their knowledge representations. From this, we illustrate the initial performance of a fine-tuned BERT model, demonstrating the possibility of systematic and rapid textbook analysis according to their knowledge content and representation.

Method

To generate a labeled synthetic dataset of CS textbook passages, we first needed to define the labels, which we refer to as knowledge types. We then evaluated sample CS textbooks to identify instances of knowledge types across sections, chapters, and books before leveraging a LLM to help consolidate these instances into a set of distinct knowledge types; each representing a category of information or understanding that serves a particular purpose in engineering education and practice. This set of knowledge types allowed us to generate 10,000 samples of labeled synthetic textbook passages, occurring in random textbook locations (e.g., section beginning, subsection end, after equation, sidebar) and contexts (worked example, failure analysis, practice application, derivation proof). We then present the initial results of a fine-tuned foundational BERT model trained on this synthetic dataset.

Defining Knowledge Types

To evaluate knowledge representations, we first needed to define them. We opted to scope this investigation to CS textbooks because of our disciplinary expertise and to narrow the potential outcome space of knowledge types, though future research could explore their transferability to other engineering disciplines. Unfortunately, we were unable to find existing research that directly proposed a model for how knowledge is conveyed in text, particularly in engineering text. Consequently, we conceptually defined a knowledge representation as a "knowledge type," or a distinct category of information or understanding that serves a particular purpose in engineering education and practice. Importantly, a knowledge type should be able to "detach" from engineering textbooks, meaning they should be observable and applicable to other curricular materials and disciplinary venues. This is because knowledge types reflect different aspects of what students need to know and understand; attributes of an epistemic climate that exhibit coherence across the environment where it operates [12]. For example, conceptual knowledge represents theoretical understanding, like what time complexity means, while practical knowledge might apply conceptual concepts and show how time complexity can be used to compare algorithms. Textbooks often use many such knowledge types throughout their material, weaving them together through examples, case studies, and foundational lessons. Accordingly, we defined knowledge types as having three qualities: 1) Knowledge types have a distinct purpose, 2) Knowledge types have characteristic ways they are expressed in text, and 3) Knowledge types have different implications for how students understand what counts as knowledge in engineering. These criteria are broken down below.

Knowledge types have a distinct purpose. Each knowledge type serves specific educational and professional goals. It has characteristic learning outcomes and contributes differently to engineering understanding and functions. This premise recognizes institutional definitions of engineering knowledge, such as the Engineering Body of Knowledge published by the National Society of Professional Engineers (NSPE), which includes capabilities such as mathematics, natural sciences, design, and communication [43]. Each of these capabilities serves unique roles in professional practice, which we interpret as representing a distinct way of understanding or doing. In a simplified example, conceptual knowledge builds the theoretical understanding necessary for engineering analysis and reasoning. This differs from practical and professional knowledge which develops the capabilities to solve real engineering problems and work efficiency in the profession. While these knowledge types are distinct in their goals and purposes, they are compatible and may be connected within a textbook lesson. Their purpose is also socioculturally and historically defined, reflecting shifting and different meanings, interpretations, and values to different individuals [44].

Knowledge types have characteristic ways they are expressed in text. Specific vocabulary, terminology, sentence structures, and rhetorical patterns indicate different knowledge types. These variations are critical because they literally define the context, subjects, emphasis, and outcomes present in text, thereby representing the mechanism by which different epistemic messages are conveyed to students [45], [46]. For example, conceptual knowledge might use abstract and theoretical language, mathematical expressions, universal statements ("In all cases…"), or particular definitional structures ("X is defined as…"). This contrasts practical and professional knowledge which might use action-oriented language, conditional statements ("If X occurs, then…"), real-world references, and procedural descriptions. Because of these linguistic variations, we can leverage NLP techniques to learn and differentiate between knowledge types. However, while linguistic variations exist between knowledge types, we assume there are fewer variations within one [47]. The smaller variations can thus be conceptualized as occupying a unique region in a high-dimensional space defined by word embeddings, which the model learns to associate and categorize [34]. Put another way, defining knowledge types as having characteristic linguistic features allows both students and NLP models to learn their purposes, features, and outcomes.

Knowledge types have different implications for how students understand what counts as knowledge in engineering. Different knowledge types present different implications for students' understanding, including what counts as knowledge in engineering, how knowledge is justified, how certain/uncertain knowledge claims are, and who has authority over knowledge [48], [49]. This premise draws from discourse analysis theory, which evaluates how text or speech conveys significance, practices, identities, relationships, politics, connections, sign systems, and knowledge [50]. Consequently, the prevalence and distribution of different knowledge types in textbooks can engender biases in how engineering knowledge is presented to students, including what types of knowledge are valued and prioritized. For example, conceptual knowledge may imply that knowledge is based on mathematical and scientific principles, inscribing a high degree of certainty and deriving authority from proofs. However, practical and professional knowledge may imply that knowledge is validated through successful applications that may have multiple solutions. Its examples may showcase engineers balancing uncertainty and trade-offs, with authority coming from professional experience, standards, and application viability. In this way, knowledge types convey different messages about the source, certainty, connectedness, and justifications of engineering knowledge, shaping students' learning outcomes and understanding of the profession.

This structured approach to defining engineering knowledge serves two purposes. First, from a computational perspective, these categories provide boundaries and characteristics that NLP model can learn to recognize. This enables both the generation of synthetic training data and the development of classification models that can distinguish between different forms of knowledge presentations in educational texts. Using a common CS concept and popular sorting algorithm—QuickSort—to illustrate, conceptual knowledge markers might include abstract principles (“divide-and-conquer paradigm”), explanatory language (“stems from”, “creates”), and theoretical concepts (“efficiency”, “partitioning strategy”). Conversely, procedural knowledge markers pertaining to QuickSort might include numbered steps, implementation verbs (“select”, “partition”, “apply”), specific operations (“recursively”), and direct instructions. Given these differences, NLP models can be trained to distinguish between knowledge types and used to generate synthetic training examples by following linguistic patterns. Second, from an educational research perspective, this framework enables a systematic analysis of how knowledge is conveyed in engineering textbooks by identifying their prevalence and variations. Ultimately, whether using a fine-tuned model or not, quantifying the distribution of knowledge representations across engineering textbooks allows researchers to examine potential biases, identify gaps in knowledge coverage, and understand the implicit messages being conveyed about what types of knowledge are valued in the field. Current textbooks appear to emphasize some knowledge types over others [3], [26], so it may be that a more inclusive and multifaceted approach to conveying knowledge defines the next generation of engineering curricular materials; a mix of theory and applications mirroring engineering practice.

Determining Knowledge Types

This knowledge type definition guided our development of ten distinct knowledge types using a combination of manual analysis and gen-AI techniques. Limiting our analysis to CS textbooks provided greater control over both the manual analysis and the subsequent generation steps which required a detailed prompt. That said, future analyses can look to replicate this process to contribute to a more representative and generalizable repository of knowledge types present across various disciplinary curricular materials. Notwithstanding, our definition of knowledge types requires that their characteristics are observable and consistent across different CS educational materials. While some natural variation inevitably exists, analyzing multiple CS textbooks across a curriculum enables us to identify common patterns in how knowledge is presented to students.

To determine different knowledge types, we began by examining CS textbooks for the authors’ university’s undergraduate curriculum. We identified the primary textbook for each required core CS course, as shown in Table I. When digital versions were unavailable, we used comparable alternatives that were accessible online. This step was to ensure the future fine-tuned model can assess actual engineering textbooks without the need for additional transcription.

TABLE I
CS COURSES AND TEXTBOOKS

Course Name	Textbook

CS 1114 – Intro to Software Design	OpenDSA CS 1114 Online Textbook – Intro to Software Design
CS 2114 – Software Design & Data Structures	OpenDSA CS 2114 Online Textbook – Software Design & Data Structures
CS 2104 – Intro to Problem-Solving for CS	Whimbey, A., Lochhead, J., & Narode, R. (2013). *Problem solving and comprehension* (7. ed). Routledge
CS 2505 – Intro to Computer Organization I	Patt, Y. N., & Patel, S. J. (2004). *Introduction to computing systems: From bits and gates to C and beyond* (2. ed). McGraw-Hill Higher Education
CS 2506 – Intro to Computer Organization II	Patterson, D. A., & Hennessy, J. L. (2021). *Computer organization and design: The hardware/software interface* (Sixth edition). Morgan Kaufmann
CS 3114 – Data Structures and Algorithms	OpenDSA CS 3114 Online Textbook – Data Structures & Algorithms
CS 3214 – Computer Systems	Bryant, R. E., & O'Hallaron, D. R. (2016). *Computer systems: A programmer's perspective* (Third edition). Pearson
CS 3604 – Professionalism in Computing	Spier, R. (Ed.). (2002). *Science and technology ethics*. Routledge
CS 3304 – Comparative Languages	Sebesta, R. W. (2019). *Concepts of programming languages* (Twelfth edition). Pearson

These textbooks were evaluated to find similarities in their content, including chapter/section structure patterns, content presentation, common transition elements, and recurring educational components. We employed a process of notetaking and clustering where passages were recorded and grouped into a priori types, such as conceptual, practical, ethical, procedural, and mathematical knowledge, while allowing for new possibilities [51]. This process yielded several passages, which were then given to Claude, the large generative model from Anthropic along with the following prompt:

> You are an expert social science researcher studying computer science textbooks. Given a collection of engineering textbook passages, let's develop a framework for categorizing them into distinct knowledge types. For each passage, consider: 1) Content characteristics: (a) What is the primary purpose of this passage? (b) What information is being conveyed? (c) How is it being presented? 2) Linguistic features: (a) What vocabulary and phrasing is used? (b) What sentence structures appear? (c) What discourse patterns are present? After examining several passages, identify emerging patterns that suggest distinct knowledge categories. For each proposed category: 1) Define its distinguishing characteristics, 2) Explain how it differs from other categories, 3) Provide example passages that typify this category. Test your categories by attempting to classify new passages. If you find many passages that could fit multiple categories, refine your

framework to make the categories more distinct. Please analyze the provided passages and propose a categorization framework.

Using a LLM for this task was partially exploratory: how well could an LLM consolidate disparate textbook samples with moderate direction? As shown in Table II, this method appeared to work well and initially yielded 14 codes, which were reduced to 10 according to the knowledge type criteria. We used a zero-shot approach based on similar NLP-based qualitative coding techniques in engineering education research [38], [52], and the set was evaluated against the initial samples to verify their correctness, distinctness, and coherence. For example, the following passage from *Computer Systems: A Programmer's Perspective* was initially labeled as conceptual and aligns with the conceptual (CON) knowledge type: "Combinational circuits that perform word-level computations are constructed using logic gates to compute the actual bits of the output word, based on the individual bits of the input words" [53, p. 404]. However, these comparisons were judgment-based, and no inter-rater reliability was calculated. Future research should continue exploring the efficacy of these tools, determining whether NLP-based techniques for qualitative analysis constitute valid alternatives while identifying potential limitations in their application.

Table II
TYPES OF CS KNOWLEDGE AND DEFINITIONS

Code	Abbreviation	Definition
Conceptual Knowledge	CON	Core theoretical principles and fundamental concepts that form the basis of CS understanding, including basic theories and principles that underpin the CS discipline, explanations of key CS concepts and how they relate to each other, fundamental laws and equations central to the field, and abstract models and frameworks used to understand CS phenomena
Historical Knowledge	HIS	Information about the development of CS concepts, techniques, and technologies over time
Procedural Knowledge	PRO	Step-by-step explanations of problem-solving methods, experimental procedures, or design processes
Interdisciplinary Knowledge	INTER	Connections between CS and other fields
Epistemic Knowledge	EPIS	Information about how knowledge is constructed, validated, and evolves within CS
Metacognitive Knowledge	META	Guidance on how to approach learning and problem-solving in CS
Ethical Knowledge	ETH	Discussion of ethical considerations in CS practice and research
Mathematical Knowledge	MATH	Equations, derivations, and mathematical models used in CS
Uncertainty and Limitations	UNC	Discussions about the limitations of current knowledge and areas of ongoing research or debate in CS

Practical and Professional Knowledge	PRAC	The application of CS concepts in real-world contexts, including the use of current technologies and adherence to professional standards and practices. This encompasses case studies and real-world problem-solving scenarios, information about current CS technologies, tools, and software, professional standards, codes, and best practices in CS, and practical design processes and decision-making in CS projects

Generating Synthetic Dataset

Equipped with distinct knowledge types, we developed a structured data generation approach using a LLM. By crafting prompts that specified the desired CS topic, textbook context, location, and target knowledge types, we were able to generate synthetic textbook passages with known labels. This approach allowed us to create a diverse, balanced dataset spanning different knowledge categories while maintaining control over the distribution and combination of knowledge types. The synthetic dataset serves dual purposes: providing training data for our fine-tuned classification model and establishing a benchmark for evaluating its performance across multiple knowledge representations and textbook contexts. However, using a synthetic dataset poses many questions regarding the data's validity. Significant research has documented limitations of LLMs, including that they can reproduce or amplify bias in their training data [54] and generate fluent and plausible-sounding text but factually incorrect [55]. While these are areas of concern for synthetic datasets broadly, LLMs have demonstrated notable proficiency in generating "textbook-like" explanations, conveying complex concepts clearly and with structures resembling textbooks [56]. These abilities likely result from extensive training data, including various educational materials, and LLMs' pattern recognition and generalization capabilities, allowing them to cue into characteristic textbook explanations and connect multiple topics. For example, LLMs have been widely used in student tutoring systems, which leverage their explanatory capabilities [57]. These tools often include instructions that specify the desired outputs. Prompts such as "Explain this for a 5th grader" or "Talk to me at a graduate level," instruct the model to vary the output's vocabulary complexity, abstraction, field-specific terminology, and amount of background information [58]. While less work has examined the accuracy and consistency of these educational personas over long conversations, with carefully designed prompts and appropriate contextual constraints, LLMs can likely capture the essential features common in educational materials, particularly for short outputs [59]. Although these findings do not substitute a fine-grained comparison between generated and authentic samples, we believe they support our approach given its textbook-specific emphasis and reliance on short LLM outputs. Future investigations should examine the authenticity of generated engineering material, particularly given its growing use among students for various educational needs [60].

We generated diverse examples of textbook content across different CS topics, contexts, and textbook locations, which employed controls to ensure an authentic representation of knowledge types. Each generated excerpt was contained by, 1) a specific CS topic (e.g., Software Design, Data Structures), 2) a textbook context (e.g., main text, example, exercise), 3) a location within the text (e.g., section beginning, after equation), and 4) one or more target knowledge types from Table II. This approach resulted from many iterative rounds of prompt designing, which began with a simple prompt focused on generating multiple examples for a single knowledge type, requiring strict JSON formatting, and including basic guidelines for authenticity and variety. However, these initial experiments showed that generating authentic textbook passages required providing more contextual information to the model. Without this context, the generated samples tended to use repetitive patterns and formulaic introductions (e.g., “In this passage...” or “[Topic X] is an approach that...”). While these patterns can appear in textbooks, they don’t reflect the full diversity of textbook writing styles. To address this limitation, we enhanced our prompts by incorporating specific contexts and passage locations. We again used the Claude model as a research assistant to help generate potential contexts, and after refinement, identified various contexts and locations which are not included in this manuscript due to space constraints. To provide some examples, the list of 27 contexts included items such as proof, case study, performance analysis, and debugging process, while items such as section beginning, paragraph end, after figure, and sidebar, were included in the list of 14 locations. Yet this approach still proved inadequate; there was too much regularity between outputs, particularly those sharing a knowledge type and context/location. To remedy this, we modified the prompt to handle multiple knowledge types simultaneously while simplifying the output structure to generate single examples rather than multiple ones. These changes improved the generated content quality, but we still observed repetitive patterns and themes in the outputs. Subsequent prompt versions incorporated more specific guidance about content placement and context within topics, explicitly discouraged common introductory patterns, and introduced the concept of assumed prior knowledge—allowing passages to begin as if continuing from previous content. We also added clarifications about both content placement and writing style to enhance contextual authenticity. In this respect, we want to underscore the importance of using gen-AI to complement, not supplement, researcher decisions. The tendency toward automation bias threatens integrity and knowledge creation if left unchecked [61].

The final prompt version (available upon request) provides more detailed guidance for natural writing style and location-appropriate content. This version achieved a better balance between sophisticated textbook writing and proper implementation of knowledge types. Importantly, the entire prompt development process used the `qwen2.5:14b-instruct` model [62], which was selected because it was large enough to follow complex prompt instructions (unlike smaller models) while remaining computationally efficient for large-scale generation. Efficiency is relative, however, as we generated 10,000 labeled textbook passages in approximately three days using an Apple M2 Ultra Mac Studio equipped with a 24-core CPU, a 60-core GPU, a 32-core Neural Engine, 192 GB of unified memory, and 8 TB of SSD storage; Table III shows a few examples. While generating such a dataset manually would require significantly more time, this duration is still substantial, particularly in light of concerns about the environmental impact of gen-AI technology [63], [64]. We take these concerns seriously and implemented several measures during the generation process to ensure output validity and correctness, minimizing the need for additional rounds of generation.

Table III
EXAMPLE GENERATED PASSAGES

Course	Knowledge Type(s)	Context	Location	Example

Computer Systems	UNC	Debugging Process	Section Beginning	Debugging in computer systems often involves identifying and resolving issues that arise during software execution. Current techniques, such as static analysis and runtime monitoring, are widely used but have limitations. For instance, while static analysis can detect potential errors before code execution, it may not capture all dynamic behaviors of the program. Ongoing research focuses on integrating machine learning algorithms to predict and mitigate these shortcomings, thereby enhancing the accuracy and efficiency of debugging processes. This field remains an active area of debate due to the complexity of balancing computational overhead with diagnostic precision.
Data Structures and Algorithms	CON, MATH, UNC, EPIS	Mathematical Derivation	After Figure	Figure 2 illustrates the time complexity of a binary search algorithm, showing how it performs O (log n) operations on an array of size n. To derive this result mathematically, consider that each step in binary search halves the remaining portion of the array to be searched. Let T(n) represent the number of steps required for binary search on an input of length n. The recurrence relation is given by $T(n) = 1 + T(n/2)$, where the constant term '1' accounts for the comparison operation at each step. By solving this recursive equation, we find that $T(n) \approx \log_2(n)$, confirming the logarithmic time complexity as observed in the figure.
Computer Organization II	PRAC, ETH	Optimization Process	Section Middle	Optimizing computer performance often involves enhancing memory access times and reducing latency. Techniques

				such as caching, which leverages faster-accessible storage for frequently used data, can significantly improve system throughput. Engineers must balance the size of cache with its speed; larger caches offer more storage but may introduce additional delays in retrieval. Other considerations also play a role, such as ensuring optimization strategies do not compromise user privacy or security by exposing sensitive information through shared memory resources.

In summary, to address the challenge of limited labeled training data for classifying knowledge types in CS textbooks, we developed a data generation pipeline using LLMs. Through iterative prompt refinement, we generated a dataset of 10,000 labeled passages, ensuring diversity across topics, contexts, and knowledge types. The final dataset serves as both training data for a classification model and a benchmark for evaluating its performance.

Model Fine-Tuning

Our study implemented a foundational model for identifying and classifying distinct knowledge types in engineering textbooks. Specifically, we utilized BERT [65] with a custom classification head, incorporating a dense layer and dropout regularization for multi-label classification. Training employed the AdamW optimizer [66] with a learning rate schedule (initial rate: 5e-5), binary cross-entropy loss function, and gradient clipping (threshold: 1.0). Key hyperparameters included a batch size of 16 (for GPU used), maximum sequence length of 512 tokens (longer than each sample), and early stopping patience of 3 epochs. This approach represents an initial exploration to determine whether fine-tuning can capture the differences between knowledge types given that additional architecture decisions could be subsequently added.

Analysis of the model's performance revealed several key patterns, with performance varying notably across different knowledge categories, as shown in Table IV. ETH showed exceptionally strong performance across all metrics, with nearly perfect precision (~0.98) and high recall (~0.95), while HIS demonstrated strong recall (~0.92) but lower precision (~0.81). MATH and PRO categories showed the largest precision-recall gaps, with MATH having notably higher recall than precision. UNC showed balanced and strong performance across metrics (~0.88 for both precision and recall). The optimization of classification thresholds revealed important differences across knowledge types. Most categories required relatively high classification thresholds (0.8), including EPIS, ETH, HIS, MATH, META, and PRO. However, PRAC showed the lowest optimal threshold at 0.55, suggesting these knowledge types may be more readily identifiable, while INTER and UNC had moderate thresholds (0.6). The variation in optimal thresholds indicates different levels of certainty needed for reliable classification across knowledge types. Analysis of co-occurrence analysis revealed several significant patterns in how different types of knowledge are presented together in the samples, as shown in Figure I. We also found strong co-occurrence between MATH and CON (338 instances), suggesting mathematical concepts are often presented alongside conceptual explanations. Following suit, there was a notable relationship between PRAC and ETH (274 instances), potentially indicating the integration of ethical considerations in practical applications. Substantial overlap between META and PRO (252 instances) suggests procedural knowledge often includes metacognitive elements, although the limited co-occurrence between META and most others may indicate a gap in the synthetic dataset or the model's inability to recognize it in non-procedural contexts. Similar non-detections across the knowledge types may indicate that more targeted samples are required or that additional architecture features must be implemented for nuanced detection. That considered, analysis of prediction confidence distributions provided additional insights into the model's classification behavior. CON showed the highest mean prediction confidence (~0.48) with substantial variance, while PRO demonstrated relatively high mean confidence (~0.43) but with the largest variance. ETH and HIS showed lower mean confidence despite their strong performance metrics. Most categories showed significant confidence variance, indicating context-dependent classification certainty.

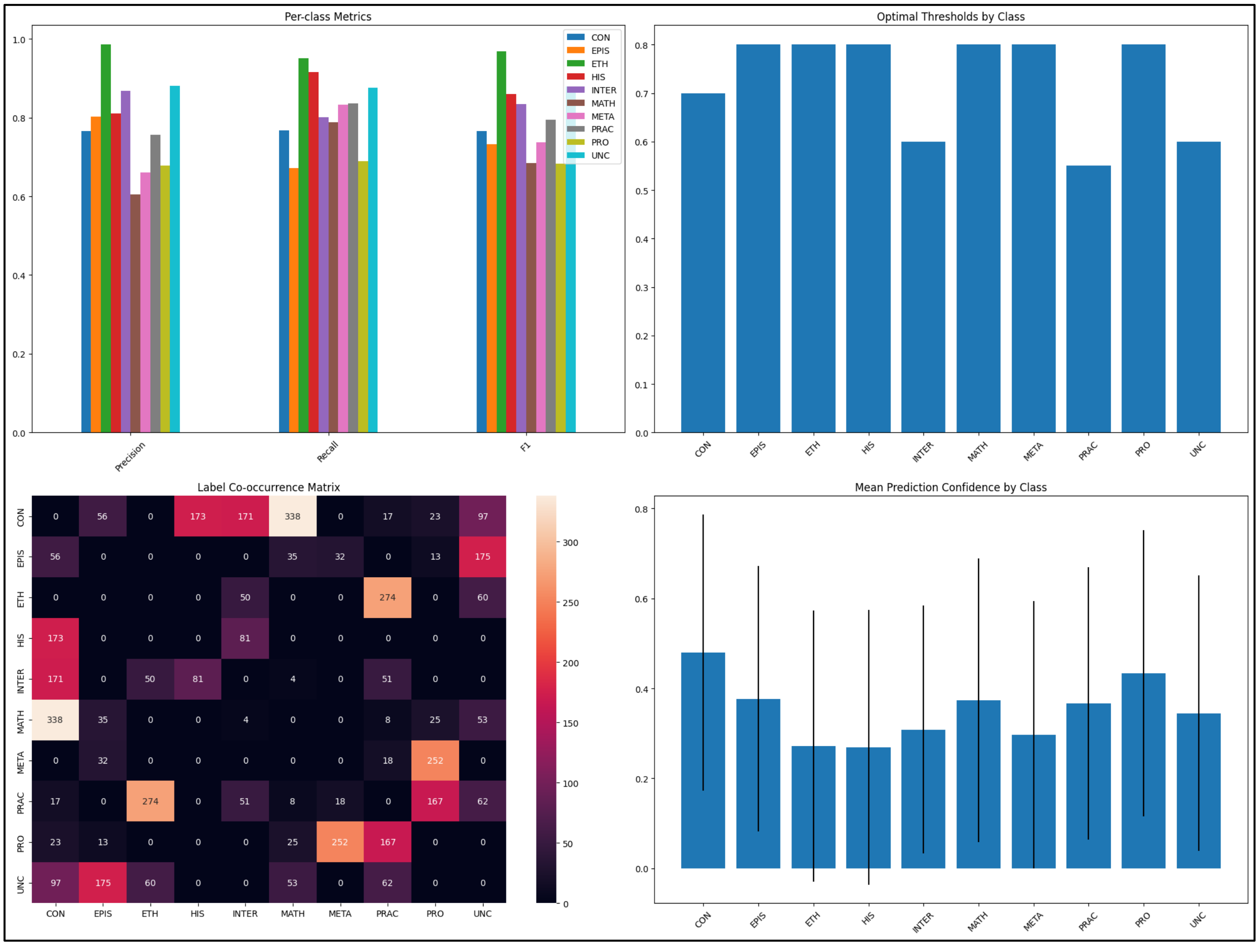
These findings provide quantitative evidence for how the CS textbook samples construct epistemic climates through various knowledge representations. The strong co-occurrence patterns between certain knowledge types suggest that the samples frequently present knowledge in integrated ways, potentially supporting the development of sophisticated epistemic understanding. Although this was partially controlled by the generation process, seeing the same observation in the training results indicates the model was able to identify knowledge type overlaps while adequately distinguishing between them. Within this co-occurrence, the varying confidence distributions and optimal thresholds across knowledge types show that some forms of knowledge are more explicitly marked in the samples than others, which has implications for how students might perceive and internalize different types of knowledge in their education. However, these results simply reflect patterns within the generated sample. To truly capture different knowledge types present in CS textbooks and comment on the resulting epistemic climate, the model must evaluate the textbooks in Table II. However, even this would be insufficient to comment on CS as a discipline. Future investigations will apply similar techniques in other CS programs and across engineering disciplines to understand the many ways engineering knowledge is presented and conveyed to students.

Table IV
PER-CLASS PERFORMANCE METRICS FOR EACH KNOWLEDGE TYPE

Label	Threshold	Precision	Recall	F1-Score
CON	0.7	0.766	0.767	0.766
EPIS	0.8	0.802	0.672	0.732
ETH	0.8	0.985	0.951	0.968
HIS	0.8	0.811	0.916	0.860
INTER	0.6	0.869	0.801	0.834
MATH	0.8	0.604	0.789	0.684
META	0.8	0.661	0.833	0.737

PRAC	0.55	0.756	0.836	0.794
PRO	0.8	0.678	0.689	0.683
UNC	0.6	0.881	0.876	0.879

Fig. 1. Initial model training results



Discussion

This research is a preliminary step toward future investigations that explore the knowledge composition of various CS curricular materials, including textbooks. Yet, the results from the generation and training processes identified present several theoretical and methodological implications for engineering education. Firstly, while not performed in this study, investigating the distribution patterns of knowledge types in engineering course materials could reveal implicit messages about what kinds of knowledge are valued and privileged in engineering education; “visibilizing” the knowledge mechanisms undergirding disciplinary practices [12]. Our computational analysis using NLP shows that while certain knowledge types (e.g., ETH, HIS, and UNC) have distinctive linguistic and structural characteristics that make them easily identifiable, there is also significant overlap and co-occurrence between knowledge types in the samples. This integration pattern provides evidence that students must learn to navigate multiple, interrelated forms of knowledge rather than developing through discrete stages or categories of understanding [25], [45], [67]. If observed across multiple textbooks and programs, this finding would highlight an area of tension between students’ epistemic beliefs and the materials they encounter. Current research shows that while students may acknowledge multiple valid perspectives can exist, they often view engineering knowledge as primarily technical, certain, and objective (e.g., [68], [69], [70]). Further, students’ epistemic beliefs can clash with their experiences or outcome expectations in educational settings, particularly when those settings differ from traditional approaches [71]. Thus, the presence of highly interconnected knowledge types across engineering curricular materials would be problematic for other aspects of the epistemic climate (instruction, support, evaluation) that do not match this interconnectedness, potentially diminishing students’ learning [72]. Furthermore, there should be an equal concern if this interconnectedness in knowledge types is not equally distributed across engineering materials. For example, the model’s varying confidence thresholds suggest what forms of knowledge are more thoroughly embedded across materials than others: low represents more embedded because it requires less “confidence.” If course materials predominantly emphasize technical knowledge types while treating other forms as supplementary, this could reinforce technically dominant disciplinary epistemologies [73], [74], [75], [76]. These epistemologies would shape students' domain-specific beliefs about knowledge [77], [78], potentially marginalizing those who approach engineering knowledge through different epistemological frameworks [9] and failing to prepare them for actual engineering practice. Practically, our quantitative approach to assessing knowledge type distributions offers new tools for curriculum development and evaluation. Authors and instructors can use these computational insights to analyze their materials and create more intentionally balanced learning experiences that incorporate diverse knowledge types [79]. For example, our analysis could help identify where ethical considerations might be integrated into technical problem-solving, or where historical context could enrich understanding of engineering principles. This systematic approach to analyzing epistemic climate could help engineering educators foster learning environments that better support the development of sophisticated epistemic beliefs while making the field more inclusive of diverse ways of knowing and learning.

Methodologically, this research offers several significant contributions to analyzing epistemic climate in engineering education curricular materials. First, the use of synthetically generated training data, validated through independent labeling, demonstrates a potential approach to overcoming the common challenge of limited labeled datasets about epistemology in educational research. This methodological process is particularly valuable for studying epistemic features of curricular materials, where manual annotation would be time-intensive and expert-dependent. Traditional methods of analyzing epistemic climates have relied heavily on qualitative analysis of classroom observations, interviews, and manual content analysis [1], [12], [78]. While these approaches provide rich insights, they demand significant resources and are difficult to scale. Our validation of synthetic data generation suggests researchers could potentially bootstrap their analysis processes, using carefully generated synthetic data to train initial models before refining them on smaller sets of authentic materials. In addition to the synthetic data validation, our NLP-based approach offers a complementary method that can analyze large volumes of curricular materials systematically. The varying model performance across knowledge types suggests both the promise and limitations of current computational approaches in capturing the nuanced ways knowledge is presented in educational materials. For instance, the co-occurrence patterns we identified might be better analyzed using more advanced techniques like topic modeling or semantic network analysis [80], [81]. Additionally, the challenge of setting appropriate confidence thresholds for different knowledge types raises important methodological questions about how to balance precision and recall when analyzing epistemic features of educational materials. Future work should explore combining computational and qualitative methods to validate and refine these approaches, perhaps through expert review of model classifications or mixed-methods analysis of how identified knowledge types manifest in actual classroom practices.

Limitations

While we tried to be transparent and explain our reasoning throughout the paper, the current prediction model exhibits several limitations that warrant careful consideration when attempting to generalize its classification capabilities across CS educational materials. A fundamental constraint emerges from the training dataset's composition, which relies on a relatively modest sample size of 10,000 training examples. Although seemingly large, this limited corpus may inadequately capture the full spectrum of knowledge representations present in CS pedagogy. The dataset's construction methodology also presents epistemological challenges, as it lacks established inter-rater reliability measures and comprehensive thematic analysis protocols that would ensure exhaustive coverage of potential knowledge types. The absence of such validation mechanisms introduces potential systematic biases in knowledge classification. Furthermore, the model's architectural framework, while leveraging BERT's natural language understanding capabilities, encounters constraints in processing extended textbook passages and may oversimplify the inherently interconnected nature of CS knowledge types. The predefined taxonomic structure, though theoretically grounded, potentially fails to accommodate emergent or hybrid forms of knowledge representation if they occur in new texts outside those surveyed. Additional limitations stem from potential sampling biases in the source materials, which may not fully represent the diverse pedagogical approaches and subject matter depth across CS subfields. These methodological constraints suggest several promising directions for future research, including: expanding the training corpus through systematic sampling strategies, implementing robust inter-rater reliability protocols, developing more nuanced knowledge type representations, and incorporating domain adaptation techniques to enhance generalization capabilities. Such methodological refinements would strengthen the model's ability to serve as a reliable tool for automated knowledge type classification in CS education materials.

Conclusion

This study introduces a novel approach to analyzing the epistemic climate in engineering education through the automated analysis of curricular materials. By developing a taxonomy of knowledge types and leveraging NLP techniques, we demonstrate that different forms of knowledge presentation in CS textbooks can be reliably identified and analyzed at scale. Our results show high classification performance across multiple knowledge types, with F1-scores ranging from 0.968 for ethical knowledge to 0.683 for procedural knowledge, validating both our theoretical framework and methodological approach. This work makes several key contributions: (1) it presents and distinguishes between distinct knowledge types in CS education curricular materials, (2) it demonstrates a scalable method for analyzing epistemic climate through these curriculum materials, and (3) it describes how future research can explore the knowledge patterns presented and interrelated in CS education. These findings have important implications for textbook authors, instructors, and researchers working to understand and improve how engineering knowledge is conveyed to students. Future work should explore the application of these methods to other engineering disciplines, investigate temporal changes in knowledge presentation across different editions of textbooks, and examine how different patterns of knowledge presentation relate to student learning outcomes. Through the continued development of such analytical approaches, we can better understand and intentionally shape the epistemic climates we create in engineering education.

References

[1] K. R. Muis, G. Trevors, and M. Chevrier, "Epistemic climate for epistemic change," in *Handbook of Epistemic Cognition*, Routledge, 2016.

[2] G. A. Valverde, L. J. Bianchi, R. G. Wolfe, W. H. Schmidt, and R. T. Houang, *According to the Book*. Dordrecht: Springer Netherlands, 2002. doi: 10.1007/978-94-007-0844-0.

[3] M. Robinson, "Analysis of Engineering Textbook Epistemologies," presented at the 2023 ASEE Annual Conference & Exposition, Baltimore , Maryland, Jun. 2023. doi: 10.18260/1-2--42648.

[4] F. Abd-El-Khalick *et al.*, "A longitudinal analysis of the extent and manner of representations of nature of science in U.S. high school biology and physics textbooks," *Journal of Research in Science Teaching*, vol. 54, no. 1, pp. 82–120, 2017, doi: 10.1002/tea.21339.

[5] B. A. Herbel-Eisenmann, "From intended curriculum to written curriculum: Examining the voice of a mathematics textbook," *Journal for Research in Mathematics Education*, vol. 38, no. 4, pp. 344–369, 2007.

[6] G. J. Stylianides, "Reasoning-and-Proving in School Mathematics Textbooks," *Mathematical Thinking and Learning*, vol. 11, no. 4, pp. 258–288, Oct. 2009, doi: 10.1080/10986060903253954.

[7] J. Lithner, "Mathematical reasoning in calculus textbook exercises," *The Journal of Mathematical Behavior*, vol. 23, no. 4, pp. 405–427, Jan. 2004, doi: 10.1016/j.jmathb.2004.09.003.

[8] M. P. Marino, "High School World History Textbooks: An Analysis of Content Focus and Chronological Approaches," *The History Teacher*, vol. 44, no. 3, pp. 421–446, 2011.

[9] B. A. Danielak, A. Gupta, and A. Elby, "Marginalized Identities of Sense-Makers: Reframing Engineering Student Retention," *Journal of Engineering Education*, vol. 103, no. 1, pp. 8–44, 2014, doi: 10.1002/jee.20035.

[10] L. Okeeffe, "A Framework for Textbook Analysis," *Int. Rev. Cont. Lear. Res.*, vol. 2, no. 1, pp. 1–13, Jan. 2013, doi: 10.12785/irclr/020101.

[11] I. Villanueva *et al.*, "What Does Hidden Curriculum in Engineering Look Like and How Can It Be Explored?," in *2018 ASEE Annual Conference & Exposition Proceedings*, Salt Lake City, Utah: ASEE Conferences, Jun. 2018, p. 31234. doi: 10.18260/1-2--31234.

[12] K. Knorr-Cetina, *Epistemic cultures: how the sciences make knowledge*. Cambridge (Mass.): Harvard university press, 1999.

[13] C. Chinn and W. Sandoval, "Epistemic Cognition and Epistemic Development," in *International Handbook of the Learning Sciences*, Routledge, 2018.

[14] C. J. Faber, R. L. Kajfez, D. M. Lee, L. C. Benson, M. S. Kennedy, and E. G. Creamer, "A grounded theory model of the dynamics of undergraduate engineering students' researcher identity and epistemic thinking," *J Res Sci Teach*, vol. 59, no. 4, pp. 529–560, Apr. 2022, doi: 10.1002/tea.21736.

[15] L. S. Vygotsky, "Interaction between learning and development," in *Mind in society: Development of higher psychological processes*, Harvard University Press, 1978, pp. 79–91. Accessed: Mar. 03, 2025. [Online]. Available: http://www.jstor.org/stable/j.ctvjf9vz4.11

[16] J. A. Greene, Ed., *Handbook of epistemic cognition*. in Educational psychology handbook series. New York: Routledge, 2016.

[17] W. Grimson and M. Murphy, “The Epistemological Basis of Engineering, and Its Reflection in the Modern Engineering Curriculum,” in *Engineering Identities, Epistemologies and Values: Engineering Education and Practice in Context, Volume 2*, S. H. Christensen, C. Didier, A. Jamison, M. Meganck, C. Mitcham, and B. Newberry, Eds., Cham: Springer International Publishing, 2015, pp. 161–178. doi: 10.1007/978-3-319-16172-3_9.
[18] R. Felder and R. Brent, “The Intellectual Development of Science and Engineering Students. Part 1: Models and Challenges,” *Journal of Engineering Education*, vol. 93, Oct. 2004, doi: 10.1002/j.2168-9830.2004.tb00816.x.
[19] C. J. Faber and L. C. Benson, “Engineering Students’ Epistemic Cognition in the Context of Problem Solving,” *Journal of Engineering Education*, vol. 106, no. 4, pp. 677–709, 2017, doi: 10.1002/jee.20183.
[20] D. Montfort, S. Brown, and D. Shinew, “The Personal Epistemologies of Civil Engineering Faculty,” *Journal of Engineering Education*, vol. 103, no. 3, pp. 388–416, 2014, doi: 10.1002/jee.20050.
[21] N. Moore, Coldwell ,Mike, and E. and Perry, “Exploring the role of curriculum materials in teacher professional development,” *Professional Development in Education*, vol. 47, no. 2–3, pp. 331–347, May 2021, doi: 10.1080/19415257.2021.1879230.
[22] M. J. Prince and R. M. Felder, “Inductive Teaching and Learning Methods: Definitions, Comparisons, and Research Bases,” *Journal of Engineering Education*, vol. 95, no. 2, pp. 123–138, 2006, doi: 10.1002/j.2168-9830.2006.tb00884.x.
[23] M. Stains *et al.*, “Anatomy of STEM teaching in North American universities,” *Science*, vol. 359, no. 6383, pp. 1468–1470, Mar. 2018, doi: 10.1126/science.aap8892.
[24] C. E. Hmelo-Silver, R. G. Duncan, and C. A. Chinn, “Scaffolding and Achievement in Problem-Based and Inquiry Learning: A Response to Kirschner, Sweller, and Clark (2006),” *Educational Psychologist*, vol. 42, no. 2, pp. 99–107, Apr. 2007, doi: 10.1080/00461520701263368.
[25] R. M. Felder, D. R. Woods, J. E. Stice, and A. Rugarcia, “The Future of Engineering Education: Part 2. Teaching Methods that Work,” *Chemical Engineering Education*, vol. 34, no. 1, Art. no. 1, Jan. 2000.
[26] J. A. Leydens and J. C. Lucena, Eds., “Introduction,” in *Engineering Justice: Transforming Engineering Education and Practice*, John Wiley & Sons, Ltd, 2017, pp. 1–43. doi: 10.1002/9781118757369.ch0.
[27] D. Riley, *Engineering Thermodynamics and 21st Century Energy Problems: A textbook companion for student engagement*. in Synthesis Lectures on Engineering. Cham: Springer International Publishing, 2012. doi: 10.1007/978-3-031-79342-4.
[28] J. C. Lucena, *Defending the nation: U.S. policymaking to create scientists and engineers from Sputnik to the “war against terrorism.”* Lanham, Md.: Univ. Press of America, 2005.
[29] D. D. Massey, “Reading History: Moving from Memorizing Facts to Critical Thinking,” in *Improving Reading Comprehension of Middle and High School Students*, Springer, Cham, 2015, pp. 19–47. doi: 10.1007/978-3-319-14735-2_2.
[30] B. M. Trofanenko, “More Than a Single Best Narrative: Collective History and the Transformation of Historical Consciousness,” *Curriculum Inquiry*, vol. 38, no. 5, pp. 579–603, Dec. 2008, doi: 10.1111/j.1467-873X.2008.00437.x.
[31] P. T. Fair and M. R. Nilsson, “Differences in the Portrayal of Female and Male Scientists in College Chemistry Textbooks,” *J. Chem. Educ.*, Apr. 2025, doi: 10.1021/acs.jchemed.4c01365.

[32] M. L. Becker and M. R. Nilsson, “College Chemistry Textbooks Fail on Gender Representation,” *J. Chem. Educ.*, vol. 98, no. 4, pp. 1146–1151, Apr. 2021, doi: 10.1021/acs.jchemed.0c01037.

[33] L. W. Lee, Low ,Hui Min, and A. R. and Mohamed, “Word count analysis of Malay language textbooks for the purpose of developing a Malay reading remedial programme,” *Writing Systems Research*, vol. 4, no. 1, pp. 103–119, Apr. 2012, doi: 10.1080/17586801.2012.690713.

[34] K. R. Chowdhary, “Natural Language Processing,” in *Fundamentals of Artificial Intelligence*, K. R. Chowdhary, Ed., New Delhi: Springer India, 2020, pp. 603–649. doi: 10.1007/978-81-322-3972-7_19.

[35] C. G. P. Berdanier, E. Baker, W. Wang, and C. McComb, “Opportunities for Natural Language Processing in Qualitative Engineering Education Research: Two Examples,” in *2018 IEEE Frontiers in Education Conference (FIE)*, San Jose, CA, USA: IEEE, Oct. 2018, pp. 1–6. doi: 10.1109/FIE.2018.8658747.

[36] S. Bhaduri, “NLP IN ENGINEERING EDUCATION: Demonstrating the use of Natural Language Processing techniques for use in Engineering Education classrooms and research,” Virginia Tech, Blacksburg, VA, Dissertation, Dec. 2017. [Online]. Available: https://vtechworks.lib.vt.edu/server/api/core/bitstreams/d70f6e0b-7ee5-4bb5-a567-ee4221937acf/content

[37] A. Katz, U. Shakir, and B. Chambers, “The Utility of Large Language Models and Generative AI for Education Research,” May 29, 2023, *arXiv*: arXiv:2305.18125. Accessed: Jan. 26, 2024. [Online]. Available: http://arxiv.org/abs/2305.18125

[38] A. Katz, M. Gerhardt, and M. Soledad, “Using Generative Text Models to Create Qualitative Codebooks for Student Evaluations of Teaching,” *International Journal of Qualitative Methods*, vol. 23, p. 16094069241293283, Jan. 2024, doi: 10.1177/16094069241293283.

[39] “AI Foundation Models: Initial Report,” Competition & Markets Authority (CMA), UK, Sep. 2023. [Online]. Available: https://assets.publishing.service.gov.uk/media/65081d3aa41cc300145612c0/Full_report_.pdf

[40] A. Rogers, O. Kovaleva, and A. Rumshisky, “A Primer in BERTology: What we know about how BERT works,” Nov. 09, 2020, *arXiv*: arXiv:2002.12327. doi: 10.48550/arXiv.2002.12327.

[41] K. W. Church, Z. Chen, and Y. Ma, “Emerging trends: A gentle introduction to fine-tuning,” *Natural Language Engineering*, vol. 27, no. 6, pp. 763–778, 2021, doi: 10.1017/S1351324921000322.

[42] H. Adam *et al.*, “Write It Like You See It: Detectable Differences in Clinical Notes by Race Lead to Differential Model Recommendations,” in *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society*, in AIES ’22. New York, NY, USA: Association for Computing Machinery, Jul. 2022, pp. 7–21. doi: 10.1145/3514094.3534203.

[43] “Professional Engineering Body of Knowledge,” National Society of Professional Engineers, 2013. [Online]. Available: https://www.nspe.org/sites/default/files/resources/nspe-body-of-knowledge.pdf

[44] P. L. Berger and T. Luckmann, *The Social Construction of Reality*.

[45] J. A. Greene and S. B. Yu, “Educating Critical Thinkers: The Role of Epistemic Cognition,” *Policy Insights from the Behavioral and Brain Sciences*, vol. 3, no. 1, pp. 45–53, Mar. 2016, doi: 10.1177/2372732215622223.

[46] G. A. Myers, “Textbooks and the sociology of scientific knowledge,” *English for Specific Purposes*, vol. 11, no. 1, pp. 3–17, Jan. 1992, doi: 10.1016/0889-4906(92)90003-S.

[47] J. A. Hartigan, *Clustering algorithms*. in Wiley series in probability and mathematical statistics. New York: Wiley, 1975.

[48] Y. Gao and M. Riley, “Knowledge and Identity: A Review,” *International Journal of Management Reviews*, vol. 12, no. 3, pp. 317–334, 2010, doi: 10.1111/j.1468-2370.2009.00265.x.

[49] D. Michalaka and E. Giogli, “Which skills are more important in the engineering world? Perceptions of college students, recent graduates, and employers,” presented at the 2020 ASEE Southeast Section Conference, 2020. [Online]. Available: https://sites.asee.org/se/wp-content/uploads/sites/56/2021/01/2020ASEESE98.pdf

[50] J. P. Gee, *An introduction to discourse analysis: theory and method*, 3rd ed. Milton Park, Abingdon ; New York: Routledge, 2011.

[51] M. Borrego, E. P. Douglas, and C. T. Amelink, “Quantitative, Qualitative, and Mixed Research Methods in Engineering Education,” *Journal of Engineering Education*, vol. 98, no. 1, pp. 53–66, 2009, doi: 10.1002/j.2168-9830.2009.tb01005.x.

[52] A. Katz, S. Wei, G. Nanda, C. Brinton, and M. Ohland, “Exploring the Efficacy of ChatGPT in Analyzing Student Teamwork Feedback with an Existing Taxonomy,” May 09, 2023, *arXiv*: arXiv:2305.11882. Accessed: Feb. 03, 2024. [Online]. Available: http://arxiv.org/abs/2305.11882

[53] R. E. Bryant and D. R. O’Hallaron, *Computer systems: a programmer’s perspective*, Third edition. Boston Amsterdam London: Pearson, 2016.

[54] M. Döll, M. Döhring, and A. Müller, “Evaluating Gender Bias in Large Language Models,” Nov. 14, 2024, *arXiv*: arXiv:2411.09826. doi: 10.48550/arXiv.2411.09826.

[55] W. Wang *et al.*, “The Earth is Flat? Unveiling Factual Errors in Large Language Models,” arXiv.org. Accessed: Jan. 15, 2025. [Online]. Available: https://arxiv.org/abs/2401.00761v1

[56] J. Leinonen *et al.*, “Comparing Code Explanations Created by Students and Large Language Models,” arXiv.org. Accessed: Jan. 15, 2025. [Online]. Available: https://arxiv.org/abs/2304.03938v1

[57] J. Paladines and J. Ramirez, “A Systematic Literature Review of Intelligent Tutoring Systems With Dialogue in Natural Language,” *IEEE Access*, vol. 8, pp. 164246–164267, 2020, doi: 10.1109/ACCESS.2020.3021383.

[58] A. Gao, “Prompt Engineering for Large Language Models,” *SSRN Journal*, 2023, doi: 10.2139/ssrn.4504303.

[59] Y.-J. Lan and N.-S. Chen, “Teachers’ agency in the era of LLM and generative AI: Designing pedagogical AI agents,” *Educational Technology & Society*, vol. 27, no. 1, p. I–XVIII, 2024.

[60] “Teen and Young Adult Perspectives on Generative AI: Patterns of use, excitements, and concerns,” Harvard University Center for Digitial Thriving, Boston, MA, 2024. [Online]. Available: https://digitalthriving.gse.harvard.edu/wp-content/uploads/2024/06/Teen-and-Young-Adult-Perspectives-on-Generative-AI.pdf#page=3.10

[61] J. Jiao, S. Afroogh, K. Chen, D. Atkinson, and A. Dhurandhar, "The global landscape of academic guidelines for generative AI and LLMs," *Nat Hum Behav*, pp. 1–5, Mar. 2025, doi: 10.1038/s41562-025-02124-6.

[62] Qwen *et al.*, "Qwen2.5 Technical Report," Dec. 19, 2024, *arXiv*: arXiv:2412.15115. doi: 10.48550/arXiv.2412.15115.

[63] A. van Wynsberghe, "Sustainable AI: AI for sustainability and the sustainability of AI," *AI Ethics*, vol. 1, no. 3, pp. 213–218, Aug. 2021, doi: 10.1007/s43681-021-00043-6.

[64] C.-J. Wu *et al.*, "Sustainable ai: Environmental implications, challenges and opportunities," *Proceedings of Machine Learning and Systems*, vol. 4, pp. 795–813, 2022.

[65] J. Devlin, M.-W. Chang, K. Lee, and K. 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 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds., Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. doi: 10.18653/v1/N19-1423.

[66] Y. You *et al.*, "Large Batch Optimization for Deep Learning: Training BERT in 76 minutes," Jan. 03, 2020, *arXiv*: arXiv:1904.00962. doi: 10.48550/arXiv.1904.00962.

[67] B. K. Hofer and P. R. Pintrich, "The Development of Epistemological Theories: Beliefs About Knowledge and Knowing and Their Relation to Learning," *Review of Educational Research*, vol. 67, no. 1, pp. 88–140, Mar. 1997, doi: 10.3102/00346543067001088.

[68] L. C. Benson *et al.*, "Understanding undergraduate engineering researchers and how they learn," in *2016 IEEE Frontiers in Education Conference (FIE)*, Oct. 2016, pp. 1–5. doi: 10.1109/FIE.2016.7757630.

[69] N. Frye, D. Montfort, S. Brown, and O. Adesope, "I'm absolutely certain that's probably true: Exploring epistemologies of sophomore engineering students," in *2012 Frontiers in Education Conference Proceedings*, Oct. 2012, pp. 1–6. doi: 10.1109/FIE.2012.6462356.

[70] J. C. Wise, S. H. Lee, T. Litzinger, R. M. Marra, and B. Palmer, "A Report on a Four-Year Longitudinal Study of Intellectual Development of Engineering Undergraduates," *Journal of Adult Development*, vol. 11, no. 2, pp. 103–110, Apr. 2004, doi: 10.1023/B:JADE.0000024543.83578.59.

[71] M. Daniels, Å. Cajander, R. McDermott, A. Vasilcheko, and D. Golay, "Why Don't You Tell Me What I Need to Know? Self-Flipped Classroom and Students' Personal Epistemology," in *2021 IEEE Frontiers in Education Conference (FIE)*, Oct. 2021, pp. 1–9. doi: 10.1109/FIE49875.2021.9637355.

[72] J. A. Greene, W. A. Sandoval, and I. Bråten, Eds., *Handbook of epistemic cognition*, First published. in Educational psychology handbook series. New York: Routledge, 2016.

[73] K. Beddoes, D. Montfort, and S. Brown, "Squaring Philosophy of Engineering Through Personal Epistemologies Research," in *Philosophy and Engineering: Exploring Boundaries, Expanding Connections*, D. P. Michelfelder, B. Newberry, and Q. Zhu, Eds., Cham: Springer International Publishing, 2017, pp. 23–41. doi: 10.1007/978-3-319-45193-0_3.

[74] E. A. Cech, "Ideological Wage Inequalities? The Technical/Social Dualism and the Gender Wage Gap in Engineering," *Social Forces*, vol. 91, no. 4, pp. 1147–1182, Jun. 2013, doi: 10.1093/sf/sot024.

[75] J. L. Hess and G. Fore, "A Systematic Literature Review of US Engineering Ethics Interventions," *Sci Eng Ethics*, Apr. 2017, doi: 10.1007/s11948-017-9910-6.

[76] K. R. Muis, L. D. Bendixen, and F. C. Haerle, "Domain-Generality and Domain-Specificity in Personal Epistemology Research: Philosophical and Empirical Reflections in the Development of a Theoretical Framework," *Educ Psychol Rev*, vol. 18, no. 1, pp. 3–54, Mar. 2006, doi: 10.1007/s10648-006-9003-6.
[77] I. Bråten, "Epistemic cognition interventions," in *Handbook of Epistemic Cognition*, 1st ed., New York: Taylor and Francis Inc., 2016, pp. 360–371. doi: 10.4324/9781315795225.
[78] K. R. Muis and M. C. Duffy, "Epistemic climate and epistemic change: Instruction designed to change students' beliefs and learning strategies and improve achievement," *Journal of Educational Psychology*, vol. 105, no. 1, pp. 213–225, Feb. 2013, doi: 10.1037/a0029690.
[79] R. M. Felder and R. Brent, "The Intellectual Development of Science and Engineering Students. Part 2: Teaching to Promote Growth," *Journal of Engineering Education*, vol. 93, no. 4, pp. 279–291, 2004, doi: 10.1002/j.2168-9830.2004.tb00817.x.
[80] M. Grootendorst, "BERTopic: Neural topic modeling with a class-based TF-IDF procedure," *arXiv preprint arXiv:2203.05794*, 2022.
[81] E. Segev, *Semantic Network Analysis in Social Sciences*, 1st ed. London: Routledge, 2021. doi: 10.4324/9781003120100.