Article

An Automatic Data Completeness Check Framework for Open Government Data

Sovit Bhandari ¹ , Navin Ranjan ¹ , Yeong-Chan Kim ^{1,2}, Jong-Do Park ^{1,3}, Kwang-Il Hwang ¹, Woo-Hyuk Kim ^{1,4}, Youn-Sik Hong ^{1,5} and Hoon Kim ^{1,2,*}

¹ IoT and Big-Data Research Center, Incheon National University, Yeonsu-gu, Incheon 22012, Korea; sovit198@gmail.com (S.B.); ranjannavin07@gmail.com (N.R.); kyc0288@naver.com (Y.-C.K.); jdp23@inu.ac.kr (J.-D.P.); hkwangil@inu.ac.kr (K.-I.H.); woohyuk@inu.ac.kr (W.-H.K.); yshong@inu.ac.kr (Y.-S.H.)

² Department of Electronics Engineering, Incheon National University, Yeonsu-gu, Incheon 22012, Korea

³ Department of Library and Information Science, Incheon National University, Yeonsu-gu, Incheon 22012, Korea

⁴ Department of Consumer Science, Incheon National University, Yeonsu-gu, Incheon 22012, Korea

⁵ Department of Computer Science and Engineering, Incheon National University, Yeonsu-gu, Incheon 22012, Korea

* Correspondence: hoon@inu.ac.kr

Abstract: In recent years, the governments in many countries have recognized the importance of data in boosting their economies. As a result, they are implementing the philosophy of open government data (OGD) to make public data easily and freely available to everyone in standardized formats. Because good quality OGD can boost a country's economy, whereas poor quality can jeopardize its efficient use and reuse, it is very important to maintain the quality of data stored in open government data portals (OGDP). However, most OGDPs do not have a feature that indicates the quality of the data stored there, and even if they do, they do not provide real-time service. Moreover, most recent studies focused on developing approaches to quantify the quality of OGD, either qualitatively or quantitatively, but did not offer an approach to automatically calculate and visualize it in real-time. To address this problem to some extent, this paper proposes a framework that can automatically assess the quality of data in the form of a data completeness ratio (DCR) and visualize it in real-time. The framework is validated using the OGD of South Korea, whose DCR is displayed in real-time using the Django-based dashboard.

Keywords: open government data; open government data portal; application programming interface; Django; data completeness ratio; data quality

Citation: Bhandari, S.; Ranjan, N.; Kim, Y.-C.; Park, J.-D.; Hwang, K.-I.; Kim, W.-H.; Hong, Y.-S.; Kim, H. An Automatic Data Completeness Check Framework for Open Government Data. *Appl. Sci.* **2021**, *11*, 9270. <https://doi.org/10.3390/app11199270>

Academic Editor: Manuel Armada

Received: 10 August 2021

Accepted: 2 October 2021

Published: 6 October 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the onset of the fourth industrial revolution (Industry 4.0), the global economy has become more data-centric. Industry 4.0 refers to the means of automation and data sharing in manufacturing technology, including internet of things (IoT), big data and analytics, augmented reality, autonomous robots, and so on [1]. According to [2], there are currently more than 10 billion active IoT devices, and this number is expected to increase to 25.4 billion by 2030. The amount of data generated by IoT devices is expected to reach 73.1 zettabytes in 2025 [2].

As data are the fuel for Industry 4.0 [1], governments in various countries are interested in using IoT devices to collect data from the public domain. In recent years, public data collected by governments through IoT and non-IoT means has been published on open government data portals (OGDPs) to make them available for citizens to use for their business or research purposes, which will ultimately contribute to a country's economic growth [3–5]. An OGDP is a web-based system that collects existing datasets from various sources and publishes them on user-friendly dashboards that users can view, download,

and retrieve via an application programming interface (API) in standardized file formats (CSV, XLSX, JSON, and XML) [6].

The key characteristics that make an OGD highly reliable are quality, completeness, accessibility, usability and comprehensibility, timeliness, value and usefulness, and granularity and comparability [7]. In addition to all these characteristics, the quality of the data plays a crucial role in the success of OGD, as high-quality data can increase the chance of achieving peak performance. The six characteristics defined by the International Data Management Association (DAMA) to ensure data quality are completeness, uniqueness, timeliness, validity, accuracy, and consistency and are illustrated in Figure 1 [8]. These characteristics are defined as follows:

- Completeness: ratio between the number of non-null values in a source and the size of the universal relation [9,10].
- Accuracy: the extent to which data are correct, reliable, and certified [11].
- Consistency: the extent to which data are presented in the same format and compatible with previous data [11].
- Validity: the extent to which data conform to the syntax (format, type, range) of their definitions [8].
- Uniqueness: the extent to which data are not duplicated [8].
- Timeliness: the extent to which age of the data are appropriate for the task at hand [11].

Figure 1. Data quality dimensions [8].

With the noticeable increase in the amount and variety of open data released by government agencies around the world, the quality of the data published in the OGD will determine its future potential [12]. Since data are the most important resource of the 21st century [13], good quality data can help users find the data they need more easily [14], whereas poor quality data in the OGD jeopardizes the efficient use and reuse of open data [12,15,16].

Furthermore, low data quality increases the cost of accessing and interpreting data [12] and provides misleading information [17], which consequently reduces the use of OGD [18].

Therefore, to address the problems identified with the poor quality of data stored in OGD, a framework that automatically examines the quality of data stored in OGD is needed to motivate the government to invest in improving the data stored in OGD.

1.1. Related Work

In this sub-section, we present some related work that deals with evaluating the quality of data stored in the OGD, and are listed in [6,12,19–22].

In [6], the quality of OGDs are assessed at the national level for 67 countries. The quality of the OGDs was assessed based on evaluation metrics such as the number of

datasets on the portals, the number of thematic groups in the portals, the number of tags associated with each dataset, the number of participating organizations, the number of licenses available for open government data (OGD) for publication and reuse, the number of users accessing the OGD, and questionnaires. The quality of the OGD cannot be determined solely by the amount of data stored in it. However, this work mainly adopted a quantitative approach to evaluate the quality of the OGD and did not consider the quality of the individual data stored in the OGD. Moreover, their work is more of a theoretical framework for evaluating the quality of OGD rather than a framework that can be converted into tools for automatically calculating the quality of OGD.

In [12], a framework for measuring the quality of Italian OGD is presented using data quality dimensions such as completeness, accuracy, traceability, currentness, expiration, conformity, and understandability at the most granular level of measurement. In this work, a qualitative approach is presented to evaluate the quality of the OGD. However, this work also presented a theoretical framework to assess the values of the data quality dimensions. As part of their long-term goal, they have considered the development of an OGD quality framework that can be transformed into a tool to automatically calculate data quality dimensions.

The framework to evaluate the quality of Chinese OGD portals at the provincial level was proposed in [19] based on metrics such as data quantity, data accessibility, and data quality. In [20], the quality of government datasets in Chinese regions of Beijing, Guangzhou, and Harbin is quantified based on seven quality dimensions such as completeness, accuracy, consistency, timeliness, uniqueness, and understandability. The conceptual framework is shown in [21] to classify 10 previous quality assessment frameworks for OGD based on six data quality indicators, namely accuracy, accessibility, completeness, timeliness, consistency, and understandability. The work presented in [19–21] consider a qualitative approach for assessing the quality of OGD; however, these studies only consider a theoretical approach, as in [6,12].

In [22], a general metadata quality assessment framework for 260 OGDs is presented. This work differs from the work presented in [6,12,19–21] as it focuses on automatic metadata quality assessment on a weekly basis. The quality of metadata is evaluated using quality metrics such as retrievability, existence, conformance, and openness. Although the work in [22] focuses on assessing the quality of metadata rather than the main data stored in OGD, it provides a framework for automatically assessing its quality on a weekly basis.

1.2. Contribution and Organization

To address the drawbacks of the work in [6,12,19–22] to some extent, this work proposes a framework that can automatically retrieve the quality of data stored in the OGD in tabular form and visualize them in real-time. In this framework, data quality is quantified using the data completeness ratio (DCR), which is one of the most important metrics for checking data quality as defined by the DAMA [8]. Here, DCR refers to the percentage of complete cells in the dataset. Mathematically, DCR can be written as [6]:

$$DCR = \left(1 - \frac{\text{Number of the incomplete cells}}{\text{Number of cells}}\right) \times 100\% \quad (1)$$

In this framework, DCR is only used to evaluate the quality of the data stored in OGD, since calculating its value does not require any manual work. Apart from DCR, the other data quality metrics cannot be automatically evaluated without manual computation.

In addition to the framework for automatic DCR calculation and real-time visualization, this paper also provides recommendations for improving the DCR of the dataset. To verify the usefulness of our framework, samples from the OGD of South Korea are used for national and provincial level analysis. With our framework, the government agencies that control OGD can evaluate the DCR of the individual dataset stored in their portal and use our recommendation methods to improve the quality of their data before releasing them to the public.

The structure of the paper is as follows. Section 2 formulates the general framework for automatic computation and visualization of DCR of data stored in OGD in real-time. Section 3 presents an implementation of this work using a Django-based framework to compute the DCR of the OGD from South Korea. Section 4 presents the DCR results of the OGD from South Korea. The obtained results are discussed in Section 5. Section 6 provides some recommendations to improve the DCR. Section 7 presents the limitations of this work. Section 8 provides a conclusion and roadmap for future work.

2. DCR Check Framework

In this section, an automatic general framework is shown for accessing the quality of OGD in terms of DCR, which is illustrated in Figure 2.

Figure 2 consists of various modular steps, such as OGD, API collection, automatic data download, automatic DCR calculation, website framework, website deployment, and website visualization.

Figure 2. A framework to check the DCR of data stored in OGD.

The tasks and functionalities of each module step are explained in the following sub-sections.

2.1. OGD

OGDP is an online data repository where the data are freely available to anyone. The data available in the OGD are called OGD and can be used or re-published without restrictions from copyright or patents. The goal of OGD is to open all non-personal and non-commercial data collected and processed by government organizations [23]. As part of this trend, public agencies have started to make government data available in standardized file formats on web portals, as web services, or via API, mostly based on open source data management systems such as CKAN or DKAN [6].

Using the API, the data stored in the OGDG can be downloaded automatically. So, in the first step of the automatic data quality checking framework, we find the OGDG whose data stored in the tabular form can be downloaded using the API. Some of the OGDG where the data can be downloaded through the API are data.go.kr (South Korea), data.gov (USA), data.gov.uk (UK), etc. The list of APIs associated with the data files can be found in the metadata section of these portals [6,24].

2.2. API Collection

In the second step, the API $\mathcal{A} = \{a_1, a_2, \dots, a_M\}$ listed in the metadata section of the OGDG, along with their corresponding data file name as $\mathcal{N} = \{N_1, N_2, \dots, N_M\}$ are stored in the API database $\mathcal{D} = \{d_1, d_2, \dots, d_M\}$ of length M in a standardized format. The API stored in the database can be used to automatically download the data files associated with it using scripting languages such as Python, Java, Ruby, etc.

2.3. Automatic Data Download

In the third step, scripting language is used to automatically download the files using the API database created in the second step. The algorithm to download list of M files $\mathcal{F} = \{f_1, f_2, \dots, f_M\}$ using the corresponding API $\mathcal{A}$ listed in the API database of length M is shown in Algorithm 1.

In Algorithm 1, the database containing the standard data name $\mathcal{N}$ and their corresponding API $\mathcal{A}$ are used to download the files f_i ($i \in \{1, 2, 3, \dots, M\}$) in the standard file format using their corresponding names N_i . The downloaded files $\mathcal{F}$ are stored in the preferred system location.

Algorithm 1 Automatic data download

Input: API database $\mathcal{D}$ containing API $\mathcal{A} = \{a_1, a_2, \dots, a_M\}$ and its corresponding data file name $\mathcal{N} = \{N_1, N_2, \dots, N_M\}$

Output: Downloaded data file $\mathcal{F} = \{f_1, f_2, \dots, f_M\}$

Initialize: $API\_response = [], f = []$

Automatically download data file $\mathcal{F}$: To download the data file f_i from its respective API a_i

```

1.  for  $i$  in range( $M$ ) do
2.       $API\_response_i \leftarrow \text{requests.get}(a_i)$ 
3.       $f_i \leftarrow \text{open}(N_i, \text{standard\_file\_format})$  from preferred system location
4.       $f_i.write(API\_response_i.content)$ 
5.       $f_i.close()$ 
6.  end for
```

2.4. Automatic DCR Calculation

After successfully downloading the data file $\mathcal{F}$ in a standard format, the data of these files are further analyzed to determine the region-based DCR of these files. The term 'region' here stands for both the administrative division of the country and the country as a whole.

The algorithm for determining the region-based DCR of a file is shown in Algorithm 2. In Algorithm 2, based on the unique regions $\mathcal{L} = \{l_1, l_2, \dots, l_L\}$ listed in the file $\mathcal{F}$, the DCR is calculated considering the case of the overall data fields and the mandatory data fields. The term 'overall data fields' refers to the list of column names in file $\mathcal{F}$, whereas the 'mandatory data fields' of file $\mathcal{F}$ are predefined and listed in the data standardization policy section of the OGDG.

Algorithm 2 Automatic region-based DCR calculation**Input:** Data file $\mathcal{F} = \{f_1, f_2, \dots, f_M\}$, list of unique regions in the file $\mathcal{L} = \{l_1, l_2, \dots, l_L\}$, list of overall and mandatory fields**Output:** Region-based DCR stored in the data file $\mathcal{R} = \{r_1, r_2, \dots, r_M\}$ **Initialize:** $N, DCR, O, M = []$ **Automatically find region-based DCR:** To find DCR in the data file f_i

```

1.  for  $i$  in range( $M$ ) do
2.       $O_{f_i} \leftarrow$  Count (data points of overall fields in the file  $f_i$ )
3.       $M_{f_i} \leftarrow$  Count (data points of mandatory fields in the file  $f_i$ )
4.      for  $j$  in range( $L$ ) do
5.           $N_{f_i}^{l_j, overall} \leftarrow$  Count (overall field null data points in  $f_i$  for region  $l_j$ )
6.           $N_{f_i}^{l_j, mandatory} \leftarrow$  Count (mandatory field null data points in  $f_i$  for region  $l_j$ )
7.           $DCR_{f_i}^{l_j, overall} \leftarrow \left(1 - \frac{N_{f_i}^{l_j, overall}}{O_{f_i}}\right) \times 100\%$ 
8.           $DCR_{f_i}^{l_j, mandatory} \leftarrow \left(1 - \frac{N_{f_i}^{l_j, mandatory}}{M_{f_i}}\right) \times 100\%$ 
9.      endfor
10.     save the data frame containing  $DCR_{f_i}^{l_j, overall}$ , and  $DCR_{f_i}^{l_j, mandatory}$  as  $r_i$  in the preferred system
        location
11.  end for

```

The DCR is calculated based on (1). After computing the region-based DCR, results of each f_i are stored in r_i .

2.5. Website Framework

After successfully calculating DCR, the data are sent to the open-source website framework for visualization. The functionalities that determine the quality of web development frameworks are AJAX support, cloud computing, comet support, custom error messages, customization and extensibility, debugging, documentation, forms validation, HTML5 support, JavaScript-based frameworks support, object relational mapping, parallel rendering, platform support, etc. [25].

Because there are many ways to build websites, some of the open source frameworks that have powerful backends for website development are Ruby on Rails, Cake PHP, ASP.NET, Django, Laravel, etc. [25].

2.6. Website Deployment

After making the website in step 5 using the data of step 4, it could only be accessed in the local server where the website was made. To deploy the website to the internet, an open-source web server should be used [26].

An open-source web server is public-domain software designed to deliver web pages over the World Wide Web. For instance, some of the popular open-source web servers are Apache HTTP Server, NGINX, Apache Tomcat, and Node.js [27].

2.7. Website Visualization

Finally, in this step, users can enter the website uniform resource locator (URL) in their web browser for visualization. When the URL is entered in the web browser, a server hosting a website receives URL requests for a resource or web page, and in response, sends the webpage over to the user's browser [26,27].

3. Django-Based DCR Check Framework for South Korea

In this section, we present a Django-based framework to automatically evaluate the DCR of data available in the OGDG from South Korea (see Figure 3). This framework is an implementation of the framework shown in Figure 2.

Figure 3. A Django-based framework to check the DCR of standard data available in the OGD of South Korea.

In Figure 3, at first, OGD of South Korea is accessed to collect the API that contains data in the standard format. The list of APIs can be found in the metadata section of the platform, as shown in Figure 4.

```
{
  "name": "National performance event information standard data",
  "description": "Provides per-formance event information (performance information, venue, schedule, fee information, etc.). Public data open standard data attribute information (expression format/unit) etc.), please refer to the [Public Data Open Standards] Notice. (Information Sharing>Resources>Laws (No-tice)/Guidelines). The standard data registered by each institution is collected and provided. The update cycle of standard data is different for each file. (Data registered by institutions is collected and the update time for each file is different)",
  "url": "https://www.data.go.kr/data/15013106/standard.do",
  "keywords": ["Art, Performance, Event"],
  "license": "https://data.go.kr/ugs/selectPortalPolicyView.do",
  "distribution": [
    {
      "encodingFormat": "CSV",
      "contentUrl": "https://www.data.go.kr/download/15013106/standard.do?dataType=csv",
      "@type": "DataDownload"
    },
    {
      "encodingFormat": "XML",
      "contentUrl": "https://www.data.go.kr/download/15013106/standard.do?dataType=xml",
      "@type": "DataDownload"
    },
    {
      "encodingFormat": "JSON",
      "contentUrl": "https://www.data.go.kr/download/15013106/standard.do?dataType=json",
      "@type": "DataDownload"
    },
    {
      "encodingFormat": "RDF",
      "contentUrl": "https://www.data.go.kr/download/15013106/standard.do?dataType=rdf",
      "@type": "DataDownload"
    },
    {
      "encodingFormat": "XLS",
      "contentUrl": "https://www.data.go.kr/download/15013106/standardExcel.do",
      "@type": "DataDownload"
    }
  ],
  "@context": "https://schema.org",
  "@type": "Dataset"
}
```

Figure 4. Metadata showing lists of API to download data in different standardized formats shown in the OGD of South Korea.

In Figure 4, the metadata section containing the list of APIs to download 'National performance event information standard data' in different standardized formats is shown. Only the data of 93 categories were found in the portal in the standardized format, which can be accessed directly via the API.

In the second step, the APIs that can download those data in the CSV format (tabular format) were stored in the API database, as shown in Table 1.

Table 1. List of standard data listed in ‘data.go.kr’ along with their download APIs stored in the API database.

SN	English Name	API
1	Standard Data for Protection Zones for the Elderly and Disabled Nationwide	https://www.data.go.kr/download/15034532/standard.do?dataType=csv (accessed on 12 July 2021)
2	National Performance Event Information Standard Data	https://www.data.go.kr/download/15013106/standard.do?dataType=csv (accessed on 12 July 2021)
3	National Library Standard Data	https://www.data.go.kr/download/15013109/standard.do?dataType=csv (accessed on 12 July 2021)
4	National Automobile Maintenance Company Standard Data	https://www.data.go.kr/download/15028204/standard.do?dataType=csv (accessed on 12 July 2021)
5	National Lifelong Learning Course Standard Data	https://www.data.go.kr/download/15013110/standard.do?dataType=csv (accessed on 12 July 2021)
⋮		
91	National Elementary School Commuting Area Standard Data	https://www.data.go.kr/download/15021149/standard.do?dataType=csv (accessed on 12 July 2021)
92	National Elementary and Secondary School Location Standard Data	https://www.data.go.kr/download/15021148/standard.do?dataType=csv (accessed on 12 July 2021)
93	National Recreational Forest Standard Data	https://www.data.go.kr/download/15013111/standard.do?dataType=csv (accessed on 12 July 2021)

In Table 1, a list of data stored in the API database is shown. A total of 93 APIs linking to different data categories were stored in the database.

In the third step, Python code is written based on Algorithm 1 to automatically download files using the APIs listed in the API database. In the fourth step, the downloaded files are analyzed using Python to determine the region-based DCR based on Algorithm 2. The DCR is calculated for the first-tier administrative division of South Korea, including the national one, for the overall and preferred data fields, which are stored in the file name r_i (CSV).

After the completion of the fourth step, i.e., the fifth step, the region-based DCR r_i is stored in the backend database of a Django-based framework for visualization. Django is a Python-based web framework that simplifies the creation of dynamic websites [28]. It follows the model–template–views (MTV) architectural pattern as shown in Figure 5.

Figure 5. MTV structure of a Django application.

In the MTV architecture, the model manages the data and is represented by a database; the view handles hypertext transfer protocol (HTTP) requests and HTTP responses; the template is the frontend layer. The view completes HTTP response by interacting with the model and the template [28].

The Django-based framework is used to visualize the r_i database as shown in Figure 3. The r_i database stored in the backend of the Django framework is accessed directly through

the views (Python file). In the views, 'context' is used to store the variables of the database in dictionary format, as shown in Figure 6.

```
context = {'items': {'SN': index, 'name': column, 'value': val}}
```

Figure 6. Context used to store the variables of the database in the dictionary format.

In Figure 6, the list data of the database are stored in the dictionary under the variable name 'items'. The dictionary 'item' is then assigned to the variable 'context'. Then the items from 'context' are rendered into the frontend template. Basic HTML is used for simple frontend templates; to make the templates more attractive and dynamic CSS and JavaScript are used. In the frontend template, Django's built-in multivariable template is used to display the list variables sent through 'context' [28,29].

The frontend template is only displayed as an HTTP response when the URL is entered. This constructed dashboard can only be accessed on the local server. To make this dashboard accessible through the client's browser, we used Apache Web Server in the sixth step.

Apache Web Server, commonly known as Apache, is a free and open-source web server that provides web content through the internet [30]. The Django-based website can be deployed using Apache and WSGI (Web Server Gateway Interface), where WSGI is an Apache module that can host any Python application [28].

After completing the sixth step, the last step of this framework is to enter the URL of this website into the web browser to get the DCR of data.go.kr. This created website can be accessed through the URL <http://cityview.inu.ac.kr> (accessed on 9 August 2021).

4. Results

In this section, we show the results of the Django-based data quality checking framework for South Korea. The result of each data file is automatically quantified in terms of DCR and visualized on the dashboard in real time. The specification of the computer server on which the dashboard is created includes an Intel Core i7 CPU, four Intel Xeon E7-1680 processors, and 128 GB memory.

4.1. DCR

In this part, based on Algorithm 2, the region-specific DCR of the OGD of South Korea is shown. The DCR is calculated for the different administrative regions of the country, including the national level, based on the overall and mandatory data fields. However, in this paper, only the administrative regions such as Incheon, Seoul, and Gyeonggi, collectively referred to as Greater Seoul, are considered for better presentation of the results.

Figures 7 and 8 show the radar plot of DCR of 93 different standardized files of National, Incheon, Seoul, and Gyeonggi regions in South Korea. The DCR values for 93 different files shown in Figure 7 were calculated considering the overall data fields in the files, whereas in Figure 8 the mandatory data fields in the files were considered.

Figure 7. Radar chart showing region-specific DCR of South Korea's OGD considering overall data fields.

Figure 8. Radar chart showing region-specific DCR of South Korea's OGD considering mandatory data fields.

The file names shown in Table 1 are mapped as N_i , where the value of i is varied from 1 to 93. The first name in Table 1 is denoted as N_1 for SN 1, whereas the last name is denoted as N_{93} for SN 93. This is done to facilitate the file-specific DCR representation on the radar chart, as shown in Figures 7 and 8.

In Figures 7 and 8, the broken lines due to missing points can be seen for the Incheon, Seoul, and Gyeonggi regions. The missing points in the graphs are due to lack of data points in the file for that region. The list of files that do not contain data for National, Incheon, Seoul, and Gyeonggi regions, considering the overall and mandatory fields, can be found in Table 2.

Table 2. List of filenames that do not contain data for National and Greater Seoul regions.

Region	Missing Dataset Filename	Count
National (Overall)		
National (Mandatory)		
Incheon (Overall)	N8, N45, N57, N62, N67, N81, N82, N83, N84, N87, N88, N89, N90, N91	14
Incheon (Mandatory)	N8, N45, N57, N62, N67, N81, N82, N83, N84, N87, N88, N89, N90, N91	14
Seoul (Overall)	N8, N29, N32, N45, N51, N57, N58, N67, N70, N76, N77, N82, N84, N87, N88, N89, N90, N91, N93	19
Seoul (Mandatory)	N8, N29, N32, N45, N51, N57, N58, N67, N70, N76, N77, N82, N84, N87, N88, N89, N90, N91, N93	19
Gyeonggi (Overall)	N8, N82, N87, N88, N89, N90, N91	7
Gyeonggi (Mandatory)	N8, N82, N87, N88, N89, N90, N91	7

From Figures 7 and 8, we can see that the DCR of the files evaluated considering the mandatory fields have a higher value than that of the total (mandatory + non-mandatory) fields.

A detailed analysis of the results from Figures 7 and 8 is performed to categorize the 93 different standardized files of National, Incheon, Seoul, and Gyeonggi regions based on different DCR values, which are shown in Figure 9.

Figure 9. Categorization of the National and Greater Seoul region data files for different values of DCR count.

To further clarify the results of Figures 7 and 8, the average DCR of National, Incheon, Seoul, and Gyeonggi regions is shown in Figure 10 for the overall and mandatory data fields.

Figure 10. National and Greater Seoul region average DCR considering overall and mandatory data fields.

Although our framework is capable of performing a minute-by-minute analysis of the DCR of the OGD, the DCR evaluation of the OGD is only presented on a daily basis in this paper. The average daily DCR of National, Incheon, Seoul, and Gyeonggi regions is shown in Figure 11.

Figure 11. Daily average DCR of National and Greater Seoul regions.

4.2. Dashboard Visualization

In this part snapshots of the dashboard under the domain name 'cityview.inu.ac.kr' are shown. Figure 12 shows the overall view of the dashboard.

Figure 12. Dashboard under the URL <http://cityview.inu.ac.kr> (accessed on 9 August 2021), where (a) is the dashboard home page, (b) is the dashboard showing standard public data information of South Korea, (c) is the dashboard showing DCR of the files listed there for National (Overall).

Figure 12a shows the home page of the dashboard. When the ‘Open Government Data’ button on the home page is clicked, the page with the items shown in Figure 12b is displayed. Figure 12b shows the region-specific list of standard public data and the corresponding DCRs. The highlighted portion of the DCR can be seen in Figure 12c for the

National (Overall), where the DCRs are displayed in different background colors to easily identify any change. The background color of the DCR is changed according to the rule shown in Table 3.

Table 3. Rule for changing DCR background color.

Condition	Background Color
$DCR \geq 90\%$	Blue
$70 \leq DCR < 90\%$	Green
$50\% \leq DCR < 70\%$	Yellow
$DCR < 50\%$	Red
No Data	Grey

5. Discussion

From the radar plots in Figures 7 and 8, we can obtain information about the completeness of the 93 individual standardized files of National, Incheon, Seoul, and Gyeonggi regions. Figure 8 shows that the result of DCR evaluated considering the mandatory fields seems to be more complete than that of Figure 7, in which DCR was calculated considering all fields. Moreover, in both figures, the missing DCR value for Incheon, Seoul, and Gyeonggi regions can be seen, which means that the data points in the respective files are not available for these regions. The standardized files that do not contain values for Incheon, Seoul, and Gyeonggi regions are 14, 19, and 7, respectively, which are identical for the overall and the mandatory case. This result is even more evident in Table 2, which also lists the names of the files that do not contain data points for the previously defined regions.

In Figure 9, the 93 files of the above regions are categorized according to the different levels of completeness. A file is considered most complete if the DCR is $\geq 90\%$, whereas the lowest degree of completeness applies to the case when the DCR is $< 50\%$. Following this rule, Figure 9 shows that for the case $DCR \geq 90\%$, the National (Mandatory) region has the highest number of files, i.e., 87, whereas Seoul (Overall) and Gyeonggi (Overall) regions have the lowest number of files, i.e., 46. For the $DCR < 50\%$ case, the National (Overall), National (Mandatory), Incheon (Mandatory), and Gyeonggi (Mandatory) regions have the highest number of files, namely 4, whereas Incheon (Overall), Seoul (Overall), and Gyeonggi (Overall) regions have the lowest number of files, namely 1. Figure 9 shows that a large proportion of files in the Seoul, Incheon, and Gyeonggi regions have a DCR of less than 90%, let alone 100%.

Figure 10 shows the average DCR of the 93 standard files based on the regions. From the results, it can be seen that the National (Mandatory) region has the highest average DCR of 94.87%, whereas the National (Overall) region has the lowest average DCR of 86.63%. It can be inferred that in 93 standardized files, the non-mandatory fields are incomplete in most of the cases as compared to the mandatory fields.

Figure 11 shows the daily average DCR for the National, Incheon, Seoul, and Gyeonggi regions for the overall and the mandatory case. The line graph in the figure shows the slight fluctuations in the DCR value, indicating that the government agencies are revising the dataset, but the desirable DCR level has not yet been reached.

The DCR results shown in Figures 7–11 are visualized in real-time using the Django-based dashboard, as shown in Figure 12. The results displayed in the dashboard show the alarming state of OGD quality of South Korea. The government agencies that control and monitor the OGD data should use this platform to improve and monitor the quality of data published in their portal before releasing it to the public.

6. Recommendations for DCR Improvement

In this section recommendations for improving DCR of files stored in the OGDP are provided. Several recommendations are listed below:

- If the DCR of the files stored in the OGDP has already been calculated, sort the files by their DCR value for the preferred region, as shown in Figure 13.

- Select the files in priority order by degree of completeness and find the headers of the files whose row is not filled, along with their frequency, as shown in Figure 14.
- After completing step b, first select the names of the mandatory headings in the list and contact the organization responsible for providing information on the missing units and complete the missing entries with the correct information.
- When the process for the mandatory list is complete, repeat the same process for the list of non-mandatory headings.
- After the completing steps c and d, upload this file to OGD for public use.
- Repeat steps b–e for each individual file in the OGD whose DCR is less than the desired value.

Figure 13. Dashboard showing sorted files based on the DCR value of OGD.

Figure 14. Dashboard showing headers of a file whose row is empty, along with their count.

7. Limitations

This framework is only applicable to data files that can be downloaded via an API and are in tabular format. Also, this framework requires manual work to store the list of standardized filenames along with their API in the API database, as shown in Table 1. Once the API database is created, this framework can be applied to all OGDs, regardless of the language they are bound to, since the count of empty fields in the file does not take the language into account. The main limitation of this framework is that only the DCR metric is used to quantify the quality of the files stored in OGD, as this is the only metric

whose value can only be computed automatically, whereas the computation of the other data quality indicators requires manual work.

8. Conclusions and Future Work

In this paper, we propose a general framework for automatically checking and visualizing the quality of data provided in OGD with respect to DCR in real-time. To validate our framework, it is applied to 93 standard tabular OGD datasets from South Korea that can be downloaded via the API. The quality of the dataset is quantified using the region-based DCR, considering the overall and mandatory data fields. For the region-based analysis, National, Incheon, Seoul, and Gyeonggi regions were considered. The simulation results show that among the 93 standard datasets, the datasets from the National (Mandatory) region gave better results, as 87 files were at least 90% complete, whereas in the case of the Seoul (Overall) and Gyeonggi (Overall) regions, only 46 files were 90% complete. The National (Mandatory) region had the highest average DCR with a value of 94.87%, and the lowest value of average DCR was found in the National (Total) region with a value of 86.63%. The results were visualized in real-time using the Django-based dashboard, which can be accessed via the URL <http://cityview.inu.ac.kr/data/> (accessed on 9 August 2021). The results show that the average DCR value of the OGD of South Korea was not even 95% even when the mandatory fields were included, and the result was even worse when the non-mandatory fields were also included.

Although we only checked the DCR of the OGD of South Korea, the value of DCR for the OGD of other countries could be even worse. Thus, in order to check and improve the quality of datasets stored in OGD in real-time before releasing them to the public, the government agencies responsible for storing and maintaining OGD could use our framework.

Since this framework only considers DCR to automatically quantify the quality of tabular datasets that can be visualized in real-time, future work will consider other metrics to calculate data quality that will be applicable to non-tabular datasets. The framework will consider OGD from different countries rather than just from one country.

Author Contributions: Conceptualization, S.B., H.K., N.R., Y.-C.K., J.-D.P., K.-I.H., W.-H.K. and Y.-S.H.; methodology, S.B. and H.K.; software, S.B. and N.R.; validation, S.B. and N.R.; resources, H.K.; data curation, S.B. and N.R.; writing—original draft preparation, S.B.; writing—review and editing, S.B., N.R., Y.-C.K., J.-D.P., K.-I.H., W.-H.K., Y.-S.H. and H.K.; visualization, S.B., H.K. and N.R.; supervision, H.K.; project administration, H.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. These data can be found here: <https://data.go.kr/> (accessed on 12 July 2021).

Acknowledgments: This work was supported by Incheon National University (Institute of Convergence Science and Technology) Research Grant in 2020.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Tay, S.I.; Lee, T.C.; Hamid, N.A.A.; Ahmad, A.N.A. An Overview of Industry 4.0: Definition, Components, and Government Initiatives. *J. Adv. Res. Dyn. Control Syst.* **2018**, *10*, 1379–1387.
2. Fascinating IoT Statistics for 2021 | The State of the Industry. DataProt. Available online: <https://dataprot.net/statistics/iot-statistics/> (accessed on 30 July 2021).
3. Kalampokis, E.; Tambouris, E.; Tarabanis, K. Open government data: A stage model. In Proceedings of the 10th IFIP WG 8.5 International Conference, Heidelberg, Germany, 28 August–2 September 2011; pp. 235–246.

4. Lněnička, M. An in-depth analysis of open data portals as an emerging public e-service. *Int. J. Soc. Behav. Educ. Econ. Manag. Eng.* **2015**, *9*, 589–599.
5. Yang, H.-C.; Lin, C.S.; Yu, P.-H. Toward automatic assessment of the categorization structure of open data portals. In *Multidisciplinary Social Networks Research, Proceedings of the International Conference on Multidisciplinary Social Networks Research, Matsuyama, Japan, 1–3 September 2015*; Wang, L., Uesugi, S., Ting, I.-H., Okuhara, K., Wang, K., Eds.; Springer: Berlin, Germany, 2015; pp. 372–380.
6. Máchová, R.; Lněnička, M. Evaluating the Quality of Open Data Portals on the National Level. *J. Theor. Appl. Electron. Commer. Res.* **2017**, *12*, 21–41. [[CrossRef](#)]
7. Lourenço, R.P. An analysis of open government portals: A perspective of transparency for accountability. *Gov. Inf. Q.* **2015**, *32*, 323–332. [[CrossRef](#)]
8. DAMA. Defining Data Quality Dimensions Data Management Association (DAMA)/UK Working Group. Available online: https://is.gd/dama_def_data_quality_dim (accessed on 30 July 2021).
9. Batini, C.; Cappiello, C.; Francalanci, C.; Maurino, A. Methodologies for data quality assessment and improvement. *ACM Comput. Surv. (CSUR)* **2009**, *41*, 1–52. [[CrossRef](#)]
10. Naumann, F. *Quality-Driven Query Answering for Integrated Information Systems*, 1st ed.; Springer: Berlin, Germany, 2002; pp. 1–168.
11. Wang, R.Y.; Strong, D.M. Beyond accuracy: What data quality means to data consumers. *J. Manag. Inf. Syst.* **1996**, *12*, 5–33. [[CrossRef](#)]
12. Vetrò, A.; Canova, L.; Torchiano, M.; Minotas, C.O.; Iemma, R.; Morando, F. Open data quality measurement framework: Definition and application to Open Government Data. *Gov. Inf. Q.* **2016**, *33*, 325–337. [[CrossRef](#)]
13. Šlibar, B.; Oreški, D.; Redep, N.B. Importance of the Open Data Assessment: An Insight Into the (Meta) Data Quality Dimensions. *SAGE* **2021**, *11*, 1–18.
14. Attard, J.; Orlandi, F.; Scerri, S.; Auer, S. A systematic review of open government data initiatives. *Gov. Inf. Q.* **2015**, *32*, 399–418. [[CrossRef](#)]
15. McBride, K.; Aavik, G.; Toots, M.; Kalvet, T.; Krimmer, R. How does open government data driven co-creation occur? Six factors and a ‘perfect storm’; insights from Chicago’s food inspection forecasting model. *Gov. Inf. Q.* **2019**, *36*, 88–97. [[CrossRef](#)]
16. Yi, M. Exploring the quality of government open data: Comparison study of the UK, the USA and Korea. *Electron. Libr.* **2019**, *37*, 35–48. [[CrossRef](#)]
17. Kubler, S.; Robert, J.; Neumaier, S.; Umbrich, J.; Traon, Y.L. Comparison of metadata quality in open data portals using the Analytic Hierarchy Process. *Gov. Inf. Q.* **2018**, *35*, 13–29. [[CrossRef](#)]
18. Dahbi, K.Y.; Lamharhar, H.; Chiadmi, D. Exploring dimensions influencing the usage of open government data portals. In *Proceedings of the 12th International Conference on Intelligent Systems: Theories and Applications, Rabat, Morocco, 24–25 October 2018*; pp. 1–6.
19. Wang, D.; Chen, C.; Richards, D. A prioritization-based analysis of local open government data portals: A case study of Chinese province-level governments. *Gov. Inf. Q.* **2018**, *35*, 644–656. [[CrossRef](#)]
20. Li, X.-T.; Zhai, J.; Zheng, G.-F.; Yuan, C.-F. Quality Assessment for Open Government Data in China. In *Proceedings of the 2018 10th International Conference on Information Management and Engineering (ICIME 2018), Salford, UK, 22–24 September 2018*; pp. 110–114.
21. Zhang, H.; Xiao, J. Quality assessment framework for open government data: Meta-synthesis of qualitative research. *Electron. Libr.* **2020**, *38*, 209–222. [[CrossRef](#)]
22. Neumaier, S.; Umbrich, J.; Polleres, A. Automated Quality Assessment of Metadata across Open Data Portals. *J. Data Inf. Qual.* **2016**, *8*, 1–29. [[CrossRef](#)]
23. Almeida, P.; Bernardino, J. A Comprehensive Overview of Open Source Big Data Platforms and Frameworks. *Int. J. Big Data* **2015**, *2*, 1–19. [[CrossRef](#)]
24. Braunschweig, K.; Eberius, J.; Thiele, M.; Lehner, W. The state of open data: Limits of current open data platforms. In *Proceedings of the 2012 International World Wide Web Conference, Lyon, France, 16–20 April 2012*; pp. 1–6.
25. Salas-Zárate, M.; Alor-Hernández, G.; Valencia-García, R.; Rodríguez-Mazahua, L.; Rodríguez-González, A.; López Cuadrado, J. Analyzing best practices on Web development frameworks: The lift approach. *Sci. Comput. Program.* **2015**, *102*, 1–19. [[CrossRef](#)]
26. Kunda, D.; Chihana, S.; Muwanei, S. Web Server Performance of Apache and Nginx: A Systematic Literature Review. *Comput. Eng. Intell. Syst.* **2017**, *8*, 43–52.
27. Jader, O.H.; Zeebaree, S.R.; Zebari, R.R. A State Of Art Survey for Web Server Performance Measurement and Load Balancing Mechanisms. *Int. J. Sci. Technol. Res.* **2019**, *8*, 535–543.
28. Django Documentation | Django Documentation | Django. Docs.djangoproject.com. Available online: <https://docs.djangoproject.com/> (accessed on 2 August 2021).
29. Bhandari, S.; Ranjan, N.; Hong, Y.-S.; Kim, H. Interactive Map-Based Framework for Visualization of Illegal Parking and CCTV CCTV Information to find CCTV Blind Spots. In *Proceedings of the 7th Online International Conference on Advanced Engineering and ICT-Convergence, Incheon, Korea, 7 July 2021*; pp. 54–57.
30. Fielding, R.; Kaiser, G. The Apache HTTP Server Project. *IEEE Internet Comput.* **1997**, *1*, 88–90. [[CrossRef](#)]