IRSTI 20.19.27 **https://doi.org/10.58805/kazutb.v.4.29-1085**

REVIEW OF REAL-WORLD APPLICATIONS OF NLP-BASED KEYWORD IDENTIFICATION

D. Ganiuly, D. Kaibassova

Astana IT University, Astana, Kazakhstan

Corresponding author: d.ganiuly@astanait.edu.kz

Keyword extraction is an important task in natural language processing, supporting summarization, indexing, content analysis, and information retrieval. Over time, a wide range of methods have been developed to automate the identification of key concepts in text. This study reviews the main families of keyword extraction techniques, including statistical methods, graph-based algorithms, embedding-based models, and approaches that use neural representations. In addition to summarizing existing research, the study evaluates several widely used tools, such as TF-IDF, YAKE, KeyBERT, KPMiner, TextRank, TopicRank, and MultipartiteRank, and examines how their strengths and limitations appear when applied to real educational materials. Special attention is given to the challenges that arise in multilingual and low-resource settings, particularly for languages with fewer annotated resources. The analysis highlights patterns in algorithm behavior, differences in semantic coverage, and factors that influence performance across syllabus formats and subject areas. The findings aim to help researchers and practitioners choose appropriate keyword extraction methods for educational, multilingual, and domain-specific applications.

Keywords: keyword extraction, low-resource languages, terms, keywords, algorithms, techniques, identification.

NLP НЕГІЗІНДЕГІ КІЛТІ СӨЗДЕРДІ АНЫҚТАУ НЕГІЗГІ СӨЗДЕРДІ ҚОЛДАНУЛАРДЫ ШОЛУ

Д. Ғаниұлы, Д. Қайбасова

Astana IT University, Астана, Қазақстан,
e-mail: d.ganiuly@astanait.edu.kz

Түйінді сөздерді шығару-бұл табиғи тілді өңдеудегі, қорытындылауды, индекстеуді, мазмұнды талдауды және ақпаратты іздеуді қолдаудағы маңызды міндет. Уақыт өте келе мәтіндегі негізгі ұғымдарды анықтауды автоматтандырудың көптеген әдістері жасалды. Бұл зерттеу кілт сөздерді алу әдістерінің негізгі топтарын, соның ішінде статистикалық әдістерді, графикалық алгоритмдерді, ендіруге негізделген модельдерді және нейрондық көріністерді пайдаланатын тәсілдерді қарастырады. Қолданыстағы зерттеулерді қорытындылаумен қатар, зерттеу TF-IDF, YAKE, KeyBERT, KPMiner, TextRank, TopicRank және MultipartiteRank сияқты кеңінен қолданылатын бірнеше құралдарды бағалайды және олардың күшті жақтары мен шектеулері нақты оқу материалдарына қолданылған кезде қалай көрінетінін зерттейді. Көптілді және ресурстары аз жағдайларда, әсіресе аннотацияланған ресурстары аз тілдер үшін туындайтын мәселелерге ерекше назар аударылады. Талдау алгоритмнің мінез-құлқындағы заңдылықтарды, семантикалық қамтудағы айырмашылықтарды және силлабус форматтары мен пәндік салалардағы өнімділікке әсер ететін факторларды көрсетеді. Нәтижелер зерттеушілер мен тәжірибешілерге білім беру, көптілді және доменге тән қолданбалар үшін кілт сөздерді шығарудың сәйкес әдістерін таңдауға көмектесуге бағытталған.

Түйін сөздер: кілт сөздерді шығару, ресурсы аз тілдер, терминдер, түйінді сөздер, алгоритмдер, тәсілдер, сәйкестендіру.

ОБЗОР РЕАЛЬНЫХ ПРИМЕНЕНИЙ ИДЕНТИФИКАЦИИ КЛЮЧЕВЫХ СЛОВ НА ОСНОВЕ NLP

Д. Ганиулы, Д. Кайбасова

Astana IT University, Астана, Казахстан,
e-mail: d.ganiuly@astanait.edu.kz

Извлечение ключевых слов является важной задачей в области обработки естественного языка, поскольку оно поддерживает процессы суммирования, индексирования, контент-анализа и информационного поиска. Со временем был разработан широкий спектр методов, позволяющих автоматически определять ключевые концепции в тексте. В данном исследовании рассмотрены основные группы методов извлечения ключевых слов, включая статистические подходы, алгоритмы на основе графов, модели на основе эмбеддингов, а также методы, использующие нейронные представления. Помимо анализа существующих исследований, работа оценивает ряд широко применяемых инструментов, таких как TF-IDF, YAKE, KeyBERT, KPMiner, TextRank, TopicRank и MultipartiteRank, и рассматривает, как их преимущества и ограничения проявляются при обработке реальных образовательных материалов. Особое внимание уделено проблемам, возникающим в многоязычной и низкоресурсной среде, особенно для языков с ограниченным количеством размеченных данных. Анализ выявляет закономерности поведения алгоритмов, различия в степени семантического охвата и факторы, влияющие на эффективность в зависимости от структуры учебных программ и дисциплин. Полученные результаты предназначены для того, чтобы помочь исследователям и практикам выбирать подходящие методы извлечения ключевых слов для образовательных, многоязычных и специализированных задач.

Ключевые слова: извлечение ключевых слов, языки с низкими ресурсами, термины, ключевые слова, алгоритмы, методы, идентификация.

Introduction. Keyword extraction plays an important role in natural language processing and is widely used to summarise documents, support information retrieval, and organise large collections of educational content. As universities continue to digitise teaching materials and online learning platforms expand, the demand for reliable and accurate keyword extraction methods grows accordingly. Well-chosen keywords make it easier for educators and students to identify essential concepts within a text, follow complex subjects, and connect related ideas across different resources.

Despite the long history of research in this area, traditional keyword extraction methods still face serious limitations. Many of them rely on surface-level statistics such as word frequency, co-occurrence, or simple graph structures. These techniques often fail to capture deeper semantic relationships between terms and do not fully reflect how meaning changes across different contexts. As a result, the extracted keywords may look correct from a technical standpoint but still fail to represent the actual content of educational texts, which frequently include abstract terminology, multi-word expressions, and domain-specific phrases.

Recent advances in transformer models and contextual embeddings have introduced new possibilities for capturing semantic information that traditional methods overlook. These models represent text in a way that preserves context and meaning, allowing them to recognize relationships that are invisible to frequency-based or graph-based approaches. A growing number of keyword extraction techniques now build on these models, yet existing studies vary widely in their datasets, evaluation methods, and goals. This makes it difficult to form a clear understanding of how different approaches compare, especially when the task involves educational materials with specialized vocabulary and varied structure.

The aim of this study is to examine several representative keyword extraction methods and evaluate their suitability for educational texts. The analysis focuses on how traditional statistical approaches behave in this setting, how contextual and transformer-based models address their shortcomings, and how different techniques perform when assessed using consistent criteria. Special attention is given to challenges specific to educational and multilingual materials, including

the presence of discipline-specific terminology and the diversity of writing styles. By bringing these perspectives together, the study seeks to provide clear guidance on selecting appropriate extraction methods for educational content and to outline areas where further improvement is needed.

Materials and Methods. To provide a comprehensive and methodologically grounded survey of keyword extraction algorithms, a systematic literature search was conducted across major databases, including IEEE Xplore, ACM Digital Library, SpringerLink, Scopus, and Google Scholar. Scopus offers a broad interdisciplinary coverage, making it particularly useful for identifying research trends in keyword extraction and machine learning. IEEE Xplore, on the other hand, is a valuable source for high-impact publications in computer science and artificial intelligence, ensuring access to the latest advancements in the field.

Scopus was used as the primary database due to its advanced search capabilities, including keyword-based queries with logical operators. An initial search using the query ”keyword identification AND machine learning” retrieved approximately 9,592 articles. To improve relevance, filters were applied to limit the selection to publications from the last five years (2021-2025) and to include only research and review articles. Given the importance of precise query formulation, a second search attempt was conducted using ”keyword extraction AND dataset,” which resulted in 6,607 articles specifically related to computer science. This refinement helped to identify studies that focus on dataset-driven approaches to keyword extraction.

To further refine the selection and focus on high-quality scientific publications, IEEE Xplore was used as a secondary source. An initial search with ”identifying keywords” for the 2020–2025 timeframe retrieved approximately 1,223 papers. A second iteration with paraphrased search terms yielded 190 research papers with a more specific emphasis on keyword extraction techniques. In a final search attempt aimed at narrowing the selection even further, 520 papers were identified from the last half-decade, ensuring that the survey included only the most relevant and recent contributions to the field.

The multi-stage filtering process was applied according to the Fig.1:

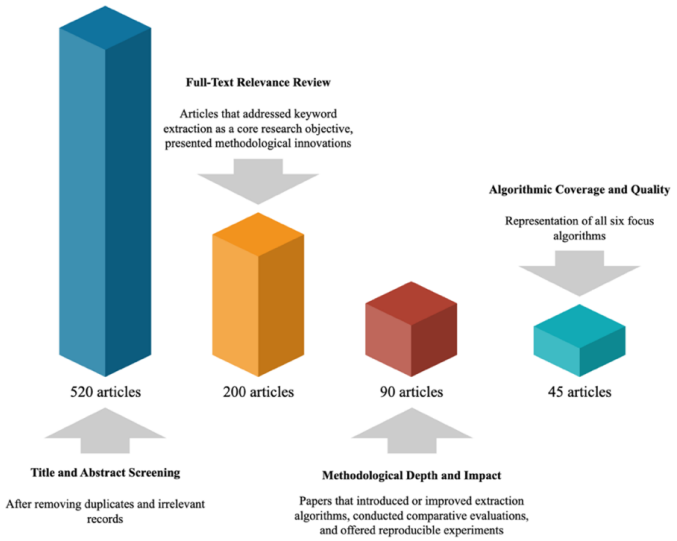


Fig.1 - The order of article selection steps

Statistical approach. The statistical approach to keyword extraction relies on analyzing patterns like word frequency and co-occurrence within a text to identify terms that are statistically significant or unusually prominent compared to the rest.

TF-IDF. One of the most well-known techniques

is Term Frequency-Inverse Document Frequency (TF-IDF), which assigns importance to words based on their occurrence in a document relative to their overall distribution in a dataset. While TF-IDF is computationally inexpensive and effective for structured text, it lacks semantic awareness and struggles with multi-word phrases [1]. TF-IDF score is calculated as follows:

$$\text{TF-IDF}(t, d, D) = TF(t, d) \cdot IDF(t, D)$$

where,

TF(t, d) - how often term *t* appears in document *d*, typically normalized by the total number of terms in *d*;

IDF(t, D) - a measure of how rare term *t* is across the entire document set *D*, reducing the weight of common terms.

YAKE. Other statistical methods include YAKE (Yet Another Keyword Extractor), which enhances keyword identification by considering statistical features such as word casing, position in text, and frequency distributions. Unlike TF-IDF, YAKE operates without requiring a large corpus, making it useful for small datasets. Additionally, RAKE (Rapid Automatic Keyword Extraction) identifies key phrases by analyzing word co-occurrences within a sliding window, enabling the detection of multi-word expressions [2,3]. Despite their effectiveness, statistical approaches often fail to capture the deeper contextual meaning of words, limiting their applicability in complex NLP tasks.

Word n-grams without punctuation and without a stop word at the beginning or end are chosen by YAKE as candidate keywords. To further filter out tokens that shouldn' t be part of a keyword (such as non-alphanumeric character sequences, etc.). The following formula is used to compute word and phrase score:

$$\text{Score}(w) = \frac{TF(w)}{\wp(w) \cdot RS(w) \cdot C_w(w) \cdot D(w)}$$

where,

TF(w) - Term Frequency – how often the word *w* appears in the document.

WP(w) - Word Position – a measure giving higher weight to words that appear earlier in the document;

RS(w) - Relatedness to Sentences – the proportion of sentences that contain the word *w*, reflecting its distribution;

Cw(w) - Word Case – the ratio of lowercase appearances of *w*, penalizing words frequently capitalized;

D(w) - Word Co-occurrence Degree – the degree of how often *w* co-occurs with other words in a local window, penalizing overly connected words.

KPMiner. KPMiner uses n-gram and is a revised form of TF-IDF. Prospective keyword selection, candidate keyword weight calculation, and keyword refinement are the three main phases of the suggested approach. KPMiner incorporated two new statistical characteristics after eliminating punctuation and stop words during the candidate keywords picking phase [4].

1) A word must appear at least n times in the document from which keywords are to be retrieved in order to qualify as a candidate keyword; this is known as the least acceptable seen frequency (lasf) factor.

2) If a term appears in a lengthy text after a certain cutoff threshold position, it will not be regarded as a keyword and will be filtered away. Since single keywords often receive greater marks due to their probable multiple existence, KPMiner employs an enhancement factor for compound keywords in the candidate keywords weight calculation step to counteract the bias of TF-IDF towards single

keywords. The top n keywords are returned after the last round of keyword refinement.

The following formula is applied to compute the frequency score:

$$\text{Score}(p) = \frac{F_p}{\log_2\left(1 + \text{first_pos}_p\right)} \cdot \text{len}_p^{\alpha}$$

where,

Fp = Frequency of phrase p – how many times the candidate phrase p appears in the document.

first_pos = First occurrence position – the position (word index) of the first appearance of p in the document, penalizing phrases that appear late;

len_p = Length of phrase p – number of words in the phrase (e.g., 2 for ”artificial intelligence”).

α = Length scaling factor – a constant (typically 1 or 2) to control the impact of phrase length.

Graph-Based Approaches. Graph-based keyword extraction techniques model text as a network of words or phrases, enabling the identification of important terms through ranking algorithms. Graph-based methods are advantageous because they do not require training data and can effectively capture multi-word phrases. Variants of TextRank, incorporate additional linguistic features (e.g., word position) to improve keyword relevance. However, these methods rely heavily on the structure of the input text, and their performance can degrade when applied to highly unstructured or domain-specific content. Once the relevant words are identified, the next step is to construct a graph where each word (or phrase) becomes a node [5,6]. Edges are created between nodes based on the co-occurrence of words within a defined window (e.g., a sentence or a fixed number of words). The weight of the edge reflects the strength of the relationship between the words, which can be measured by metrics like cosine similarity or simple co-occurrence.

TextRank. One of the most widely used graph-based methods is TextRank, which applies Google’s PageRank algorithm to textual data. In this approach, words or phrases are represented as nodes in a graph, and their relationships (e.g., co-occurrence within a sentence or paragraph) form weighted edges. The algorithm iteratively assigns importance scores to nodes, ultimately ranking the most relevant keywords.

$$S\left(w_i\right) = (1-d) + d * \sum_{w_j \in N\left(w_i\right)} \frac{S\left(w_j\right)}{C\left(w_j\right)}$$

where,

$S\left(w_i\right)$ - the score importance of word w_i;

d - The damping factor, typically set to 0.85;

$N\left(w_i\right)$ - The set of neighboring words for word w_i ; $C\left(w_j\right)$ - The number of outgoing edges from word w_j.

TopicRank. A graph-based KE technique called TopicRank extracts significant keywords from a manuscript by using its thematic representation. By grouping similar terms in the document into clusters, the approach first finds potential subjects. Then, it uses a variation of the TextRank method to determine which keywords in each cluster are most crucial. It creates a graph with nodes standing for the subjects that have been discovered and edges for how related the subject areas are to one another [7]. The correlation between the terms that comprise each subject is used to calculate the similarity between topics. To automatically aggregate related terms into themes, the Hierarchical Agglomerative Clustering (HAC) method is utilized. TopicRank is very helpful for locating significant subjects that individual keywords can miss [8]. After scoring with the following formula, TopicRank selects one representative phrase per topic (often the first occurrence or most frequent one).

$$\text{Score}\left(t_i\right) = \sum_{t_j \in N\left(t_i\right)} w_{ij}$$

where,

t_i - a topic node in the graph, representing a cluster of semantically related keyphrase candidates.

$N\left(t_i\right)$ - the set of neighboring topic nodes that co-occur with t_i in the text;

w_{ij} - the weight of the edge between topic t_i and neighbor t_j, typically based on proximity or co-occurrence distance within the document.

MultipartiteRank. The graph-based keyword extraction technique MultipartiteRank creates a multipartite graph with nodes representing unique texts as well as phrases that occur inside the documents [9]. A modified version of the PageRank algorithm, MultipartiteRank takes into account both the bipartite structure of the network and the relative relevance of each term in its corresponding page. Instead of individual papers, the end product is a collection of keywords that accurately reflect the content of the full document collection [10]. Because it has a stage where edge weights are adjusted to take location information into account, this approach is more intricate. Relevant keywords that occur first in the text are therefore given preference.

The following formulas are used to construct graphs. Let G = (V, E) be an undirected, complete graph with V as a set of vertices and E as a subset3 of V*V for the edges.

$$\text{Score}\left(c_i\right) = (1-\lambda) + \lambda \cdot \sum_{c_j \in N\left(c_i\right)} \frac{w_{ji} \cdot \text{Score}\left(c_j\right)}{\sum_{c_k \in N\left(c_j\right)} w_{jk}}$$

where,

c_i = candidate keyphrase node whose importance is being computed;

$N\left(c_i\right)$ = set of neighboring candidate keyphrases from other topics (no intra-topic edges):

w_{ji} = weight of the edge from candidate c_i to c_j, based on co-occurrence or positional proximity.

λ = damping factor (usually 0.85), balancing random walk with link-based recommendation;

Score(c_j) = current score of neighbor node c_j, recursively computed;

$\sum_{c_k \in N\left(c_j\right)} w_{jk}$ = normalization factor: sum of edge weights from $c_j$to its neighbors.

Embedding-Based Methods. To overcome the limitations of purely statistical techniques, researchers have developed embedding-based methods that leverage vector representations of words to enhance keyword extraction [11,12]. These methods utilize pre-trained word embeddings such as Word2Vec, FastText, and GloVe to capture semantic similarities between words. By representing words in a continuous vector space, embedding-based models improve keyword selection by considering contextual relationships rather than just frequency.

KeyBERT. A prominent example in this category is KeyBERT, which builds upon transformer-based embeddings such as BERT (Bidirectional Encoder Representations from Transformers). KeyBERT generates candidate keywords and ranks them based on cosine similarity between document embeddings and potential keywords, ensuring that extracted terms are contextually meaningful [13]. The ability of embedding-based models to preserve semantic relationships makes them particularly useful for tasks requiring high precision in keyword extraction. However, these methods often require significant computational resources and large pre-trained models, which may limit their real-time applicability.

$$\text{Score}(p) = \cos\left(\vec{p}, \vec{d}\right)$$

where, $\vec{p}$ = embedding vector of the candidate phrase p, typically obtained using a BERT model (or similar transformer);

$\vec{d}$ = embedding vector of the full document (or a central sentence representation), also computed using the same model;

$\cos\left(\vec{p}, \vec{d}\right)$ = cosine similarity between the phrase and document embeddings, producing a value between -1 and 1, where values closer to 1 indicate higher semantic similarity.

To provide a clearer overview of how the examined algorithms differ in their underlying principles and practical behavior, we summarise the methods used in this study in a comparative form. This overview helps to highlight the contrast between statistical, graph-based, and embedding-based approaches before moving to the discussion of transformer-based techniques. The comparison of keyword extraction approaches employed in this research is presented in Table 1.

Table 1 - Overview of Keyword Extraction Approaches

Approach Type	Algorithms Used	Strengths	Limitations	Suitability for Educational Texts
Statistical	TF-IDF, YAKE, KPMiner, RAKE	Fast; interpretable; no training needed; effective on structured text	No semantic knowledge; may miss multi-word concepts; unstable for abstract terminology	Works moderately well on short or well-formatted syllabuses; weaker for concept-heavy subjects
Graph-based	TextRank, TopicRank, MultipartiteRank	Detects multi-word expressions; structured keyword ranking; unsupervised	Dependent on text cohesion; limited semantic depth; sensitive to preprocessing	Useful for longer syllabuses or materials with strong internal structure
Embedding-based	BERT/SBERT embeddings, KeyBERT	Captures semantic meaning; strong performance on abstract academic terms; handles rephrasing well	Requires more computation; quality relies on transformer model	Provides most consistent and contextually accurate results on educational content

Experiments. To evaluate the effectiveness of different keyword extraction algorithms for educational content generation, we conducted a series of experiments on a curated dataset comprising 255 course syllabuses from various disciplines, including computer science, business, psychology, and engineering. These syllabuses ranged from short outlines to detailed week-by-week breakdowns and were sourced from publicly available higher education platforms.

Each syllabus was preprocessed using standard NLP techniques: lowercasing, tokenization, stopword removal, and sentence segmentation. For algorithms that required part-of-speech tagging (e.g., MultipartiteRank, TopicRank), we used spaCy' s English language model. The primary goal was to extract representative keywords or keyphrases that would serve as seeds for downstream educational content generation [14,15].

Preprocessing. The importance of the preprocessing step was illustrated by results obtained in previous studies related to keyword extraction. Document preprocessing includes cleaning and transforming the text into a processable format for keyword spotting instruments [16].

The requirements to follow the pre-processing stage include the elimination of redundant fragments from text, such as stop words, special characters, suffices, and punctuation marks that can directly or indirectly influence the outcome [17]. In order to make it simpler for the approaches to find and extract important terms, document preprocessing can also include methods like stemming and lemmatization to reduce words to their root form. Adequate document preprocessing minimizes noise and guarantees consistent input data, which can enhance the performance of the KE techniques.

It was revealed that utilizing data pretreatment procedures boosts the performance of RAKE on the SemEval2017 dataset. The F1-score increases by nearly 10 percent if the data is preprocessed. Additionally, the preprocessed data results in an almost 22 percent reduction in execution time. YAKE further examines the effects of data preparation and assesses them using the SemEval2010 dataset.

Among all found articles from various data sources of publications, nearly 90 percent of all studies included or at least mentioned preprocessing as an important stage in their experiments. This is due to the fact that written information should correspond to a format that is processable by NLP instruments for keyword extraction. Therefore, the majority of previous research included data preprocessing to improve the effectiveness and increase the accuracy of results.

In the scope of this study, syllabuses on university courses will be processed by several techniques to extract keywords. Since course syllabuses are created based on a single template, there is information that can be processed erroneously, which will lead to inaccurate results. For this reason, a custom list of stopwords will be used in the preprocessing step to avoid keyword spotting of irrelevant data. This list will be complemented to decrease the probability of processing redundant words and phrases and leave space only for necessary information.

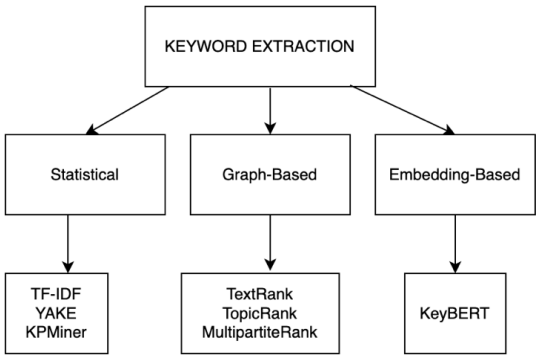


Fig.2 - Classification of keyword extraction algorithms

Applications and use cases. Scientists used TF-IDF, TextRank, SingleRank, ExpandRank, RAKE, CO, Semantic Relationship-based methods, and Combination methods to extract keywords in the scope of their experiments to process information from the Hulth 2003 dataset. This study also concentrated on comparing various import terms spotting, so numerous methods were included. Each document in the Hulth2003 dataset, which is a document summary dataset from the Inspec physical and engineering literature database, has two sets of keywords assigned to it: the unregulated keywords are openly assigned by the editors, and the manually controlled keywords

are displayed in the Inspec thesaurus instead of the document.Another dataset that was common between studies is Krapivin2009.

Fig.2 illustrates classifications of different algorithms used for keyword extractions. Three major classes of keyword extraction algorithms are found: statistical, graph-based, and embedding-based. While statistical algorithms consist of TF-IDF, YAKE, and KPMiner, instruments such as TextRank, TopicRank, and MultipartiteRank were classified as graph-based. Embedding-based instrument: KeyBERT was also commonly used by previous studies.

Comparative analysis. According to Table 1, the applicability and behavior of the examined keyword extraction algorithms vary considerably depending on their underlying approach. Traditional statistical methods such as TF-IDF provide very fast results but lack semantic awareness, which limits their ability to identify abstract or domain-specific concepts. Heuristic multi-feature methods like YAKE improve robustness without relying on external corpora, making them effective for structured and domain-independent texts. Graph-based algorithms, including TextRank, TopicRank, and MultipartiteRank, offer a balanced solution by capturing co-occurrence patterns and multi-word expressions, although their performance depends on text cohesion. Embedding-based approaches such as KeyBERT deliver the most contextually accurate keywords by leveraging transformer-based semantic representations, making them particularly suitable for educational materials that contain concept-heavy terminology.

Table 2 - Qualitative Performance of Methods on Educational Content

Algorithm	Relevance of Extracted Keywords	Ability to Capture Semantic Meaning	Stability Across Different Subjects	Notes
TF-IDF	Low–medium	Low	Medium	Extracts frequent terms rather than conceptual keywords; often misses multi-word expressions
YAKE	Medium	Low–medium	High	Performs reliably on clean and structured syllabuses; multi-feature scoring helps reduce noise
KPMiner	Medium	Low–medium	Medium	Better for longer documents due to frequency and position adjustments; still lacks semantic awareness
RAKE	Medium	Low	Medium	Good for detecting simple multi-word phrases, but performance varies based on text formatting
TextRank	Medium	Low–medium	Medium	Works well with cohesive text, but lacks contextual understanding
TopicRank	Medium	Medium	Medium	Topic clustering improves conceptual grouping but depends heavily on preprocessing quality
MultipartiteRank	Medium–high	Medium	Medium	Topic-aware graph helps detect diverse keyphrases; sensitive to sentence segmentation
KeyBERT	High	High	High	Produces the most contextually meaningful keywords; robust across different academic domains

Since the algorithms differ in their ability to capture context, semantic meaning, and domain-specific terminology, their qualitative behavior shows noticeable variation across subjects. A summary of these tendencies is provided in Table 2, offering a high-level understanding of the relative strengths and weaknesses of each technique.

Results and Discussion. The analysis of 255 course syllabuses showed noticeable differences in how each keyword extraction algorithm behaves. Because the syllabuses varied in length, structure, and writing style, the methods performed differently depending on the discipline and the level of conceptual detail.

Algorithm Behavior in Educational Contexts. KeyBERT delivered the strongest overall results. Since it relies on contextual embeddings, it was able to identify terms that were central to a course even when they appeared only once. For example, in machine learning syllabuses, KeyBERT detected keywords such as overfitting, feature engineering, and gradient descent simply because they aligned well with the main theme of the document [18]. This ability to capture meaning rather than frequency made it especially effective in subjects where important concepts are not repeated many times but are essential for understanding the course.

KPMiner performed well on syllabuses that were written in a structured and repetitive format. In courses such as databases or operating systems, where key terms naturally appear early and are repeated across weekly modules, KPMiner consistently identified fundamental concepts like SQL, normalization, and paging. This made it particularly suitable for STEM syllabuses that follow a predictable instructional pattern.

YAKE, although purely statistical, showed good adaptability. Its multi-feature scoring allowed it to extract relevant phrases from shorter and less organized syllabuses. However, because it does not consider semantic meaning, it sometimes selected general or administrative words such as assessment, outline, or unit, which reduced its usefulness for generating educational content.

Redundancy, Diversity, and Pedagogical Coherence. Graph-based methods such as TextRank and TopicRank often produced redundant results. For example, they might output both machine learning and learning as separate keywords, which creates noise when these terms are used to build lesson plans or learning objectives. MultipartiteRank handled this problem better by separating candidate terms into distinct topics, which reduced redundancy and increased variety. Although its accuracy was lower than that of KeyBERT and KPMiner, MultipartiteRank offered broader topic coverage-a useful property when the goal is to outline course themes rather than extract a small set of precise concepts.

KeyBERT and MultipartiteRank also produced more coherent keyword sets. When grouped into clusters, the selected terms reflected the conceptual structure of the syllabuses and aligned with typical learning outcomes used in curriculum planning [19]. This coherence makes these models particularly helpful for generating learning materials such as quizzes, concept maps, or weekly module themes.

Domain Sensitivity. The results also varied across academic fields. In humanities and social sciences, where terminology depends heavily on context and is often less repetitive, KeyBERT had a clear advantage thanks to its semantic representation. In contrast, KPMiner and YAKE worked better in STEM subjects, where technical terms tend to appear consistently throughout the syllabus. Graph-based methods performed moderately across all domains but lacked the depth needed to capture more abstract concepts [20].

Accuracy and efficiency. Three algorithms stood out for practical applications:

1. KeyBERT provided the most accurate and pedagogically meaningful keywords. Its semantic understanding makes it ideal for generating study guides, concept maps, and learning objectives.

2. KPMiner was reliable for structured syllabuses and works well when the goal is to extract core terms for summaries or review sheets.

3. MultipartiteRank offered strong topic diversity, which is useful for designing course outlines or identifying broader thematic areas.

Traditional methods such as TF-IDF and TextRank were less effective for educational purposes. TF-IDF tended to select frequent but less meaningful words, while TextRank often lacked conceptual focus. As a result, both methods are less suitable for tasks related to automated curriculum development.

The redundancy analysis shown in Fig.3 highlights these differences. KeyBERT produced the lowest redundancy (18%), meaning it generated the most distinct and semantically rich keyword sets. TF-IDF (42%) and TextRank (38%) showed high redundancy, often repeating variations of the same idea. These results reinforce the conclusion that semantic models provide more useful and diverse keywords for educational applications.

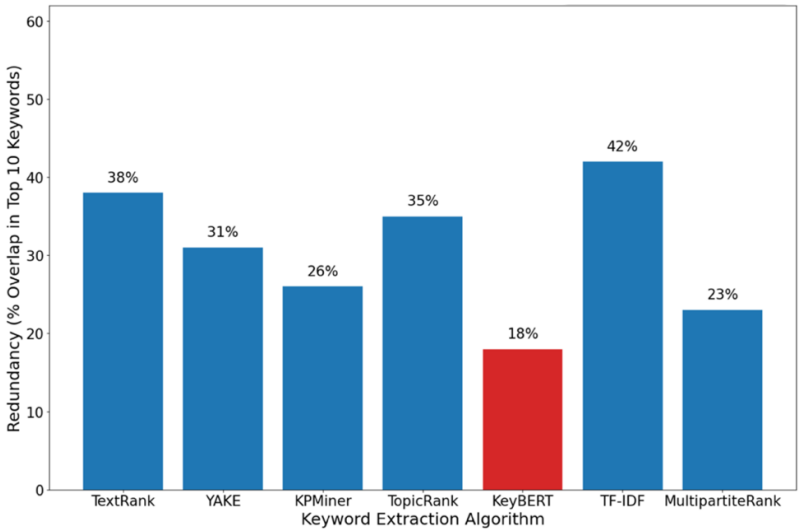


Fig.3 - Keyword Redundancy Across Extraction Algorithms

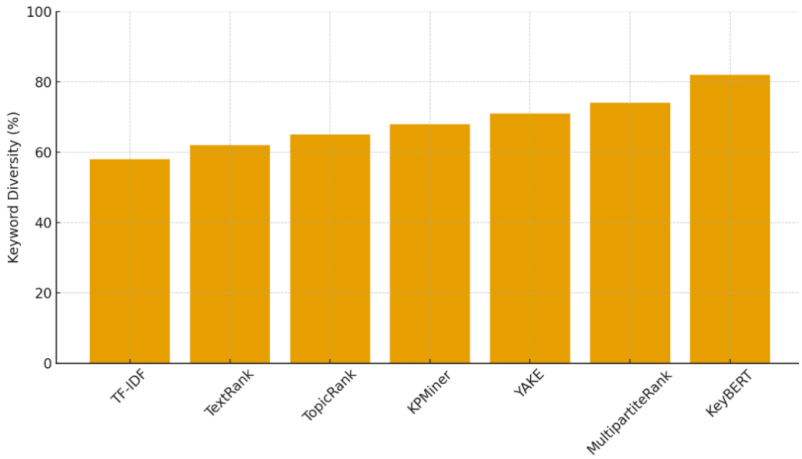


Fig.4 - Keyword Diversity Across Extraction Algorithms

To complement the redundancy analysis, we also measured the diversity of extracted keywords, defined as the proportion of distinct terms within each algorithm's top-10 output. As shown in Fig.4, diversity levels vary substantially across methods. KeyBERT achieved the highest diversity (82%), reflecting its ability to capture semantically distinct concepts even when key terms appear infrequently in the syllabus. MultipartiteRank and YAKE also performed well (74% and 71%, respectively), indicating strong topic coverage without excessive repetition. In contrast, older statistical and graph-based approaches such as TF-IDF (58%) and TextRank (62%) produced more overlapping or closely related terms, limiting their usefulness for downstream educational tasks that require broad conceptual representation. These results reinforce the conclusion that semantic and topic-sensitive models provide richer keyword sets that better reflect the structure and thematic variety of university syllabuses.

Conclusion. This study examined the performance of several keyword extraction algorithms on a collection of 255 course syllabuses and demonstrated clear differences in their ability to identify meaningful and pedagogically relevant terms. The results confirm that semantic modeling plays a central role in educational text processing: transformer-based approaches,

particularly KeyBERT, consistently produced the most accurate, diverse, and conceptually coherent keyword sets. This is significant for applications in educational technology, where extracted keywords often serve as the basis for generating study materials, aligning learning outcomes, and supporting automated curriculum design.

The findings also provide practical recommendations for selecting appropriate methods. KeyBERT is the most effective choice when the goal is to extract concept-level information and handle abstract terminology across various disciplines. KPMiner offers reliable precision for structured STEM syllabuses written in a repetitive week-by-week format. MultipartiteRank is valuable for tasks that require broad thematic coverage, such as constructing course outlines or identifying clusters of related topics. These insights can assist educators, developers, and researchers in choosing suitable tools based on the structure and purpose of their educational content.

Future work may explore explore adaptive keyword extraction models that dynamically adjust to different syllabus structures and subject domains to enhance consistency across diverse educational materials.

References

1. Nadim M., Akopian D., Matamoros A. A Comparative Assessment of Unsupervised Keyword Extraction Tools//IEEE Access. -2023. -Vol.11.-P.144778-144798. DOI 10.1109/ACCESS.2023.3344032.

2. Khan M.Q., et al. Impact Analysis of Keyword Extraction Using Contextual Word Embedding//PeerJ Computer Science. -2022. - Vol.8: e967. DOI 10.7717/peerj-cs.967.

3. Zhang Y., Tuo M., Yin Q., Qi L., Wang X., Liu T. Keywords Extraction with Deep Neural Network Model // Neurocomputing. - 2020. -Vol.383 (C). -P.113-121. DOI 10.1016/j.neucom.2019.11.08.

4. Guo W., Wang Z., Han F. Multifeature Fusion Keyword Extraction Algorithm Based on TextRank// IEEE Access. -2022. -Vol.10. -P. 71805-71813. DOI 10.1109/ACCESS.2022.3188861

5. Mao X., Huang S., Li R., Shen L. Automatic Keyword Extraction Based on Co-Occurrence and Semantic Relationships Between Words//IEEE Access. -2020. -Vol.8.-P.117528-117538. DOI 10.1109/ACCESS.2020.3004628.

6. Papagiannopoulou E.,Tsoumakas G. A Review of Keyphrase Extraction//Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. -2019. -Vol.10(2): e1339. DOI 10.1002/widm.1339.

7. Chen L.C. An Extended TF-IDF Method for Improving Keyword Extraction in Traditional Corpus-Based Research: An Example of a Climate Change Corpus//Data & Knowledge Engineering. -2024. -Vol.153: 102322. DOI 10.1016/j.datak.2024.102322.

8. Zhang L., Li Y., Li Q. A Graph-Based Keyword Extraction Method for Academic Literature Knowledge Graph Construction//Mathematics. -2024. -Vol.12(9): 1349. DOI 10.3390/math12091349.

9. Song M., Feng Y., Jing L. A Survey on Recent Advances in Keyphrase Extraction From Pre-Trained Language Models // Findings of the Association for Computational Linguistics (EACL). -2023. -P.2153-2164. DOI 10.18653/v1/2023.findings-eacl.161.

10. Gupta A., Chadha A., Tewari V. A Natural Language Processing Model on BERT and YAKE Technique for Keyword Extraction on Sustainability Reports // IEEE Access. -2024. -Vol.12. -P.7942-7951. DOI 10.1109/ACCESS.2024.3352742.

11. Sarwar T.B., Noor N.M., Miah M.S. Evaluating Keyphrase Extraction Algorithms Using Lexical Similarity and Semantic Relatedness // PeerJ Computer Science. -2022. -Vol.8: e1024. DOI 10.7717/peerj-cs.1024.

12. Amur Z.H., Hooi Y.K., Soomro G.M., et al. Unlocking the Potential of Keyword Extraction: The Need for Access to High-Quality Datasets//Applied Sciences. -2023. -Vol.13(12): 7228. DOI 10.3390/app13127228.

13. Ahmed U., Alexopoulos C., Piangerelli M., Polini A. BRYT: Automated Keyword Extraction for Open Datasets//Intelligent Systems with Applications. -2024. -Vol.23: 200421. DOI 10.1016/j.iswa.2024.200421.

14. Nomoto T. Keyword Extraction: A Modern Perspective//SN Computer Science. -2023. -Vol.4(1): 92. DOI 10.1007/s42979-022-01481-7.

15. Liu Q., Hui Y., Liu S., Ji Y. Y-Rank: A Multi-Feature-Based Keyphrase Extraction Method for Short Text//Applied Sciences. -2024. -Vol.14(6): 2510. DOI 10.3390/app14062510.

16. Bai R., Liu F., Zhuang X., Yan Y. MICRank: Multi-Information Interconstrained Keyphrase Extraction//Expert Systems with Applications. -2024. -Vol.249: 123744. DOI 10.1016/j.eswa.2024.123744.

17. Bennani-Smires K., Musat C., Hossmann A., Baeriswyl M., Jaggi M. Simple Unsupervised Keyphrase Extraction Using Sentence Embeddings // Proceedings of the Workshop on Automated Knowledge Base Construction (AKBC). -2018. -Vol. arXiv:1801.04470. DOI 10.48550/arXiv.1801.04470.

18. Campos R., Mangaravite V., Pasquali A., Jorge A.M., Nunes C., Jatowt A. YAKE! Collection-Independent Automatic Keyword Extractor // European Conference on Information Retrieval (ECIR). -2018. -P.806–810. DOI 10.1007/978-3-319-76941-7_80.

19. Kaibassova D., Nurtay M. The Comparative Analysis of Machine Learning Models for Quality Assessment of Textual Academic Works // Proc. Int. Conf. On Smart Information Systems and Technologies (Sist). -2022. -P.1-4. DOI 10.1109/sist54437.2022.9945714.

20. Kaibassova D., Ashimbekova A. Data Acquisition Model for an Intelligent System Integrating the Creation of Educational Programs and Professional Standards // Wseas Transactions On Computer Research. -2024. -Vol.12. -P.443 - 449. DOI 10.37394/232018.2024.12.43.

Information about the authors

Ganiuly D. - Master of Technical Sciences, Astana IT University, Astana, Kazakhstan, e-mail: d.ganiuly@astanait.edu.kz;

Kaibassova D. - PhD., Ass. Professor, Astana IT University, Astana, Kazakhstan, e-mail: d.kaibasova@astanait.edu.kz.

Информация об авторах

Ганиулы Д. - магистр технических наук, Astana IT University, Астана, Казахстан, e-mail: d.ganiuly@astanait.edu.kz;

Кайбасова Д. - доктор PhD, Acc. профессор, Astana IT University, Астана, Казахстан, e-mail: d.kaibasova@astanait.edu.kz.