# An Extensible General-Purpose Data Gathering and Classification Platform: Maestro v2023

***António Miguel Martins***
*INESC-ID, Instituto Superior Técnico, Universidade de Lisboa*
*Lisboa, Portugal* *antonio.valente.martins@tecnico.ulisboa.pt*

***Alberto Rodrigues da Silva***
*INESC-ID, Instituto Superior Técnico, Universidade de Lisboa*
*Lisboa, Portugal* *alberto.silva@tecnico.ulisboa.pt*

***Jacinto Estima***
*INESC-ID, Lisboa, Portugal*
*CISUC, Dep. of Informatics Engineering, University of Coimbra*
*Coimbra, Portugal* *estima@dei.uc.pt*

**Abstract**

The Maestro platform has been developed with the purpose of helping researchers automatically gather and classify data on specific topics, which has commonly been done using several purpose-specific tools in a tedious process. This paper introduces and discusses Maestro v2023, an improvement over the original Maestro platform. This new version of the platform enhances its capabilities and extends its applicability in various scenarios. Notable improvements include the introduction of text data types, the addition of a new analysis stage to Maestro's pipeline, numerous usability-oriented enhancements, and the implementation of various other features. Furthermore, several conceptual and architectural aspects of Maestro were re-designed and extended. This research follows the design science research methodology, an iterative methodology that combines principles, practices, and procedures to provide guidance for research in information systems. Maestro v2023's capabilities are demonstrated by showcasing a scenario focused on streamlining literature reviews. Maestro v2023's usability and relevance are also evaluated through a user assessment session and the results are compared to those obtained in a similar user assessment of Maestro's original iteration. The results of this evaluation demonstrate a significant improvement in the perceived usability of the platform, as well as a continued interest in using Maestro v2023 in the future.

**Keywords:** Data gathering, Data classification, Data analysis, Text analytics, Machine learning.

## 1. Introduction

Thanks to The World Wide Web (commonly referred to as the Web), we now have unprecedent access to vast amounts of specialized knowledge that was once arduous to share and discover.

The Web is home to an immense amount of information that is stored in various forms of data, including text, sound, video, and other representative structures [1], [2]. The field of ML has harnessed this ability to access vast amounts of digitized data, leading to the exponential growth and rapid improvement in major tasks. The area of data classification stands out as a prime beneficiary of this approach, as complex classification algorithms require a large amount of quality-data to be trained effectively. As the amount of digitized data continues to grow, their use in research will only become more prevalent, driving the development of new mechanisms to facilitate the process.

Maestro was developed to streamline some of these processes, serving as a modular, extensible, and configurable platform for data gathering and data classification [3], [4]. Although the platform's primary goal is data gathering and classification, it also allows for additional modular steps, such as data filtering and post-processing, further expanding

its range of applications. This paper aims to present how Maestro v2023, an improved version of the original platform, can be helpful to the scientific community, namely by automating several aspects of the literature review process. From now on, we will refer to the original Maestro platform as "Maestro v2022" and the proposed revamped version resulting from the expansion as "Maestro v2023". The term "Maestro" will refer to the overall concept of the platform.

## 2. Background

This section describes various aspects and concepts related to Maestro v2023, including some of the approaches and tools employed, as well as related areas of research.

### 2.1. Data Classification

Data classification is a crucial process in ML projects. Essentially, it involves using a classification algorithm (i.e., a classifier) to assign a label to each object in a given dataset. To do this, the algorithm must undergo a training phase using a labeled dataset, in which each data item is paired with its respective label. Once trained, the classifier can then classify new, unlabeled data items [5] with a high degree of accuracy. In addition to the initial training phase, classification algorithms can be further adapted through a process called "fine-tuning", where the outputs are modified or extended to meet specific needs.

Data classification is a crucial process in various fields as it helps establish relationships between data and their corresponding classes. For Maestro, we place significant emphasis on this process as it serves as a bridge between data collection and classification.

### 2.2. Data Gathering

Gathering data is another process that Maestro handles directly. With the amount of available data on the web, it only makes sense that modern data collection techniques have evolved to collect it. This can be done in various ways. One such method would be through the usage of web crawlers. They allow for the discovery of very large amounts of data, which can be specialized through the usage of preferential crawlers. However, web crawling is mostly focused on discovering URLs. To extract the data itself, another method that can be used is data scraping.

Web data scraping focuses on methods of extracting data from specific sources, such as websites or databases. One such method revolves around using specialized APIs. The Twitter API [6] allows users to retrieve data from the Twitter platform, such as media, tweets, or users. The Bing Images API [7] is another example of a scraping API - in this case specialized for the retrieval of image type data.

Some web crawler frameworks recognize the data structure of a page automatically, removing the need to differentiate between the steps of web crawling and scraping. This is the case for Scrapy [8], a python-based web crawling framework. By utilizing the correct set of tools for data gathering, one can eliminate many of the intermediate steps in this process, enabling researchers and users to decrease the time and effort needed to produce datasets.

### 2.3. Text Analytics

Text analytics, or text mining, is the process of examining and extracting meaningful insights, patterns, and information from unstructured text data [9]. The fields of ML and natural language processing (NLP) have become intimately tied to the field of text analytics, driving research in the advancement of several tasks associated with text analytics, such as text summarization and simplification. These tasks are relevant to the development of Maestro v2023, as it introduces new text data types and, consequently, may aid in scenarios that require these tasks to be pursued.

The area of text summarization aims to allow the condensation of documents and publications. When done correctly, the produced summaries are expected to highlight the critical aspects of these artifacts, effectively undermining the need to sift through a large amount of redundant information.

Different trends and techniques form the basis for research within this field. A recent study by Widyassari et al. [10] systematically reviews automatic text summarization by analyzing different publications published from 2008 to 2019. They identified ML approaches as the most predominant technique, being used in more than half of the studies analyzed.

Text simplification aims to make complex language easier to understand by rephrasing it into simpler terms and typically involves making use of three core elements: splitting, deletion, and paraphrasing. Splitting involves breaking lengthy sentences into several smaller sentences that enhance the readability of the overall text. Deletion discards a sentence's extraneous and less consequential parts, thereby reducing its complexity. Finally, paraphrasing is used to reorder, substitute, and, in some cases, expand sentence constructs to achieve a simplified version of the original text [11].

## 3. Maestro Key Concepts and Workflow

This section presents the underlying philosophy and architecture of Maestro, along with an overview of the major improvements introduced in Maestro v2023.

### 3.1. Maestro v2023's Architecture

Maestro was created to gather and classify data as a service. It functions in a modular, extensible, and configurable fashion, enabling users within an organization to automatically collect and perform ML related tasks on data of various types (e.g., images, sound, text). Fig. 1 represents Maestro's domain model in UML, highlighting its top-level concepts.

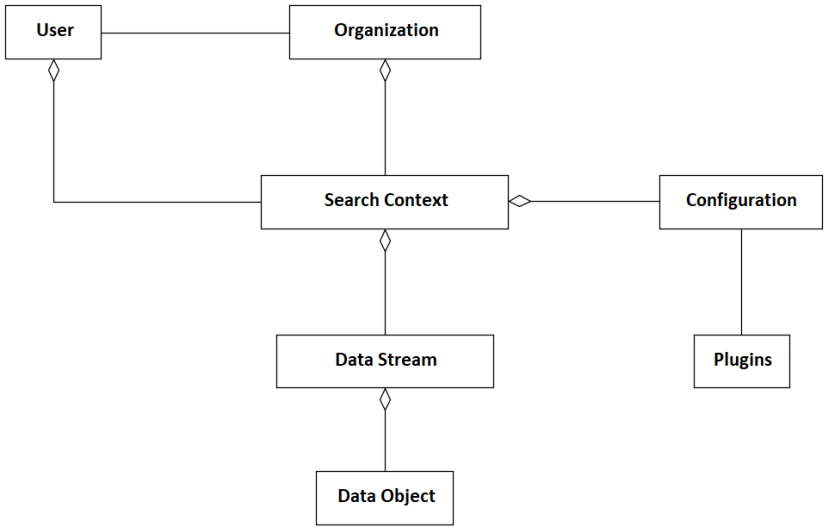


**Fig. 1.** Maestro v2023 top-level concepts (UML diagram).

Maestro possesses three top-level key concepts: organizations, users, and search contexts. Maestro users can be associated with one or more organizations, which exist to facilitate collaboration and simultaneous workflows. Users can define multiple workflows configured to gather, classify (or perform other ML processes), analyze the results, and deliver their target data. These workflows are named search contexts, and function as declarative expressions of the tasks to be run through Maestro's pipeline.

Maestro's behavior during the execution of a search context can be configured by the

user, who may select several plugins to be used during each stage of Maestro's pipeline. Furthermore, the user defines the data type used, what data to search for, and the sources for this data, during the configuration of their search context.

### *Plugins-based Pipeline*

As illustrated in Fig. 2, Maestro v2023 supports a pipeline that, once run, results in a classified dataset that can be provided to external services or analyzed within the platform itself. Maestro v2023's pipeline comprises ten essential steps or stages, namely: (1) Create / Configure a search context; (2) Start a configured search context; (3) Fetch URLs pointing to objects of the desired data type, as well as additional information regarding the object; (4) Gather the resources or data items from the fetched URLs; (5) Review the gathered data items and manually discard those deemed irrelevant; (6) Post-process the gathered data with the use of plugins, acquiring additional parameters for the subsequent stages (e.g., adding metadata to image data); (7) Filter the data according to the specified plugins and parameters defined by the user (e.g., filtering based on the date and location of a given data item); (8) Classify/Perform ML tasks on the dataset items using the desired plugins; (9) Provide the resulting classified dataset to external services; (10) Produce data regarding the data stream using the specified plugins and generate charts that allow the user to visually analyze their data.

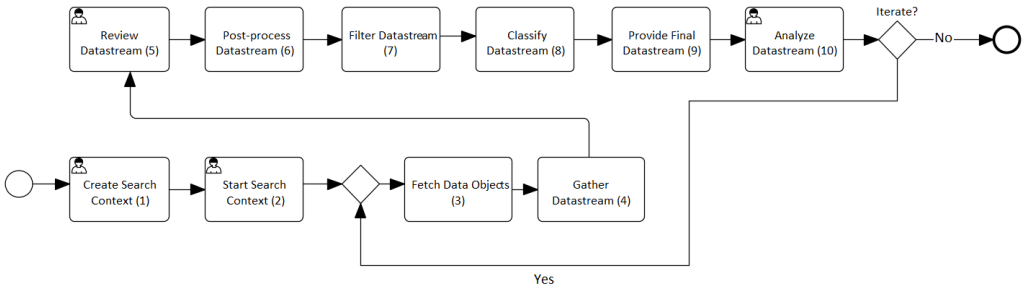


**Fig. 2.** Maestro v2023's Workflow (BPMN Process diagram).

Plugins are user-made scripts that follow a common interface that Maestro understands. Similarly to Maestro v2022, in Maestro v2023 plugins can be applied to most stages of the pipeline (stage three and six to ten), dictating how the system should handle the data objects when that stage is reached. It is relevant to mention that, despite its name, other ML processes beyond classification can be performed during the "Classify Data stream" (8) stage.

## 3.2. Major Modifications

The original iteration of Maestro, Maestro v2022, implemented the majority of systems detailed previously. It allows users to use the system to gather and classify both image data and sound data types.

However, Maestro v2022 still bore some limitations that needed to be addressed, namely: limited types of data, inability to do in-depth analysis of the pipeline, usability issues, and other constraints. Thus, a new iteration of the platform, Maestro v2023, began being developed to expand and refine Maestro's capabilities.

**Text data types**. One of the main extensions added to Maestro v2023 was the introduction of text data types. In Maestro v2022, only image and sound data was available. As text represents a significant amount of the available data on the Web, a general text data type, as well as two specialized text data types, news article data and scientific paper data, were introduced.

**Analysis stage**. The analysis of gathered data, as well as the remaining outputs provided by Maestro during a search context run, allows users to better understand the nature of their data, and how to improve the configurations for future runs. As such, an

analysis stage was introduced to Maestro's pipeline, focusing on the analysis of information regarding the data objects, as well as additional information produced by Maestro's pipeline. This is achieved through the usage of plugins, which allow for the generation of charts with which users may visually analyze their data streams.

**Usability improvements**. For Maestro to be useful to as many different types of users and scenarios as possible, it ought to remain undemanding and intuitive. While Maestro v2022's usability was generally well rated by users, it was still the platform's lowest ranking aspect. By harnessing the feedback provided in the evaluation phase of Maestro v2022's development, several changes to the UI and structure were introduced in Maestro v2023, including: changing several aspects of the platform's UI; re-writing and expanding the documentation for Maestro; removal of unnecessary options, features, and pages; redesigning several features, such as the review process and manual submission of data objects into a search context, for improved usability.

**Other changes**. Several other changes were also introduced into Maestro v2023, such as advanced search strings, which allows you users to utilize propositional logic when defining a search string, utilization of multiple classifiers in a run, and allowing classification plugins to perform other ML tasks, beyond classification. Each of these individual aspects may not single-handedly make a big impact on the platform's quality, however, when all their contributions are accounted for, we believe they provide significant benefit to Maestro's design.

## 4. Demonstration

To demonstrate Maestro v2023's usefulness, we showcase how the platform can be used to streamline the literature review process. To do this, we developed a set of plugins to be used in search contexts for scientific paper data. Table 1 describes the developed plugins. The following steps were followed to make use of Maestro in this context, along with the developed plugins.

### 4.1. Essential Configurations

The user creates a search context through Maestro's interface. The user must define an owner, title, unique code, and a description for their search context.

Once created, the user must configure their search context. As shown in Fig. 3, the user defines the essential configurations, which are mandatory. They define the search string for finding the data as "(automatic $OR semi-automatic) $AND literature review $AND (system $OR program)", the data type ("Scientific Paper"), as well as other options that allow the search context to automatically run again after a certain amount of time (in this case, we set it to "Don't repeat").

**Table 1.** Description of the developed plugins for supporting scientific publications.

| Name | Plugin Type | Description |
|---|---|---|
| Elsevier Fetcher | Fetcher | Queries the Elsevier API [12] for scientific publications using a search string. |
| ArXiv Fetcher | Fetcher | Queries the ArXiv API [13] for scientific publications using a search string and tags. |
| Duplicate Filter | Filter | Removes duplicates of scientific publications by comparing the descriptor (title or DOI). |
| Paper Summarizer | Classifier | Generates summaries of scientific publications using the BARTxiv model [14]. |
| Abstract Simplifier | Classifier | Rewrites difficult-to-understand scientific abstracts into simpler, easier-to-read versions using the SAS model [15]. |
| Keyword Extractor | Classifier | Extracts relevant keywords from paper abstracts using KeyBERT [16]. |

### 4.2. Advanced Configurations

The user can then define the advanced configurations. In spite of these settings not being mandatory, the system will do nothing if they are not configured.

In this phase, the user shall proceed by conducting the following tasks: select the "Elsevier API" and "ArXiv API" fetching plugins for fetching URLs of scientific papers related to the search string; manually submit any previously gathered data objects to be included in the data stream; select the "Yield data after gathering" option to allow for manual data stream review; select the "Paper Summarizer", "Abstract Simplifier", "Keyword Extractor" plugins, to be considered during the classification step (see Fig. 4); apply filtering configurations in order to discard any duplicates of gathered articles, by checking their DOI and/or Title; though optional, the user may also specify the configurations for an HTTP Rest endpoint to which the data will be sent during the providing step; finally, the user selects the "Citations Analyzer" and "Publication Date Analyzer" plugins to generate charts for data object analysis.

### 4.3. Search Context Run and Results

Once the configurations have been defined, the user triggers the run of the search context, and waits for the results. This process runs in the background, and may take some time to complete. Once the classification stage ends, the user accesses the "Results" page to inspect the data object that were not filtered, as well as the respective results provided by the classification plugins. Furthermore, the system sends the results to the configured endpoint, which the user may then utilize as input for external tools and processes.

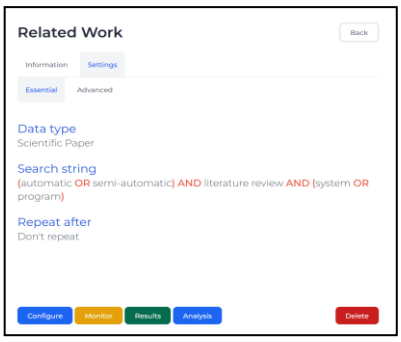


**Fig. 3.** Configuration of search context in the Maestro platform.

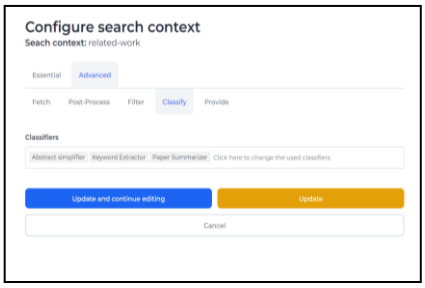


**Fig. 4.** Configuration of three data classification plugins for the classification stage of a search context: Paper Summarizer, Abstract Simplifier and Keyword Extractor.

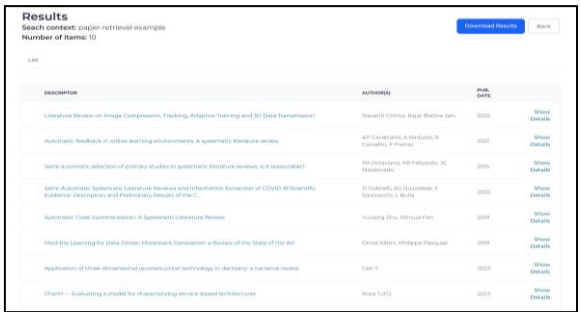


**Fig. 5.** Maestro v2023's results page for a scientific paper data stream.


**Fig. 6.** View of the charts produced using the "Citations Analyzer" and "Publications Date Analyzer" plugins.

As shown in Fig. 5, the system provided the user with multiple scientific publications related to the defined search string, summarized them, simplified the abstract, and extracted relevant keywords from the original abstract. Selecting the "Show Details" option allows users to see each data object in finer detail.

### 4.4. Data stream Analysis

After the analysis stage ends, the user may then access the charts generated with the help of the data produced using the analysis plugins. As presented in Fig. 6, the user is able to visually analyze information regarding the produced data stream. The generated charts present a distribution of the number of citations found for the gathered papers, as well as the gathered papers' publication year.

## 5. Literature Review

This section introduces, analyses, and discusses research and work that relates to Maestro, providing a comparison between Maestro's iterations and these related works.

### 5.1. Related Work

Data gathering and data classification are highly researched areas of study. Several research projects and works on these areas exist, either focusing on a specific one or combining them.

Much research has been done on the process of training ML classification algorithms using both supervised and unsupervised [17] techniques. In these works, there is often a data gathering step, followed by a training phase, with the ultimate aim of training a classification algorithm.

Dilrukshi et al. [18] follows this structure, where the Twitter API [6] was used to gather tweets containing news headlines from certain micro-blogs. They then manually labelled these short texts into different categories and, using a Support Vector Machine [19] method of supervised learning, trained a classifier with the purpose of labelling news articles into categories.

Various data gathering systems have also been developed and applied with the purpose of categorizing retrieved data for various purposes. In [20], [21], K. Yanai proposes an image collecting system, which allows for the gathering of web images employing keyword-based search engines. Later, the same author expanded upon his work by proposing a system [22] that used the gathered images to train a generic image classification algorithm, thus allowing images to be provided to the system for classification.

The InTime [23] platform is another project with relevant similarities. It allows users to configure web sources from where to identify and extract data related to the Cyber-Threat Intelligence (CTI) domain. This data can then be classified using ML algorithms to extract relevant CTI artifacts and export them to a MISP database [24], an open-source collaborative threat intelligence sharing platform.

Another popular tool is the "Publish or Perish" (PoP) software, developed by Anne-Wil Harzing [25]. This software program retrieves and analyzes academic citations from multiple sources, presenting academics with several citation metrics. Though it does not make use of classification algorithms, the analysis step derives additional information based on the retrieved citations, similarly to what is done in Maestro's pipeline during its post-processing and proposed analysis phase.

### 5.2. Critical Analysis

Both data gathering and data classification are fields that continue to grow. New data gathering techniques often arise with the purpose of addressing different challenges, especially with the increasing need for high amounts of data in machine learning paradigms, data classification being no exception.

Research projects related to data gathering, data classification, and the development of systems that integrate both fields of study, show significant diversity and provide valuable contributions to these fields. There are, however, limitations to all the previously showcased systems. The ability to provide this data to external services, the ability to do statistical analysis on the data, and collaborative approaches, such as Maestro's organizational framework, are generally unavailable. Furthermore, all of these works are highly specialized, focusing on specific types of data and/or domains in which they can be applied. Maestro aims to go beyond individual use-cases, allowing organizations to apply the system in many research contexts. A summary of the comparison between these systems and both iterations of Maestro is provided in Table 2.

Table 2. Comparison between Maestro's iterations and related works.

| | **Processes Supported** | | | | **Features Supported** | | | |
|---|---|---|---|---|---|---|---|---|
| **Work** | Gather Data | Classify Data | Analyze Data | Automatically Provide Data | Collaborative Capabilities | Multiple Data types | Simultaneous use of Multiple Classifiers | Extensible Plugins |
| Maestro V2023 | Yes | Yes | Yes | Yes | Yes | Yes[a] | Yes | Yes |
| Maestro V2022 [3] | Yes | Yes | No | Yes | Yes | Yes[b] | No | Yes |
| InTime [23] | Yes | Yes | Yes | No | No | No | No | Yes |
| PoP [25] | Yes | No | Yes | No | No | No | No | No |
| Extended image collector [22] | Yes | Yes | No | No | No | No | No | No |
| Image collector [20], [21] | Yes | No | No | No | No | No | No | No |

[a] *Image data, sound data, and text data.*
[b] *Image data and sound data.*

## 6. Evaluation

To assess Maestro v2023's usability and relevance, we followed a similar approach to the one employed in Maestro v2022, in which a user test session was prepared. To support this test session, a user guide was produced, detailing the tasks necessary to perform the case study described in section 4, as well as a questionnaire for the testers to fill out regarding their perceived experience when following the guide. In total, 18 users participated in the test session and filled out the questionnaire.

When asked to rate Maestro v2023 in different categories, using a Likert scale from 1 to 5 (where 1 is Very Poor and 5 Very Good), usability received an average score of 4.5, compared to the value of 4.05 in the original evaluation of Maestro v2022. This demonstrates a marked improvement in the perceived usability of Maestro. Furthermore, when asked whether they could see themselves using Maestro v2023 in their daily lives, 72.2% of users reported a definitive interest, 11.1% reported some interest, with only 16.6% of users reporting no interest in using the platform further.

Participants were also asked to rate their experience using the platform using a Likert scale from 1 to 5 (where 1 is Very Poor and 5 Very Good), and the results show an average score of 4.55. The feedback provided by the participants further reflects their overall positive impression when using Maestro v2023. "The UI is easy on the eyes", "I would definitely use Maestro", and "This was surprisingly fast" are all examples of the feedback provided by the users. Nonetheless, the respondents did provide bug reports and

feedback regarding the UI, which were taken into account and fixed, posteriorly.

In summary, the results of this user assessment seemed positive and promising. Most participants seemed to like Maestro v2023, both in terms of usability, as well as usefulness. Though some negative outliers exist, we believe the results were still well within our expected outcomes. The biggest limitation we found in regard to this approach was the relatively small number of testers. For future iterations of Maestro, attracting a larger number of testers would surely be beneficial.

## 7. Conclusion

In this work we present the work done during the development of Maestro v2023, the second iteration of the Maestro platform. A wide array of improvements to Maestro were added in this new iteration, greatly expanding the number of scenarios in which Maestro may be of use. We believe the addition of text data types, introduction of a new analysis stage to Maestro's pipeline, several usability-oriented improvements, along with numerous other modifications and features, have led to an increase in Maestro's quality and usefulness.

This expectation is supported by the results attained from the evaluation of Maestro v2023, during which several users participated in a test session, following the steps necessary to perform the scenario outlined in the scenario proposed in section 4. Analysis of these results shows that, in general, some of Maestro v2022's biggest limitations were successfully addressed, as the perceived usability of the platform increased. Furthermore, several of the participants showed interest in Maestro v2023's potential to be of use in their daily lives.

The increase in Maestro's quality following this new iteration, as well as the positive results of our evaluation, leads us to believe in Maestro's potential to impact several different fields, such as business management and scientific research. Maestro v2023's flexibility and depth of functionalities allows it to serve as an intermediary tool for many different projects, bridging the gap between data gathering and the manipulation of data using different ML mechanisms.

## References


1. Internet Assigned Numbers Authority [IANA]. (2023, January 4). Media Types. Retrieved August 25, 2023, from https://www.iana.org/assignments/media-types/media-types.xhtml.
2. Nagel, S. (2022). Statistics of Common Crawl Monthly Archives by commoncrawl. Retrieved August 25, 2023, from https://commoncrawl.github.io/cc-crawl-statistics/plots/mimetypes.
3. Serra, Alexandre & Estima, Jacinto & Rodrigues da Silva, Alberto. (2022). Maestro: An Extensible General-Purpose Data Gathering and Classification System. Proceedings of ISD'2022. A15. 10.13140/RG.2.2.26824.80646.
4. Magalhães Serra, A., Estima, J., & Rodrigues da Silva, A. (2023). Evaluation of Maestro, an extensible general-purpose data gathering and data classification platform. Information Processing & Management [in Press].
5. Aggarwal, C. C. (2020). Data Classification: Algorithms and Applications (Chapman & Hall/CRC Data Mining and Knowledge Discovery Series) (1st ed.). Chapman and Hall/CRC.
6. Twitter API | Products. Twitter Developer Platform. Retrieved May 25, 2023, from https://developer.twitter.com/en/products/twitter-api.
7. Bing Image Search API | Microsoft Bing. Bingapis. Retrieved May 25, 2023, from https://www.microsoft.com/en-us/bing/apis/bing-image-search-api.
8. Scrapy | A Fast and Powerful Scraping and Web Crawling Framework. (2022). Retrieved November 25, 2022, from https://scrapy.org/.
9. Chakraborty, G., Pagolu, M., Garla, S. (2013). Introduction to Text Analytics. In Text Mining and Analysis: Practical Methods, Examples, and Case Studies Using SAS (pp. 1–

17). essay, SAS Institute Inc.

10. Widyassari, A. P., Rustad, S., Shidik, G. F., Noersasongko, E., Syukur, A., Affandy, A., & Setiadi, D. R. I. M. (2020). Review of automatic text summarization techniques & methods. Journal of King Saud University - Computer and Information Sciences, 34(4), 1029–1046. doi: 10.1016/j.jksuci.2020.05.006.
11. Xu, W., Callison-Burch, C., & Napoles, C. (2015). Problems in Current Text Simplification Research: New Data Can Help. Transactions of the Association for Computational Linguistics, 3, 283–297. doi: 10.1162/tacl_a_00139.
12. Elsevier Developer Portal. (2023). Elsevier. Retrieved August 20, 2023, from https://dev.elsevier.com/.
13. ArXiv API (2023). ArXiv. Retrieved August 20, 2023, from https://info.arxiv.org/help/api/index.html.
14. Du, J. (2022, December). BARTxiv. Hugging Face. Retrieved August 10, 2023, from https://huggingface.co/kworts/BARTxiv.
15. Wang, H. (2022). Scientific abstract simplification. Hugging Face. Retrieved August 10, 2023, from https://huggingface.co/haining/scientific_abstract_simplification.
16. Grootendorst, M. (2022). KeyBERT: Minimal keyword extraction with BERT. GitHub. Retrieved August 10, 2023, from https://github.com/MaartenGr/KeyBERT.
17. Badillo, S., Banfai, B., Birzele, F., Davydov, I. I., Hutchinson, L., Kam-Thong, T., Siebourg-Polster, J., Steiert, B., & Zhang, J. D. (2020). An Introduction to Machine Learning. Clinical pharmacology and therapeutics, 107(4), 871–885. doi: 10.1002/cpt.1796.
18. I. Dilrukshi, K. De Zoysa and A. Caldera, "Twitter news classification using SVM," 2013 8th International Conference on Computer Science & Education, 2013, pp. 287-291, doi: 10.1109/ICCSE.2013.6553926.
19. Han, J., Kamber, M., & Pei, J. (2006). Data Mining: Concepts and Techniques, Second Edition (The Morgan Kaufmann Series in Data Management Systems). Chapter 6 - Classification and Prediction. Morgan Kaufmann.
20. Yanai, K.: Image collector: an image-gathering system from the world-wide web employing keyword-based search engines. In: IEEE International Conference on Multimedia and Expo, 2001. ICME 2001. pp. 523–526 (2001).
21. Yanai, K.: Image collector ii: a system for gathering more than one thousand imagesfrom the web for one keyword. In: 2003 International Conference on Multimedia and Expo. ICME ’03. Proceedings
22. Generic image classification using visual knowledge on the web. In: Proceedings of the eleventh ACM international conference on Multimedia. pp. 167–176 (2003)
23. P. Koloveas, T. Chantzios, S. Alevizopoulou, S. Skiadopoulos, and C. Tryfonopoulos, “intime: A machine learning-based framework for gathering and leveraging web data to cyber-threat intelligence,” Electronics, vol. 10, no. 7, p. 818, 2021.
24. “MISP Open-Source Threat Intelligence Platform: Open Standards For Threat Information Sharing.” Retrieved December 29, 2022, from https://www.misp-project.org.
25. Harzing, A.W. (2007) Publish or Perish. Retrieved May 20, 2023, from https://harzing.com/resources/publish-or-perish.