Computing Education Research Landscape through an Analysis of Keywords

Zacharoula Papamitsiou
Norwegian University of Science and Technology
Norway
papamitsiou.zacharoula@ntnu.no

Michail Giannakos
Norwegian University of Science and Technology
Norway
michailg@ntnu.no

Simon
University of Newcastle
Australia
simon@newcastle.edu.au

Andrew Luxton-Reilly
University of Auckland
New Zealand
a.luxton-reilly@auckland.ac.nz

ABSTRACT

Authors of academic papers are generally required to nominate several keywords that characterize the paper, but are rarely offered guidance on how to select those keywords. We analyzed the keywords in the past 15 years of selected computing education publications: the 1274 papers published in the proceedings of ICER and ITiCSE, including the ITiCSE working group reports. As well as the keywords assigned by the authors, we mined the abstracts of these papers to extract a separate list of keywords. Our work has two goals: to frame the thematic landscape of the field, using keywords that communicate the work conducted; and to detect differences between the human judgement and interpretation of keywords and the machine 'intelligence' on handling those keywords, with respect to the clusters of thematic topics identified in each case. The analysis shows that the field is dominated by learning approaches (e.g., active learning, collaborative learning), aspects of programming (e.g., debugging, misconceptions), computational thinking, feedback, and assessment, while other areas that have attracted attention include academic integrity (e.g., plagiarism) and diversity (e.g., female students, underrepresented groups). It was observed that the keywords chosen by authors are often too general to provide information about the paper (e.g., 'concerns', 'course', 'fun', 'justice'). We elaborate on the findings and begin a discussion on how authors can improve the communication of their research and make access to it more transparent.

CCS CONCEPTS

• **Social and professional topics → Computing education**.

KEYWORDS

Computing education, keywords, dominant themes, bibliometrics

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

ACM Reference Format:
Zacharoula Papamitsiou, Michail Giannakos, Simon, and Andrew Luxton-Reilly. 2021. Computing Education Research Landscape through an Analysis of Keywords. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Over the past two decades, computing education research (CER) has striven to increase its rigor and validity [3, 33] and to mature into a respectable field of research [40]. A number of recent reviews have set out to capture and present the state of the art in various areas of CER, such as introductory programming [21], K-12 [10, 46], and teaching assistants [27], and exploring such aspects as measurements [23], replications [13], empiricism [2], and inferential statistics [35].

CER as a field of study is growing in popularity, as evidenced by the numbers of submissions to ITiCSE and the SIGCSE Technical Symposium (TS), both of which have more than doubled over the past 20 years [1], and the number of publications concerning introductory programming, which has tripled over the period from 2003 to 2017 [21]. To help capture the big picture of CER, the present work identifies the thematic areas of interest of the CER community, using hierarchical clustering, strategic diagrams, and graph theory.

To achieve this objective, the paper employs a method called 'co-word analysis' combined with social network analysis, with a focus on the core-periphery structure (i.e., the frequency, centrality, and interconnectivity of themes). This scientometric method examines the associations and networks among concepts, ideas, and issues that have contributed to the maturation of the field to date [6]. Co-word analysis relies on the assumption that an article's keywords or phrases adequately summarize its content, and can therefore be used to represent the article [7]. Co-word analysis also assumes that co-occurrence of keywords within a paper indicates a linkage between the topics represented by those keywords. Therefore, co-word analysis can help researchers to identify patterns that point to changes in a research topic (such as emerging or declining research interests) or changes in research direction (such as paradigm changes), based on the graph of keywords [7].

The current study aims to map the intellectual progress of the CER landscape, as reflected in the proceedings of ITiCSE (Innovation and Technology in Computer Science Education) and ICER (International Computing Education Research Conference), which provide a solid foundation for the related work published since the first offering of ICER in 2005 — the so-called 'modern era' of computing education research [12]. Considerable work has been published during this period, allowing us to observe where the field currently stands, what challenges and opportunities researchers are facing, and what the potential driving forces will be in the near future. This work contributes by documenting the intellectual progress of the scientific area of computing education; by providing evidence-based insights of the community's research themes; and by highlighting individual topics as popular, core, or backbone research topics within the discipline. Furthermore, a classification schema of the author-assigned keywords was developed, relying exclusively on the judgement and interpretation of human experts in the field. This allows us to synergistically combine the fully automated approach and the human understanding to provide a more holistic understanding of the CER landscape.

Our research questions are thus:

- RQ1 What computing education topics emerge from an analysis of key phrases extracted automatically from abstracts?
- RQ2 What computing education topics emerge from human classification of author-assigned keywords?
- RQ3 What do the lists of topics from RQ1 and RQ2 have in common, and how do they differ?

2 BACKGROUND AND RELATED WORK

Various standardized methods and processes, both qualitative and quantitative, have been applied to study scientific communities. Different review and bibliometric measures, such as inclusion index, centrality, and density, have been developed to quantify and evaluate the impact of scientific communities [14]. This section summarizes previous reviews and bibliometric efforts to investigate the CER landscape and gives a brief overview of co-word analysis.

There have been a number of reviews of the computing education literature, some focusing on particular aspects of computing education while others aim to develop frameworks, taxonomies, or definitions. In a review of articles published from 2000 to 2005, Randolph et al. [32] focused on the methodological approaches reported. They collected 352 papers from a large number of venues – SIGCSE Bulletin, Computer Science Education (CSEd), SIGCSE TS, ITiCSE, ICER, Koli Calling, and Australasian Computing Education Conference (ACE) — and concluded that a relatively high proportion (40%) of studies with human subjects used solely anecdotal evidence. Kinnunen et al. [17] created a theoretical categorization of didactic-oriented CER. Examining papers published at ICER between 2005 and 2009, they found that the most common educational topics were those relating to students' actions and understanding of learning outcomes and to pedagogical activities used in the classroom.

Other review works have focused on the use of supporting theories in CER, clustering them into three main theory communities focused on social theories, experiential theories, and theories of mind [45]. In a review of computing education publications from 1976 to 2000, Holmboe et al. [15] noted the limited references to pedagogical theory and the fact that the majority of papers provided reflections from computer scientists on their teaching. More than a decade later, Malmi et al. [22] conducted a literature review on the theoretical underpinnings of CER, covering works published from 2005 to 2011 in ICER, Transactions on Computing Education (TOCE), and CSEd. They found that as many as 60% of publications refer to external theories, most of which are drawn from education, psychology, or other relevant disciplines; only 16% of the theories used were developed in CER.

Simon [37] devised a classification scheme for computing education papers and applied it to all of the computing education papers published in three years of ACE and New Zealand's conference of the National Advisory Committee for Computing Qualifications (NACCQ). With various co-authors he applied the scheme to all of the papers over eight years of NACCQ [44], three years of ICER [42], and a number of other bodies of work [38–41, 43]. The classification included an assessment of what each paper is about, but by examining the paper itself, rather than its keywords or abstract.

Miró Julià et al. [26] analysed the author networks of a number of computing research conferences, a number of computing education conferences, and ICER, finding that on various measures ICER lies between the group of computing research conferences and the group of CER conferences. McCartney and Sanders [24] extended the analysis to include ITiCSE's working groups, which Miró Julià et al. [26] had not considered because they had no counterpart in the other conferences that they studied. McCartney and Sanders [24] found that the author networks of ITiCSE working groups are more like those of ICER than of the standard ITiCSE papers.

All of these bibliometric analyses rely on human examination of the papers. In our work, we are interested in classifying publications based on automatic and semi-automatic analysis of their keywords, a process that entails no examination of the papers themselves.

Our goal is to characterize the CER landscape through the lens of two of the main research conferences, ICER and ITiCSE, and their entire proceedings for 15 years. We have not included the SIGCSE TS because the volume of its papers would dominate the two we have chosen. If these two are reasonably representative, analyzing them will give us some insight into the thematic areas and landscape of the CER community, and its intellectual progress.

Co-word analysis is a content analysis technique that maps the strength of relationships between terms in texts and traces patterns and trends in term association [6]. In particular, the extracted keywords are seen as the basic building blocks of the structure of a research field and their dynamics are represented as an interaction between keywords: keywords that co-occur within a paper show a link between the research topics they represent. Co-word analysis can thus reveal patterns and trends in CER based on the co-occurrence patterns of pairs of words. The same approach has been used to discover connections and interactions among research themes in various areas such as the UbiComp community [19], the CHI community [20] and the learning analytics community [31].

3 METHOD

3.1 Data Collection

The data analyzed in this study, all 1274 peer-reviewed full and short papers published in ICER, ITiCSE, and ITiCSE working group reports between 2005 and 2019, were extracted from the ACM Digital Library. The author-assigned keywords were extracted from the metadata of each paper and were used as a unit of analysis. However, the granularity of the keywords may not be consistent since they are subjectively selected by the human authors. For example, the authors might describe their work in fairly generic terms to enhance its visibility, to categorize and link their work to a broader research domain, or to synopsize the sub-topics, replacing specific terms such as ‘Java’ and ‘Python’ with more generic ones such as ‘programming languages’. Therefore, the abstracts of the papers were also text-mined in order to automatically extract from them key-phrases that can describe their contents, based on the understanding that the abstract can be seen as a ‘standalone’ summary of the paper, a coherent synopsis of the paper. The 1274 papers, containing 5601 author-assigned keywords (M=4.40 per article) and 6308 machine-extracted key-phrases (M=4.95 per article), are distributed by year of publication as shown in Figure 1.

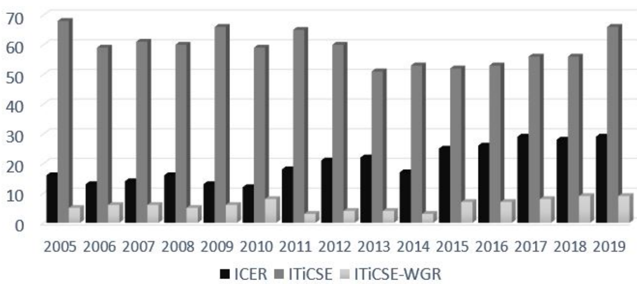

Figure 1: Number of publications (ICER, ITiCSE, and ITiCSE$_{\mathbf{WGR}}$) per year, 2005–2019

3.2 Data Preprocessing

The author-assigned keywords were manually preprocessed and standardized by merging singular and plural forms of nouns and words that convey similar meaning (e.g., ‘computing education’ and ‘IT education’, ‘TDD’ and ‘test-driven development’), fixing spelling errors (e.g., ‘internet of ings’), and combining UK and US terms (e.g., ‘behaviour’ and ‘behavior’), following previously recommended approaches [16, 20, 31]. At the end of this phase, 2821 (50% of the original author-assigned keywords) were identified as *unique* keywords, and were retained for further analysis.

To extract the key-phrases from abstracts we used a Python implementation of the TextRank algorithm for text summarization [25]. TextRank is fully unsupervised: no training is necessary, and instead of n-grams, it can tokenize words and phrases and annotate the tokens with parts of speech (PoS). In this study, the TextRank sliding window was set to 4, for the PoS we included nouns (NOUN), adjectives (ADJ) and proper nouns (PROPN), and we requested the top 15 phrases. After manually removing phrases that carry little semantic significance (e.g., ‘general goal’, ‘first iteration’, ‘contribution’), we were left with 6308 key-phrases, and we repeated the same preprocessing as for the author-assigned keywords, ending up with 4127 (65%) key-phrases identified as *unique*.

3.3 Co-word Analysis and Strategic Diagram

This study employs co-word analysis to summarize the big picture of computing education research, mapping the strength of relationships between terms in texts and tracing patterns and trends in term associatedness [6]. Co-word analysis relies on the assumptions that key-terms identified within an article (either as author-assigned keywords or as machine-extracted key-phrases) can adequately describe and communicate the content of that article, and that the co-occurrence of two or more key-terms in the same article indicates a linkage between those topics, known as a *theme* [5]. The main units of analysis are *key-terms*, *clusters* (sets of closely related key-terms), and *key-term networks* [20].

Co-word analysis is applied to reduce the broad network of key-terms into a smaller network of related topics using graph theory [8]. Graphs consist of nodes that represent the key-terms and links that represent the interactions between the nodes. Given a network of key-terms, a combination of clustering, network analysis, and strategic diagrams is used to model the conceptual structure of a field [5]. The graph theory concepts employed to map the research field are *centrality*, the strength of the links from one research theme or cluster to others, indicating its significance in the development of the community [20]; and *density*, the coherence of a cluster and a measure of a theme’s development [14]. A two-dimensional *strategic diagram* [5] combines centrality on the horizontal axis and density on the vertical axis: the position of a cluster in the diagram shows the importance of the cluster in the whole network (its centrality) in relation to how well its theme is developed (its density), as shown in Figure 2.

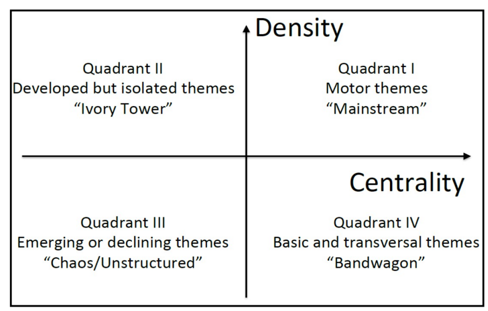

Figure 2: Strategic diagram of density and centrality [20]

In the strategic diagram, *Quadrant I (Q1)* holds the motor themes (i.e., mainstream themes) that have strong centrality and high density. *Quadrant II (Q2)* contains themes that are more specialized and peripheral to the mainstream work, and are internally well-structured but have weak external ties. *Quadrant III (Q3)* includes the themes with low density and low centrality, which are either emerging or declining. Finally, *Quadrant IV (Q4)* covers basic and transversal themes, central to the community or borrowed from other disciplines, that have the potential to become significant.

3.4 Data Analysis – Machine-Extracted Key-phrases

To identify the major research themes in the computing education domain, we performed hierarchical clustering analysis on a correlation matrix with the retained terms, using Ward’s method with squared Euclidean distance as the distance measurement [28]. This supervised clustering method allows the maintenance of content validity and cluster fitness for the greatest number of clusters [20, 31] or research themes. We further analyzed the co-word network using the following measures [31]:

- **Key-terms:** set of terms that constitute a cluster.
- **Size:** number of key-terms in the cluster.
- **Frequency:** how many times all key-terms in a cluster appear in the full data set.

- **Co-word frequency:** how many times at least two key-terms from a cluster appear in the same paper. This frequency leads to a symmetrical co-occurrence matrix [18], in which values on the diagonal are term frequencies and values off the diagonal are co-word frequencies. High co-occurrence between terms indicates a connection between the topics they represent.
- **Transitivity:** a number in the range [0, 1] representing the tightness of a cluster's connection (its *clustering coefficient*). Transitivity is the frequency in the cluster of loops of length three: sequences of nodes x, y, z such that (x, y), (y, z) and (z, x) are edges of the graph [36].
- **Centrality:** the number of other clusters that a cluster connects to [5]. Centrality comprises a group of metrics that aim to quantify the 'importance' of a particular node or cluster within a network; examples are betweenness centrality, closeness centrality, eigenvector centrality, and degree centrality) [29]. Here we used betweenness centrality (C), with $0 \leq C \leq 1$.
- **Density:** the cohesiveness of the cluster of terms, the number of direct ties observed for the cluster divided by the maximum number of possible ties [5]. Density is graph-dependent, and can be any positive real number [9].

Based on the clustering results, we plotted the strategic diagram for the years 2005–2019 to visualise the cohesion and maturity of the research themes in computing education [5, 20].

3.5 Data Analysis – Human-Rationalized Keywords

For the author-assigned keywords we applied a distinctly different approach. Two researchers with many years of experience in the computing education literature manually grouped the keywords into related themes. This entailed merging different keywords that represent the same concept, as in the automated preprocessing described in Section 3.2. But beyond that, it entailed the grouping of distinct but related terms such as 'academic integrity', 'plagiarism', 'collusion', 'cheating', 'program similarity', and 'attribution'. Grouping of this sort is indisputably subjective, and was in this case carried out by consensus. This sets the semantically-based grouping of keywords in sharp contrast to the automated clustering of the key-phrases that were extracted automatically from the abstracts of the papers, and thus provides an interesting basis for comparison of the two sets of terms.

3.6 Data Analysis – Network Graphs

From both the human-generated and machine-generated lists we generated key-term network graphs. In these graphs each key-term is represented as a node, and the key-terms that co-appear in a paper are joined by a line. The associations between key-terms lead to the creation of multiple networks associated with different themes. In this case, bridges are built between the nodes of key-terms, to allow communication and information flow between isolated regions of the network. Those nodes are known as *structural holes* [30]. Key-terms acting as structural holes also serve as the 'backbone' of a network: if they are removed, the network loses its cohesion and disintegrates into separate and disconnected concepts. Thus the network's core-periphery structure needs to be computed in order to determine whether nodes are part of a densely connected core (one with a higher number of bridges) or a sparsely connected periphery [34]. Core nodes are reasonably well connected to peripheral nodes, while peripheral nodes are sparingly connected to a core node or to one another. Hence, a node belongs to a core only if it is well connected to other core nodes and to peripheral nodes [34]. A follow-up core-periphery analysis was performed to identify the core research topics from the perspective of the whole network. In this analysis, key-terms were categorized according to their popularity, coreness (connectedness with other topics), and constraint (backbone topics). The whole approach is illustrated in Figure 3. in which the nodes marked 'SNA' refer to the production and analysis of the network graphs.

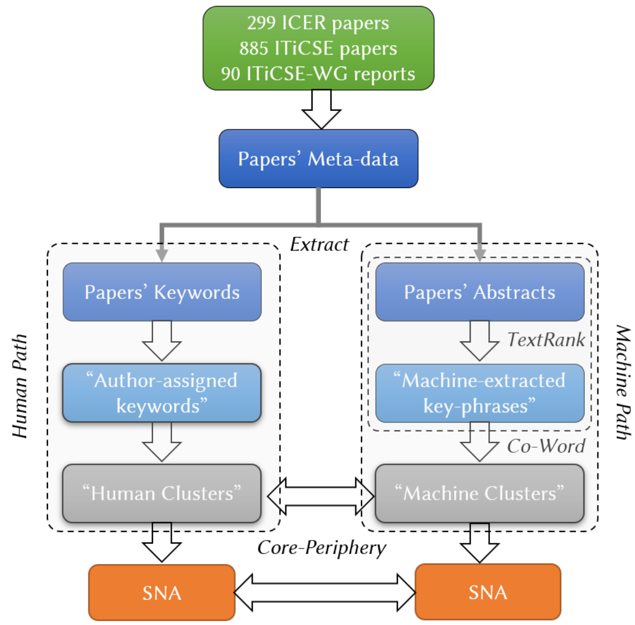

Figure 3: Research method; 'SNA' refers to semantic network analysis, the production and analysis of network graphs

4 RESULTS

To investigate how each year's publications contribute to CER development through the various research topics, we performed a correspondence analysis (CA) between the publication years and the identified keywords. CA employs a homogeneity analysis of an indicator matrix to obtain a low-dimensional Euclidean representation of the original data [11]. CA uses the frequencies formed by categorical data (i.e, a contingency table) and provides factor scores (coordinates) for both the rows and the columns of the indicator matrix (i.e., the contingency table). These coordinates are used to graphically visualize the association between the row and column variables in the contingency table in a two-dimensional space, based on the chi-squared statistic associated with the contingency table. In the two-dimensional outcome chart, all rows of the contingency table (i.e., a set of variables in the original data set) and all columns of the contingency table (i.e., a different set of variables in the original data set) can be displayed on the same axes. All data should be on the same scale for CA to be applicable, keeping in mind that the method treats rows and columns equivalently. The results of the CA for CER for the years 2005–2019 are illustrated in Figure 4. The CA factor map positions the most common keywords and years on a common set of orthogonal axes. The percentages depicted on the axes correspond to the proportions of the variance in the data that can be explained by the visualization. In this study, the visualization displays 27.5% of the variance in the data.

Based on the results of the CA depicted in Figure 4, different years have contributed to the development of different topics. We see that the publications of the first five years of our analysis (2005–2009) are positioned in the lower right quadrant of the CA. These first five years of our analysis contributed heavily to knowledge on the topics of visualization, introductory course curriculum, software engineering, and dropout. The triangle of the years 2010, 2013, and 2014 contributed to games, game development, CS2, and motivation, positioned in the upper right quadrant. In the upper left quadrant is another triangle, 2012, 2015, and 2018, covering subjects such as computational thinking, open source, multiple-choice questions, and novice programmers. In the lower left quadrant, recent publications (2017 and 2019) address the topical areas of teaching assistants, Python, assignments, and misconceptions. The years 2011 and 2016, close to the centre of the map, cover CS majors, programming concepts, and exams.

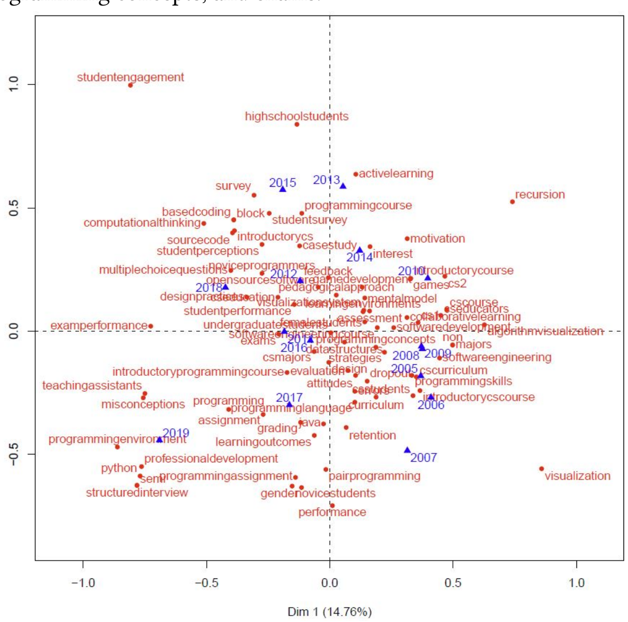

Figure 4: Correspondence analysis map for CER, 2005—2019

4.1 Major Research Themes, 2005–2019, Based on Machine-Extracted Key-Phrases

Of the 4127 unique machine-extracted key-phrases (Section 3.2), there are 94 that occur in six or more papers, together covering 85% of the papers. Clustering analysis on these 94 keywords leads to 14 clusters (labeled as C01-C14 in Table 1), each representing a research theme or a subfield. In order to better understand the relative 'positions' of these clusters within the overall CER field (their distance from one another in terms of cohesion and the maturity of the research themes they correspond to), and in order to create the conceptual structure of the CER discipline, we constructed a strategic diagram using the centrality and density of each cluster [6]. In this plot (Figure 5), the axes are centered to the average centrality (0.59) and density (1.36). The overall network density, representing the cohesiveness of the whole research field, was found to be 0.057. To understand the results, the reader needs to consider Figure 5 and Table 1 together.

Figure 5 shows that the CER field, as portrayed by ICER and ITiCSE proceedings, has two mainstream research (Q1) themes, represented by clusters C02 (e.g., introductory programming courses, exam performance, CS1) and C11 (e.g., CS2, game development, games), with C09 (e.g., design, evaluation, computational thinking, professional development) being very close to Q1. There are also some developed but isolated research themes (Q2), which are internally well-structured, but have rather weak external ties; these are represented by the clusters C4 (e.g., algorithm visualization, visualization system, block-based programming), C6 (e.g., software development, software engineering), and C8 (e.g., dropout, student satisfaction). The third quadrant (Q3) includes several themes that are either emerging or disappearing: C1 (e.g., programming languages, OOP, data structures, ITS), C10 (e.g., non-majors, CS majors, introductory programming, CS curriculum), and C13 (e.g., cognitive skills, cognitive load, instructional material). The final quadrant (Q4) includes a relatively high number of basic and transversal themes, themes that are strongly linked to specific research interests throughout the network, yet are only weakly linked together: C3 (e.g., errors, pair programming, compiler errors, syntax errors), C5 (e.g., performance, assessment, exams, assignment, feedback), C7 (e.g., survey, student perceptions), C9 (e.g., design, evaluation, computational thinking, professional development), C12 (e.g., CS education, CS students), and C14 (e.g., novice students, high school students). The detailed results are listed in Table 1.

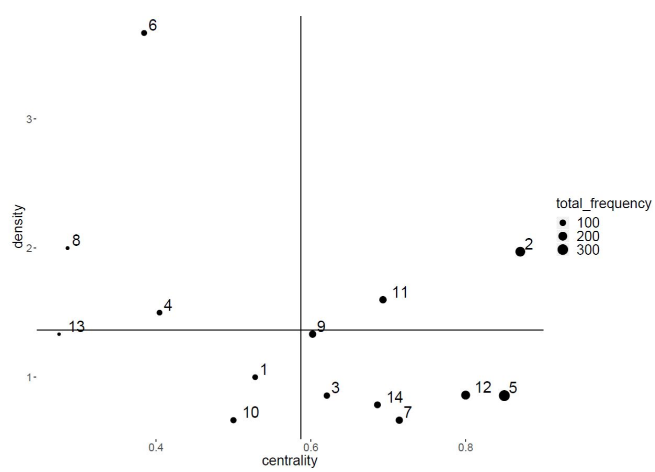

Figure 5: Strategic diagram for CER, 2005–2019, based on machine-extracted key-phrases; numbers correspond to cluster IDs in Table 1

4.2 Major Research Themes, 2005–2019, Based on Author-Assigned Keywords

The author-assigned keywords were grouped into 23 different clusters by two experienced computing education researchers. For each cluster, we calculated its size (how many unique keywords belong to the cluster), its frequency (how many times the keywords in

Table 1: Clusters of topics in CER, 2005–2019, machine-extracted keywords, including their quadrant on the strategic diagram (Figure 5)

Q	ID	Key-terms (the most frequent in bold)	Size	Freq†	CW-Fr†	T†	C†	D†
Q1	C2	**introductory programming course**, programming concepts, student performance, exam performance, cs1, misconceptions, multiple choice questions, student understanding, python	9	251	437	0.72	0.87	1.97
Q1	C11	**games**, motivation, introductory cs course, cs2, interest, game development	6	132	184	0.73	0.69	1.60
Q1-Q4	C9	**computational thinking**, design, evaluation, cs concepts, professional development, k12	6	129	163	0.56	0.60	1.33
Q2	C4	**algorithm visualization**, visualization, visualization system, block-based coding, scratch	5	73	95	0.87	0.40	1.50
Q2	C6	**software development**, software engineering, software engineering course	3	80	89	1.00	0.38	3.67
Q2	C8	**drop-out**, student satisfaction, information technology	3	28	47	1.00	0.29	2.00
Q3	C1	**programming language**, object-oriented programming, data structures, intelligent tutoring system, learning styles	5	74	108	0.87	0.53	1.00
Q3	C10	**cs curriculum**, non-majors, student success, gender, cs majors, introductory programming	6	88	126	0.25	0.50	0.67
Q3	C13	**cognitive skills**, cognitive load, instructional material	3	21	44	1.00	0.27	1.33
Q4	C3	**novice programmers**, errors, empirical study, pair programming, compiler errors, syntax errors, collaborative learning	7	95	146	0.64	0.62	0.86
Q4	C5	**assessment**, performance, programming, introductory course, programming course, exams, assignment, feedback, java, programming assignment, summative assessment, peer review, source code, grading	14	322	424	0.59	0.85	0.86
Q4	C7	**survey**, undergraduate students, learning environments, semi-structured interview, solo taxonomy, teaching assistants, design practices, student engagement, student perceptions, student survey	10	130	191	0.59	0.71	0.67
Q4	C12	**cs education**, active learning, cs students, programming skills, attitudes, female students, retention, pedagogical approach, cs course	9	207	265	0.64	0.80	0.86
Q4	C14	**novice students**, programming environment, mental model, case study, introductory cs, learning outcomes, recursion, high school students	8	112	185	0.48	0.69	0.79

† **Freq:** Total frequency of all key-terms in cluster; **CW-Fr:** Co-word Frequency; **T:** Transitivity; **C:** Centrality; **D:** Density

the cluster appeared in the papers), and its strength (the ratio of frequency to size). The strongest cluster (CH6) comprises generic terms such as computing education, computer science education, and education. Another cluster, almost equally strong but with only three keywords (CH11), comprises the terms evaluation, performance, and course performance. A cluster that is both very strong and large is introductory programming (CH12), which includes terms such as CS1, novice programmer, and introductory programming course. There are several moderately strong clusters (strength ≥ 3), such as computational thinking (CH4), computing (CH5), educational data (C10), pedagogy (CH14), and programming (CH15). The remaining clusters are relatively weak, although some of them are quite large (e.g., general terms, specific courses or topics, and learning approaches). Table 2 lists the clusters arising from the analysis of the author-assigned and expert-ranked keywords.

4.3 Keyword Network Map

In order to better understand the CER research themes presented in Tables 1 and 2, we visualized their relationship through network analysis and the development of two granular network maps of the keywords. Figures 6 and 7 display the networks of machine-extracted and author-assigned keywords respectively. Each node in the graphs represents a keyword that is linked to other keywords that appear in the same paper. The size of the nodes is proportional to the frequency of the keywords, the color of the node corresponds to the cluster the keyword has been classified in, and the thickness of the links between nodes is proportional to the co-occurrence correlation for that pair of keywords. From this analysis, keywords that appeared less than six times in the initial data set were excluded (as previously explained), and keywords with fewer than six strong ties were excluded to avoid a highly disconnected network.

Our last analysis was to identify the core research topics in the field from a whole-network perspective, as individual keywords, regardless of the cluster they belong to (this is known as core-periphery analysis). We performed this analysis separately for the machine-extracted keywords (Table 3) and the author-assigned keywords (Table 4). The core-periphery analysis yielded ten core research topics in each of the following categories:

- **Popularity:** how frequently a keyword is used;
- **Core:** how connected a keyword is with other topics; coreness is measured on a [0−1] scale;
- **Structural holes (constraint):** how connected a research keyword is with otherwise distinct topics (i.e., if the topic creates a backbone of the field); constraint is measured on a [0−1] scale.

A higher core value indicates a topic that is well connected to other topics. Higher structural holes indicate keywords that brings together otherwise isolated topics. Burt's constraint [4] is commonly used as a measure of structural holes: the larger the constraint value, the fewer structural opportunities a node may have for bridging structural holes, and so keywords that act as bridges between topics have lower constraint values. Topics with high scores on popularity

Table 2: Clusters of topics in computing education, 2005–2019, human processing of author-assigned keywords

ID	Cluster Name (alpha order)	Popular Keywords (ordered based on their frequency)‡	Size†	Freq†	Str†
CH1	academic integrity	**academic integrity, plagiarism**, cheating, collusion, program similarity, ethical hacking, copying	19	40	2.11
CH2	aspects of programming	**debugging**, recursion, misconceptions, testing, simulation, mental models, software testing, test-driven development (tdd), polymorphism	244	480	1.97
CH3	assessment	**assessment**, feedback, automated grading, peer assessment, SOLO taxonomy, exam, programming assignments, multiple choice questions	156	327	2.09
CH4	computational thinking	**computational thinking**, problem solving, abstraction, alg. thinking	26	105	4.04
CH5	computing discplines	**software engineering, computer science**, introductory computer science, CS, informatics, Information technology	15	91	6.07
CH6	computing education	**computer science education**, education, computing education, software engineering education, CS education, informatics education	29	266	9.17
CH7	course management	**Moodle**, course management, mobile devices, assistive technology, learning environments, classroom management, content management	54	82	1.52
CH8	curriculum	**curriculum**, curriculum design, course design, curriculum issues, instructional design, learning outcomes, curricula, learning objectives	54	117	2.17
CH9	diversity	**gender**, diversity, women in computing, broadening participation, gender issues, girls, women, minorities, under-represented group, disability	73	138	1.89
CH10	educational data	**educational data mining**, learning analytics, data mining, big data	11	38	3.45
CH11	evaluation	**evaluation**, course evaluation, course performance	3	26	8.67
CH12	introductory programming	**CS1**, introductory programming, learning to program, introductory cs, novice programmer, introductory programming course	31	206	6.65
CH13	learning approaches	**active learning**, collaborative learning, e-learning, pair programming, games, constructivism, peer instruction, cognitive load theory	299	622	2.08
CH14	pedagogy	**pedagogy**, computer science pedagogy, course pedagogy, educational model, pedagogical approach	16	66	4.13
CH15	programming	**programming**, novice programmers, programming education, object-oriented programming, block-based programming, coding	53	256	4.83
CH16	programming languages and environments	**Java**, python, scratch, alice, BlueJ, Jeliot, interactive learning environments, visual programming language, app inventor	64	132	2.06
CH17	research and approaches	**CS ed research**, experimental evaluation, phenomenography, qualitative research, empirical research, grounded theory	150	279	1.86
CH18	school	**K-12**, high school, middle school, outreach, secondary education, children, high school curriculum, elementary school	54	129	2.39
CH19	specific courses or topics	**CS2**, data structures, algorithms, research methods, artificial intelligence, CS1/2, operating systems, game development, security	301	607	2.02
CH20	students	**motivation**, retention, self-efficacy, non-majors, accessibility, attitudes, engagement, creativity, students, cognition	224	408	1.82
CH21	teachers	**pedagogical content knowledge**, professional development, teaching assistants, teachers, programming knowledge, teacher training	53	86	1.62
CH22	visualisation	**visualization**, algorithm visualization, program visualization, graphics/visualization, software visualization, visual representations	26	76	2.92
CH23	ungroupable (general) terms	**experience report**, higher education, tools, educational technology, interaction, data, practitioner	862	1024	1.19

† **Size:** How many unique keywords belong to the cluster; **Freq:** How many times the keywords that belong to the cluster are found; **Str:** The ratio of frequency to size, indicating that the keywords in that cluster are commonly used in the community (i.e., strong).

‡ Other low-frequency keywords are omitted to reduce visual clutter.

and coreness and a low score on constraint can be considered as the driving force for advances in the field: without these topics, a research field would be fragmented.

Table 3 shows the results of this analysis of the machine-extracted keywords. We can see that the term 'introductory programming courses' dominates in the machine-extracted keywords (is the most popular, core, and backbone topic), while topics such as assessment, software development, CS1, exams, and assessment were also identified as significant keywords (top 10 in popularity, coreness, and connectivity with other topics). The results of the analysis of the author-assigned keywords can be seen in Table 4. Topics identified as significant (top 10 in popularity, coreness, and connectivity with other topics) are specific courses or topics, learning approaches, aspects of programming, students, programming research and approaches, and programming languages and environments.

Comparing the two keyword networks, we see that while introductory programming dominates in the machine-extracted keywords, it also has a central role and is very popular in the author-grouped keywords. In the machine-extracted keywords (Figure 6), besides the dominant cluster of introductory programming, we can see several relatively large clusters: one in the areas of software development and software engineering; one in the areas of computational thinking, K-12, and high school students; and one in feedback, grading, and assessment. There is also a good number of smaller clusters (e.g., games and game development; novice programmer and syntax errors; and programming languages, recursion and mental models). On the author-assigned and expert-grouped keywords (Figure 7), we can see several central nodes, such as introductory programming, learning approaches, assessment, students, as well as some generic terms such as computing education. We can also see several nodes incorporating thematic areas that are less popular (depicted by their size), central (depicted by their position), and interconnected (depicted by their connections): these themes include evaluation, educational data, academic integrity, visualization, teachers, and course management.

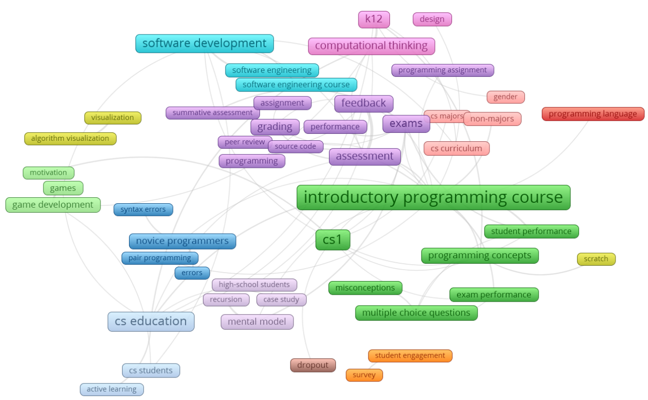

Figure 6: Keyword network map for CER, 2005—2019, based on machine-extracted keywords; each line links two keywords with correlation coefficient ≥ 0.24

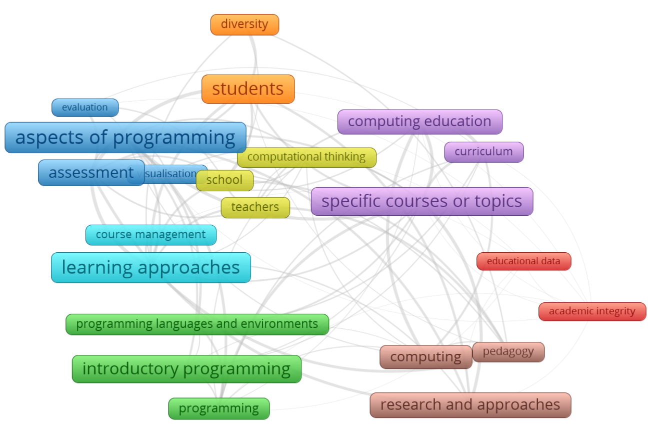

Figure 7: Keyword network map for CER, 2005-–2019, based on human-processed author keywords; each line links two keywords with correlation coefficient ≥ 0.18

5 DISCUSSION

The CER community has witnessed steady growth over the past two decades, as evidenced by the initiation and development of new conferences (ICER and CompEd) and the growing number of submissions and participants in all major computing education conferences. As we see from the results of this analysis, CER has also developed several mainstream/established areas where research is mature. Since our research field is constantly growing and progressing, it is important to map the landscape as well as to identify the popular and core research topics in order to facilitate an understanding of our community and its subfields.

5.1 Machine 'Intelligence': from Text Summarization to Unsupervised Clustering

Transitivity and density are used to measure the degree to which the key-terms within a cluster are related to one another. The clusters with the highest levels of density ($density \geq 1.5$) also have high levels of transitivity ($transitivity > 0.7$). Such clusters include C2 (introductory programming), C11 (games), C4 (algorithm visualization), C6 (software development), and C8 (drop-out). These clusters represent groups of topics that are closely related or often appear together in published work.

As shown from our analysis (Table 1), some of the CER thematic areas have reached a relatively high level of maturation and centrality in the field (e.g., introductory programming, computational thinking), although some researchers might be disappointed that some areas were not found to be as mature as they might have liked (e.g., algorithm visualization, assessment).

Table 3 identifies the most popular (high frequency), core (high connection with other topics), and backbone (connection with otherwise isolated topics) thematic areas that emerged during the period 2005–2019. Six of the most popular themes (identified in bold) are also in the top ten core and backbone themes in the field, suggesting a high consistency between research interests and scientific efforts to maintain the sustainability of the field. However, one of these themes, *CS education*, is very general and tells us relatively little about the content of the corresponding papers in this analysis.

Introductory programming appears to be the most frequent topic, has the highest connection with other topics, and connects otherwise isolated topics. The keyword *CS1*, which is considered a proxy for introductory programming, also appears in the top ten most popular, core, and backbone themes. The analysis of clusters described in Table 1 illustrates that the cluster containing introductory programming is one of the largest clusters, and appears in Q1 with strong centrality and high density — this is a mainstream theme. Our analysis confirms findings from a recent substantive review of this topic which found that there is a large (and growing) body of research into introductory programming, and that the research relates to a wide range of other topic areas that impact students, teachers, curriculum, and assessment [21].

Two related topics that demonstrate high levels of popularity, core, and backbone characteristics are the themes of *assessment* and *exams*. Table 1 shows that the terms *assessment* and *exams* appear in the same cluster, and that cluster has very high centrality, connecting to a large number of other clusters. This is unsurprising, as assessment is a central component of education and forms one of the main data sources that can be analysed in computing education. This suggests a substantial focus in the community on the way that student performance is evaluated.

Table 3: Summary of popular, core, and backbone topics of computing education, 2005–2019, machine-extracted keywords

#	Popular Topic	Frequency	Core Topic	Coreness [0-1]	Backbone Topic	Constraint [0-1]
1	**introductory programming course**	68	**introductory programming course**	0.445	**introductory programming course**	0.116
2	**assessment**	63	**cs1**	0.224	**SW development**	0.154
3	feedback	58	**cs education**	0.148	**cs1**	0.171
4	**cs education**	57	**SW development**	0.145	**cs education**	0.202
5	**cs1**	55	**exams**	0.111	programming concepts	0.227
6	**SW development**	45	programming concepts	0.096	multiple choice ques.	0.250
7	student performance	44	**assessment**	0.095	**exams**	0.256
8	games	41	novice programmers	0.075	k12	0.286
9	game development	35	k12	0.070	computational thinking	0.301
10	**exams**	34	computational thinking	0.062	**assessment**	0.309

Table 4: Summary of popular, core, and backbone topics of computing education, 2005–2019, human-processed keywords

#	Popular Topic	Frequency	Core Topic	Coreness [0-1]	Backbone Topic	Constraint [0-1]
1	**specific course/topic**	441	**aspects of prog/ming**	0.048	**students**	0.238
2	**learning approaches**	406	assessment	0.042	**specific course/topic**	0.288
3	**aspects of prog/ming**	357	**specific course/topic**	0.034	school	0.291
4	**students**	299	**students**	0.020	teachers	0.293
5	computing education	253	**learning approaches**	0.020	**learning approaches**	0.294
6	assessment	229	**research/approaches**	0.014	**aspects of prog/ming**	0.294
7	**programming**	229	computing education	0.014	**programming**	0.298
8	**research/approaches**	201	**programming**	0.013	computational thinking	0.300
9	introductory prog/ming	175	introductory prog/ming	0.008	research/approaches	0.304
10	**prog/ming languages & environments**	115	**prog/ming languages & environments**	0.007	**prog/ming languages & environments**	0.305

The final theme identified as popular, core, and backbone is the broad category of *software development*. It could be argued that much of what we do as computing educators relates to software development, or is motivated by producing graduates capable of software development, which may explain the significance of this theme in the data.

However, it is worth noting that the machine-extracted keywords may unfairly emphasize topic areas for which there are commonly used and broadly applicable terms that have few synonyms. Very general terms, such as *CS education* and *computing education*, are frequently used by authors and therefore identified as popular topics. Such terms are useful to distinguish different disciplinary areas (e.g., computing education compared with cybersecurity) in computing databases such as the ACM Digital Library, but provide little value in distinguishing topic areas *within* computing education in an analysis of papers published in venues associated with the computing education discipline.

As the machine analysis does not use semantic clustering during calculation of popularity, core, and backbone, topics such as *CS1* and *introductory programming* are treated as distinct. If a topic area had a diverse range of synonyms that were all used in the abstract, the machine analysis would record the topic area as diffused into a wide range of lower-frequency keywords.

5.2 Human Perspective: from Selection of Keywords to Abstract (Conceptual) Schema

The strongest cluster of the 23 groups described in Table 2 is the computing education cluster (CH6). This cluster comprises generic keywords that relate to education (e.g., computer science education, computing education, education), which serve a valuable purpose in distinguishing education-focused publications from other disciplinary areas, but are of limited use within the CER community.

We observe that CER authors use relatively few theory-related keywords (CH17 and CH14). This is an interesting observation, since we know from the literature that approximately six out of ten CER papers acknowledge a theoretical underpinning [22]. Nevertheless, authors do not appear to be using keywords to categorize their papers according to the theories they employ.

Besides the thematically constructed clusters, the cluster CH23 consists of generic terms (e.g., experience report, tools, data). There is a very large number of such generic terms (862), with very low frequency (1.19 on average). The use of such generic and low-frequency terminology in CER contributes to a long-tailed distribution of keywords. One possible explanation for this observation is that the community's interests are broad and disparate. However, a closer look at the keywords shows that the long tail is due both to the selection of keywords that do not necessarily characterize the contribution of the paper (e.g., data, report, technology, tools) and to the absence of a common nomenclature to describe common concepts (e.g., introductory CS, introductory programming, introductory courses, CS1).

The most popular (highest-frequency) clusters identified through manual coding (Table 2) are learning approaches (CH13, 622), specific courses or topics (CH19, 607), aspects of programming (CH2, 480), students (CH20, 408), and assessment (CH3, 327). These clusters reflect the most active areas of publication – our teaching strategies (learning approaches), the content we focus on (specific courses and topics, aspects of programming), our students, and how we assess those students.

5.3 Comparing Machine Intelligence and Human Perspective

The analysis involving machine extracted keywords is objective, yet unaware of context. This results in clusters of keywords that may frequently appear together, but have different semantic meaning (e.g., CS1 and exam performance). The authors' choice of keywords is more context-aware, but while the human clustering of those keywords adds subjective bias, it can more easily group semantically similar ideas together. This ability to form abstract conceptualizations of the keywords results in a different clustering of topics that focuses on semantics rather than structural relationships between the keywords.

Despite the differences in approaches, assessment was identified as a significant topic area by both the manual and automated analyses. Introductory programming, which was identified as popular, core, and backbone by the machine analysis, also includes programming concepts within the same cluster. The manual analysis includes *aspects of programming* and *programming* within the top ten most popular, core, and backbone clusters. The manual categorization also results in introductory programming being in the top ten popular and core clusters, but not a backbone cluster. This provides a high degree of confidence that both assessment and programming form a strong core of computing education research.

5.4 Limitations

Although this work considers a substantial portion of the published work of the past 15 years, we do not claim that it provides a comprehensive review of the field; rather, it provides insights from quantification of the author-assigned keywords and key-phrases extracted automatically from the papers' abstracts, in order to map the landscape of the CER community. The selection and execution of each step of our methodology was extensively discussed by the authors. However, as with any methodological decisions, we are aware that our choices also pose certain limitations.

First and foremost, the analysis includes only ICER and ITiCSE proceedings. Although these conferences are principal CER venues, the selection brings some bias to the study by excluding papers published in other computing education conferences, computing education journals, and indeed other computing, software engineering, HCI, and engineering education venues. These factors introduce a selection bias to our work — but the inclusion of CER contributions from other conferences and journals would also introduce a selection bias. Nevertheless, the papers included in our analysis (ICER papers, ITiCSE papers, and ITiCSE working group reports), lead to clear insights on the CER landscape seen through the lens of those particular publications.

Another crucial issue is the extent to which author-assigned or machine-extracted keywords accurately reflect a paper. Authors do not all follow the same approach when writing their abstracts or choosing keywords for their papers; they use different terminology, different focus, and different backgrounds, and this might lead to inconsistencies. Although our analysis takes care of some of the inconsistencies with dedicated protection mechanisms, such as disregarding very low-frequency keywords, there is still a certain bias coming from authors' habits and perceptions. Nevertheless, in order to map the landscape of a research community, it is important to consider how the main actors of this community, the authors, perceive the various thematic areas and consequently select their keywords and write their abstracts.

5.5 Conclusions and Future Work

CER is a growing community with several annual conferences (e.g., ICER, ITiCSE, SIGCSE TS) and and journals (e.g., TOCE, CSEd). As the community grows, there is potential benefit in mapping the landscape and progress of the various topic areas, in discussing where we are, where we want to be, and what it takes to get there. This study performed a co-word analysis on two CER publication channels (ICER and ITiCSE) in order to map the landscape and progress of the field via various metrics (e.g., core, popular, and emerging topics) and visualizations (e.g., keyword networks). The findings of our study suggest that recent growth in CER includes several mainstream themes (programming concepts, introductory programming, student performance, exam performance, CS1, misconceptions, student understanding, to mention a few), that are summarized in three clusters (Table 1). The results from the analysis of the two different perspectives (machine-extracted keywords and author-assigned expert-grouped keywords) show that introductory programming seems to dominate (with very high frequency in both analyses); that topics such as software engineering, evaluation, and assessment are identified as significant; while topics such as academic integrity, diversity, and educational data seem to be in the periphery.

Future work can further our understanding of CER development by conducting analyses such as authorship analysis and citation analysis, or more qualitative approaches such as systematic and narrative reviews. In addition, future work can consider CER publications from other venues such as SIGCSE TS, TOCE, and CSEd. As our findings demonstrate a reasonable overlap between the machine-generated and human-annotated analysis, future work can apply the machine-generated analysis to larger data sets, such as the SIGCSE TS corpus, where human analysis would be highly laborious. Finally, further analysis should consider investigating potential differences of the use of terms between the different periods (for example, in five-year windows). Such an analysis would reveal areas that had recently emerged, areas that had disappeared, and areas that had transformed into something new. This could be particularly interesting, since besides the traditional research areas of CER, we would be able to see research approaches arising from bridging between CER and other areas such as learning analytics, ITS, HCI, and K-12 education.

A recommendation for authors is that they devote more time and consideration when choosing keywords and writing abstracts. Can these fields be seen as providing a plausible, albeit brief, summary of the paper? If so, they are more likely to be helpful, not just for automated analysis but also for informing prospective readers.

ACKNOWLEDGMENTS

We thank Prof Vassilis Kostakos from the University of Melbourne and his team for sharing the initial version of the co-word analysis code with us.

REFERENCES

[1] 2020. *51st ACM Technical Symposium on Computer Science Education*. ACM. https://doi.org/10.1145/3328778

[2] Ahmed Al-Zubidy, Jeffrey C Carver, Sarah Heckman, and Mark Sherriff. 2016. A (updated) review of empiricism at the SIGCSE Technical Symposium. In *47th ACM Technical Symposium on Computing Science Education (SIGCSE 2016)*. ACM, 120—-125. https://doi.org/10.1145/2839509.2844601

[3] Vicki L Almstrum, Orit Hazzan, Mark Guzdial, and Marian Petre. 2005. Challenges to computer science education research. *ACM SIGCSE Bulletin* 37, 1 (2005), 191–192. https://doi.org/10.1145/1047344.1047415

[4] Ronald S Burt. 2004. Structural holes and good ideas. *Amer. J. Sociology* 110, 2 (2004), 349–399. https://doi.org/10.1086/421787

[5] M Callon, J-P Courtial, and F Laville. 1991. Co-word analysis as a tool for describing the network of interactions between basic and technological research: the case of polymer chemistry. *Scientometrics* 22, 1 (1991), 155–205. https://doi.org/10.1007/BF02019280

[6] Michel Callon, Jean-Pierre Courtial, William A Turner, and Serge Bauin. 1983. From translations to problematic networks: an introduction to co-word analysis. *Information (International Social Science Council)* 22, 2 (1983), 191–235. https://doi.org/10.1177/053901883022002003

[7] Alberto Cambrosio, Camille Limoges, J Courtial, and Françoise Laville. 1993. Historical scientometrics? Mapping over 70 years of biological safety research with coword analysis. *Scientometrics* 27, 2 (1993), 119–143.

[8] MJ Cobo, AG López-Herrera, E Herrera-Viedma, and F Herrera. 2011. Science mapping software tools: review, analysis, and cooperative study among tools. *Journal of the American Society for Information Science and Technology* 62, 7 (2011), 1382–1402. https://doi.org/10.1002/asi.21525

[9] M de Laat, V Lally, L Lipponen, and R-J Simons. 2007. Investigating patterns of interaction in networked learning and computer-supported collaborative learning: a role for social network analysis. *Journal of Computer-Supported Collaborative Learning* 2, 1 (2007), 87–103.

[10] Varvara Garneli, Michail N Giannakos, and Konstantinos Chorianopoulos. 2015. Computing education in K-12 schools: a review of the literature. In *IEEE Global Engineering Education Conference (EDUCON)*. IEEE, 543–551. https://doi.org/10.1109/EDUCON.2015.7096023

[11] Albert Gifi. 1990. *Nonlinear multivariate analysis*. Wiley.

[12] Mark Guzdial and Benedict du Boulay. 2019. The history of computing education research. Cambridge University Press, Cambridge, UK, 11–39.

[13] Qiang Hao, David H. Smith IV, Naitra Iriumi, Michail Tsikerdekis, and Andrew J. Ko. 2019. A Systematic Investigation of Replications in Computing Education Research. *ACM Transactions on Computing Education (TOCE)* 19, 4 (2019), 1–18. https://doi.org/10.1145/3345328

[14] Qin He. 1999. Knowledge discovery through co-word analysis. *Library Trends* 48, 1 (1999), 133–159. http://hdl.handle.net/2142/8267

[15] Christian Holmboe, Linda McIver, and Carlisle E George. 2001. Research agenda for computer science education. In *Psychology of Programming Interest Group (PPIG)*, Vol. 13. 207–223.

[16] C-P Hu, J-M Hu, S-L Deng, and Y Liu. 2013. A co-word analysis of library and information science in China. *Scientometrics* 97, 2 (2013), 369–382. https://doi.org/10.1007/s11192-013-1076-7

[17] Päivi Kinnunen, Veijo Meisalo, and Lauri Malmi. 2010. Have we missed something? Identifying missing types of research in computing education. In *Sixth International Computing Education Research Workshop (ICER 2010)*. ACM, 13–22. https://doi.org/10.1145/1839594.1839598

[18] L Leydesdorff and L Vaughan. 2006. Co-occurrence matrices and their applications in information science: extending ACA to the web environment. *American Society for Information Science and Technology* 57, 12 (2006), 1616–1628. https://doi.org/10.1002/asi.20335

[19] Yong Liu, Jorge Goncalves, Denzil Ferreira, Simo Hosio, and Vassilis Kostakos. 2014. Identity crisis of UbiComp? Mapping 15 years of the field's development and paradigm change. In *ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 75–86. https://doi.org/10.1145/2632048.2632086

[20] Yong Liu, Jorge Goncalves, Denzil Ferreira, Bei Xiao, Simo Hosio, and Vassilis Kostakos. 2014. CHI 1994-2013: mapping two decades of intellectual progress through co-word analysis. In *32nd Conference on Human Factors in Computing Systems*. ACM, 3553–3562. https://doi.org/10.1145/2556288.2556969

[21] Andrew Luxton-Reilly, Simon, Ibrahim Albluwi, Brett A Becker, Michail Giannakos, Amruth N Kumar, Linda Ott, James Paterson, Michael James Scott, Judy Sheard, and Claudia Szabo. 2018. Introductory programming: a systematic literature review. In *ITiCSE Working Group Reports (ITiCSE WGR 2018)*. ACM, 55–106. https://doi.org/10.1145/3293881.3295779

[22] Lauri Malmi, Judy Sheard, Simon, Roman Bednarik, Juha Helminen, Päivi Kinnunen, Ari Korhonen, Niko Myller, Juha Sorva, and Ahmad Taherkhani. 2014. Theoretical underpinnings of computing education research: what is the evidence?. In *Tenth International Computing Education Research Conference (ICER 2014)*. ACM, 27–34. https://doi.org/10.1145/2632320.2632358

[23] Lauren Margulieux, Tuba Ayer Ketenci, and Adrienne Decker. 2019. Review of measurements used in computing education research and suggestions for increasing standardization. *Computer Science Education* 29, 1 (2019), 49–78.

[24] Robert McCartney and Kate Sanders. 2018. ITiCSE working groups and collaboration in the computing education community. In *23rd Conference on Innovation and Technology in Computer Science Education (ITiCSE 2018)*. ACM, 332–-337. https://doi.org/10.1145/3197091.3197143

[25] R Mihalcea and P Tarau. 2004. TextRank: bringing order into text. In *Conference on Empirical Methods in Natural Language Processing*. 404–411.

[26] Joe Miró Julià, David López, and Ricardo Alberich. 2012. Education and research: evidence of a dual life. In *Ninth International Computing Education Research Conference (ICER 2012)*. ACM, 17–-22. https://doi.org/10.1145/2361276.2361281

[27] Diba Mirza, Phillip T Conrad, Christian Lloyd, Ziad Matni, and Arthur Gatin. 2019. Undergraduate teaching assistants in computer science: a systematic literature review. In *15th International Computing Education Research Conference (ICER 2019)*. ACM, 31–40. https://doi.org/10.1145/3291279.3339422

[28] F Murtagh and P Legendre. 2014. Ward's hierarchical agglomerative clustering method: which algorithms implement Ward's criterion? *Journal of Classification* 31, 3 (2014), 274–295. https://doi.org/10.1007/s00357-014-9161-z

[29] ME Newman. 2005. A measure of betweenness centrality based on random walks. *Social Networks* 27, 1 (2005), 39–54.

[30] AE Nielsen and C Thomsen. 2011. Sustainable development: the role of network communication. *Corporate Social Responsibility and Environmental Management* 18, 1 (2011), 1–10.

[31] Zacharoula Papamitsiou, Michail N Giannakos, and Xavier Ochoa. 2020. From childhood to maturity: are we there yet? Mapping the intellectual progress in learning analytics during the past decade. In *Tenth International Conference on Learning Analytics & Knowledge (LAK 2020)*. ACM, 559–568. https://doi.org/10.1145/3375462.3375519

[32] Justus J Randolph, George Julnes, Erkki Sutinen, and Steve Lehman. 2008. A methodological review of computer science education research. *Journal of Information Technology Education: Research* 7, 1 (2008), 135–162. https://doi.org/10.28945/183

[33] Anthony Robins. 2015. The ongoing challenges of computer science education research. *Computer Science Education* 25, 2 (2015), 115–119. https://doi.org/10.1080/08993408.2015.1034350

[34] P Rombach, MA Porter, JH Fowler, and PJ Mucha. 2017. Core-periphery structure in networks (revisited). *SIAM Review* 59, 3 (2017), 619–646.

[35] Kate Sanders, Judy Sheard, Brett A Becker, Anna Eckerdal, Sally Hamouda, and Simon. 2019. Inferential statistics in computing education research: a methodological review. In *ACM Conference on International Computing Education Research (ICER 2019)*. ACM, 177–185. https://doi.org/10.1145/3291279.3339408

[36] T Schank and D Wagner. 2005. Approximating clustering coefficient and transitivity. *Journal of Graph Algorithms and Applications* 9, 2 (2005), 265–275. https://doi.org/10.7155/jgaa.00108

[37] Simon. 2007. A classification of recent Australasian computing education publications. *Computer Science Education* 17, 3 (2007), 155–169. https://doi.org/10.1080/08993400701538021

[38] Simon. 2009. Informatics in Education and Koli Calling: a comparative analysis. *Informatics in Education* 8, 1 (2009), 101–114. https://doi.org/10.15388/infedu.2009.07

[39] Simon. 2009. Ten years of the Australasian Computing Education Conference. In *11th Australasian Conference on Computing Education (ACE 2009)*. Australian Computer Society, Inc., 157–164.

[40] Simon. 2015. *Emergence of Computing Education as a Research Discipline*. Ph.D. Dissertation. Aalto University.

[41] Simon. 2020. Twenty-Two Years of ACE. In *22nd Australasian Computing Education Conference (ACE 2020)*. ACM, 203–-210. https://doi.org/10.1145/3373165.3373188

[42] Simon, Angela Carbone, Michael de Raadt, Raymond Lister, Margaret Hamilton, and Judy Sheard. 2008. Classifying computing education papers: process and results. In *Fourth International Workshop on Computing Education Research (ICER 2008)*. ACM, 161–172. https://doi.org/10.1145/1404520.1404536

[43] Simon and Judy Sheard. 2020. Twenty-four years of ITiCSE papers. In *ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE 2020)*. ACM, 5–-11. https://doi.org/10.1145/3341525.3387407

[44] Simon, Judy Sheard, Angela Carbone, Michael de Raadt, Margaret Hamilton, Raymond Lister, and Errol Thompson. 2008. Eight years of computing education papers at NACCQ. *National Advisory Committee on Computing Qualifications* (2008), 101–107. https://www.citrenz.ac.nz/conferences/2008/101.pdf

[45] Claudia Szabo, Nickolas Falkner, Andrew Petersen, Heather Bort, Kathryn Cunningham, Peter Donaldson, Arto Hellas, James Robinson, and Judy Sheard. 2019. Review and use of learning theories within computer science education research: primer for researchers and practitioners. In *ITiCSE Working Group Reports*. ACM, 89–109. https://doi.org/10.1145/3344429.3372504

[46] Claudia Szabo, Judy Sheard, Andrew Luxton-Reilly, Simon, Brett A. Becker, and Linda Ott. 2019. Fifteen Years of Introductory Programming in Schools: A Global Overview of K-12 Initiatives. In *19th Koli Calling International Conference on Computing Education Research (Koli Calling 2019)*. ACM. https://doi.org/10.1145/3364510.3364513