Intelligent Systems with Applications 23 (2024) 200421

Contents lists available at ScienceDirect

Intelligent Systems with Applications

journal homepage: www.journals.elsevier.com/intelligent-systems-with-applications

BRYT: Automated keyword extraction for open datasets

Umair Ahmed [a,*], Charalampos Alexopoulos [b], Marco Piangerelli [a], Andrea Polini [a]

[a] Computer Science Division, University of Camerino, Via Madonna Delle Carceri, 7, Camerino, 62032, Macerata, Italy
[b] Department of Information and Communication Systems Engineering (ICSE), University of the Aegean, Karlovasi, 83200, Samos, Greece

ARTICLE INFO

Dataset link: https://github.com/umairahmedq/BRYT-AKE

Keywords:
Open data
Metadata
European data portal
Keyword extraction
Large language models
ChatGPT

ABSTRACT

In today's information-driven world, open data is crucial in making valuable structured data freely accessible to the public. However, the absence of quality metadata often hinders the findability and representation of this data. In this study we specifically focus on keywords, proposing a strategy for their automatic generation. In particular, we employed five existing keyword extraction methodologies (BERT, RAKE, YAKE, TEXTRANK, and ChatGPT) and proposed a novel hybrid methodology, named BRYT (read as bright). Our evaluation of these algorithms was conducted using Gestalt String Matching and Jaccard Similarity techniques. We validated our study using a selection of datasets from the EU data portal, specifically choosing those that exhibited potentially high-quality metadata. This included datasets that contained a substantial number of keywords and had comprehensive, relevant metadata. The results showed that 69.1% of the dataset keywords majorly matched (more than 50% or 5 keywords), 24.7% minorly matched (up to 50% or 5 keywords), and 6.2% did not match. The proposed hybrid model, BRYT, outperformed other algorithms in the major matches, while ChatGPT was a close second. YAKE outperformed the others in minor matches, and ChatGPT was again a close second. The evaluations concluded that BRYT consistently extracted more representative keywords in major matches, highlighting its effectiveness in improving findability. This study sets up a favorable field for further representative metadata extraction and population, making the data more findable, discoverable, and accessible.

1. Introduction

Motivations. In the information age, there has been substantial growth in the volume and circulation of data. As a result, numerous organizations and stakeholders have come to appreciate the significance of open data in fostering transparency and generating added value. To realize these benefits, they aim to make their data freely available, which demands rapid and efficient ways of locating the information (Kubler, Robert, Neumaier, Umbrich, & Le Traon, 2018). In 2020, Europe's data economy was estimated to be worth around 739 billion. Opening up this data can lead to various benefits, such as enhancing transparency, accountability and fostering greater participation. Furthermore, it encourages stakeholders to extract more value from the data and promotes increased accountability and transparency to refine processes (Lnenicka & Nikiforova, 2021; Loenen et al., 2021).

According to the Open Knowledge Foundation, data is considered open if it can be "freely accessed, used, modified, and shared by anyone for any purpose" (OpenKnowledge, 2023). Most of the data portals provide mechanisms to host open data sets and make them available, but they severely lack in terms of findability. Users with less than adequate technical skills struggle with findability, and eventually, it affects the accessibility and usability of open data sets that are already available.

Findability can be enhanced through intuitive search engines and intelligent recommendation systems for laymen users (Brickley, Burgess, & Noy, 2019). Quality metadata plays a significant role in the effectiveness of search engines, making it essential for improving findability (Firoozeh, Nazarenko, Alizon, & Daille, 2020). This serves as a compelling motivation for our research, emphasizing the need to potentially enhance data discoverability by extracting quality metadata unleashing the full potential of open data.

Innovations. We examined 15 notable data portals in this study and discovered they lacked intelligent and intuitive search mechanisms. The analysis of these data portals is presented in Table 1. None of the portals featured a recommender system, and most of them relied on basic keyword searches, with some offering autocomplete suggestion mechanisms related to dataset titles. The search engines typically matched metadata elements such as titles, categories, or keywords. The majority of searches were not intuitive, mainly due to the lack of representative metadata for the given datasets. Most search mechanisms concentrated

* Corresponding author.
E-mail address: umair.ahmed@unicam.it (U. Ahmed).

https://doi.org/10.1016/j.iswa.2024.200421
Received 29 November 2023; Received in revised form 1 July 2024; Accepted 22 July 2024
Available online 24 July 2024

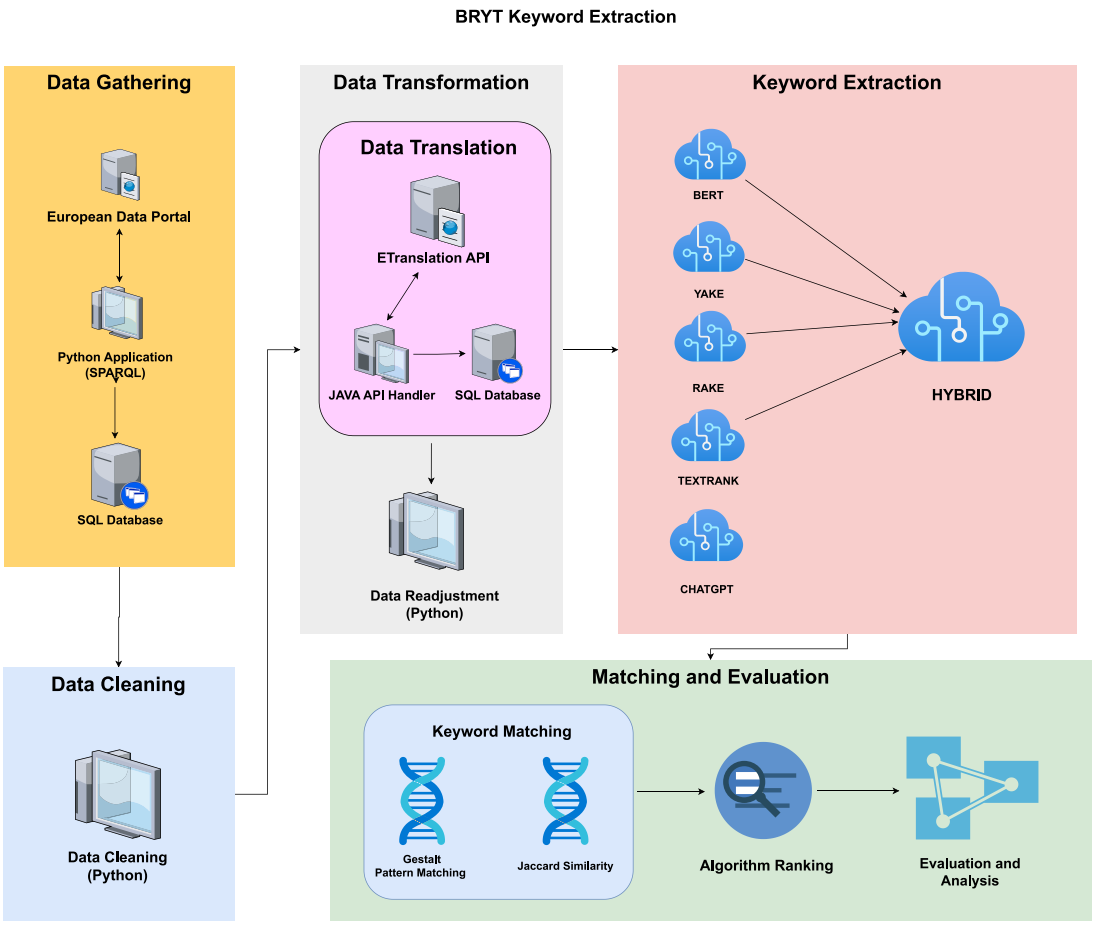

Fig. 1. BRYT Automated Keyword Extraction.

on metadata, and a large number of datasets lacked sufficient metadata to be easily findable. Although not all metadata elements contribute to findability, titles, keywords, and categories significantly enhance it. In this research, we emphasize keywords, given their strong association with search engines and their ability to concisely represent the document (Merrouni, Frikh, & Ouhbi, 2016). This study explores methods to automate the keyword extraction process, aiming to recommend keywords to publishers for new datasets and populate missing keywords in existing datasets.

Contributions. We selected the European Data Portal (EDP) as our case study for this study. The EDP is regarded as one of Europe's most extensive data portals, aggregating data from 73 other data portals in the European area (Correa, Zander, & Da Silva, 2018). According to the findability index published by EDP, keywords and categories of datasets contribute to 60% of the findability score (EU, 2023). The details of the index are illustrated in Fig. 1. Focusing on the findability score, we analyzed 37,306 datasets in the portal. Among them, 11,958 (32%) had three or fewer keywords, while the remaining 25,348 (68%) datasets had more than three. However, a significant number of these datasets exhibited non-representative traits and flaws. The flaws observed include redundant keywords, column names used as keywords, other exact metadata used as keywords, title words, and numerical data as keywords. These findings indicate that many publishers may not have paid sufficient attention to the keywords, either entering them hastily or skipping them entirely. It rendered the findability of those datasets to be difficult.

As manual keyword provision proved insufficient, we proposed generating and recommending automated representative keywords to make data more findable. The most comprehensive metadata attribute for a dataset is its description, so we used this to extract keywords for the datasets. For our experiment, we collected 69,423 metadata records from 13 different themes within the EDP. After gathering the data, we cleaned it by removing duplicates, empty entries, garbage characters, non-representative data, and items that could not be translated. Following the cleaning process, the metadata records were reduced to 1,393. We established that these cleaned datasets contained representative descriptions and keywords, which form the foundation of our experimental and evaluative approach. The objective of our research was to produce keywords that were as representative as the original keywords.

In our study, each algorithm we used, including BERT, RAKE, YAKE, TEXTRANK, and ChatGPT, generated between 2 and 10 keywords from the dataset metadata. Our proposed hybrid methodology, BRYT, combined the results of the first four algorithms and employed cosine similarity to rescore and select the top 10 keywords from the generated list. To evaluate the effectiveness of our approach, we used Gestalt pattern matching and Jaccard similarity to score the matching with the original keywords. Our results showed that 69.1% of keywords matched majorly (more than 50% or 5 keywords) while 24.7% matched minorly (less than or equal to 50% or 5). Moreover, BRYT outperformed the other algorithms in major matches (27.1%), while YAKE had better results in minor matches (35.5%). Our study established that the proposed hybrid methodology was superior to other algorithms, particularly in major matches.

Paper organization. In the subsequent sections of this article, we delve into the intricacies of our study, beginning with the **Background** to establish the context and significance of our research. We then detail our **Methodologies**, explaining the processes and techniques we have employed. Our **Results** section presents our findings, while the **Related Research** section situates our work within the broader academic

research. The **Discussion and Future Works** section discusses the implications of our findings and suggests directions for further study. Finally, the **Conclusion** section encapsulates the study's key findings and its contributions to the field of open data and automated metadata extraction.

2. Problem Statement

In the context of an open data platform, represented as $\mathcal{D}$, we focus on enhancing the findability of the datasets $\{d_1, d_2, \dots, d_n\}$ uploaded on it. Given that mechanisms for dataset retrieval are generally based on the manipulation of associated keywords, we focused our research on the definition and assessment of approaches able to automatically generate "good" sets of keywords from a data set description. As the objective of our work is to enrich the list of keywords associated to a datasets, by generating an appropriate set of keywords, we need to check and compare that the algorithms we propose are indeed effective in such a task. For doing this we selected, at first, a set of datasets for which we manually checked the quality of the associated keywords, according to the linked descriptions and topics. Successively we evaluated the capability of the different algorithms to automatically generate words in line with those manually assigned by the operators, and we could derive a ranking including the different approaches. The assessment was performed according to the following mathematical formulation.

Given a dataset $d \in \mathcal{D}$ we represent the set of extracted keywords as K_d^e and the set of reference keywords, the ones assigned by the operator, as K_d^r. Given two keywords $k_e \in K_d^e$ and $k_r \in K_d^r$ we compute the alignment $m(k_e, k_r)$ on the base of the Gestalt ($G(k_e, k_r)$) and Jaccard ($J(k_e, k_r)$) scores, taking the maximum among the two values, as summarized in the following formula:

$$m(k_e, k_r) = max\{G(k_e, k_r), J(k_e, k_r)\} \tag{1}$$

The function M is then used to measure the similarity of an extracted keyword with respect to the set of keyword assigned by the operator. In particular we consider a keyword $k \in K_d^e$ to be a good match when the following function returns the value 1:

$$M(k, K_d^r) = \begin{cases} 1, & \text{if } \max\limits_{c \in K_d^r} \{m(k, c)\} \geq 0.6, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

So, an extracted keyword is considered somehow good if it is already included in K_d^r, admitting some tolerance possibly due to misspelled words or the use of more generic/specific terms. Given the definition of M we can use it to define if a set of extracted keyword can be considered a "Major match", a "Minor match" or a non matching set. Im particular such decision is derived on the base of the following formula:

$$\text{MajorMatch}(K_d^e, K_d^r) = \begin{cases} \text{True}, & \text{if } \sum\limits_{k \in K_d^e} M(k, K_d^r) > \min(\frac{1}{2} \cdot |K_d^r|, 5), \\ \text{False}, & \text{otherwise.} \end{cases} \tag{3}$$

$$\text{MinorMatch}(K_d^e, K_d^r) = \begin{cases} \text{True}, & \text{if } 0 < \sum\limits_{k \in K_d^e} M(k, K_d^r) \leq \min(\frac{1}{2} \cdot |K_d^r|, 5), \\ \text{False}, & \text{otherwise.} \end{cases} \tag{4}$$

To be considered a "Major match" a set of extracted keywords as to count at least for a half of the keywords assigned by the operator. It is worth noting that the functions we defined could lead to somehow unreliable results, if applied to arbitrary set of keywords. This is due to the intrinsic behavior of the two metrics we used, as they could over estimate a match as they work on the set of characters or on the sub-strings appearing in the keywords (see Section 3 for detail on the two metrics). Nonetheless, as the datasets selected for the experiment have carefully selected keywords, a "Major match" unlikely emerges when K_d^e is not representative of the dataset. In such a sense it is also worth noting that the metrics have a good behavior when you want to consider misspelled or derivative words.

3. Background

This section focuses on establishing the background necessary to understand the problem statement, its implications, and the methodologies adopted to solve it.

3.1. Similarity measures

This section discusses different measures that help measure the relevancy or distance between texts. In our case, the particular focus was on keywords and descriptions. We have employed Cosine Similarity, Gestalt string matching, and Jaccard Similarity.

Cosine Similarity. Cosine similarity measures the affinity between two vectors. It calculates the cosine angle between two vectors (Lahitani, Permanasari, & Setiawan, 2016). It also involves vectorizing the data using TF-IDF. In our study, it shows the cosine angle between a vector of collective keywords and the dataset's description. This is used to figure out how relevant the keywords are, give them a score, and sort them to find the most relevant ones.

$$\cos(\mathbf{t}, \mathbf{e}) = \frac{\mathbf{te}}{\|\mathbf{t}\|\|\mathbf{e}\|} = \frac{\sum_{i=1}^{n} \mathbf{t}_i \mathbf{e}_i}{\sqrt{\sum_{i=1}^{n} (\mathbf{t}_i)^2} \sqrt{\sum_{i=1}^{n} (\mathbf{e}_i)^2}} \tag{5}$$

Cosine similarity is useful for detecting matches, even when there are no explicit pattern matches between documents. This makes it highly robust for document matching tasks, particularly when matching documents with keywords. In contrast to word-to-word matching, cosine similarity considers the overall distribution of terms in a document, making it well suited for identifying documents that contain related concepts or ideas. We used the spatial module provided by SciPy, an open source Python-based library. When applying it, we initially imported the library, vectorized the words or sentences, then calculated the cosine distance with the "distance.cosine()" function, and subsequently determined the cosine similarity by subtracting the cosine distance from 1. Consider the following examples to see how cosine similarity is applied in practice:

- "happy birthday" and "birthday wishes" have a score of 0.91
- "I adopted a cute little kitten" and "cat" have a score of 0.72
- "I picked some fresh apples from the orchard today" and "apple" have a score of 0.6
- "I went for a walk in the park today" and "pizza" have a score of 0.6

Jaccard Similarity. Jaccard similarity measures the similarity between two sets of items by calculating the ratio of the size of their intersection to the size of their union:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{6}$$

where A and B are the two sets being compared, $|A|$ and $|B|$ denote their cardinalities (i.e., the number of elements in each set), and $\cap$ and $\cup$ denote the intersection and union operations, respectively (Niwattanakul, Singthongchai, Naenudorn, & Wanapu, 2013). It performs well in matching words with similar characters, regardless of their order. This makes it particularly useful for tasks where words may be mispelled or have small variations in spelling. By considering the shared characters between two words, it can identify patterns that would be missed by other matching techniques that rely solely on exact character matches. Our approach generated sets of unique n-grams from each string and then determined the similarity ratio as

the size of the intersection of these sets divided by the size of their union. This ratio reflects the proportion of n-grams the strings share, measuring their similarity from 0 (no overlap) to 1 (identical). Consider the following examples to contextualize different aspects of jaccard similarity in practice:

- "hello world" and "world hello" have a score of 1
- "green apple" and "red apple" have a score of 0.625
- "cat" and "bat" have a score of 0.5
- "computer science" and "data science" have a score of 0.5
- "Spicy Tofu Stir Fry" and "Vegetable Curry" have a score of 0.25
- "apple" and "banana" have a score of 0.16
- "dog" and "cat" have a score of 0

Gestalt Matching. The gestalt pattern matching is based on recognizing patterns as functional units having unique properties. The Ratcliff/Obershelp pattern-matching algorithm uses this concept to compare the similarity between two one-dimensional patterns, such as text strings, by counting the number and length of common sub-patterns (k_m). (Ratcliff & Metzener, 1988):

$$G(S_1, S_2) = \frac{2k_m}{\|S_1\| + \|S_2\|} \tag{7}$$

where the function $\|\cdot\|$ returns the length of a string, and the value k_m is given by the sum of the lengths of common sub-strings.

Gestalt matching is well-suited to situations with a clear pattern or structure underlying the data being analyzed. Unlike other matching techniques that focus on individual words or small text segments, Gestalt matching can analyze larger structures and patterns in the data, making it ideal for tasks that involve more complex or multi-part information. We employed "SequenceMatcher" from Python's "difflib" module which used "Ratcliff/Obershelp" algorithm to compute the similarity ratio between two sequences by identifying and comparing the largest contiguous matching subsequences without considering any junk elements. It yielded a ratio ranging between 0 to 1, with 1 indicating an exact match. For instance the metrics will return the value 0.8 when comparing the two strings "green apple" and "red apple". In such a case the two substrings 're', of length 2, and ' apple' (initial space included) of length 6 are present in both strings. So in the formula we will have a numerator equal to 2*(2+6) = 16 and a denominator equal to 20. The following list provides further examples on how the method can be applied in practice:

- "computer science" and "data science" have a score of 0.64
- "apple" and "papel" have a score of 0.6
- "Blender Master 2000" and "Mixer Pro 3000" have a score of 0.5
- "Spicy Tofu Stir Fry" and "Vegetable Curry" have a score of 0.3
- "dog" and "cat" have a score of 0

3.2. Algorithms

In improving the findability of datasets our approach foresees the usage of five different algorithms permitting to extract representative keywords from a given text. In particular we used the dataset descriptions to extract representative keywords that accurately reflected the dataset's contents. Once we have generated a set of keywords, we evaluated their effectiveness by comparing them to the ones provided by the person who uploaded the dataset. The main characteristics of the used algorithms are described in the following subsections.

BERT. BERT is a transformer-based algorithm that derives context-based relations between words and subwords of a text. Contrary to common directional and sequential models, BERT adopts a bidirectional approach and reaches the text in both directions to gain the entire context before making predictions. It defies the conventional sequential approach for more contextual understanding, which is imminent in NLP (Devlin, Chang, Lee, & Toutanova, 2018). The Python library we utilized for accessing BERT's implementation was Keybert.

Computational complexity. The computational complexity of BERT (Devlin et al., 2018), considering its dependence on the number of layers (L), size of hidden layers (d), number of attention heads (h), and sequence length (n), can be approximated by the following formula:

$$O(L \cdot n \cdot d \cdot (n/h + d)) \tag{8}$$

where:

- L = Number of Layers
- d = Size of Hidden Layers
- h = Number of Attention Heads
- n = Sequence Length

RAKE. The Rapid Automatic Keyword Extraction (RAKE) algorithm also utilizes statistical methods. It is a single document method and works independently of the domain. It utilizes a stop list to delimit a candidate list. It then calculates the cooccurrence of candidate items and scores their weight. The delimited candidate words yield more meaningful context and scoring (Campobello, Segreto, Zanafi, & Serrano, 2017; Rose, Engel, Cramer, & Cowley, 2010).

Computational complexity. The computational complexity of the Rapid Automatic Keyword Extraction (RAKE) algorithm primarily depends on its process of candidate keyword identification and scoring based on co-occurrence matrices (Campobello et al., 2017). The complexity can be described by the formula:

$$O(W + C^2) \tag{9}$$

where:

- W represents the total number of words in the document, affecting the initial processing and identification of candidate keywords.
- C stands for the number of unique candidate keywords, with the complexity of scoring each based on their co-occurrences and properties such as frequency and degree.

YAKE. YAKE is an unsupervised keyword extraction method that employs statistical techniques to determine the textual importance of subtexts. Unlike other methods, YAKE does not require specific training on a corpus of data, making it a robust approach for keyword extraction. Furthermore, it is not dependent on external databases, languages, domains, or document sizes, making it a versatile tool for multilingual open data portals. Its ability to effectively extract keywords without relying on external factors makes YAKE a useful tool for improving the searchability and discoverability of datasets. (Campos et al., 2020). The Python library we utilized for accessing Yake's implementation was YAKE.

Computational complexity. The computational complexity of the YAKE! keyword extraction algorithm (Campos et al., 2020) is given by the formula:

$$O(n) + O(mlogm) + O(p) + O(q^2) + O(qlogq) \tag{10}$$

where:

- n is the number of characters in the text,
- m is the number of unique words in the document,
- p is the number of generated n-grams, and
- q is the number of candidate keywords to be extracted.

TextRank. TextRank is an algorithm used for keyword and sentence extraction from a given text document. It creates a graph of the sentences in the document, assigns a score to each sentence based on its importance, and then extracts the top-ranked sentences as a summary of the document or the most important keywords from the text. It is a simple and effective algorithm that is unsupervised and can be applied to a wide range of text datasets (Mihalcea & Tarau, 2004).

Computational complexity. The computational complexity of the TextRank algorithm depends on its iterative process of calculating word importance scores and propagating these scores through the graph of word relationships (Mihalcea & Tarau, 2004). The complexity can be described by the formula:

$$O(n^2) + O(kn^2) + O(nlogn) \tag{11}$$

where: where:

- n is the number of nodes,
- k is the number of iteration in updating the nodes' weights

ChatGPT. ChatGPT is a large-scale language model based on the GPT (Generative Pre-trained Transformer) architecture, which is trained on a diverse corpus of text using unsupervised learning techniques. While ChatGPT is primarily known for its ability to generate human-like text, it can also be used for other NLP tasks, including keyword extraction. In the context of this study, the description of the dataset is provided to ChatGPT that then returns the top N candidate words based on their relevance (Song et al., 2023). The "get-3.5-turbo" model has been used in our experiments.

Computational complexity. The computational complexity of GPT, a large language model developed by OpenAI, is challenging to quantify precisely as no precise information has been released with respect to such an aspect, due both to its complex architecture, and proprietary nature.

The algorithms above yield their respective keywords. In our proposed methodology, we combine the output of four of these pre-trained representative algorithms and calculate their similarity with the document. We have employed cosine similarity and TF-IDF for the final similarity measure due to their robustness.

3.3. Adjustable Control Parameters

Our methodology incorporates several adjustable control parameters critical for tailoring the algorithm to specific research needs, enhancing the precision and relevance of outcomes. These parameters include the matching threshold (Gestalt and Jaccard) for keyword similarity, the number of keywords extracted per dataset, and the minimum description length for keyword extraction. Each parameter has been thoughtfully established to ensure a balance between comprehensive representation and analytical clarity.

Matching Threshold (Gestalt and Jaccard Index). The decision to use a 0.6 threshold for similarity matching between original and predicted keywords, employing both Gestalt and Jaccard similarity methods, was grounded in extensive empirical testing. Trials at 0.4, 0.6, and 0.8 thresholds highlighted the importance of striking a balance between precision and recall.

At a 0.4 threshold, while recall was high, precision suffered significantly, leading to mismatches like "climate change" and "weather forecast" – a clear overreach in terms of relevance. On the other end, a 0.8 threshold yielded high precision but at the expense of recall, excluding closely related terms such as "global warming" and "climate warming" due to slight linguistic variations, thereby narrowing our research's applicability.

The 0.6 threshold, therefore, represents a carefully calibrated balance, effectively capturing a broad array of relevant keywords while maintaining a strong guard against irrelevant matches. It allows for meaningful matches like "solar power" and "solar energy" to be recognized as similar, ensuring our findings are both accurate and inclusive, thereby enhancing the credibility and utility of our research.

Number of Keywords. The number of keywords extracted per dataset description was another critical adjustable parameter. Setting this number at ten allowed us to capture a dataset's main themes comprehensively and succinctly, including both primary topics and significant secondary themes. This approach ensured a depth of analysis while avoiding the inclusion of peripheral or less relevant keywords, thereby maintaining the thematic focus and potentially facilitating efficient retrieval across the datasets.

Description Length. The minimum description length, set at 1000 characters, ensured that the text base for keyword extraction was sufficiently detailed to yield meaningful and contextually rich keywords. This threshold guaranteed that the extracted keywords reflected well-developed themes and concepts, providing a robust basis for extraction.

3.4. Findability score

Findability is a critical factor that affects the accessibility and usefulness of datasets in the context of open data. The findability score measures how easily and quickly a user can locate a dataset. This score is often calculated using various metrics derived from metadata. Improving the findability score is essential for increasing the accessibility and potential usability of open data.

This study focuses on the European Data Portal as a case study with the goal of improving the findability of datasets. To achieve this, we first investigated the factors that affect findability and found that they are mainly related to keyword usage and categories. Together, these two elements account for 60% of the findability score, each contributing 30% (EU, 2023). Since keywords are more specific to individual datasets than categories, this study places particular emphasis on analyzing the role of keywords in improving findability. The findability score chart is presented in Table 1.

4. Methodology

This section outlines our methodology, in which we analyzed 15 different open data portals from the findability perspective, as reflected in Table 2. Following the analysis, we found there to be a lack of effective search mechanisms and representative metadata. Consequently, it directed our research towards automated metadata extraction. In the initial phase, we considered EDP as our use case, collected metadata records from it, and then cleaned the data, ensuring consistency in language and structure. All metadata records were converted to a common language for consistency and ease of analysis.

Next, we experimented with various NLP algorithms to extract up to 10 keywords from the descriptions of datasets. This number was chosen to cover a broad range of topics from different perspectives, offering publishers diverse choices. The efficiency of these algorithms was assessed by comparing the extracted keywords against the original ones found in the metadata records. This comparison helped us determine the accuracy of each algorithm in replicating the original keywords. Through this process, we identified the algorithms that provided the most accurate matches and analyzed their performance in different situations.

We structured our approach into a multi-step process, which encompassed an initial Preliminary Analysis of Data Portals followed by a five-step methodology: Data Gathering, Data Cleaning, Data Transformation, Keyword Extraction, and Evaluation, all of which are delineated in Fig. 1. Analysis of Data Portals involved surveying different portals from the perspective of intelligence, search mechanisms, and recommendation systems, Data Gathering involved collecting metadata records from the EDP, while Data Cleaning ensured consistency in structure and appropriate representation of the metadata records. Data Transformation involved converting the metadata records to a common language for consistent analysis. Keyword Extraction involved experimenting with various NLP algorithms to extract keywords from dataset

Table 1
European data portal findability score.

Indicator	Description	Metrics	Weight
Keyword usage	Keywords directly support the search and thus increase the findability of the data dataset.	The system checks whether keywords are defined. The number of keywords has no impact on the score.	30
Categories	Categories help users to explore datasets thematically.	It is checked whether one or more categories are assigned to the dataset. The number of assigned categories has no impact on the score.	30
Geo search	Usage of spatial information would enable users in order to find the dataset with a geo-facetted search.	It is checked whether the property is set or not.	20
Time-based search	Usage of temporal information would enable users for a timely based facetted search.	It is checked whether the property is set or not.	20

Table 2
Data portals technical analysis.

Data portal	Region	Search engine	Autocomplete	Recommendations	Multilanguage
https://data.europa.eu/en	Europe	Basic	No	No	Yes
https://www.dati.piemonte.it/	Italy	Basic	Yes	No	No
https://www.dati.gov.it/	Italy	Basic	No	No	No
https://www.data.gouv.fr/en/	France	Basic	No	No	Yes
https://opendata.paris.fr/	France	Basic	No	No	No
https://data.overheid.nl/	Netherlands	Basic	Yes	No	No
https://www.data.gov.uk/	UK	Basic	No	No	No
https://data.gov/	US	Basic	No	No	No
https://data.gov.ie/	Ireland	Basic	No	No	No
https://africaopendata.org/	Africa	Basic	No	No	Yes
https://dataportal.asia/home	Asia	Basic	No	No	Yes
http://data.un.org/	UN World	Basic	No	No	No
https://data.nasa.gov/	US	Basic	Yes	No	No
https://data.gov.be/	Belgium	Basic	Yes	No	Yes
https://www.data.gv.at/	Austria	Basic	No	No	Yes

descriptions, while Evaluation involved comparing the extracted keywords with the original keywords in the records and determining the accuracy of each algorithm.

Following this methodology, we formed an analysis framework and gained insight into the effectiveness of different NLP algorithms to extract representative keywords from metadata records.

Preliminary step: Analysis on data portals. We analyzed 15 well known and largely used data portals for this study to assess findability, as reflected in Table 2. All of the considered portals were lacking in intuitive and intelligent search mechanisms. None of them had an intuitive search engine or a recommendation system to improve findability. The search engines in them mostly performed one-to-one matching of basic metadata elements such as titles, categories, or keywords. However, a few of them provided an autocomplete search that suggested exact dataset titles upon entering text in the search box. The majority of searches were non-intuitive, either because the metadata was missing or because the metadata that was provided was not representative of the datasets. It furthered our motivation to look into the formation of the problem statement.

Step 1: Data gathering. In the initial phase, our goal was to gather varying data from the European Data Portal for the study to be more representative. It had 3 technical substeps, namely European Data Portal, Application and Database. The application was created in Python, which used SPARQL endpoints to access EDP and retrieve metadata and store it in a SQL database. As reflected in Listing 1, we used the following SPARQL query to retrieve data for each theme/category. The listing is made available so to permit the replication of the experiment by the interested reader. Indeed, using the data freely available, as indicated in Data availability section, it is possible to fully replicate the experiment to confirm the discussion we report in this paper, and to possibly provide further insights on the subject. The listing could be also useful for successive analysis on the evolution of the metadata associated to a data set.

For data gathering, we planned the following strategy:

1. Data strategy & SPARQL endpoint: We accessed EDP through the SPARQL endpoint. We created a Python application to interact with EDP using SPARQL queries and stored it in an SQL database. There were 13 EDP themes containing populated data, and we downloaded representative data from each category. After the data dump from each category, it amounted to 69423 metadata.
2. Metadata attributes: The metadata attributes we downloaded were title, description, publisher, keywords, theme, dataset URI, language, and themes. The imminent attributes were description (for algorithms to process), original keywords (to match and evaluate results), and themes.
3. SQL Database: We created a table that accommodated all the columns above, the results of data transformation, keyword extraction, and keyword matching. Following are the columns attributed to each category:
    (a) SPARQL Retrieved: Title, description, publisher, keywords, theme, dataset URI, language, and themes
    (b) Data Transformation: English Description and English keywords.
    (c) Keyword Extraction: BERT keywords, RAKE keywords, YAKE keywords, TEXTRANK keywords, ChatGPT keywords, and BRYT keywords.
    (d) Keyword Matching: BERT matching, RAKE matching, YAKE matching, TEXTRANK matching, ChatGPT matching, and BRYT matching.

Step 2: Data cleaning. Our data cleaning phase focused on eliminating duplicate, incomplete, garbage, and nonrepresentative data. The

Listing 1: SPARQL query for retrieving datasets with specific conditions.

```
PREFIX dct: <http://purl.org/dc/terms/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX dcat: <http://www.w3.org/ns/dcat#>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
PREFIX foaf: <http://xmlns.com/foaf/8.1/>
PREFIX adms: <http://www.w3.org/ns/adms#>

SELECT DISTINCT ?datasetURI ?datasetTitle ?datasetDescription ?publisher (GROUP_CONCAT(?
    keyword; SEPARATOR=", ") AS ?keywords)
WHERE {
  ?datasetURI a dcat:Dataset;
              dcat:theme <http://publications.europa.eu/resource/authority/data-theme/TRAN>;
                  # This is the theme for Transport
              dcat:keyword ?keyword;
              dct:title ?datasetTitle;
              dct:publisher ?publisher;
              dct:description ?datasetDescription;
              dct:language <http://publications.europa.eu/resource/authority/language/ENG>.
  FILTER (LANG(?datasetTitle) = "en")
  FILTER (strlen(str(?datasetDescription)) > 1000)
}
GROUP BY ?datasetURI ?datasetTitle ?datasetDescription ?publisher
LIMIT 10000
```

objective was to trim down to a sample of datasets with representative metadata that could be used as our base for evaluating our methodologies. We performed the following cleaning strategy:

- Initially, we removed the rows that were exactly alike, and to our surprise, there were quite a few.
- Moving forward, we deleted rows that were similar but only had minimal information changed, for example, a similar kind of data set but just with differences in month, date, place, or a similar attribute. This allowed us to eliminate a lot of data that inherently seemed alike.
- We also deleted data where they had put column names in keywords or had put other metadata as it is.
- We also removed data that had one or two numbers as their keywords, which might not have been suited our study.
- We also removed the instances where they had just put a variation of the title into the description and the description was really short.
- We removed the white spaces or garbage characters in the data.

Step 3: Data transformation. In the transformation phase, we recognized data that was not in English despite the query filters and needed to be translated. It needed to be translated using the API that EDP uses for our study to be more accurate. EDP uses E-translation API; hence, we accessed the same API. This section has the following substeps:

- ETranslation API: Etranslation API is available for Public administrations, SMEs, and CEF Projects. It was particularly allowed for research and development purposes for this project. It allows you to send text to be translated along with source and target languages. It can also detect the source language by itself most of the time. In your request to it, you also send your public IP address and address to your response handler, which sends the response once it is ready.
- Java API Handler: We had two components of Java API Handler: Request and Response Handler. They are defined as follows:

  1. Request Handler: It accesses the database and retrieves records that need to be translated, sends them to the API, receives "Request id", and saves it against the record in the database.
  2. Response Handler: It receives the response from the API, which contains translated text and request id that it returned in the request handler. It receives and stores the translated text against the request-id.

Step 4: Keyword extraction. Our objective for this phase was to assess and compare the efficiency of five keyword extraction algorithms: BERT, RAKE, YAKE, TEXTRANK, and ChatGPT. We applied these algorithms to extract keywords from the text description and then used the output of the four initial algorithms as input to our proposed hybrid method, BRYT. BRYT utilizes cosine similarity to score the collective keywords generated by the other algorithms and selects the top 10 most relevant keywords as the final output.

Computational Complexity. The computational complexity of the BRYT, which integrates BERT, RAKE, YAKE, and TextRank, followed by a cosine similarity scoring, primarily hinges on the algorithm with the highest complexity due to its parallel execution strategy. Given the complexities, BERT is typically the most computationally demanding due to its quadratic relationship with sequence length and model dimensions. It also has an additional but less significant load from the cosine similarity calculations which in larger picture could prove to be negligent. The mathematical notation for complexity of BRYT is reflected below:

$$O(L(n^2 \cdot \frac{d}{h} + n \cdot d^2))$$

Where:

- L represents the number of layers in BERT.
- n is the sequence length for BERT and corresponds to the number of nodes for TextRank.
- d denotes the size of the hidden layers in BERT and the dimensionality of the vector space for cosine similarity.
- h is the number of attention heads in BERT.

Excluding the negligible computational cost of the cosine similarity, BRYT's computational complexity mirrors that of BERT, mainly due to the parallel processing of the constituent algorithms.

Step 5: Matching and evaluation. During our final phase, we evaluated the performance of our algorithms using two metrics: Gestalt matching and Jaccard similarity. To assess the similarity between the

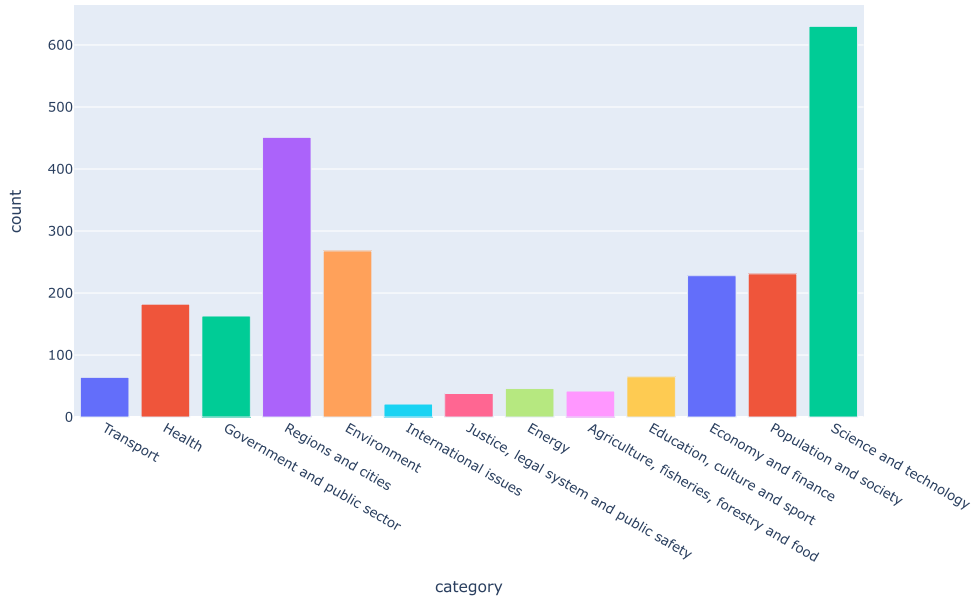

Fig. 2. Data distribution across themes/categories.

extracted keywords and the original keywords, we calculated the similarity score between the extracted keywords and the original keywords for each algorithm using these measures. If the similarity score for a keyword was greater than or equal to 0.6 for either metric, we consider that keyword a match. Using Gestalt matching and Jaccard similarity, we determined which algorithm performed better in each case by evaluating the number of matches. We categorized matches into three categories, namely major match, minor match, and no match. They are defined as follows:

- Major Match: If, using a particular algorithm, more than 50% or 5 keywords matched in any given instance (keyword), then the result was categorized as a major match under that algorithm.
- Minor Match: If, using a particular algorithm, less than 50% or 5 keywords matched in any given instance (keyword), then the result was categorized as a minor match under that algorithm.
- No Match: If no keywords matched under the defined criteria, it was categorized as no match.

It is worth mentioning that for this research, the computational framework was powered by an 11th Gen Intel(R) Core(TM) i7-1185G7 processor, clocked at 1.80 GHz, with turbo speeds up to 3.00 GHz, and supported by 16.0 GB of RAM (15.4 GB usable), ensuring swift data processing and analysis. The system, leveraging a 64-bit OS with an x64-based architecture, was augmented by a total storage capacity of approximately 436.64 GB, distributed across three hard drives, and an Intel Iris Xe Graphics card for advanced graphical computations. Some of the experimentation was also carried out on Google Colab with 12.7 RAM, 107.7 GB disk storage capacity and Googles backend compute engine TPU. Given that our work did not require intensive training of algorithms, these specifications proved to be more than sufficient, enabling swift and efficient performance throughout the research activities.

5. Results

In the results section, we analyzed our resulting data. We evaluated the performance of our methodology from different angles to gain extensive insight into the patterns of data and their effects on keyword scoring. Apart from employing NLP techniques, we also proposed a hybrid methodology, BRYT, which combined the output of four algorithms (BERT, YAKE, RAKE, TEXTRANK), recalculated their relevance using cosine similarity, and selected the top 10 keywords from the rescored keywords. We employed gestalt pattern matching and jaccard similarity to score the matching with the original keywords. Our evaluation of the results from the lens of each algorithm, considering each category and varying lengths of description, reflected that our hybrid methodology outperformed other algorithms in having more representative keywords.

5.1. Data analysis

Initially, our analysis focused on the structure of the cleaned data, which comprised 1393 datasets across 13 themes/categories, reduced from an original count of 69423 datasets. Each dataset had one or more than one theme/category associated with it. The distribution of data sets according to categories is reflected in Fig. 2. The distribution appears to be skewed, as a significant portion of the datasets are associated with multiple themes/categories. For example, the categories “Science and Technology” and “Regions and Cities” are notably dense, but 88% and 97% of these instances, respectively, are associated with at least one other theme/category. Apart from the categories, most datasets had description lengths between 500 and 3000, while a considerable number of datasets also had greater than 3000. Following the data analysis, we moved on to evaluating our methodology.

5.2. Methodology analysis

At the beginning of our evaluation, we first analyzed our general results, which reflected that 69.1% of instances were majorly matched using one of the algorithms. In comparison, 24.7% were minorly matched and 6.2% had no matches, as reflected in Fig. 3.

While comparing the major matches of each algorithm, we found that our proposed hybrid methodology BRYT performed better than other algorithms, with 27.1% of times having more efficient or equally efficient matches, while ChatGPT was a close second with 23.8%. This percentage also entails that there were instances where other algorithms matched equally, and in that case, it counted both algorithms as an efficient/winner algorithm in that particular instance.

While comparing the minor matches of each algorithm, we found YAKE performed efficiently with 35.5% matches while ChatGPT was again a close second with 29.4%. We focused this study more on major matches because 69. 1% of all the data were major matches, and our

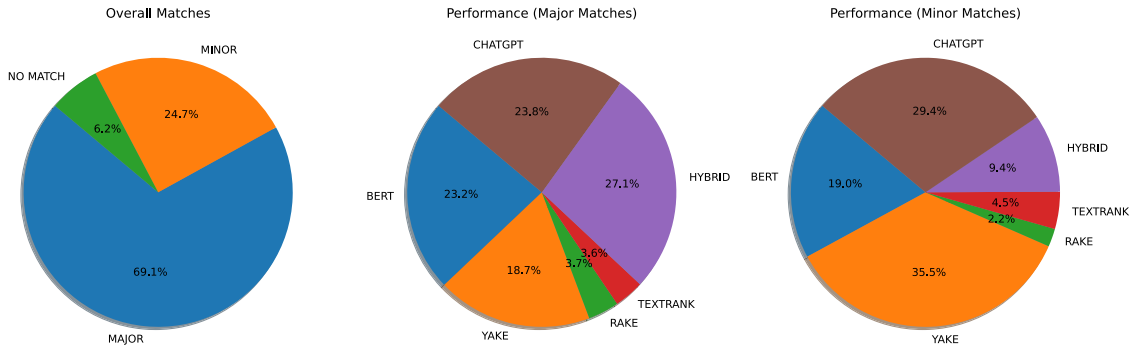

Fig. 3. Overall matching.

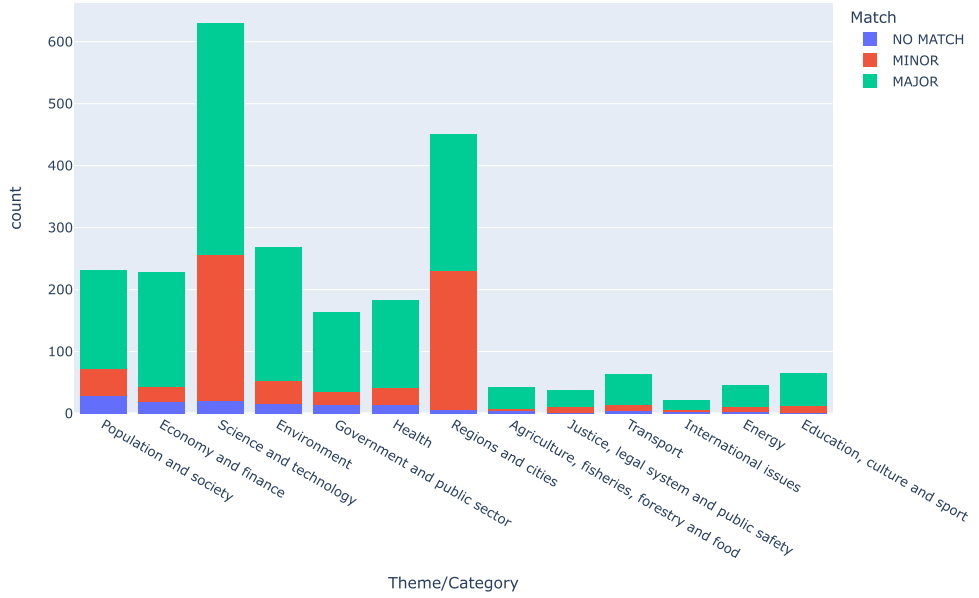

Fig. 4. Number of matches in each theme/category.

proposed methodology performed better in those instances, deeming it a more fitting evaluation for this problem and for the particular open data environment. Moving further in our evaluation, we analyzed our results more deeply to gain more intuitive and meaningful insights. As previously discussed, there were 13 themes/categories that data in European Data Portal belonged to, and in Fig. 4, we analyzed the number of instances majorly matched, minorly matched or did not match at all in each theme/category.

During our analysis, we found that in most of the categories, there were heavily major matches except "Regions and Cities". As we analyzed further, we found that the majority of descriptions in that category were more structured than textual. Most of them defined data in terms of column and value about some informative attributes, followed by minimal descriptive text. This implied that the studied algorithms worked better with more textual descriptions.

After our findings about algorithms being more efficient with textual descriptions, we analyzed the matches from the perspective of the length of descriptions. As reflected in Fig. 5, we found that description length had near to no effect on the efficiency of the algorithms. All the ranges of length above 1000 characters seemed to have a similar pattern in terms of matches i.e. on average, around 67% were major matches irrespective of description length, and around 25% were minor matches.

Considering our insights above, we moved on to plot data of each category and description length to evaluate the length of description each category had. As reflected in Fig. 6, the majority of data(description length) was plotted within the range of 0 to 3000 characters. While there were quite a few that exceeded 3000 characters, and most of them were major matches. On the other hand, most of the matches under 3000 characters were major matches except in the category "Regions and Cities". Most matches in this category were minor, as established, potentially owing to their description format.

Moving forward, we analyzed, then, the pattern of all the algorithms in each category during major and minor matches. As reflected in Table 3, we first analyzed how each algorithm performed in each category during all major matches. In most categories we found that our proposed methodology performed more or equally efficiently except in the categories "Regions and Cities" and "Science and Technology" where ChatGPT performed better. As discussed before, the majority of descriptions in the category "Regions and Cities" had a more structured format than textual. In contrast, in "Science and Technology" category, there were descriptions with consistently varying lengths, which either matched majorly or minorly.

Following it, we also analyzed how they performed during minor matches. As reflected in Table 4, the majority of minor matches were in the categories "Regions and Cities" and "Science and Technology". ChatGPT dominated these categories in minor matches. While in other categories there were minimal minor matches where either ChatGPT or BRYT performed well.

After our findings about major and minor matches, we scatter-plotted them separately according to description length, categories, and matches with each algorithm. First, we plotted the instances of major

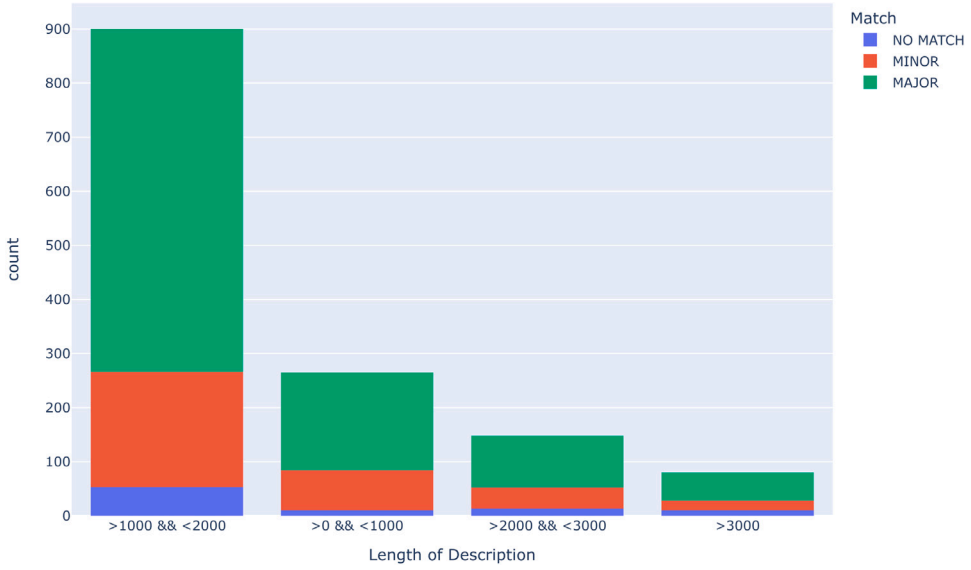

Fig. 5. Number of matches with description length.

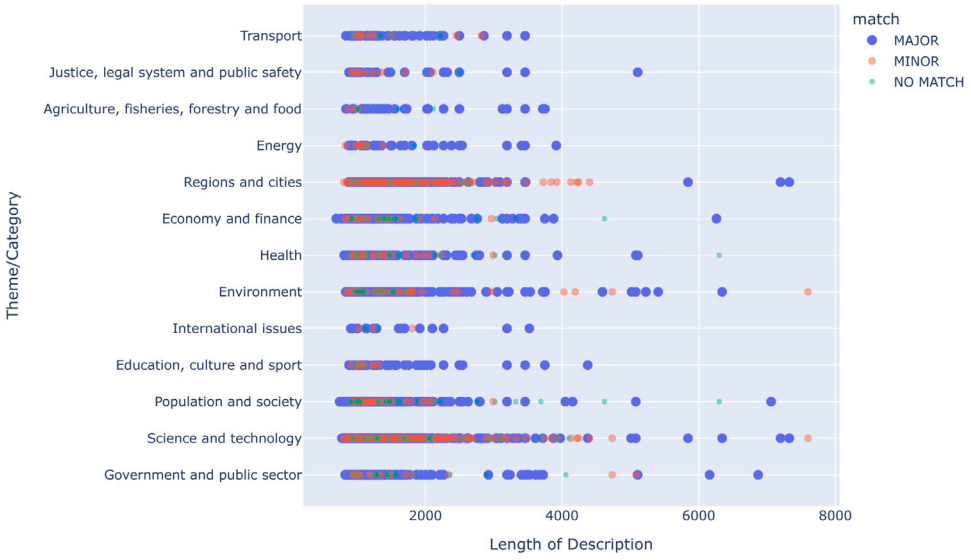

Fig. 6. Scattered data plot of matches in categories according to the description lengths.

Table 3
Performance of algorithms during major matches in each category.

Category/Theme	BERT	YAKE	RAKE	TEXT RANK	CHATGPT	Hybrid
Agriculture, fisheries, forestry and food	7	10	0	4	7	**15**
Economy and finance	72	42	11	5	44	**82**
Education, culture and sport	17	15	1	2	14	**32**
Energy	6	6	2	2	14	**17**
Environment	52	53	10	12	81	**93**
Government and public sector	36	34	10	7	38	**53**
Health	52	40	4	13	31	**64**
International issues	3	5	1	3	2	**9**
Justice, legal system and public safety	8	3	2	2	8	**14**
Population and society	64	33	5	7	32	**88**
Regions and cities	43	73	6	2	**110**	38
Science and technology	94	104	16	18	**157**	96
Transport	15	11	5	3	12	**20**

Table 4
Performance of algorithms during minor matches in each category.

Category/Theme	BERT	YAKE	RAKE	TEXT RANK	CHATGPT	Hybrid
Agriculture, fisheries, forestry and food	**2**	1	0	1	**2**	1
Economy and finance	10	10	1	3	**13**	9
Education, culture and sport	**4**	**4**	0	1	**4**	**4**
Energy	**4**	3	2	3	**4**	2
Environment	14	12	4	4	**17**	10
Government and public sector	7	9	1	2	**13**	7
Health	11	**12**	2	6	**12**	11
International issues	2	1	1	1	**3**	0
Justice, legal system and public safety	5	5	0	1	3	**7**
Population and society	19	14	1	6	**22**	17
Regions and cities	82	93	5	10	**134**	25
Science and technology	85	95	8	14	**143**	25
Transport	**6**	4	0	2	8	4

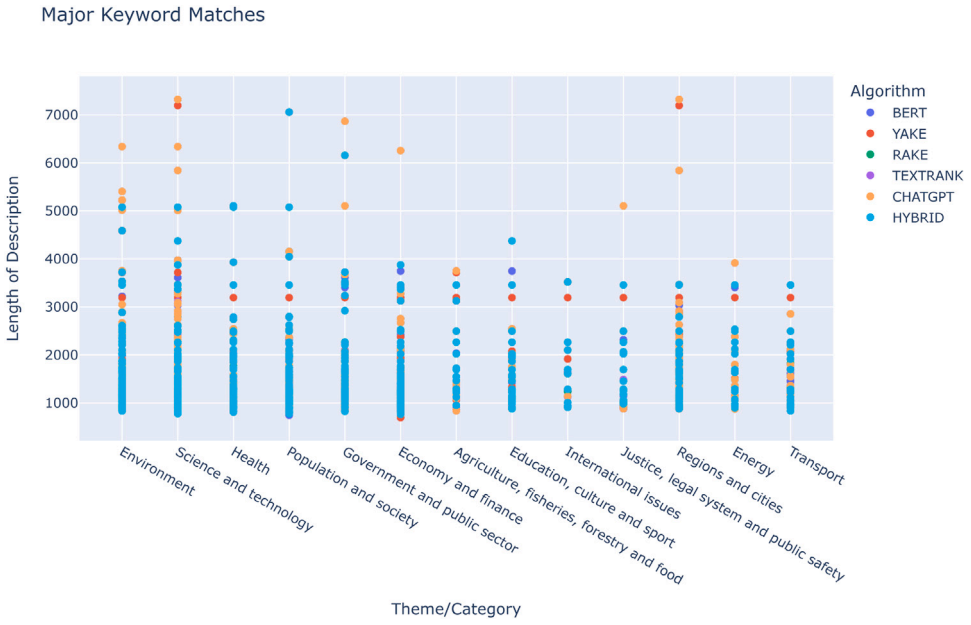

Fig. 7. Data plot of major matches by all algorithms in each category by description length.

matches with respect to the description lengths in each theme/category considering all algorithms. As reflected in Fig. 7, most major matches went up to the description length of 4000. Except for the category of "International Issues", which had minimal data points as major matches, most were populated in a similar pattern and the majority of them had a dense population of matches up to a description length of 2500.

While in minor matches most data points in the majority of categories were densely populated, up to 2000 characters. As reflected in Fig. 8, except for two categories namely "Regions and Cities" and "Science and Technology", where it went up till 4000. The analysis made us look more intuitively towards description length in major matches. While as established, it looked inconsequential but it also entailed that a minor match is more probable to have a description of under 2500 characters while a major match could have a description of up to 4000 characters.

During our analysis, we found some insightful findings and correlations. They entailed that BRYT performed better in most of the major matching cases where the description was in a proper textual format while ChatGPT performed better in instances where there were minor matches even if the description was in a structured format. We also found that in the bigger picture description length had no effect on the efficiency of algorithms in terms of major matches. However, in minor matches, it was more probable for them to have a description under 2000 characters than longer. Additionally, we found that minor matches were minimal in other categories except for "Regions and Cities" and "Science and Technology", where there were a lot, and ChatGPT flourished in them, while BRYT mostly dominated major matches. Our analysis entailed more confidence in our proposed methodology and its implication in European Data Portal.

5.3. Runtime Analysis

Our runtime analysis, as reflected in Table 5, focuses on comparing the computational efficiency of our proposed hybrid model (BRYT) against traditional methods, calculating the time taken to process 1393 instances. This section aims to highlight the model's time efficiency.

The hybrid model, which integrates outputs from multiple algorithms, achieves an impressive balance of speed and effectiveness, outperforming all other methodologies in this regard. Notably, its runtime is under one second and quite close to BERT, making it a highly viable option for practical use.

We calculated the hybrid model's runtime based on BERT's processing time, the most expensive part of our parallel processing of four algorithms (B.R.Y.T) at once. Following it we added BRYT's own runtime to rescore the outputs. The selection of BERT as the baseline for BRYT's completion time provides a realistic evaluation of the model's efficiency.

Overall, the analysis underscores our BRYT's superior efficiency, establishing it as an optimal choice for keyword extraction, particularly when the balance between runtime and efficiency is key.

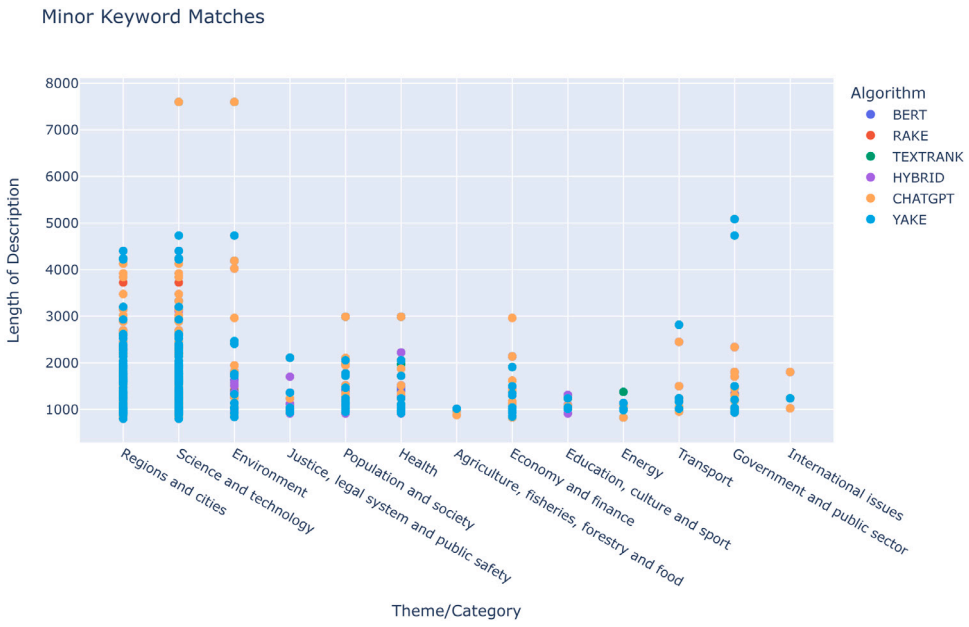

Fig. 8. Data plot of minor matches by all algorithms in each category by description length.

Table 5
Runtime performance comparison.

Method	Total instances	Total time taken (s)	Runtime per instance (s)
BERT	1393	1007	0.723
YAKE	1393	43.28	0.031
RAKE	1393	4.02	0.003
TEXTRANK	1393	173.1	0.124
CHATGPT	1393	2463	1.76
HYBRID	1393	1007 + 121 = **1128**	0.8

6. Related research

In this section, we surveyed various research works to further our hypothesis and methodology. We have considered research articles related to auto-tagging/keyword extraction in the context of automated metadata generation and their use in pertinence to open data portals.

Most open data portals have a very basic search mechanism which contributes to the findability problem. Despite the availability of search engines, the problem of findability persists, mainly due to the presence of selective metadata elements (Ahmed, 2023). Metadata mostly consists of title, description, and keywords (Škoda, Bernhauer, Nečaskỳ, Klímek, & Skopal, 2020). When metadata, even the basic one, is not properly defined, it directly affects the discoverability of datasets and hence hinders its usability (Nogueras-Iso, Lacasta, Ureña-Cámara, & Ariza-López, 2021). According to the European Data Portal, keywords and categories each contribute 30% to the findability score and make data more discoverable (EU, 2023). It drives our research to focus on automatic keyword extraction to improve findability.

Keyword extraction mostly employs supervised or unsupervised algorithms. Most of the time, supervised algorithms turn out to be inconsistent in the context of keyword extraction (Beliga, 2014). They require a huge corpus of textual data and fail when it is tested with cases outside of the corpus domain. This necessitates the consideration of unsupervised approaches which do not rely solely on corpus or thesaurus, they use provided documents to make statistical inferences (Baruni & Sathiaseelan, 2020). To address these challenges, heuristic and evolutionary search optimization algorithms have been proposed to enhance algorithmic efficiency and adaptability in keyword extraction tasks (Shahraki & Zahiri, 2017; Yadav & Deep, 2013). A significant contribution in this area is the Gravitational Search Algorithm (GSA), introduced by Esmat Rashedi et al. which is based on the law of gravity and mass interactions, showing promising results in solving various nonlinear functions(Rashedi, Nezamabadi-Pour, & Saryazdi, 2009). Additionally, they also conducted an extensive review of GSA's developments in engineering optimization, highlighting its utility in similar diverse applications(Rashedi, Rashedi, & Nezamabadi-Pour, 2018).

Advanced approaches explore multi-task learning paradigms and pre-training models to enhance the generalization and robustness of keyword extraction methods, adapting evolutionary principles to refine these processes (Guo, Sun, Qi, & Wang, 2022). Moreover, real-time heuristic search strategies are evolving through simulated evolution, demonstrating their potential in dynamic applications such as automated keyword extraction (Bulitko, 2016). The application of heuristic and evolutionary algorithms in keyword extraction is further exemplified by the development of systems like EA-SOM, which combines evolutionary algorithms with self-organizing maps to optimize the extraction process (Cuevas et al., 2020). Similarly, adaptive heuristic functions have been tailored through evolutionary strategies to improve the accuracy and efficiency of various algorithms, including those used in keyword extraction (Yiu, Du, & Mahapatra, 2019).

When it comes to unsupervised approaches, apart from basic methodologies, the most contemporary choice is transformers while tackling natural language processing problems (Lin, Wang, Liu, & Qiu, 2022). A novel methodology employing linguistic approaches and evolutionary graphs has been designed specifically for the educational context, showing significant improvements in extracting relevant keywords from educational texts(Espada, Martínez, Rico, & Sánchez, 2023).

Automated tagging or keyword extraction is not traditionally deployed with open data portals in practice. One of such studies presents an improved algorithm to extract objective keywords from Chinese academic text. They use an improved text rank to overcome traditional limitations and achieve better results (Liu, 2023). Another such study focuses on the fact that to use supervised learning, you need a good amount of training data which might come with its own set of limitations and biases (Wu et al., 2023). Notably, the use of Gravitational Search Algorithm (GSA) has been explored to optimize the parameters in keyword extraction algorithms, providing a novel approach to enhance precision and reduce reliance on large training datasets (Xueqiang, Zhi'an, & Xindong, 2019). Focusing on unsupervised learning, Campos et al. introduce YAKE!, a tool for unsupervised, lightweight

keyword extraction. It leverages statistical text features from individual documents, facilitating efficient, domain-independent keyword extraction without external data sources (Campos et al., 2020).

Zaira Hassan Amur et al. (2023) in essence highlight the importance of quality datasets in keyword extraction research, focusing on factors like structure, complexity, and dataset quality for efficient keyword extraction (Amur et al., 2023). Enes Altuncu et al. (2023) suggest improvements in Automatic Keyword Extraction (AKE) through PoS-tagging and semantic enhancement, demonstrating considerable gains in various AKE approaches (Altuncu, Nurse, Xu, Guo, & Li, 2022). M. Saef Ullah Miah et al. (2021) present an experimental investigation in the Electric Double-Layer Capacitor (EDLC) domain, highlighting the effectiveness of unsupervised techniques like YAKE and the necessity of appropriate similarity indices for evaluating keyword similarity. Their research underscores the advantage of algorithms such as MultipartiteRank (Miah, Sulaiman, Sarwar, Zamli, & Jose, 2021).

Swagata Duari and Vasudha Bhatnagar (2018, 2019) discuss challenges in graph-based keyword extraction methods and introduce novel approaches like sCAKE and a complex network-based supervised keyword extractor, showing effectiveness across multiple languages and independence from the domain, collection, and training language (Duari & Bhatnagar, 2019, 2020). Nieman (2015) focuses on the educational domain, and employs object-to-object similarity for tags. It evaluates tags based on other tags with which they have been used. Its focus is on the history of tags, which could make it inefficient in terms of new data and new tags (Niemann, 2015). Another such study focuses on experimental approach combining statistical methods and Wordnet-based pattern evaluation for automatic keyphrase extraction. This work emphasizes the limitations of manually assigned keyphrases and introduces a novel combination of graph and statistical methods for enhanced keyphrase extraction (Dostal & Ježek, 2011).

Tkaczyk et al. (2015) highlight the challenges of extracting metadata from scientific articles and utilizing open data sources and presents an open-source system that addresses these issues. CERMINE, the proposed system, employs machine learning and a modular workflow to extract various types of metadata with accuracy (Tkaczyk, Szostek, Fedoryszak, Dendek, & Bolikowski, 2015). Another such system, Roomba, automates the extraction, validation, correction, and generation of descriptive linked dataset profiles to overcome the variations in size, language, and freshness of open data sources (Assaf, Senart, & Troncy, 2015). The study focuses on CKAN-based data portals and validates Roomba against a set of open data portals, including the Linked Open Data Cloud. The findings suggest that most open datasets suffer from poor metadata quality and lack informative metrics, highlighting the need for improved dataset search and metadata quality control (Assaf et al., 2015).

Open Data portals require publishers to enter keywords manually. They might ask for them to be different than themes/categories since they are chosen from a predefined vocabulary while keywords are not (Lnenicka, 2015). The European Data Portal was analyzed and there were found to be severe inconsistencies with regard to keywords. 90 of the most frequent keywords could be reduced to basic 18 nonredundant keywords. Apart from that different variations of the same keywords were entered which could potentially lead to search bias (Schauppenlehner & Muhar, 2018). Given there was no keyword extraction or recommendation mechanism found on surveyed data portals we found the hypothesis for our study and proceeded to experiment. We have summarized the major contribution of each study in the Table 6 below.

7. Discussion and future works

This study highlights the importance and problem of findability in open data portals. Furthermore, it underscores the importance of high-quality metadata in relation to the findability of datasets. The primary focus of this study pertains to a particular element of metadata, namely keywords. Keywords are identified as one of the most crucial and efficient elements in enhancing the findability of datasets. The significance of metadata is sometimes overlooked, and it is hastily completed with nonrepresentative keywords throughout the publication process, compromising their findability.

This article proposes an approach to extracting representative keywords and their subsequent recommendation to publishers, with the aim of improving metadata and findability. Although we have proposed a solution to address the issue, it is important to recognize that this approach is not without its inherent limits. The method depends on the preexisting information of the dataset, namely the description, which is expected to be available and indicative of the dataset's characteristics. When confronted with brief or unrepresentative descriptions, our technique was demonstrated to be less efficient.

This study also reviewed and acknowledged the role and relevance of heuristic and evolutionary search optimization AI-based algorithms such as the Gravitational Search Algorithm and Inclined Planes System Optimization. Although not utilized in this study, we plan to explore these methods in our future work to further refine our metadata extraction and recommendation system.

In the context of Europe's data economy, which was valued at approximately 739 billion in 2020, enhancing metadata quality can have significant economic impacts (Loenen et al., 2021). By automating metadata extraction, operational costs are reduced by minimizing manual labor and improving the efficiency of data management. For instance, if we estimate a modest 5% increase in data usability due to improved metadata across major European data portals, this could potentially translate into millions of euros in additional economic activity annually. This is driven by enhanced decision-making and innovation due to better data accessibility (Firoozeh et al., 2020). Furthermore, on a global scale, implementing high-quality metadata standards could help align international data practices, supporting economic development especially in regions where data-driven decision-making is nascent.

In our forthcoming research, we intend to incorporate the actual dataset sample in order to extract metadata rather than only relying on textual descriptions. Our main objective regarding it is to create a specialized Large Language Model (LLM) that has the capability to autonomously produce all essential metadata elements related to findability, including the dataset's title, description, categories/themes, and keywords. This approach is intended to facilitate a seamless and efficient process for generating comprehensive and pertinent metadata, enhancing the findability and accessibility of the data set. This foundation also paves the way for our anticipated future research, which involves utilizing LLMs to develop more sophisticated and intelligent search mechanisms of interactive nature for discovering open datasets.

8. Conclusions

This study sheds light on the critical importance of high-quality metadata for findability. Our analysis of 15 data portals revealed that inadequate metadata significantly hindered effective search mechanisms. To address this, we focused on enriching metadata through keyword extraction. To this end, we evaluated five popular algorithms – BERT, RAKE, YAKE, TEXTRANK, and ChatGPT – for their effectiveness. Following it, we proposed our hybrid methodology BRYT, which outperformed other approaches in generating more representative keywords. Our evaluation reflected that the optimal description length for major matches ranged between 1000 to 4000 characters, and the most effective descriptions for it were those that were textual rather than structured or formatted. Furthermore, our findings showed that BRYT performed better with textual descriptions, while ChatGPT performed better with more formatted descriptions. Moreover, our analysis found the number of 10 keywords to be an efficient and effective recommendation. These findings have significant implications for data portal designers and administrators, emphasizing the importance of robust metadata and the potential benefits of keyword extraction. They will potentially help publishers annotate new datasets with representative keywords and enrich the metadata of existing datasets.

Table 6
Comprehensive summary of literature review.

Title (Year)	Authors	Contribution
Comparison of metadata quality in open data portals using the Analytic Hierarchy Process **(2018)**	Kubler, Sylvain; Robert, Jeremy; Neumaier, Sebastian; Umbrich, Jürgen; Le Traon, Yves	Evaluated metadata quality across various open data portals.
Google Dataset Search: Building a search engine for datasets in an open Web ecosystem **(2019)**	Brickley, Dan; Burgess, Matthew; Noy, Natasha	Introduced Google Dataset Search, enhancing dataset discoverability.
BERT: Pre-training of deep bidirectional transformers for language understanding **(2018)**	Devlin, Jacob; Chang, Ming-Wei; Lee, Kenton; Toutanova, Kristina	Revolutionized NLP approaches with the introduction of BERT.
YAKE! Keyword extraction from single documents using multiple local features **(2020)**	Campos, Ricardo; Mangaravite, Vítor; Pasquali, Arian; Jorge, Alípio; Nunes, Célia; Jatowt, Adam	Proposed YAKE!, an unsupervised, lightweight tool for keyword extraction.
Keyphrase Extraction from Document Using RAKE and TextRank Algorithms **(2020)**	Baruni, J; Sathiaseelan, J	Explored the effectiveness of RAKE and TextRank algorithms for keyphrase extraction.
Born for Auto-Tagging: Faster and better with new objective functions **(2022)**	Liu, Chiung-ju; Shieh, Huang-Ting	Discussed advancements in auto-tagging using novel objective functions.
Automatic keyword extraction based on dependency parsing and BERT semantic weighting **(2023)**	Liu, HuiXin	Enhanced keyword extraction through dependency parsing and BERT.
Automated metadata annotation: What is and is not possible with machine learning **(2023)**	Wu, Mingfang; Brandhorst, Hans; Marinescu, Maria-Cristina; Lopez, Joaquim More; Hlava, Margorie; Busch, Joseph	Investigated the capabilities and limitations of ML in metadata annotation.
Unlocking the Potential of Keyword Extraction: The Need for Access to High-Quality Datasets **(2023)**	Amur, Zaira Hassan; Hooi, Yew Kwang; Soomro, Gul Muhammad; Bhanbhro, Hina; Karyem, Said; Sohu, Najamudin	Highlighted the importance of high-quality datasets for keyword extraction.
Improving Performance of Automatic Keyword Extraction (AKE) Methods Using PoS-Tagging and Enhanced Semantic-Awareness **(2022)**	Altuncu, Enes; Nurse, Jason RC; Xu, Yang; Guo, Jie; Li, Shujun	Proposed enhancements in AKE through PoS-tagging and semantic awareness.
Extracting keywords of educational texts using a novel mechanism based on linguistic approaches and evolutive graphs **(2023)**	Espada, Jordán Pascual; Martinez, Jaime Solis; Rico, Irene Cid; Sánchez, Luis Emilio Velasco	Proposes innovative techniques involving linguistic and evolutionary graph approaches to enhance keyword extraction in educational texts.
Keyword Extraction Algorithm Based on Pre-training and Multi-task Training **(2022)**	Guo, Lingqi; Sun, Haifeng; Qi, Qi; Wag, Jingyu	Details the use of pre-training and multi-task learning to improve the performance and robustness of keyword extraction systems.
Evolutionary Heuristic A* Search: Pathfinding Algorithm with Self-Designed and Optimized Heuristic Function **(2019)**	Yiu, Ying Fung; Du, Jing; Mahapatra, Rabi N.	Explores the development of heuristic functions through evolutionary strategies, applicable to keyword extraction.
Knowledge-Based Optimization Algorithm **(2020)**	Cuevas, Erik; Gálvez, Jorge; Avalos, Omar	Introduces a system combining evolutionary algorithms with knowledge-based systems to enhance data extraction processes.
Inclined planes optimization algorithm in optimal architecture of MLP neural networks **(2017)**	Shahraki, Najmeh Sayyadi; Zahiri, Seyed Hamid	Illustrates the use of IPSO in optimizing neural architectures, relevant for algorithmic efficiency in keyword extraction.
Evolving Real-time Heuristic Search Algorithms **(2016)**	Bulitko, Vadim	Shows the potential of real-time heuristic search strategies evolved through simulation for dynamic keyword extraction.

(*continued on next page*)

Table 6 (*continued*).

Title (Year)	Authors	Contribution
sCAKE: semantic connectivity aware keyword extraction **(2019)**	Duari, Swagata; Bhatnagar, Vasudha	Introduced sCAKE, focusing on semantic connectivity for keyword extraction.
Complex network-based supervised keyword extractor **(2020)**	Duari, Swagata; Bhatnagar, Vasudha	Developed a supervised keyword extraction method using complex network analysis.
CERMINE: automatic extraction of structured metadata from scientific literature **(2015)**	Tkaczyk, Dominika; Szostek, Paweł; Fedoryszak, Mateusz; Dendek, Piotr Jan; Bolikowski, Łukasz	Introduced CERMINE for automatic metadata extraction from scientific literature.
Roomba: Automatic validation, correction, and generation of dataset metadata **(2015)**	Assaf, Ahmad; Senart, Aline; Troncy, Raphaël	Developed Roomba for automated dataset metadata management.
A comprehensive survey on gravitational search algorithm **(2018)**	Rashedi, Esmat; Nezamabadi-pour, Hossein; Saryazdi, Saeid	Explored Gravitational Search Algorithm (GSA) as a novel optimization tool inspired by Newtonian laws of gravity and motion.
Constrained Optimization Using Gravitational Search Algorithm **(2013)**	Yadav, Anupam; Deep, Kusum	Discusses the application of GSA to optimize various algorithmic tasks, potentially applicable to keyword extraction.
GSA: A Gravitational Search Algorithm **(2009)**	Rashedi, Esmat; Nezamabadi-pour, Hossein; Saryazdi, Saeid	Developed the foundational principles of GSA, highlighting its effectiveness in various optimization scenarios.
Automatic keyphrase extraction based on NLP and statistical methods **(2011)**	Dostal, Martin; Ježek, Karel	Combined statistical methods and NLP for enhanced keyphrase extraction.

9. Declarations

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 955569 and from European Union – NextGenerationEU - Piano Nazionale di Ripresa e Resilienza, Missione 4 Istruzione e Ricerca - Componente 2 Dalla ricerca all'impresa - Investimento 1.5, ECS_00000041 VITALITY - Innovation, digitalization and sustainability for the diffused economy in Central Italy.

CRediT authorship contribution statement

Umair Ahmed: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original oraft, Writing – review & editing Visualization. **Charalampos Alexopoulos:** Conceptualization, Resources, Writing – review & editing, Supervision. **Marco Piangerelli:** Conceptualization, Resources, Writing – review & editing, Supervision. **Andrea Polini:** Conceptualization, Methodology, Resources, Writing – review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The dataset reflecting the conclusions of this article are available in the BRYT-AKE repository at https://github.com/umairahmedq/BRYT-AKE.

Appendix A. List of Abbreviations

Table A.7 reports the list of abbreviations used in this document:

Table A.7
List of abbreviations.

Abbreviation	Description
AI	Artificial Intelligence
CI	Collective Intelligence
ML	Machine Learning
NLP	Natural Language Processing
LLM	Large Language Model
ChatGPT	Chat Generative Pre-trained Transformer
BERT	Bidirectional Encoder Representations from Transformers
RAKE	Rapid Automatic Keyword Extraction
YAKE	Yet Another Keyword Extractor
EDP	European Data Portal

Appendix B. List of variables

Table B.8 reports the list of variables used in this study:

Appendix C. List of control parameters

Table C.9 reports the list of control parameters used in this study:

Table B.8
List of variables.

Variable	Description
Title	Original title fetched from EDP
Description	Original description fetched from EDP
Translated Description	Description translated into English
Keywords	Original keywords fetched from EDP
Translated Keywords	Keywords translated into English
BERT Keywords	Keywords extracted using BERT
YAKE Keywords	Keywords extracted using YAKE
RAKE Keywords	Keywords extracted using RAKE
TEXTRANK Keywords	Keywords extracted using TEXTRANK
ChatGPT Keywords	Keywords extracted using ChatGPT
BRYT Keywords	Keywords extracted using BRYT
No of Keywords	Total no of keywords to be extracted
Jaccard Threshold	Jaccard threshold to match original and extracted keywords
Gestalt Threshold	Gestalt threshold to match original and extracted keywords
Matching score (Each Method)	Score of no of matching keywords extracted by each method

Table C.9
List of control parameters.

Control parameter	Description
Jaccard Similarity Threshold	Defines the minimum score for considering two keywords as matching
Gestalt Matching Threshold	Defines the minimum score for considering two keywords as matching
Number of Keywords	Specifies the maximum number of keywords to be extracted
Description Length	Defines the minimum description lengths considered for keyword extraction
Prompt	It defines the input prompt that we engineered for out methodologies

References

Ahmed, U. (2023). Reimagining open data ecosystems: a practical approach using AI, CI, and Knowledge Graphs. In *Joint proceedings of the BIR 2023 workshops and doctoral consortium co-located with 22nd international conference on perspectives in business informatics research* (pp. 235–249). CEUR-WS.

Altuncu, E., Nurse, J. R., Xu, Y., Guo, J., & Li, S. (2022). Improving performance of automatic keyword extraction (AKE) methods using PoS-tagging and enhanced semantic-awareness. arXiv preprint arXiv:2211.05031.

Amur, Z. H., Hooi, Y. K., Soomro, G. M., Bhanbhro, H., Karyem, S., & Sohu, N. (2023). Unlocking the potential of keyword extraction: The need for access to high-quality datasets. *Applied Sciences*, *13*(12), 7228.

Assaf, A., Senart, A., & Troncy, R. (2015). Roomba: Automatic validation, correction and generation of dataset metadata. In *Proceedings of the 24th international conference on world wide web* (pp. 159–162).

Baruni, J., & Sathiaseelan, J. (2020). Keyphrase extraction from document using RAKE and TextRank algorithms. *International Journal of Computing Science and Mobile Computing*, *9*, 83–93.

Beliga, S. (2014). *Keyword extraction: a review of methods and approaches*: *vol. 1*, (no. 9), Rijeka: University of Rijeka, Department of Informatics.

Brickley, D., Burgess, M., & Noy, N. (2019). Google dataset search: Building a search engine for datasets in an open web ecosystem. In *The world wide web conference* (pp. 1365–1375).

Bulitko, V. (2016). Evolving real-time heuristic search algorithms. In *Artificial life conference proceedings* (pp. 108–115). Cambridge, MA 02142-1209, USA journals-info . . . : MIT Press One Rogers Street.

Campobello, G., Segreto, A., Zanafi, S., & Serrano, S. (2017). RAKE: A simple and efficient lossless compression algorithm for the internet of things. In *2017 25th European signal processing conference* (pp. 2581–2585). IEEE.

Campos, R., Mangaravite, V., Pasquali, A., Jorge, A., Nunes, C., & Jatowt, A. (2020). YAKE! Keyword extraction from single documents using multiple local features. *Information Sciences*, *509*, 257–289.

Correa, A. S., Zander, P. O., & Da Silva, F. S. C. (2018). Investigating open data portals automatically: a methodology and some illustrations. In *Proceedings of the 19th annual international conference on digital government research: governance in the data age* (pp. 1–10).

Cuevas, E., Gálvez, J., Avalos, O., Cuevas, E., Gálvez, J., & Avalos, O. (2020). Knowledge-based optimization algorithm. *Recent metaheuristics algorithms for parameter identification*, 245–277.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Dostal, M., & Ježek, K. (2011). Automatic keyphrase extraction based on NLP and statistical method. In *Proceedings of the dateso 2011: annual international workshop on databases, texts, specifications and objects* (pp. 140–145). Západočeská univerzita v Plzni.

Duari, S., & Bhatnagar, V. (2019). sCAKE: semantic connectivity aware keyword extraction. *Information Sciences*, *477*, 100–117.

Duari, S., & Bhatnagar, V. (2020). Complex network based supervised keyword extractor. *Expert Systems with Applications*, *140*, Article 112876.

Espada, J. P., Martínez, J. S., Rico, I. C., & Sánchez, L. E. V. (2023). Extracting keywords of educational texts using a novel mechanism based on linguistic approaches and evolutive graphs. *Expert Systems with Applications*, *213*, Article 118842.

EU (2023). European data portal. https://data.europa.eu/en. (Accessed 28 November 2023).

Firoozeh, N., Nazarenko, A., Alizon, F., & Daille, B. (2020). Keyword extraction: Issues and methods. *Natural Language Engineering*, *26*(3), 259–291.

Guo, L., Sun, H., Qi, Q., & Wang, J. (2022). Keyword extraction algorithm based on pre-training and multi-task training. In *Proceedings of sixth international congress on information and communication technology: ICICT 2021, London, vol. 1* (pp. 723–734). Springer.

Kubler, S., Robert, J., Neumaier, S., Umbrich, J., & Le Traon, Y. (2018). Comparison of metadata quality in open data portals using the Analytic Hierarchy Process. *Government Information Quarterly*, *35*(1), 13–29.

Lahitani, A. R., Permanasari, A. E., & Setiawan, N. A. (2016). Cosine similarity to determine similarity measure: Study case in online essay assessment. In *2016 4th international conference on cyber and IT service management* (pp. 1–6). IEEE.

Lin, T., Wang, Y., Liu, X., & Qiu, X. (2022). A survey of transformers. *AI Open*.

Liu, H. (2023). Automatic keyword extraction based on dependency parsing and BERT semantic weighting. In *Third international seminar on artificial intelligence, networking, and information technology, vol. 12587* (pp. 325–330). SPIE.

Lnenicka, M. (2015). An in-depth analysis of open data portals as an emerging public e-service. *International Journal of Humanities and Social Sciences*, *9*(2), 589–599.

Lnenicka, M., & Nikiforova, A. (2021). Transparency-by-design: What is the role of open data portals? *Telematics and Informatics*, *61*, Article 101605.

Merrouni, Z. A., Frikh, B., & Ouhbi, B. (2016). Automatic keyphrase extraction: An overview of the state of the art. In *2016 4th IEEE international colloquium on information science and technology* (pp. 306–313). http://dx.doi.org/10.1109/CIST.2016.7805062.

Miah, M. S. U., Sulaiman, J., Sarwar, T. B., Zamli, K. Z., & Jose, R. (2021). Study of keyword extraction techniques for electric double-layer capacitor domain using text similarity indexes: an experimental analysis. *Complexity*, *2021*, 1–12.

Mihalcea, R., & Tarau, P. (2004). Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing* (pp. 404–411).

Niemann, K. (2015). Increasing the accessibility of learning objects by automatic tagging. In *Proceedings of the fifth international conference on learning analytics and knowledge* (pp. 414–415).

Niwattanakul, S., Singthongchai, J., Naenudorn, E., & Wanapu, S. (2013). Using of jaccard coefficient for keywords similarity. In *Proceedings of the international multiconference of engineers and computer scientists, vol. 1, no. 6* (pp. 380–384).

Nogueras-Iso, J., Lacasta, J., Ureña-Cámara, M. A., & Ariza-López, F. J. (2021). Quality of metadata in open data portals. *IEEE Access*, *9*, 60364–60382.

OpenKnowledge (2023). Open data definition. https://opendatahandbook.org/glossary/en/terms/open-data/. (Accessed 28 November 2023).

Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2009). GSA: a gravitational search algorithm. *Information Sciences*, *179*(13), 2232–2248.

Rashedi, E., Rashedi, E., & Nezamabadi-Pour, H. (2018). A comprehensive survey on gravitational search algorithm. *Swarm and Evolutionary Computation*, *41*, 141–158.

Ratcliff, J. W., & Metzener, D. E. (1988). Pattern-matching-the gestalt approach. *Dr Dobbs Journal*, *13*(7), 46.

Rose, S., Engel, D., Cramer, N., & Cowley, W. (2010). Automatic keyword extraction from individual documents. *Text Mining: Applications and Theory*, 1–20.

Schauppenlehner, T., & Muhar, A. (2018). Theoretical availability versus practical accessibility: The critical role of metadata management in open data portals. *Sustainability*, *10*(2), 545.

Shahraki, N. S., & Zahiri, S. H. (2017). Inclined planes optimization algorithm in optimal architecture of MLP neural networks. In *2017 3rd international conference on pattern recognition and image analysis* (pp. 189–194). IEEE.

Škoda, P., Bernhauer, D., Nečaskỳ, M., Klímek, J., & Skopal, T. (2020). Evaluation framework for search methods focused on dataset findability in open data catalogs. In *Proceedings of the 22nd international conference on information integration and web-based applications & services* (pp. 200–209).

Song, M., Jiang, H., Shi, S., Yao, S., Lu, S., Feng, Y., et al. (2023). Is ChatGPT a good keyphrase generator? A preliminary study. arXiv preprint arXiv:2303.13001.

Tkaczyk, D., Szostek, P., Fedoryszak, M., Dendek, P. J., & Bolikowski, Ł. (2015). CERMINE: automatic extraction of structured metadata from scientific literature. *International Journal on Document Analysis and Recognition (IJDAR)*, *18*, 317–335.

Van Loenen, B., Zuiderwijk, A., Vancau-Wenberghe, G., Lopez-Pellicer, F. J., Mulder, I., Alexopoulos, C., et al. (2021). Towards value-creating and sustainable open data ecosystems: A comparative case study and a research agenda. *JeDEM-eJournal of eDemocracy and Open Government*, *13*(2), 1–27.

Wu, M., Brandhorst, H., Marinescu, M. C., Lopez, J. M., Hlava, M., & Busch, J. (2023). Automated metadata annotation: What is and is not possible with machine learning. *Data Intelligence*, *5*(1), 122–138.

Xueqiang, L., Zhi'an, D., & Xindong, Y. (2019). Keywords extraction method. Patent No. CN102541910A, Available at https://patents.google.com/patent/CN102541910A/en?oq=CN102541910A.

Yadav, A., & Deep, K. (2013). Constrained optimization using gravitational search algorithm. *National Academy Science Letters*, *36*, 527–534.

Yiu, Y. F., Du, J., & Mahapatra, R. (2019). Evolutionary heuristic A* search: Pathfinding algorithm with self-designed and optimized heuristic function. *International Journal of Semantic Computing*, *13*(01), 5–23.