Applied Computing and Informatics

Revealing essential notions: an algorithmic approach to distilling core concepts from student and teacher responses in computer science education

Received 31 December 2023
Revised 14 September 2024
Accepted 1 October 2024

Zaira Hassan Amur and Yew Kwang Hooi
Department of Computer and Information Sciences, Universiti Teknologi PETRONAS, Seri Iskandar, Malaysia
Gul Muhammad Soomro
Department of Artificial Intelligence, Tomas Bata University in Zlin, Zlin, Czech Republic, and
Hina Bhanbhro
Universiti Teknologi PETRONAS, Seri Iskandar, Malaysia

Abstract

Purpose – This study aims to assess subjective responses in computer science education to understand students' grasp of core concepts. Extracting key ideas from short answers remains challenging, necessitating an effective method to enhance learning outcomes.

Design/methodology/approach – This study introduces KeydistilTF, a model to identify essential concepts from student and teacher responses. Using the University of North Texas dataset from Kaggle, consisting of 53 teachers and 1,705 student responses, the model's performance was evaluated using the F1 score for key concept detection.

Findings – KeydistilTF outperformed baseline techniques with F1 scores improved by 8, 6 and 4% for student key concept detection and 10, 8 and 6% for teacher key concept detection. These results indicate the model's effectiveness in capturing crucial concepts and enhancing the understanding of key curriculum content.

Originality/value – KeydistilTF shows promise in improving the assessment of subjective responses in education, offering insights that can inform teaching methods and learning strategies. Its superior performance over baseline methods underscores its potential as a valuable tool in educational settings.

Keywords Key concepts, Teacher-student model, Core ideas, Concept detection, Dynamic of learning

Paper type Full length article

1. Introduction

There has been a notable increase in initiatives aimed at providing comprehensive computer science (CS) education to all students [1,2]. The crucial idea of extracting core concepts is essential to this advancement. Through analyzing and deriving key concepts from the responses of both students and teachers, educators can improve curricula, offer focused feedback and ensure that the methods of instruction align with core values. The process of distilling essential concepts into a manageable form can help teachers grow professionally. Teacher growth and improvement are facilitated by the analysis of student responses, which enables them to pinpoint areas for improvement in their teaching and learning methods. The

Applied Computing and Informatics
Emerald Publishing Limited
e-ISSN: 2210-8327
p-ISSN: 2634-1964
DOI 10.1108/ACI-12-2023-0207

answers given by students are in variable context compared to the key solutions provided by teachers. Evaluating and assessing the very short subjective answers is quite challenging, as they contain very little domain and key information. These answers are natural language responses given by the students. The key concepts in student and teacher responses are known as unigrams (one word), bigrams (two words) and trigrams (three words). In addition to this, bigrams and trigrams provide more nuances and context information as compared to unigrams. These language units are essential for improving written content [3–5]. These fundamental ideas enable teachers to evaluate language proficiency and comprehension in the classroom while also empowering students to express themselves clearly. Teachers can develop students' textual analysis and critical thinking abilities by stressing the significance of bigrams, unigrams and trigrams. This will ultimately lead to clearer communication and more insightful assessments in the classroom [6,7]. A few CS examples to highlight the significance of unigrams, bigrams and trigrams in student and teacher responses are as follows:

Unigram: In the realm of programming, a unigram is a fundamental linguistic unit representing a single word or term that encapsulates a specific concept. In simpler terms, it serves as a basic building block of language, allowing for concise expression and understanding of key ideas. An example of a unigram in programming is the word "algorithm." This phrase refers to a methodical set of guidelines or directives intended to resolve a computational issue. Students can use the idea of an "algorithm" that resembles a unigram to quickly explain the reasoning or procedures required to solve challenging computational problems [8]. A unigram is essentially a linguistic shortcut that helps students express and understand the basic ideas that are necessary for solving problems in the field of computation.

Bigram: Building on the example of programming, a bigram such as "machine learning" presents a particular area of CS. By combining two terms, it emphasizes the connection between algorithms and data-driven decision-making and conveys a more specific field of study. By using these paired terms, students can converse and investigate more complex subjects, fostering a more thorough and sophisticated discourse within the classroom. Essentially, bigrams act as links between discrete ideas and open the door to a more in-depth investigation of the various aspects that CS encompasses [9].

Trigram: Following that, a trigram denoting "natural language processing" delves into a specific field within machine learning. This word combination sheds light on the connection between computational techniques, algorithms and language comprehension, providing a more comprehensive understanding of the student's knowledge.

The objective of this study is as follows:

(1) To develop a model that can extract the important key concepts from student and teacher answers to identify the patterns and variations in provided answers and

(2) To compare the developed model for other key concepts and baseline methods.

Meanwhile, there are many advantages of keyword extraction methods in education settings [10–12] such as curriculum can be organized through keyword search and textbooks and materials can be sorted through key concept ideas. Even researchers used the keywords to search the relevant articles and case studies for their domain of interest. Keyword extraction can also be incorporated in grading and assessment tasks to understand how well students align the concept with a teacher-centric domain. Based on the matched keywords, teachers can grade student answers accordingly.

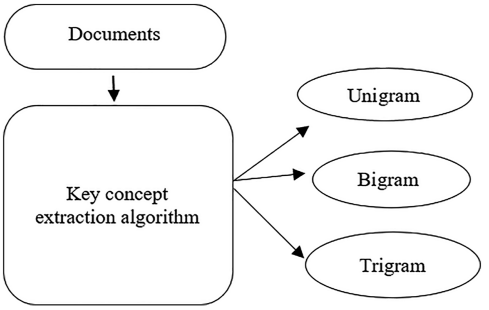
2. Related works

Previous studies have explored various methods for extracting key concepts in the forms of unigrams, bigrams and trigrams from diverse sources, including journal articles, newspapers and blogs. For instance, Ref. [13] proposed YAKE, a lightweight and unsupervised method

capable of extracting keywords from multiple documents without the need for prior training. YAKE can handle different sizes of key concepts, offering a versatile solution for key phrase extraction. In addition to YAKE, other methods such as KP-Miner, Multipartite Rank (MR) and TeKET have been widely used for key concept detection across various domains [14]. KP-Miner, known for its unsupervised and lightweight nature, initially selects candidate documents or sentences, assigns weights and then extracts frequently used terms. It incorporates a modified version of term frequency-inverse document frequency (TF-IDF) along with N-Gram analysis in its ranking methodology, adjusting the weights of multi-word candidates based on their single-word counterparts [15]. Similarly, TeKET is a tree-based key phrase extraction algorithm that operates independently of domain constraints, requiring minimal statistical knowledge. It employs a binary tree, KePhEx, to efficiently select key phrases, excelling in extracting significant terms from the initial candidate pool. MR further advances the field by implementing a two-step process: document graph conversion and relevance scoring. It leverages positional information to assign edge weights, demonstrating a preference for early text key phrases and forming a directed graph across different topics [16]. This complexity allows MR to surpass previous graph-based algorithms. Likewise, the widely known TF-IDF method evaluates term significance by considering both term frequency within a document and its rarity across the dataset [17], aiding in tasks like document similarity and information retrieval. Advanced methods like KeyBERT, as used by Ref. [18], incorporate the contextual understanding of Bidirectional Encoder Representations from Transformers (BERT) to extract key phrases. KeyBERT goes beyond frequency-based techniques by leveraging pre-trained language models to identify the most contextually relevant terms, assisting in tasks such as summarization and topic modeling [19]. KEA, another powerful algorithm designed for keyword extraction [20], utilizes a multimodal approach that integrates both linguistic and statistical features to identify key phrases effectively [21].

Despite the success of these methods in various applications, there remains a gap in handling short, i.e. subjective responses commonly found in educational settings. Many of these algorithms require further adaptation to effectively capture key concepts in concise, domain-specific answers provided by students, which this study aims to address.

Figure 1 presents the general concept of extracting the key concepts from algorithms. The unsupervised or supervised nature of algorithms can detect one word (unigram), two words (bigram) and three words (trigram). Moreover, the short nature of subjective answers makes the machine learning algorithms challenging to extract the key concepts. This study follows Bloom's taxonomy level 1 "remembering" type questions from the CS domain. Level 1 of Bloom Taxonomy contains the answers that are concise and hard to assess for several algorithms. Level 1 concentrates on memorization of information, which might not help develop a thorough comprehension of the subject. Students may find it difficult to understand



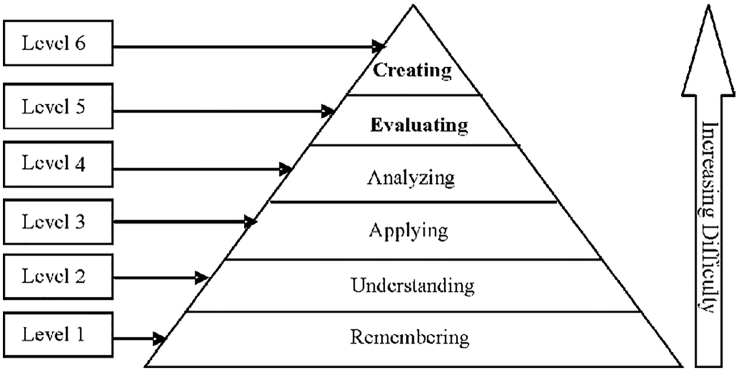
Source(s): Authors' own creation

Figure 1. Example of the general pipeline of key concept extraction

the larger meaning or context. Such types of questions contain very little information to respond to the answer. Figure 2 mentions the levels of Bloom's Taxonomy.

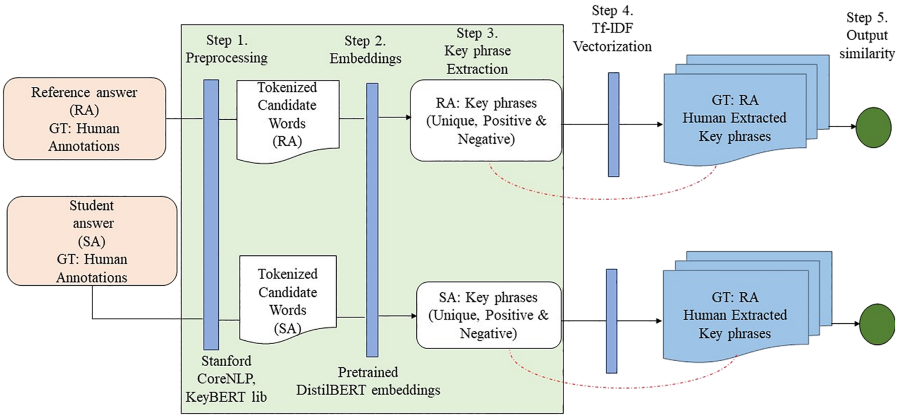
3. Methodology

KeydistilTF is a model specifically designed to extract key phrases from teacher and student responses with high effectiveness. As shown in Figure 3, the model operates through several distinct phases: preprocessing, embedding, key phrase extraction, TF-IDF vectorization and output similarity calculation. The primary contribution of KeydistilTF lies in its integration of distilBERT embeddings into the key phrase extraction process, which allows it to move beyond traditional frequency-based methods. Unlike other models such as RAKE, YAKE and TF-IDF, which rely on statistical approaches and frequency co-occurrence to identify key concepts, KeydistilTF captures the semantic relationships between words. Moreover, its unique ability to extract both positive and negative key points sets it apart, enabling a more holistic evaluation of responses. This robust approach ensures a deeper and more nuanced



Source(s): Authors' own creation

Figure 2. Example of bloom's taxonomy levels



Source(s): Authors' own creation

Figure 3. Example of proposed KeydistilTF model

analysis of student and teacher answers, offering a comprehensive solution for key concept extraction that goes beyond mere statistical patterns.

Input: The model accepts the input in the form of reference answers and student answers. We have annotated the reference answer with ground truth key phrases repeated the same for the student answers.

Step 1: The model uses preprocess techniques from the Sandford CoreNLP and KeyBERT libraries. To reduce the noise from text, the model removed the stop words, punctuations and spelling mistakes and applied case-folding techniques to make the text more readable for the algorithm. Table 1 illustrated the example of noise removal techniques.

After removing the noise, the model applied the natural language processing (NLP) tokenizer from the preprocessing library to split the sentences into chunks. Tokenization enhances text processing efficiency by breaking down input text into smaller units, such as words or phrases, facilitating more effective analysis and computation. For example, after applying the tokenizer to the teacher answer, it looks like "location," "memory," "store," and "value." Same applied on the student answer: "variable," "location," "computer," "memory," "value," "stored" and "program."

Step 2: In NLP, embeddings are essential because they represent words or phrases in a continuous vector space and capture their semantic relationships [22] Embeddings are numerical representations of textual elements within the KeyBERT model. A significant change happened in the second step of our procedure when we switched from BERT embeddings to distilBERT embeddings. This change is significant because it uses the faster and more simplified distilBERT version of the BERT model to create embeddings [23]. This modification contributes to the overall efficacy of the KeyBERT algorithm by improving the model's efficiency and guaranteeing the extraction of more dependable and contextually rich embeddings.

Step 3: In this phase, the algorithm proceeded to methodically recognize and extract important terms from the answers provided by teachers and students. These crucial phrases were carefully chosen to avoid repetition because of their rarity and significance to the corresponding sentences. Most remarkably, the model showed that it could identify negative terms when there was negation in the sentences. In addition to ensuring that key terms are extracted, this nuanced approach takes into consideration the finer points of negation, resulting in a more thorough comprehension of the information presented in both teacher and student responses.

Step 4: After this, the model went through a procedure where the teachers' and students' key concepts were transformed into TF-IDF vector representations. The next stage was to determine how similar each key concept was to the corresponding annotated ground truth

Table 1. Example of noise removal techniques from student and teacher answers

Question	Answer	Case folding (lower case)	Stop word, punctuation, and spelling errors
What is a variable?	Teacher answer: A location in memory that can store a value Student answer: A variable is the location in computer's memory where a value can be stored for use by a program	a location in memory that can store a value a variable is the location in computer's memory where a value can be stored for use by a program	location memory store value variable location computer memory value stored program

Source(s): Authors' own creation

key phrases, as depicted in Table 2. Section 4 provides a detailed explanation of the data collection ground truth annotation. Moreover, the similarity between the ground truth and model extracted bigrams has been done with the help of cosine similarity as depicted in the formula:

3.1 Equation

$$Cosine\,\theta = \frac{A.B}{||A||||B||} = \frac{\sum_{i=1}^{n} AiBi}{\sqrt{\sum_{i=1}^{n} A_i^2}\ \sqrt{\sum_{i=i}^{n} B_i^2}} \tag{1}$$

The above equation (1) explains the formula of cosine similarity. The symbol A · B denotes the dot product of vectors ||A ||and ||B||. The corresponding components of the two vectors are multiplied, and the results are then added up to determine the dot product. In essence, the cosine of the angle formed by two vectors is measured by the cosine similarity formula. A value between −1 and 1 is the outcome, where a value of 1 signifies that the vectors are the same. The vectors are orthogonal (no similarity) when the value is 0. The vectors are opposed, as indicated by the number −1. This is particularly important when dealing with text data, where words, phrases or documents are often represented as vectors in a high-dimensional space.

4. Data collection and experimentation details

In our investigation, we gathered sample data from the University of North Texas dataset [24], specifically focusing on 12 assignments within the realm of data structures in the field of CS. Approximately 29–30 students actively participated in responding to the assigned questions.

Table 2. Example of key concept extraction from model KeydistilTF

Question	Answers	Model extracted key phrase by KeydistilTF	Human assigned key phrases	Negative Terms	Cosine similarity
Question 1: What is a variable?	*Reference answer:* A location in memory that can store a value	Memory location store value memory store	Memory location store value	N/a	0.95
	Student answer: A variable is the location in computer's memory where a value can be stored for use by a program	Variable location computer memory value stored stored program	Variable location computer memory, stored value	N/a	0.89
Question 2: What are the main advantages associated with object-oriented programming?	*Reference answer:* Abstraction and reusability	Abstraction reusability	Abstraction reusability	N/a	0.1
	Student answer: Always scalable, no optimization needed	Always scalable, no optimization	Always scalable, no optimization	No optimization	0.1

Source(s): Authors' own creation

The responses underwent grading by two human evaluators; the dataset is freely available online on Kaggle source. This dataset was employed for the extraction of key concepts in this study, involving manual annotation of both teacher and student answers. Notably, a great deal of research has made use of this dataset, which has helped to resolve several research gaps. We randomly selected the number of answers from the dataset for key concept extraction as depicted in Table 3. In addition to this, Figure 4 presents the libraries and experimentation details used in this study. This depicts that along with preprocessing libraries, a list of negative terms has been created, which include "no, doesn't, don't, nothing, not, didn't, had not, not been, never, cannot and couldn't." According to this list, we have seen rare negative terms inside the student as well as teacher answers.

The approach used in our analysis is depicted in Figure 4, which also shows how Python 3 and Jupyter Notebook work together seamlessly. Spelling errors were corrected to improve the textual data's quality, and the KeyBERT library was essential in making key concept extraction easier. Furthermore, the addition of TF-IDF vectors raised the level of complexity in our method. This vectorization method allowed for a quantitative evaluation of similarity with the ground truth. By embracing both linguistic refinement and contextual similarity assessment, the application of these cutting-edge tools and techniques highlights the comprehensive nature of our analytical framework and ensures a strong evaluation of key concept extraction.

5. Comparison and findings

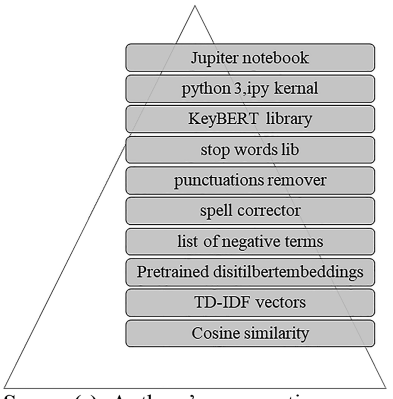
The model has been compared with a few baseline studies. Such as YAKE [13] RAKE [25] KeyBERT [18] and widely used method TF-IDF vectors [26]. Few examples of questions from the UNT dataset that have been used for evaluation.

Question 1. What is the role of a prototype program in problem-solving?

Table 3. Illustrates the available data used by the model for key concept extraction

Total teacher answers	Human annotated key concepts	Total student answers	Human annotated key concepts
53	324	1705	3,254

Source(s): Authors' own creation



Source(s): Authors' own creation

Figure 4. Example of libraries used in the experiment

Question 2. What is a variable?

Question 3. What are the main advantages associated with object-oriented programming?

Question 4. What is the scope of global variables?

Initially, we assessed the outcomes pertaining to teacher responses and student responses. The findings are summarized in Supplementary material Appendix 1, showcasing key concepts extracted by the model from teacher answers and student answers in Tables A1 and A2 across different models, as previously indicated. Additionally, we delve into the nuanced insights gleaned from this comparative analysis. Table A1 provides a comprehensive overview of the key concepts extracted by several models. Our analysis has revealed that RAKE and TF-IDF models would benefit significantly from further optimization techniques to enhance their precision in extracting the most relevant key terms. The extraction criteria were based on the conditions and size of bigrams (2,2), specifically targeting the extraction of the top five bigrams. Upon closer examination, it was observed that KeyBERT, while proficient in key concept extraction, exhibited a tendency to extract some terms in reverse and included a few rare words that were not present in the ground truth concepts. This observation highlights areas for potential refinement in the algorithm to ensure a more accurate alignment with the expected key concepts. Furthermore, within the context of Table A3, the abbreviation "TA" is utilized to denote teacher answers, adding clarity to the source of the extracted key concepts. This categorization aids in distinguishing between key concepts derived from teacher responses and facilitates a more nuanced understanding of the model's performance across different inputs. Moreover, Table A3 presents the key concepts extracted from students' answers. Moreover, Table A2 serves as a detailed showcase of the key concepts derived from student responses, shedding light on substantial performance variations among the models employed. Notably, YAKE and KeyBERT outperformed RAKE and TF-IDF in this context, demonstrating a higher efficacy in capturing essential information from the student answers. The noteworthy aspect of these models lies in their ability to successfully extract stop words, as well as unigram, bigram and trigrams, showcasing their versatility in handling diverse linguistic structures. Despite these achievements, it is crucial to acknowledge the ongoing need for refinement and advancement, particularly in the realm of optimizing methods for extracting negative key terms. While the current models excel in identifying positive or neutral concepts, the process of effectively capturing negative key terms remains an area with considerable room for enhancement. This recognition underscores the evolving nature of these methodologies and the imperative to continually refine them to ensure a more comprehensive and nuanced understanding of textual content.

5.1 Performance comparison of key concept extraction algorithms

The evaluation of performance involved a thorough comparison using machine learning evaluation metrics, namely precision, recall and F1 score. These metrics provide a quantitative measure of how well the models are performing in terms of accuracy and completeness, with a balanced combination of both the results have been presented in tabular form in Table A3. Key concept extraction from teacher answers are presented in Table A4. Key concept extraction from student answers in Supplementary material Appendix 1.

5.1.1 Equations for average similarity score, precision, recall and F1.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

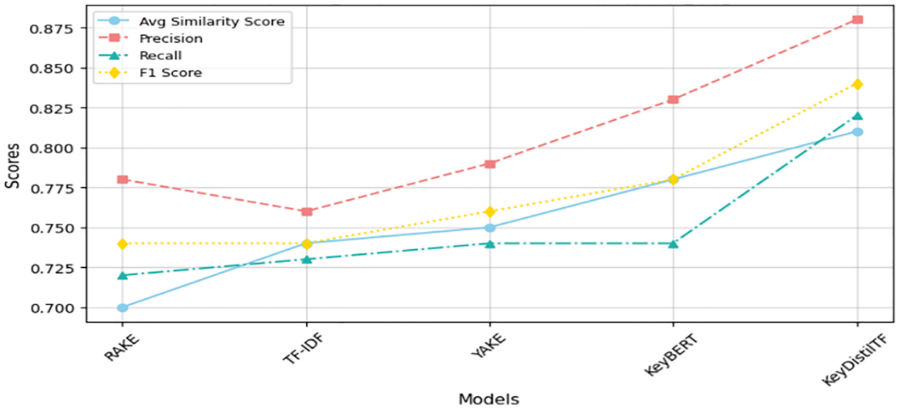
$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$\mathrm{F1} = 2\frac{(Precision * Recall)}{Precision + Recall} \tag{4}$$

$$Avg\ similairty = \frac{answerscore1 + answerscore2 + answerscore3}{total\ answers} \tag{5}$$

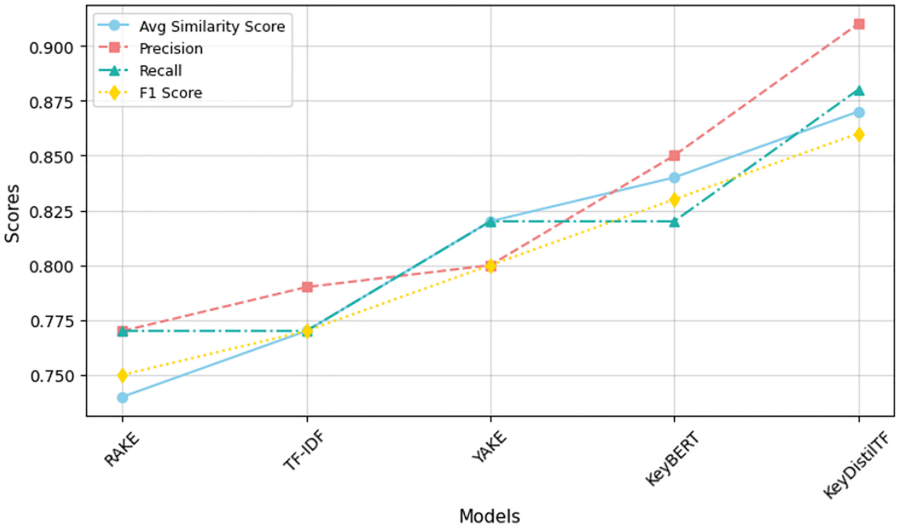
In addition, to this, Figures 5 and 6 present a detailed comparison with baseline methods.

Figure 5 offers a detailed look into the comparative analysis results for extracting key concepts from teacher answers. The findings suggest that the models generally yield satisfactory results; however, there are notable areas where optimization is still needed.



Source(s): Authors' own creation

Figure 5. Key concept extraction from teacher answers



Source(s): Authors' own creation

Figure 6. Key concept extraction from student answers

Particularly, RAKE and TF-IDF models exhibit room for improvement, as they occasionally include rare words that were not initially part of the ground truth. Moreover, in this evaluation, YAKE and KeyBERT continue to outperform TF-IDF and RAKE. It's worth noting that the improved performance of KeyBERT can be attributed to the strategic adjustment of its pre-trained model and the incorporation of TF-IDF vectors. This modification has resulted in markedly superior outcomes compared to other models, underscoring the significance of thoughtful model selection and feature integration in enhancing the accuracy and relevance of extracted key concepts. The ongoing pursuit of refinement and optimization, as evidenced by the insights from Table A4, highlights the dynamic nature of key concept extraction models. By addressing specific challenges and fine-tuning model parameters, we aim to consistently enhance the overall efficacy and reliability of these models in accurately capturing key concepts from teacher responses.

Furthermore, Figure 6 showcases the comparative results of extracting key concepts from student answers. Interestingly, the performance aligns closely with that observed in teacher answers, suggesting a consistent and comparable effectiveness across both sets of responses. This parallel performance indicates that the models are operating similarly on both teacher and student inputs. Despite the overall results, there remains a way of improving the models, particularly the RAKE and TF-IDF. These models need various optimization techniques for refining the key concepts. Several duplicate key phrases have been identified from these methods. This repetition causes the model to reduce its effectiveness. A standout in this evaluation is the KeydistilTF model, which has demonstrated substantial improvement over other baseline methods. This noteworthy advancement underscores the model's effectiveness in extracting key concepts from student responses, marking it as a promising approach in the landscape of key concept extraction. The recognition of commonalities in model performance across teacher and student answers, coupled with the identified areas for improvement, contributes valuable insights for future refinements in key concept extraction techniques. Continued efforts in optimizing models like KeydistilTF and addressing specific challenges with RAKE and TF-IDF can further elevate the overall efficacy of key concept extraction from diverse textual inputs.

The comparative analysis that follows provides results expressed as percentages and provides a thorough analysis of the data. These findings shed light on the comparative aspects of the analysis and offer insightful information while also advancing a thorough comprehension of the percentages. Additional investigation of the results adds to the richness of this synopsis by providing a more nuanced viewpoint on the comparative features noted in the data analysis.

Table 4 explains the percentage difference between our model and with other methods. The average similarity is 11% over RAKE, 11% similarity over YAKE, 14% similarity over TF-IDF and 7% similarity over the KeyBERT model. This similarity indicates that the overmodel has captured unique and representable key phrases that match with the ground truth key concepts presented in human annotations. Likewise, 10% precision over RAKE, 9% precision over YAKE, 12% precision over TF-IDF and 5% precision over the KeyBERT model. Precision determines that the positive instances have been carefully selected by our model. In addition to this, 10% recall over RAKE, 8% recall over YAKE, 9% recall over TF-IDF and 8% recall over the KeyBERT model. Recall rate on teacher answers indicate the actual instances are selected as presented in human annotations. However, 10% f1 over RAKE, 8% f1 over YAKE, 10% over TF-IDF and 6% over the KeyBERT model. F1 here indicated the combination of both precision and recall over teacher answers key concepts detection. Furthermore, Table 5 shows the same process applied to teacher answers on student answers. We have received the 13% similarity over RAKE, 11% similarity over YAKE, 14% similarity over TF-IDF and 7% similarity over the KeyBERT. In addition to this, 10% precision over RAKE, 9% over YAKE, 12% over TF-IDF vectors and 5% over the KeyBERT model. Likewise, 6% recall rate over RAKE, 4% recall rate over YAKE, 5% recall rate over TF-IDF and 4% over the KeyBERT model. However, 8% f1 score has been received over RAKE, 6%

Table 4. KeyDistilTF % comparison on a teacher set with other unsupervised models of keyword extraction

	RAKE				YAKE				TF-IDF				KeyBERT			
	Sim	*P*	*R*	*F1*	*sim*	*P*	*R*	*F1*	*Sim*	*P*	*R*	*F1*	*sim*	*P*	*R*	*F1*
KeyDistilTf (KDTF)	13%	10%	6%	8%	11%	9%	4%	6%	14%	12%	5%	8%	7%%	5%	4%	4%

Source(s): Authors' own creation

Table 5. KeyDistilTF % comparison on a student set with other unsupervised models of keyword extraction

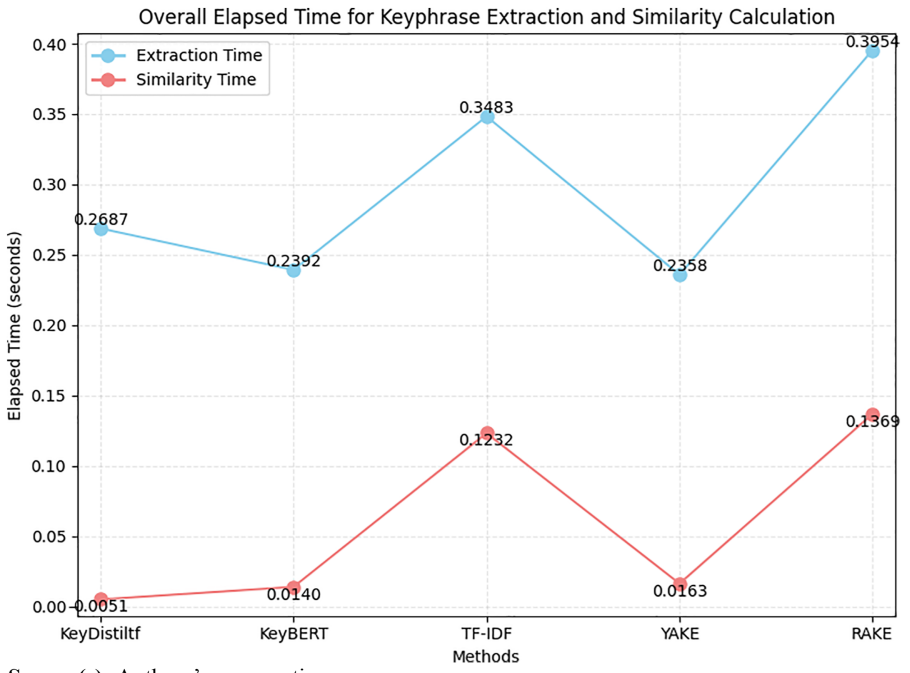
	RAKE				YAKE				TF-IDF				KeyBERT			
	sim	*P*	*R*	*F1*	*sim*	*P*	*R*	*F1*	*Sim*	*P*	*R*	*F1*	*sim*	*P*	*R*	*F1*
KeyDistilTF (KDTF)	11%	10%	10%	10%	11%	9%	8%	8%	14%	12%	9%	10%	7%	5%	8%	6%

Source(s): Authors' own creation

over YAKE, 8% over TF-IDF and 4% over the KeyBERT model. Both Tables 4 and 5 present detailed comparative analysis with key concept detection methods.

5.1.2 Elapsed time: key concept extraction and similarity extraction time. In many situations and applications, elapsed time extraction is crucial because it offers insightful information and facilitates effective decision-making. A thorough temporal analysis of events, procedures or activities is made possible by elapsed time extraction. It makes it possible to measure the amount of time that passes between various events, which aids in understanding the order and length of events. To maximize real-time performance, it is essential to keep track of the amount of time that passes between model deployment and inference. It makes it possible to locate inefficiencies and bottlenecks, which permits modifications to improve the responsiveness of machine learning applications.

Figure 7 presents the elapsed time of every model used in this study. As for key concept extraction time, the KeydisitilTF extracted the key concepts within 0.2687 seconds, and KeyBERT has a slightly better extraction time than our model, which is 0.2392 seconds. However, TF-IDF is a bit higher, which is 0.3483 seconds. Moreover, YAKE extraction is 0.2358, which is still better than our model, but RAKE extraction is higher than that of every model used. We consider this a limitation of our model for time extraction using key concept extraction. In addition, the similarity extraction time of our model is better than other methods used. Our model extracted the similarity score in 0.0051 seconds, whereas KeyBERT extracted it in 0.0140 seconds. Other methods such as TF-IDF extracted the similarity score in 0.1232 seconds, and YAKE performed slightly better than KeyBERT but was still lower than our model. The similarity time extraction is 0.0163 seconds, and the RAKE similarity score is 0.1369 seconds. The elapsed time used in this study can help to improve the model's performance over other methods. In addition to this, the summary of comparative findings has



Source(s): Authors' own creation

Figure 7. Example of overall elapsed time for key phrase extraction and similarity calculations

been presented in Table A5 in Supplementary material Appendix 1. Table A5 showcase the thorough representation of key differences between various models. It presents the key limitations of various models. KeydistilTF uses the context information with the help of distilBERT embeddings and can extract 2, 3 and 5 top key concepts from answers. The similarity value is 0.6 to 1.0, which is considered a moderate value. However, the gram size is settled 2, by 2, and the model can remove the stop words and remove the duplicate phrase inside the text. A few rare words were identified that were not included in human annotations. The model can identify the negative terms inside the text. The KeyBERT model also extracts relevant key phrases as it utilizes the BERT language model embeddings. The BERT model is a robust language model that is trained on Google Corpus, Wikipedia and Book Corpus. Extract the 2, 3 and 5 top key phrases from the answers. The similarity is also from moderate to high. The same size has been used for extracting the bigrams like 2, 2. Few stop words, as well as few unique words, are identified in the model. The mode has also extracted the reverse-order words from the data. However, the model is unable to detect the negative terms. In addition, the YAKE is also a good unsupervised model for key phrase extraction tasks [27]. The model doesn't use context information [28]. It is based on word co-occurrences and can extract the 2, 3 and 5 bigrams from the corpus. The model also uses low to moderate and average similarity values. Despite setting gram size 2, we have seen a few words that are unigram and tri grams. A few duplicated key terms were also identified. The model still needs to be optimized to detect the negative terms from the data. Likewise, TF-IDF is known as the term frequency and inverse document frequency method that detects the terms based on their co-occurrences [11]. The model can detect the top 2, 3 and 5 key concepts from answers. The similarity range is low, high and moderate. The model detects the key concepts with stop words, and a few duplications are also identified by the model. However, the model didn't detect the negative terms from the list of answers provided. Moreover, the RAKE model needs a lot of improvements to extract the unique phrases. The model is based on word occurrence terms. The model extracts the top 2 and 3 key terms from the answers. The similarity of the model ranges from low to high, with the model set to a bigram size of 2. It detects the key terms based on the co-occurrences. The majority of duplication has been identified inside the model; however, negative terms have not been detected by the model.

6. Conclusion

The model used in this study is a cutting-edge model that helps to extract the relevant key phrases from student and teacher answers. The model is further able to identify the negative key phrases. The student answers from the University of Texas dataset contain very little information on negative terms; hence, we have identified a few of them successfully, as the negative terms list has been developed inside the algorithm. KeyDistilTf utilizes the Tf-IDF vectors to calculate the similarity between ground truth key phrases and extracted model key phrases. Moreover, the model has been enhanced to utilize the distilBERT embeddings instead of BERT embeddings because this model can extract the unique phrases. The model can extract deep and core ideas from the CS domain. The model can be further utilized by the grading methods, and by extracting the negative and unique terms, the model can be used to identify sentiments from social sites or customer feedback with little enhancement. The implication of this model has a broader educational impact, which helps to improve the instructional strategies and teacher-student content. The model can also be further enhanced by implementing various other preprocessing techniques such as lemmatization, truncation, cleaning, stemming, tokenization and part-of-speech tagging. The model is still flexible to adopt such techniques to improve the student and teacher answers. The key limitation of the model is that it works in an unsupervised fashion. In the future, the model can be enhanced with fine-tuning techniques for the extraction of key concepts from corpora.

References

1. Nguyen HT, Duong PH, Cambria E. Learning short-text semantic similarity with word embeddings and external knowledge sources. Knowl Base Syst. 2019; 182. 104842. doi: 10.1016/j.knosys.2019.07.013.
2. Ni L, Bausch G, Benjamin R. Computer science teacher professional development and professional learning communities: a review of the research literature. Comput Sci Educ. 2023; 33(1): 29-60. doi: 10.1080/08993408.2021.1993666.
3. Haller S, Aldea A, Seifert C, Strisciuglio N. Survey on automated short answer grading with deep learning: from word embeddings to Transformers. 2022.
4. Hamed SK and Ab Aziz MJ. A question answering system on Holy Quran translation based on question expansion technique and neural network classification. J Comput Sci. 2016; 12(3): 169-77.
5. Han M, Zhang X, Yuan X, Jiang J, Yun W, Gao C. A survey on the techniques, applications, and performance of short text semantic similarity. Concurr Comput Pract Exp. 2021; 33(5): e5971. doi: 10.1002/cpe.5971.
6. Bachman LF, Carr N, Kamei G, Kim M, Pan MJ, Salvador C, Sawaki Y. A reliable approach to automatic assessment of short answer free responses. In: COLING 2002: The 17th International Conference on Computational Linguistics: Project Notes, 2; 2002. 1-4. doi: 10.3115/1071884.1071907.
7. Baruni JS, Sathiaseelan JGR. Keyphrase extraction from document using RAKE and TextRank algorithms. Int J Comput Sci Mob Comput. 2020; 9(9): 83-93.
8. Bhanbhro H., Kwang Hooi Y., Kusakunniran W., Amur ZH. A symbol recognition system for single-line diagrams developed using a deep-learning approach. Appl Sci. 2023; 13(15): 8816. doi: 10.3390/app13158816.
9. Alsalami AI. Challenges of short sentence writing encountered by first-year Saudi EFL undergraduate students. 2022.
10. Amur ZH, Hooi Y, Sodhar IN, Bhanbhro H, Dahri K. State-of-the art: short text semantic similarity (STSS) techniques in question answering systems (QAS). In: International Conference on Artificial Intelligence for Smart Community. Springer; 2022. p. 1033-44.
11. Amur ZH, Hooi YK, Soomro GM, Bhanbhro H, Karyem S, Sohu N. Unlocking the potential of keyword extraction: the need for access to high-quality datasets. Appl Sci. 2023; 13(12): 7228. doi: 10.3390/app13127228.
12. Galhardi LB, Brancher JD. Machine learning approach for automatic short answer grading: a systematic review. In: Ibero-american conference on artificial intelligence. Springer; 2018. 380-91.
13. Campos R, Mangaravite V, Pasquali A, Jorge AM, Nunes C, Jatowt A. Yake! collection-independent automatic keyword extractor. In: Advances in Information Retrieval: 40th European Conference on IR Research, ECIR 2018, Grenoble, France, March 26-29, 2018, Proceedings, 40. Springer; 2018. 806-10. doi: 10.1007/978-3-319-76941-7_80.
14. Sarwar TB, Noor NM, Miah MSU. Evaluating keyphrase extraction algorithms for finding similar news articles using lexical similarity calculation and semantic relatedness measurement by word embedding. PeerJ Comput Sci. 2022; 8: e1024.
15. Bhanbhro H, Hooi YK, Hassan Z. Modern approaches towards object detection of complex engineering drawings. In: 2022 International Conference on Digital Transformation and Intelligence (ICDI). IEEE; 2022. 1-6.
16. Priyanshu A., Vijay S. AdaptKeyBERT: an attention-based approach towards few-shot\& zero-shot domain adaptation of KeyBERT. 2022.
17. Amur ZH, Kwang Hooi Y, Bhanbhro H, Dahri K, Soomro GM. Short-text semantic similarity (STSS): techniques, challenges and future perspectives. Appl Sci. 2023; 13(6): 3911. doi: 10.3390/app13063911.
18. Khan MQ, Shahid A, Uddin MI, Roman M, Alharbi A, Alosaimi W, Almalki J, Alshahrani SM. Impact analysis of keyword extraction using contextual word embedding. PeerJ Comput Sci. 2022; 8: e967. doi: 10.7717/peerj-cs.967.

19. Burrows S, Gurevych I, Stein B. The eras and trends of automatic short answer grading. Int J Artif Intell Educ. 2015; 25(1): 60-117. doi: 10.1007/s40593-014-0026-8.

20. Witten IH, Paynter GW, Frank E, Gutwin C, Nevill-Manning CG. KEA: practical automatic keyphrase extraction. In: Proceedings of the fourth ACM conference on Digital libraries; 1999. 254-5.

21. Church KW. Word2Vec. Nat Lang Eng. 2017; 23(1): 155-62. doi: 10.1017/s1351324916000334.

22. Bin L, Jun L, Jian-Min Y, Qiao-Ming Z. Automated essay scoring using the KNN algorithm. In: 2008 International Conference on Computer Science and Software Engineering, 1. IEEE; 2008. 735-8. doi: 10.1109/csse.2008.623.

23. Cer D. Universal sentence encoder. 2018.

24. Mohler M, Bunescu R, Mihalcea R. Learning to grade short answer questions using semantic similarity measures and dependency graph alignments. In: Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies; 2011. 752-62.

25. Thushara M, Mownika T, Mangamuru R. A comparative study on different keyword extraction algorithms. In: 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC). IEEE; 2019. 969-73.

26. Zhuohao W, Dong W, Qing L. Keyword extraction from scientific research projects based on SRP-TF-IDF. Chin J Electron. 2021; 30(4): 652-7. doi: 10.1049/cje.2021.05.007.

27. Campos R, Mangaravite V, Pasquali A, Jorge A, Nunes C, Jatowt A. Yake! Keyword extraction from single documents using multiple local features. Inf Sci. 2020; 509: 257-89. doi: 10.1016/j.ins.2019.09.013.

28. Zhang Y, Tuo M, Yin Q, Qi L., Wang X, Liu T. Keywords extraction with deep neural network model. Neurocomputing. 2020; 383: 113-21. doi: 10.1016/j.neucom.2019.11.083.

Supplementary material
The supplementary material for this article can be found online.

Corresponding author
Zaira Hassan Amur can be contacted at: zaira_20001009@utp.edu.my