Faculty of Engineering

The University of New South Wales

By

Jay Patel, z5309776

Thesis submitted as a requirement for the degree of Bachelor of Engineering in Software Engineering

Submitted: 26/04/23

Supervisors: Dr. Helen Paik, A/Prof. Samsung Lim, Dr. Deepti Gurdasani

Abstract

This thesis addresses challenges attributed to detecting and extracting disease information tables from PDF documents, specifically in the domain of reports published by the World Organisation for Animal Health. PDFs are a popular method of information storage and exchange, however, due to the document's complex structure and limited semantic information, gathering information from information tables is challenging. Hence, extracting information tables is significant to information retrieval, data analysis, and decision-making. In the context of EPIWATCH, an artificial intelligence-driven system used to generate early warnings for epidemics, this thesis aims to provide an open-source tool that can collect animal disease information from the World Animal Health Information System database. With a focus on parsing PDF documents, the research proposes a methodology for detecting and extracting data tables using a heuristic rule-based technique. As automated table detection and extraction systems can both rule-based and machine learning-based techniques, this thesis will include a literature review of existing works in this field and determine its effectiveness in extracting data from the published PDF documents. Alas, this paper summarises the findings made in discussing several proposed approaches while illustrating potential areas for future research in the domain of extracting tabular information from PDF documents.

Acknowledgements

I would like to thank my supervisors, Dr. Helen Paik, A/Prof. Samsung Lim, and Dr. Deepti Gurdasani, as well as my thesis assessor, Dr. Xiaoyang Wang, for their guidance and advice throughout my thesis.

Abbreviations

PDF Portable Document Format

API Application Programming Interface

HTML HyperText Markup Language

IMG Image

CNN Convolutional Neural Network

OCR Optical Character Recognition

XML Extensible Markup Language

RAM Random Access Memory

WOAH World Organisation for Animal Health

WAHIS World Animal Health Information System

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

PDF documents have become a prevalent medium for storing and publishing structured data. Health organisations and government bodies alike use PDF documents to deliver relevant and up-to-date disease information. Despite their popularity, the inherent complexity and little semantic information contained in PDF documents pose significant challenges for accurately detecting and extracting relevant information from tables. Nevertheless, as a common means of distributing critical disease information, extracting data from PDFs has become essential for information retrieval, data accessibility, and data use.

In the framework of systems such as EPIWATCH—an artificial intelligence-driven platform providing global epidemic early warnings—extracting disease information from official PDF reports is critical for analysing epidemiological data. To enable EPIWATCH to enhance its disease detection and prediction systems, there is a need for an automated tool to collect and extract data from WAHIS reports. Hence, this thesis aims to provide an open-source solution that the EPIWATCH team can use to automatically collect and extract disease information data from WAHIS reports.

Thus, this thesis aims to investigate and propose an automated method for collecting and extracting data tables from PDF reports published through the WAHIS animal disease database. This paper will provide a literature review of related works in the field of table detection and extraction from PDF documents. Additionally, it will present the methodology used to collect and extract disease information from the WAHIS database. Moreover, this paper will provide an analysis and discussion of the solution's results. Lastly, this report will emphasise the value added for EPIWATCH and the broader field of PDF table extracting, concluding with a discussion of potential areas for future work.

Chapter 2

Background

The following background chapter will provide background knowledge for this thesis. Specifically, it will identify the significance of tabular PDF data, outline the structure of WAHIS reports and challenges associated with reading PDF reports, present methods of automatically collecting reports (pre-processing), provide a literature review of existing methods for detecting and extracting tables, and outline various post-processing output formats.

2.1 Significance of PDF Documents

PDF documents have become a ubiquitous medium for storing and delivering structured data across diverse domains. In addition to the versatile and consistent nature of PDF documents, there exist several factors that enforce its popularity as the preferred method of releasing critical disease information. These factors include high versatility and cross-compatibility, strong document integrity, ease of document archiving, standardisation of PDF use, and seamless interactivity. Consequently, PDF documents have common practice in various industries, including healthcare.

The structure of a PDF document is characterised by a hierarchical organisation of elements. As illustrated by Figure 1, the hierarchy involves a PDF catalogue that contains references to other objects, pages that provide a visualisation of the document, content streams that outline instructions for rendering pages, and the resources used for rendering such as fonts and images (Whitington, 2012). As such, the uses of resources, cross-referencing, compression and tagging allow for consistent representations of information across various platforms.

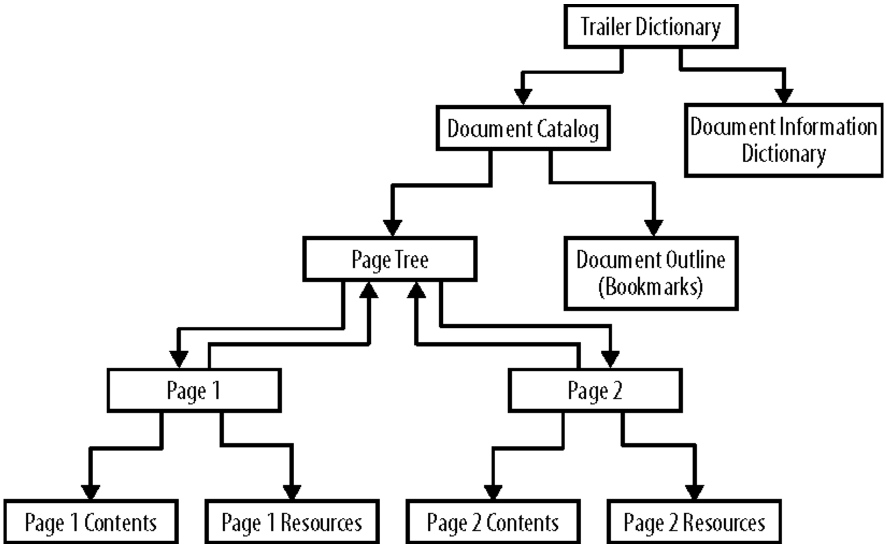


Figure 1: Diagram of the Structure of PDF Documents

Health organisations and governmental bodies rely heavily on PDF documents to publically release critical information. The significance of PDF files in delivering disease information is underscored by several factors that support the unique needs and requirements of healthcare and research. The key factors that PDF documents play in disease reporting include the preservation of document integrity, consistent visual representation across different platforms, secure distribution of sensitive information, efficient archiving and retrieval, and regulatory compliance. Consequently, disease information pertaining to human and animal infectious diseases is commonly released through PDF documents by official sources. Thus, using data information requires the processing of tabular information from PDF documents.

2.2 Challenges With Extracting Information From PDF Reports

The purpose of the following section is to outline the challenges associated with extracting information from PDF reports. The section will provide a background on the issues that arise when processing PDF documents, and the problems involved with reading tables from PDF documents.

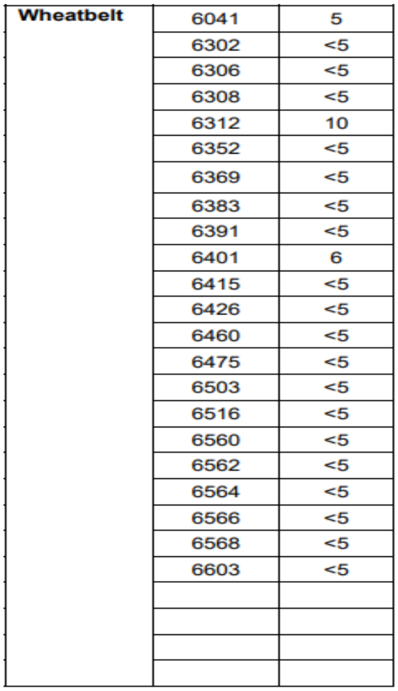
2.2.1 Challenges with PDF Document Structures

Despite PDF documents providing a robust method of storing structured data, inherent complexities arise when detecting and extracting information from the documents. These challenges involve understanding the complex relationship between elements and the hierarchical organisation of the document, variations in document layouts, limited semantic information, security measures and encryptions, and lack of standardisation in metadata (Mazrui, 2005). To overcome these complex challenges for automated document processing systems, advanced solutions involving rule-based and machine-learning-based approaches are required.

2.2.1 Challenges of Reading Tables from PDF Documents

Tables contained within PDF documents, particularly tables that relate to disease information found in surveillance reports, present several challenges when attempting to accurately extract information. The complex nature of reading tables from documents is heightened when dealing with compound tables, rules, nested headings, and varied presentations.

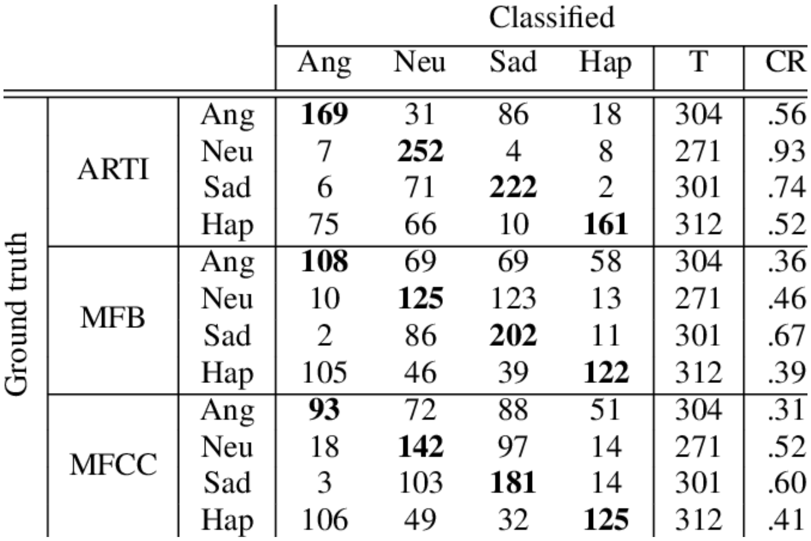
Compound tables with varied structures involve tables that may contain blank lines or unwritten rules, obfuscating the boundaries of individual cells. For example, in Table 1, empty cells and blank lines obscure the reading order for the table. Hence, the use of unique rules and odd layouts requires additional considerations for tables to be automatically processed.



Wheatbelt	6041	5
	6302	<5
	6306	<5
	6308	<5
	6312	10
	6352	<5
	6369	<5
	6383	<5
	6391	<5
	6401	6
	6415	<5
	6426	<5
	6460	<5
	6475	<5
	6503	<5
	6516	<5
	6560	<5
	6562	<5
	6564	<5
	6566	<5
	6568	<5
	6603	<5

Table 1: Table Containing Empty Cells

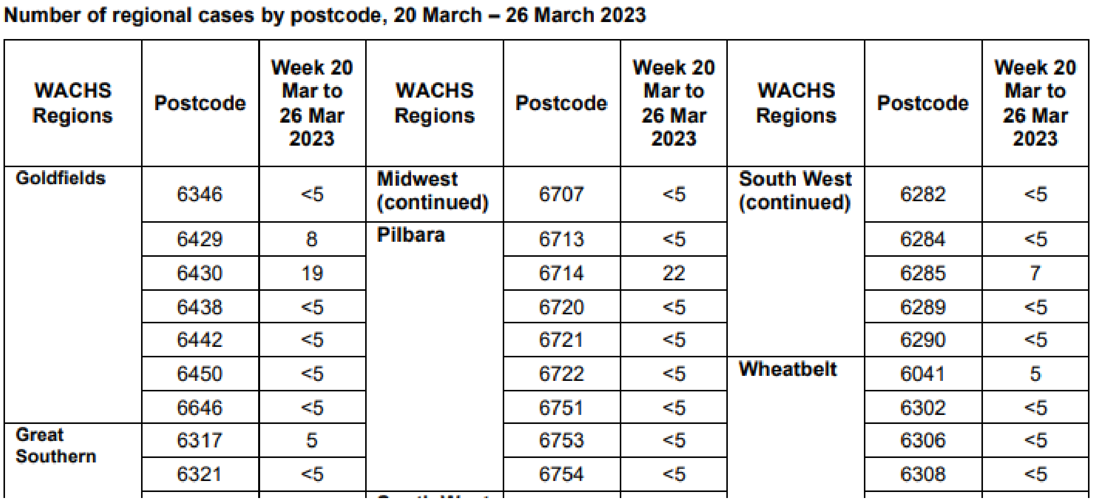
Additionally, the tables presented in PDF documents may contain obscure borders as horizontal and vertical lines may not encapsulate table cells. In Table 2, a lack of horizontal and vertical lines creates a confusing cell structure. When automatically processing tables such as Table 2, solutions cannot easily determine individual cells through the use of clearly defined lines.



			Classified					
			Ang	Neu	Sad	Hap	T	CR
Ground truth	ARTI	Ang	**169**	31	86	18	304	.56
		Neu	7	**252**	4	8	271	.93
		Sad	6	71	**222**	2	301	.74
		Hap	75	66	10	**161**	312	.52
	MFB	Ang	**108**	69	69	58	304	.36
		Neu	10	**125**	123	13	271	.46
		Sad	2	86	**202**	11	301	.67
		Hap	105	46	39	**122**	312	.39
	MFCC	Ang	**93**	72	88	51	304	.31
		Neu	18	**142**	97	14	271	.52
		Sad	3	103	**181**	14	301	.60
		Hap	106	49	32	**125**	312	.41

Table 2: Table Containing Obscure Boarders

Moreover, tables with nested headings can obscure the reading order of the table as various levels of information convey different information. In Table 3, nested headings and column headings are used to attach different information points to cell values. However, in using nested headings, the reading order becomes confusing and the step-by-step process needed to extract information becomes unclear.



Number of regional cases by postcode, 20 March – 26 March 2023

WACHS Regions	Postcode	Week 20 Mar to 26 Mar 2023	WACHS Regions	Postcode	Week 20 Mar to 26 Mar 2023	WACHS Regions	Postcode	Week 20 Mar to 26 Mar 2023
Goldfields	6346	<5	Midwest (continued)	6707	<5	South West (continued)	6282	<5
	6429	8	Pilbara	6713	<5		6284	<5
	6430	19		6714	22		6285	7
	6438	<5		6720	<5		6289	<5
	6442	<5		6721	<5		6290	<5
	6450	<5		6722	<5	Wheatbelt	6041	5
	6646	<5		6751	<5		6302	<5
Great Southern	6317	5		6753	<5		6306	<5
	6321	<5		6754	<5		6308	<5

Table 3: Table Containing Confusing Reading Order

To summarise, confusing reading orders pose significant challenges when automatically extracting tabular information. A lack of straightforward reading sequences draws issues when deciphering the logical flow of information. Consequently, the extracted information is often outputted in a similarly confusing and incomprehensible manner. For example, Table 3 demonstrates a confusing reading order by combining nested row and column headings with empty cells. As the reading order for a table becomes more confusing, reading tables automatically becomes significantly harder.

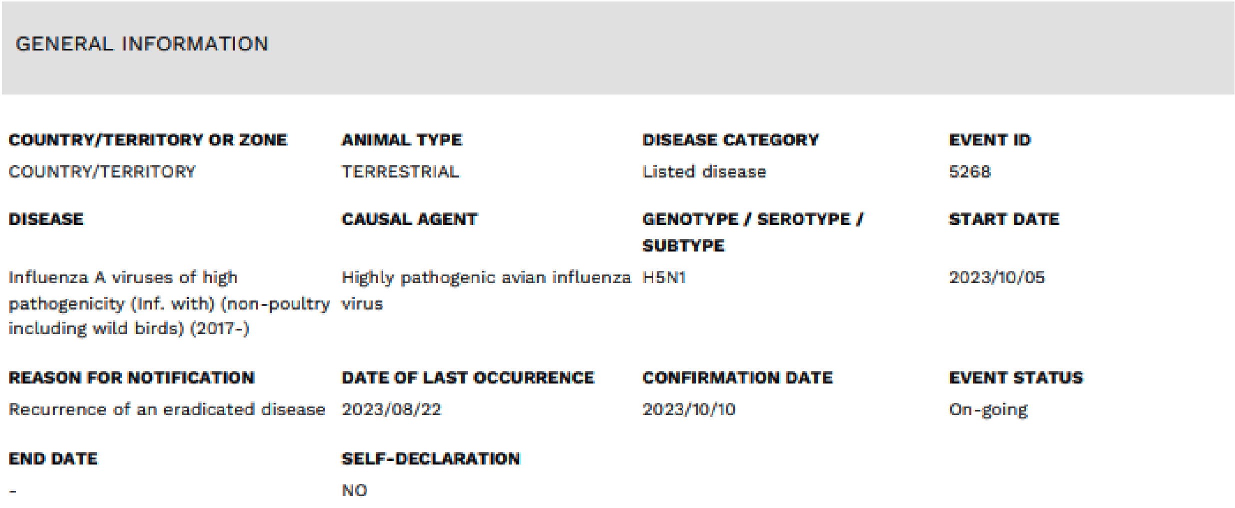
To effectively extract information from WAHIS reports, it is necessary to develop methods of addressing these challenges.

2.4 Structure of WAHIS Reports

The structure of WAHIS reports follows a well-defined framework that conveys new and previous animal disease outbreak information. To effectively extract information from this domain of reports, it is critical to understand the components of the report and its structure. Additionally, extracting tabular information from WAHIS reports requires the collection of reports from the WAHIS website.

In order to extract information from WAHIS reports, it is necessary to navigate and potentially download reports from the animal database. Hence, it is necessary to understand how elements in the website are associated with reports. Specifically, the WAHIS database attaches an EventId to each disease event so that new outbreaks can be linked to previous outbreak data. Hence, using specific EventIds, particular disease reports can be identified and collected for extraction.

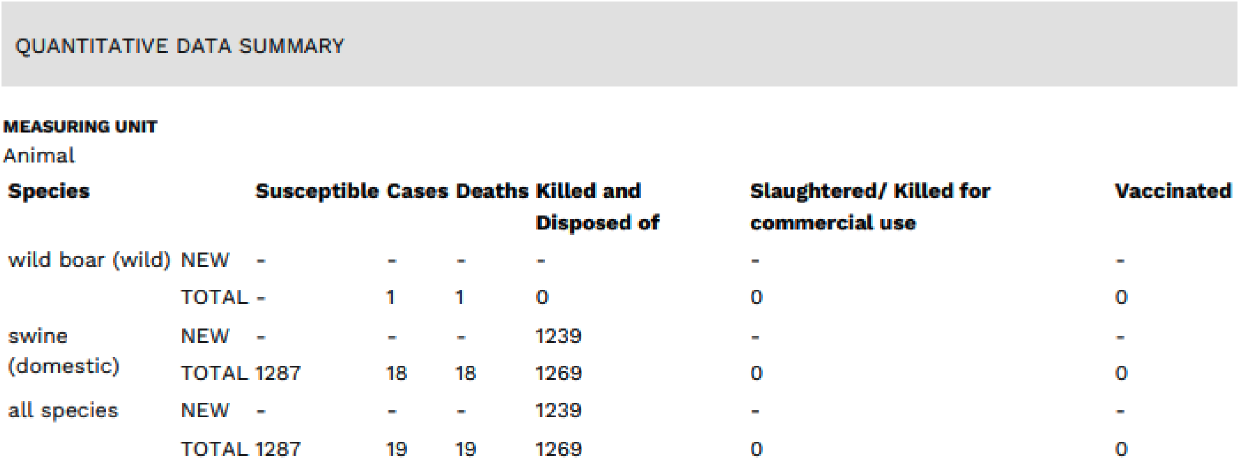
All downloadable reports are sectioned by titles with subsections either containing key-value pairs of information, or a table that contains row and column headings as well as empty cells and no cell lines. Table 4 demonstrates the structure of tables that contain key-value pairs which is standardised across all WAHIS reports. Additionally, Table 5, illustrates the table structure of WAHIS tables. Although the structure of WAHIS tables is standardised, the lack of clear borders, empty cell information and row headings creates a difficult reading order. However, as each table is standard, all WAHIS tables for certain sections share similar column headings.



GENERAL INFORMATION

COUNTRY/TERRITORY OR ZONE	**ANIMAL TYPE**	**DISEASE CATEGORY**	**EVENT ID**
COUNTRY/TERRITORY	TERRESTRIAL	Listed disease	5268
DISEASE	**CAUSAL AGENT**	**GENOTYPE / SEROTYPE / SUBTYPE**	**START DATE**
Influenza A viruses of high pathogenicity (Inf. with) (non-poultry including wild birds) (2017-)	Highly pathogenic avian influenza virus	H5N1	2023/10/05
REASON FOR NOTIFICATION	**DATE OF LAST OCCURRENCE**	**CONFIRMATION DATE**	**EVENT STATUS**
Recurrence of an eradicated disease	2023/08/22	2023/10/10	On-going
END DATE	**SELF-DECLARATION**		
-	NO		

Table 4: WAHIS Key-Value Pairs Table



QUANTITATIVE DATA SUMMARY

MEASURING UNIT
Animal

Species		**Susceptible**	**Cases**	**Deaths**	**Killed and Disposed of**	**Slaughtered/ Killed for commercial use**	**Vaccinated**
wild boar (wild)	NEW	-	-	-	-	-	-
	TOTAL	-	1	1	0	0	0
swine (domestic)	NEW	-	-	-	1239	-	-
	TOTAL	1287	18	18	1269	0	0
all species	NEW	-	-	-	1239	-	-
	TOTAL	1287	19	19	1269	0	0

Table 5: WAHIS Quantitative Data Summary Table

In addition to the basic disease information, each report can contain all previous outbreaks. The previous outbreaks are presented in a similar format, therefore extracting information from WAHIS reports requires handling these particular tables. As previous outbreaks are necessary when observing trends and developing predictions on the spread of infectious animal diseases, all data should be extracted from these reports as certain sections give different insights into a disease's behaviour.

Chapter 3

Literature Survey

The strong need to extract structured and unstructured data from PDF documents has led to a growing interest in the field of PDF table detection and data extraction. Several methods have been developed to successfully extract tabular information from PDF documents in response to this challenge. Throughout this literature review, this paper will analyse current methods of automatically collecting online reports, methods of table detection, and methods of table extraction.

3.1 Related Works for Automated Collection Methods

There exist several methods of collecting information from online sources. However, the most prevalent approaches involve web scraping techniques and accessing information through application programming interfaces (APIs). As WAHIS does not provide an API that allows for quick access to its disease information, web scraping techniques are required. To automatically web scrape components of WAHIS's website, it is necessary to use tools to process HTML documents as well as an automated tool to navigate the website systematically.

Web scraping techniques require the use of algorithms to navigate websites, retrieve information and compile data for further use. Several web scraping techniques such as text pattern matching, HTTP parsing, HTML parsing, DOM parsing, vertical aggregation, semantic annotation recognising, and computer vision web-page analysis offer various methods of processing web pages (Glez-Peña et al., 2013). As WAHIS features reports through a web format, it is appropriate to use an HTML parsing tool. Several frameworks and libraries exist that help facilitate HTML crawling techniques. Python libraries such as BeautifulSoup, a tool for parsing HTML and XML documents, create parse trees facilitating web scraping techniques (Richardson, 2019). Whereas, web scraping frameworks such as Scrapy allow a root URL to be defined with additional parameters allowing a program to crawl, download and save content from websites (Pavlovskytė, 2023). Hence, differences between libraries and frameworks that offer HTML parsing tools lie in the surrounding tools and additional benefits. For example, while BeautifulSoup efficiently fetches information requested, Scrapy offers the ability to crawl and save xHTML information.

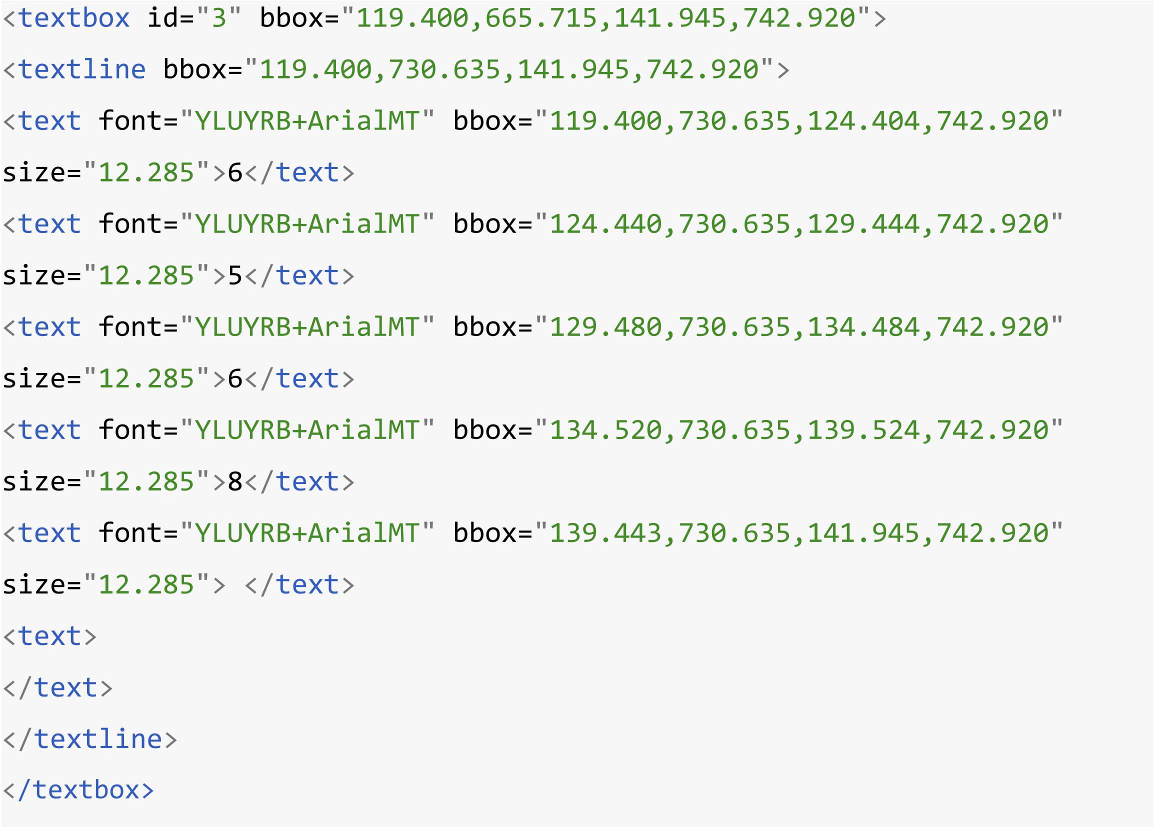
Automated webdrivers that require HTML information to operate often include webscraping capabilities as webdrivers often rely on HTML and CSS elements to identify components of a webpage. The most popular automated web driver management tool is Selenium, which is a framework that is used to control websites automatically. Hence, Selenium can navigate webpages following a script across various browser types. Selenium's compatibility, ease of use, and additional IDE and Grid automation tools have solidified it as an extremely popular tool for web driver automation (García et al., 2021). Alternative open-source tools that are used for automatisation such as Appium provide means of locating elements by their XPath, and CSS. Tools such as Selenium and Appium are often used for automated testing and web scraping, hence, these tools, alongside competitors, are often compared on the basis of time efficiency and accuracy.

Using a combination of HTML parsers and automated tools can provide the automatic and efficient collection of WAHIS reports. Due to the incredible popularity, support and usefulness of tools such as BeautifulSoup and Selenium, automating report collection using these tools offers an efficient and effective method.

3.2 Related Works for Table Detection Methods

There have been several approaches proposed for table detection. These approaches can be classified into rule-based and machine-learning-based methods.

Rule-based table detection methods refer to identifying tables using certain tabular characteristics such as semantic elements, the presence of horizontal lines, vertical lines, and white spaces (Kim & Hwang, 2020). Rule-based methods are highly dependent on the pre-processing of the PDF document. As HTML files contain clear and accessible semantic elements, rule-based methods are highly accurate and precise when converting such file types. However, applying rule-based methods to a PDF table by comparing coordinate information is extremely difficult and will result in poor accuracy and precision. Works that proposed rule-based solutions include a paper by Embley et al. [2] which aims to automatically extract data from HTML tables that contain an unknown structure. Although this proposal expresses a strong extraction method, this implementation struggles to respond to tables that contain cut-off borders and skewed rotations. Moreover, a related work by Mikhailov & Shigarov [3] presents TAO which proposes a method to detect and extract information from XML type files. TAO is highly successful in determining table locations as XML files can offer highly accurate page coordinates and highlight semantic elements. As presented in Figure 2, textbox locations and coordinates of individual text elements are presented within the XML's inner code. Although this implementation can easily identify tables and tabular elements, it is limited when determining the size and scope of a full table.



```
<textbox id="3" bbox="119.400,665.715,141.945,742.920">
<textline bbox="119.400,730.635,141.945,742.920">
<text font="YLUYRB+ArialMT" bbox="119.400,730.635,124.404,742.920"
size="12.285">6</text>
<text font="YLUYRB+ArialMT" bbox="124.440,730.635,129.444,742.920"
size="12.285">5</text>
<text font="YLUYRB+ArialMT" bbox="129.480,730.635,134.484,742.920"
size="12.285">6</text>
<text font="YLUYRB+ArialMT" bbox="134.520,730.635,139.524,742.920"
size="12.285">8</text>
<text font="YLUYRB+ArialMT" bbox="139.443,730.635,141.945,742.920"
size="12.285"> </text>
<text>
</text>
</textline>
</textbox>
```

Figure 2: Sample Textbox Output for PDF to HTML Conversion

However, machine learning-based methods refer to trained algorithms to identify tables. When given an IMG containing a table, these methods have higher accuracy and precision compared to rule-based methods. Proposed machine learning-based methods employ CNNs, SVMs, or decision trees to identify tables. Works such as DeepDeSRT by Schreiber et al. [6] and TableNet by Paliwal et al. [5] offer deep learning methods that can detect tables from IMG tables with great precision and accuracy. DeepDeSRT and TableNet implement similar deep-learning models that can identify tables by using Faster R-CNN (FRCN) which is a CNN model that is commonly used for efficient object detection. FRCN models will generate region proposals based on an input image by a region proposal network and then classify proposals using a Fast-RCNN network. DeepDeSRT and TableNet offer state-of-the-art precision and recall measures when given an IMG table. CNN and deep learning models are highly effective when handling IMG tables.

Rule-based methods offer a highly accurate and efficient solution for table detection when handling a strict input of tables. However, as the domain of reports increases, rule-based methods struggle to offer a generalised solution to all table types. Complex tables that contain confusing reading orders, odd presentations, and missing information pose significant challenges for rule-based methods. On the other hand, machine learning-based responses offer state-of-the-art solutions that can identify a wide variety of tables. However, machine learning-based solutions pose greater time and accuracy costs depending on the range of tables inputted when compared to rule-based solutions. Ultimately, both approaches do not offer a perfect solution to table detection and are dependent on the domain of tables being detected.

3.3 Related Works for Table Extraction Methods

Once tables have been identified and detected, extracting information from such tables will be required. Table extraction methods can be categorised into layout-based methods and content-based methods.

Layout-based methods refer to an approach that relies on the spatial layout of the table to extract it. Often such methods require pre-processing involving deskewing and de-tilting to ensure the table aligns with image axes. Works that proposed solutions for data extraction include a paper by Embley et al. [2] to automatically extract data from HTML tables with unknown structures offered a highly accurate solution that was able to distinguish tables from surrounding text and layouts. This implementation extracted tabular information by mapping headings and cells into a tuple format. Although this implementation suggests a different approach to table identification, the solution is highly effective when handling tables which like within the given scope. The proposed solution boasted a 96% precision score. The proposed approach was limited by the table identification process as tables that appeared as skewed or cut off by page borders failed to be extracted. Moreover, this approach will fail when given a table in the form of an image. Thus, the limitations of this layout-based approach can be addressed by implementing a highly effective pre-processing conversion process and an accurate table identification process.

Other methods that implemented table extraction by identifying table layouts and features included PDF-TREX, a paper by Oro & Ruffolo [4], and TAO, a paper by Mikhailov & Shigarov [3]. Both methods converted tables into XML formats and compared locations of row headings and text boxes to extract table data. Both PDF-TREX and TAO offer a heuristic algorithm that can efficiently extract information from simple tables. However, this approach is extremely limited when handling complex tables with row and column headings. Figure 2 illustrates an example of how page coordinates are associated with table elements.

Content-based methods refer to extraction using the table's content. Such methods rely on OCR and semantic analysis to identify table headings and cells for extraction. Tablext, a paper by Colter et al. [1] presents an OCR as a neural network model to identify and separate all potential tables. Tablext identifies the high-level structure through computer vision methods, then applies a CNN model to extract cell locations, and finally uses OCR to extract content (Colter et al., 2022). The processes required to extract table information follow real line identification, inferred line identification, final structure line identification, neural network correction, and finally OCR to gather data. Tablext uses You Only Look Once (YOLO) as an object detection algorithm. Similar tools to YOLO include Tesseract which is an open source OCR engine.

Hence, relying on layout-based methods may offer more efficient and accurate results when extracting tables from a narrow domain. Whilst, content-based extraction methods offer a generic solution that can extract information tables from a wide variety of domains with differing accuracies and precision.

3.4 Synthesis of Literature Review

The literature review has explored various automated data collection, table detection, and table extraction methods. This synthesis aims to analyse key insights and underscore the nuances necessary for determining an appropriate solution tailored for WAHIS tables.

As part of detecting and extracting tabular information, the solution will be required to automatically collect input data. In the absence of WAHIS APIs, tailored heuristic solutions, particularly web scraping techniques, emerge as crucial for automatic data collection. Tools such as BeautifulSoup and Selenium have been popularised due to their efficient web scraping capabilities.

Additionally, table detection has been classified into rule-based and machine-learning-based approaches. Where rule-based approaches rely on tabular characteristics, exhibiting high accuracies when given a specific input type. While machine learning-based approaches utilise CNNs, SVMs, and decision trees to achieve a generalised solution to table detection.

After identifying the location of tables, table extraction processes involving layout-based and content-based methods are necessary to gather information. Rule-based approaches are effective in handling specified domains of tables. Whist, content-based approaches employ OCR and semantic analysis to offer a generic solution with varying accuracy to extracting tabular data from tables across different domains.

Hence, determining a suitable approach is dependent on the domain of the tables being extracted. Given a specified domain, where all tables follow similar or standardised rules, a rule-based and layout-based extraction method should be used. While solutions that require considering tables from a wide variety of inputs should involve a machine learning-based and content-based extraction method

While contributions to the field of automated collection methods, table detection, and table extraction have been made, challenges persist. Due to the incredible variety of inputs and various types of tables, producing a generalised solution that can efficiently extract all table types is difficult. Thus, the discussion made in this literature review underscores the need for a nuanced approach that involves tailored methods for each phase. This synthesis provides a background for the subsequent chapters, the proposed methodology and a discussion of results, where solution approaches have been made with great consideration to the domain of tables being extracted.

Chapter 4

Methodology

The following methodology chapter will provide the methodology taken to develop this thesis project. Specifically, it will outline the automated report collection process, table detection method, table extraction method, and post-processing stage.

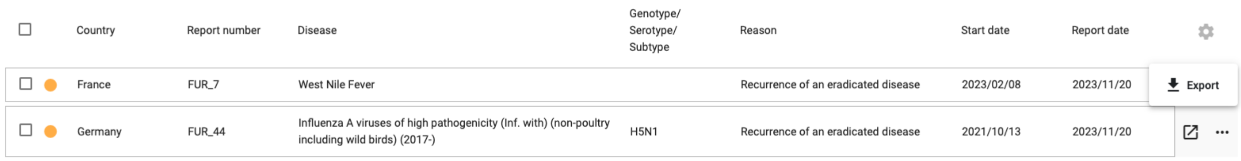
4.1 Overview of the Approach

To effectively collect and extract information from WAHIS reports, the following steps are required: pre-processing; table detection; table extraction; and post-processing. To effectively extract information from WAHIS reports, a heuristic rule-based method has been used to address WAHIS-specific reports.

4.2 Preprocessing

As part of this thesis, this solution is required to automatically collect reports from the WAHIS website. This process can be summarised into steps: collecting exported event IDs; iterating through the list of event IDs, and downloading the latest reports using event IDs. Hence, this preprocessing stage required an automated browser driver tool that could navigate a given website to iterate through these steps. To achieve this, Selenium was used as an automated web browser due to its ability to wait for elements to load and to be clickable. As such, issues involving cookie popups, and lengthy database loading times were mitigated.

To collect event IDs from the WAHIS website, the Selenium web driver tool navigated to a hidden export button found on the website as shown in Figure 3. By identifying elements by XPaths, and CSS selectors the Selenium web driver was programmed to wait until specific elements were clickable. As the WAHIS database contained almost 100,000 reports, loading main pages to collect event lists was time costly and pre-emptively clicking elements would cause errors with the program. Hence, efficiently collecting a list of event lists was necessary to reduce time costs when downloading several hundreds of reports.



□		Country	Report number	Disease	Genotype/ Serotype/ Subtype	Reason	Start date	Report date
□	●	France	FUR_7	West Nile Fever		Recurrence of an eradicated disease	2023/02/08	2023/11/20
□	●	Germany	FUR_44	Influenza A viruses of high pathogenicity (Inf. with) (non-poultry including wild birds) (2017-)	H5N1	Recurrence of an eradicated disease	2021/10/13	2023/11/20

Figure 3: WAHIS Export Event List Button

In exporting the list of reports provided on the current page, event IDs could be collected from the column heading ‘eventId’ as illustrated in Table 6. As such, by iterating through WAHIS pages and exporting event lists, an extensive list of event IDs could be collected and prepared for the next stage of downloading reports. Tools such as Pandas offered means of reading CSV files to allow the quick collection of event IDs.

country	eventId	reportId	disease	subType	eventStartDate	reason	reportStatus	submissionDate	reportNumber
Italy	5074	163768	African swine fever virus (Inf. with)		20/05/2023	First occurrence in a zone or a compartment	Validated	08/11/2023	FUR_23
Italy	5044	163767	African swine fever virus (Inf. with)		26/04/2023	First occurrence in a zone or a compartment	Validated	08/11/2023	FUR_25
Ukraine	5317	163758	African swine fever virus (Inf. with)		03/11/2023	Recurrence of an eradicated disease	Validated	08/11/2023	FUR_1
Ukraine	5126	163757	African swine fever virus (Inf. with)		16/07/2023	Recurrence of an	Validated	08/11/2023	FUR_13

						eradicated disease			

Table 6: Sample from WAHIS Export Event List

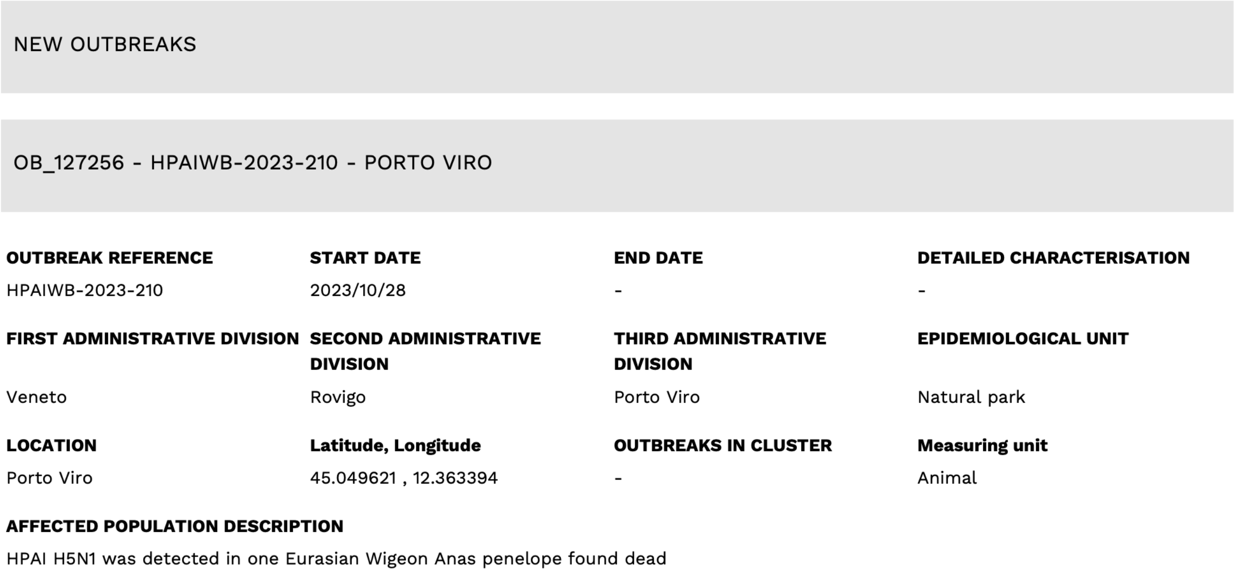
Collecting a list of event IDs allows will allow the web driver to jump directly to report specific dashboard pages using unique URLs. As each report contains a unique URL in the format of, 'https://wahis.woah.org/#/in-review/XXXX?fromPage=event-dashboard-url', where 'XXXX' are unique event IDs, specific reports can be efficiently downloaded. By identifying buttons by XPaths and CSS elements, the Selenium driver could respond to download confirmation popups, significantly reducing the time needed to download a single report. By jumping directly to a specific report dashboard, and by waiting until necessary elements are visible or clickable, the web driver could almost instantaneously download a given report. Hence, by iterating through a list of event IDs, the Selenium web driver can download countless reports listed on the WAHIS website. As such, after downloading the reports until a given date or number, the program can begin the table detection and extraction process.

4.3 Table Detection Method

After collecting and downloading a number of WAHIS reports, the table detection process can be conducted. Using a powerful Python tool, Tabula-py is a method of reading tables from PDF documents and converting them into pandas' DataFrame (Ariga, 2019). The key steps involved with the tabular algorithm involve a rule-based identification process that involves analysing visual and geometric elements to determine table locations. Specifically, by parsing PDF files and extracting text and visual elements, Tabula can conduct visual analysis and geometric recognition on PDF pages.

By analysing the visual elements on a page, Tabula identifies regions that exhibit certain characteristics. These characteristics can include table borders, cells, section headings, or table lines. Similarly, implementing geometric analysis to identify spatial relationships between text and graphical elements helps determine the boundaries of tables. Thus, by determining how a potential table visually and geometrically sits within a PDF page, Tabula compares conventional table construction rules to identify potential tables.

In detecting all pages from every WAHIS report downloaded, the Tabula algorithm is able to identify tables by their unique geometric spacing and specific cellular information by their visual position. For example, Figure 4 demonstrates how a presented table is clearly separated geometrically by section headings, with table content being clearly confined within certain regions on the PDF.



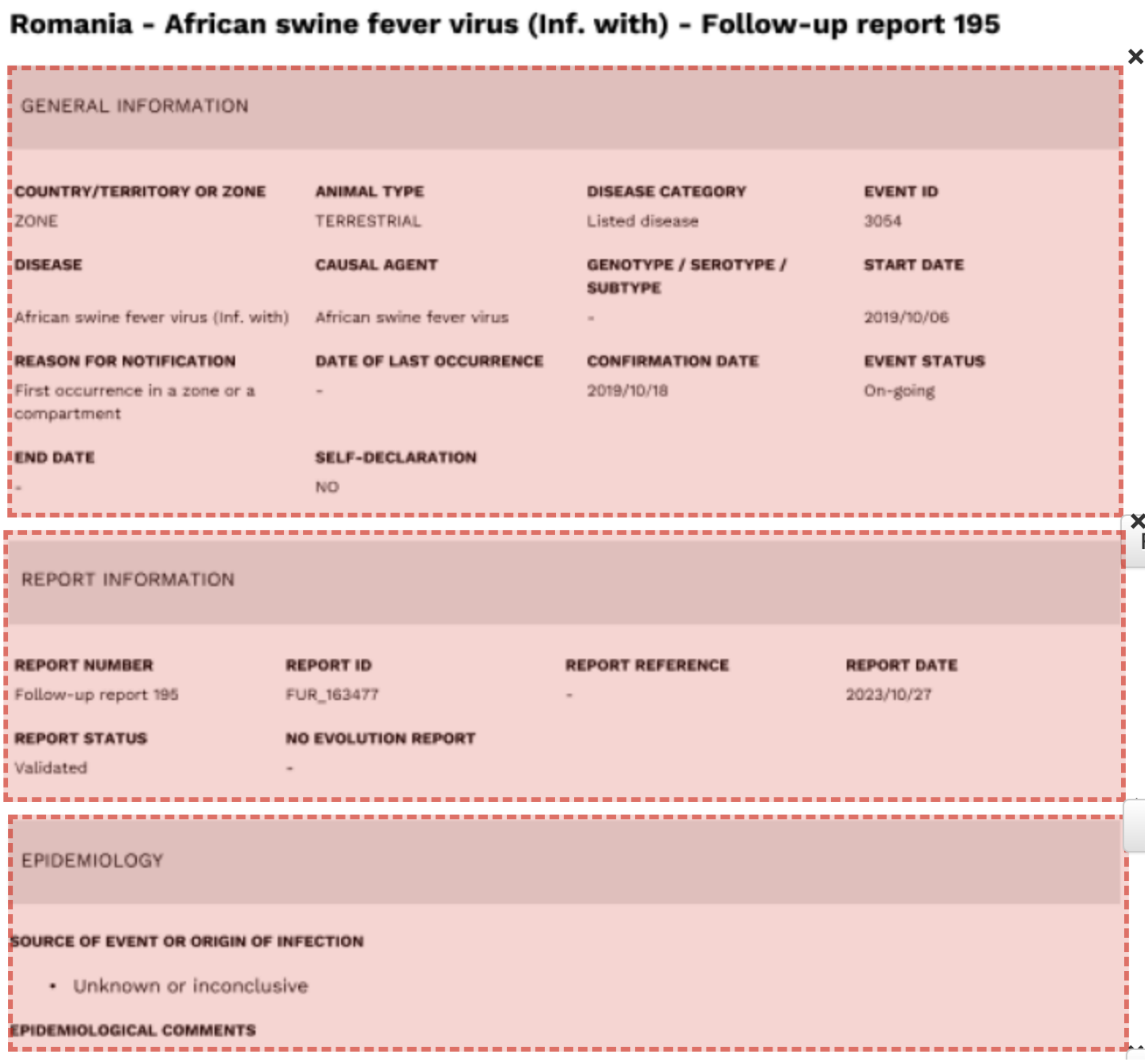
NEW OUTBREAKS

OB_127256 - HPAIWB-2023-210 - PORTO VIRO

OUTBREAK REFERENCE	**START DATE**	**END DATE**	**DETAILED CHARACTERISATION**
HPAIWB-2023-210	2023/10/28	-	-
FIRST ADMINISTRATIVE DIVISION	**SECOND ADMINISTRATIVE DIVISION**	**THIRD ADMINISTRATIVE DIVISION**	**EPIDEMIOLOGICAL UNIT**
Veneto	Rovigo	Porto Viro	Natural park
LOCATION	**Latitude, Longitude**	**OUTBREAKS IN CLUSTER**	**Measuring unit**
Porto Viro	45.049621 , 12.363394	-	Animal
AFFECTED POPULATION DESCRIPTION			
HPAI H5N1 was detected in one Eurasian Wigeon Anas penelope found dead			

Figure 4: WAHIS Visual and Geometric Table Presentation

With the standardised and clear presentation of WAHIS reports, using a rule-based detection process is effective and highly accurate. As demonstrated in Figure 5, Tabula is successful in detecting tables from WAHIS PDF reports. Thus, by using Tabula's python wrapper, tabula-py, tables from reports can be detected by their visual elements and geometric position.



Romania - African swine fever virus (Inf. with) - Follow-up report 195

GENERAL INFORMATION

COUNTRY/TERRITORY OR ZONE	ANIMAL TYPE	DISEASE CATEGORY	EVENT ID
ZONE	TERRESTRIAL	Listed disease	3054
DISEASE	CAUSAL AGENT	GENOTYPE / SEROTYPE / SUBTYPE	START DATE
African swine fever virus (Inf. with)	African swine fever virus	-	2019/10/06
REASON FOR NOTIFICATION	DATE OF LAST OCCURRENCE	CONFIRMATION DATE	EVENT STATUS
First occurrence in a zone or a compartment	-	2019/10/18	On-going
END DATE	SELF-DECLARATION		
-	NO		

REPORT INFORMATION

REPORT NUMBER	REPORT ID	REPORT REFERENCE	REPORT DATE
Follow-up report 195	FUR_163477	-	2023/10/27
REPORT STATUS	NO EVOLUTION REPORT		
Validated	-		

EPIDEMIOLOGY

SOURCE OF EVENT OR ORIGIN OF INFECTION

- Unknown or inconclusive

EPIDEMIOLOGICAL COMMENTS

Figure 5: Tabula Table Recognition

4.4 Table Extraction Method

Following the successful detection of tables using Tabula, the extraction process focuses on processing the tabular data into a data frame. To extract table information, the process extracts information according to conventional top-down reading orders to insert tables into a particular data frame. In doing so, tabula refines data by aligning headers, addressing inconsistent cell formatting, and other anomalies identified during PDF parsing. Thus, during the table extraction process, tables are outputted into a data frame and are validated so that outputted results reflect WAHIS tabular data.

For example, Table 7 demonstrates how certain disease information requires adjustments and post-processing. During basic extraction, certain titles and values are unaligned, creating unintuitive key-value pairs. Although this particular table has a clear top-down reading order, as column headings for the genotype/serotype/subtype incorrectly bleed into other rows, the alignment of extract information changes.

COUNTRY/TERRITORY OR ZONE	ANIMAL TYPE	DISEASE CATEGORY	EVENT ID
ZONE	TERRESTRIAL	Listed disease	5274
DISEASE	CAUSAL AGENT	GENOTYPE / SEROTYPE /	START DATE
		SUBTYPE	
Bluetongue virus (Inf. with)	Bluetongue virus	3	2023/10/10
REASON FOR NOTIFICATION	DATE OF LAST OCCURRENCE	CONFIRMATION DATE	EVENT STATUS
Recurrence of an eradicated disease	2021/04/05	2023/10/12	On-going

Table 7: Tabula Output Sample #1

Thus, after detecting tables found in WAHIS reports, the extraction process that follows requires converting detected tables into a dataframe where further processing can be conducted. Due to the complexity of PDF table extraction, gathering data accurately remains challenging and may require post-processing of data after extraction.

4.5 Postprocessing

To complete the extraction of WAHIS reports, all reports require conversion from the Pandas data frame to another format to ensure that extracted information is accessible and usable. As EPIWATCH's work in disease prevention and detection requires data from WAHIS to be in the form of JSON or CSV, extracted data has been converted into CSV formats. Extracting information into a CSV format allows non-technical stakeholders the ability to read and work with data. As extracting a large number of reports may result in a large number of CSV files, this thesis has outputted all files into a singular CSV worksheet for convenient reading and greater accessibility. Thus, each PDF report extracted is contained in a unique worksheet with a workbook containing all results from the extraction.

Although CSV files provide great usability and accessibility to a variety of potential end users, extracted information can be outputted into other file formats. These other formats may include JSON or a database that reflects WAHIS's internal information storage format. Storing information in a database provides great advantages, such as maintenance and storage of records. However, as storage methods for WAHIS reports were not within the scope of this thesis, developing a database and potentially a usable API remains an area for future work.

Chapter 5

Discussion of Results

This thesis has provided an approach to extracting tabular information from WAHIS reports. In using Tabula and Selenium as primary tools during the approach, insights into the effectiveness of the proposed methodology have been made. Specifically, the consistency of the solution's ability to identify and extract tabular data, the approach's ability to adapt to diverse table formats shown in WAHIS reports, and the quality of extracted data, reveal the effectiveness of the approach.

5.1 Expected Results vs Actual Outcomes

Throughout this thesis, as the initial aims and expectations of the thesis evolved, new areas of research were added and specific methodologies were refined. In comparing initial expectations with actual results, observations can be made between the alignment with initial predictions, the robustness across report types, the ability of the solution to adapt to variations with WAHIS and the quality of tabular results.

Throughout this thesis, the initial aims evolved from developing a fully generalised solution to PDF table extraction to extracting PDF tables from WAHIS reports. Narrowing the domain of reports from all PDF reports to WAHIS-specific reports provides greater feasibility to this solution and value to EPIWATCH as developing this tool will solve specific problems certain teams within EPIWATCH face as a result of poor access to WAHIS reports. Initially, the proposed methodology involved a hybrid approach to provide a solution to all types of tables. However, this hybrid method required implementing two potentially state-of-the-art approaches to solve the problem of access to tabular information from WAHIS reports. Thus, in narrowing the domain of reports, the method of the solution also changed to better suit the new requirements and objective of the thesis.

Initially, the expected outcomes in terms of robustness in extracting reports across various domains were extremely high. The preliminary thesis aimed to extract reports from various types of domains, including those with nested tables, varying column structures, and differing layouts. As the domain of reports narrowed to reports published by WAHIS, the need to support extraction across unique report types was reduced. Hence, less emphasis was placed on the table extraction method, as clear table structure and headings allowed for less post-processing.

Similarly, as the domain of reports was reduced, the actual outcomes contained a lower adaptability to report variations in comparison to the expected initial outcomes. As WAHIS reports provide a clear structure with little variation, all reports extracted from the website database will follow a similar format. Reducing the scope of reports needed to be extracted reduced the solution's adaptability, but increased its accuracy and effectiveness.

Lastly, the quality of reports needed for output remained stagnant throughout the thesis. The primary aim remained as extracting information from PDF reports throughout the course of the thesis. Thus, the expected outcomes involved outputting tabular information to another readable format that can be easily parsed by a computer such as JSON or CSV. Thus, actual outcomes surpassed expected outcomes in terms of the variety of output options. However, the actual solution fails to perfectly extract all tabular information.

To summarise, as the scope of the thesis became clearly defined by the domain of reports being extracted, the expected results of the thesis changed. Ultimately, the expected outcome involved extracting information from PDF reports into a JSON or CSV format. With varying degrees of accuracy and success, this solution provides a means of achieving these results. However, due to the complex challenge that is posed by PDF table extraction, perfectly parsing all tabular information from reports remains an obstacle. For EPIWATCH, the expected results exceed use cases, where critical disease information is more important to be extracted over all data points provided in the report.

5.2 Results Analysis

The analysis of results involves a qualitative assessment of the extracted tables including the accuracy of the extraction, time costs, and memory costs. This section analyses aspects of the result's outcomes to understand the performance of the automated solution.

Given a variety of tables, the solution successfully processes and outputs extracted data to a combined CSV workbook. The solution overcomes challenges involving nested headings, merged cells, and diverse column arrangements providing a tool for extraction. The table provided in Appendix A.1, Sample Extracted Table from WAHIS Report, demonstrates sample output from the solution. The PDF report being extracted contained various types of tables including tables for key-value pairs, tables with nested headings, and tables that contained merged or empty cells.

Analysing the qualitative attributes of the table presented in Appendix A.1 reveals the completeness of the solution's ability to extract all data from the PDF report. Specifically, analysing the quality of the data outputted by the completeness of the data, accuracy of the results, timeliness of the data, the data's relevance, and clarity are measures of its quality. Relevant information such as the date of the outbreak, the affected species, outbreak locations, and control measures implemented are all noted by the extracted table's information. Additionally, as the solution is successful in collecting recent outbreak information from a highly reputable animal disease source, the data collected is accurate, timely and relevant. Lastly, as each report contains several tables, significant challenges were faced when determining the best method of presenting the extracted information. However, to represent data, key-value paired information has been associated by a top-down reading order in the table with unique column headings containing differing information to associate row headings and columns with specific cells. Due to the different presentations of the tables, the extracted data is not presented in a straightforward manner as empty cells and filler data points extracted from the WAHIS report create a confusing association between values.

Moreover, the process of collecting numerous reports is extremely time-consuming. As connecting to WAHIS's website database will make it automatically load potentially hundreds of thousands of reports, repetitively downloading event lists from the website is time-costly. However, to mitigate this issue and improve the efficiency of the application, the Selenium driver has managed to click on elements as soon as they are clickable. Hence, one's internet speed is the greatest determiner of the collection speed. Similarly, as the number of reports increases, the time required to download and extract information from reports increases linearly. To reduce time costs, efficient conversion to data frames has reduced the need to iterate and fully extract each report one by one. Saving time at each phase of the collection and extraction process will significantly reduce time costs when extracting a large number of reports.

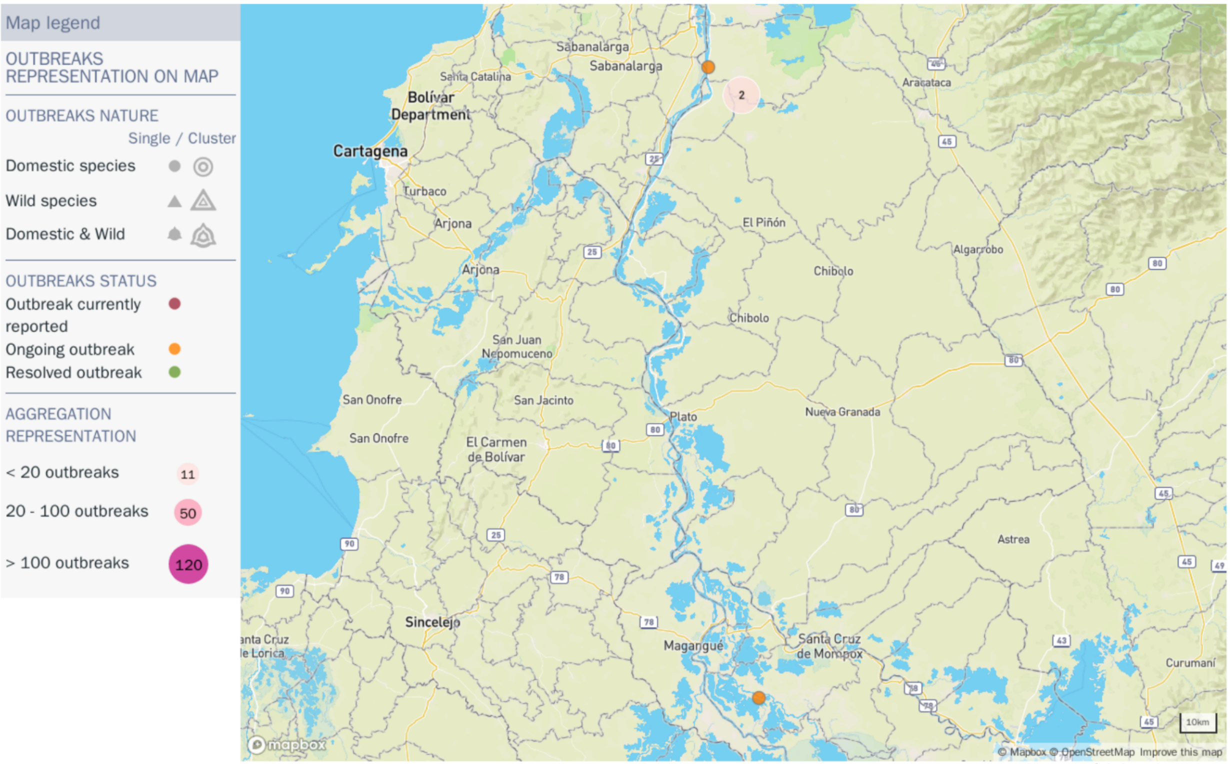
Additionally, as the number of reports being extracted increases, the memory costs of storing information increases. Downloading countless event lists and reports will require substantial memory space and RAM as extracted reports will be stored in a data frame in local memory during the extraction process. Thus, as the number of reports increases, the demand for a system's requirements increases. If a system does not contain sufficient space of RAM for the solution, the extraction process will fail and only some of the reports will be extracted. Lastly, as the number of reports being extracted increases, the final file will increase in size. To mitigate these memory costs, storing information in a combined CSV reduces the amount of physical memory needed to store all information extracted. Similarly, extracting information iteratively will only require RAM to be used to store information in the short term during the extraction process. Moreover, by deleting downloaded reports and event lists, the solution can clean up a machine saving memory and improving the usability of the application. Hence, memory costs pose issues when the solution extracts a large number of reports, however, mitigates made during the extraction process and the post-processing process reduce the system's requirements needed to run the solution.

5.3 Challenges and Limitations

Despite the method's effectiveness in extracting a large quantity of disease information from WAHIS reports, several challenges and limitations need to be addressed for future work. These challenges include handling complex table structures, extracting information from non-tabular data, and maintaining performance in the event of changes to WAHIS's website or PDF structure. As a result, these challenges limit the solution in terms of accuracy, scalability, and generalisability.

As the domain of reports widens to meet new WAHIS published reports or to extract disease information from other sources, the solution will need to overcome challenges faced by handling unique tables with complex structures and changes to WAHIS's website. As extracting tabular information from PDF reports is extremely difficult due to the complex nature and presentation of tables, the accuracy of the report extraction varies heavily depending on the inputted table. Even within WAHIS tables, empty cells and nested headings can create confusing extraction results. Thus, challenges involving handling complex table structures pose accuracy limitations to the solution. Moreover, as the WAHIS database evolves, changes to its web page and PDF structure can break the proposed methodology. Hence, to maintain a working collection process, continuous monitoring of the website will require updates to the collection method as necessary.

Additionally, to improve the collection of disease information, the solution should aim to interpret non-tabular information in addition to PDF reports. As WAHIS delivers maps that describe the movement of outbreaks, critical information that may help epidemic detection and prevention may not be currently extracted. Figure 6 presents other informational sources that may convey the location range an outbreak is affecting, giving greater insight into the scale of an outbreak. Hence, expanding the domain of extractable information from PDF reports to encompass IMG and map information may improve the scalability of the solution.



© Mapbox | © OpenStreetMap
Prints use map data from Mapbox and OpenStreetMap and their data sources.
To learn more, visit https://www.mapbox.com/about/maps/ and http://www.openstreetmap.org/copyright.
Les localisations des foyers ont été renseignées par les Services vétérinaires compétents et peuvent ne pas représenter l'emplacement exact d'un foyer. L'OIE n'assume aucune responsabilité quant aux données affichées.

Figure 6: Map of Disease Outbreak

To summarise, due to the significant challenge posed by extracting tabular data from PDF reports, the solution has become limited by its accuracy of extracted information, scalability over time and when extracting numerous reports, and its generalisability when extracting information from a wider domain of information sources.

Chapter 6

Conclusion

This thesis has investigated and proposed an approach for automatically collecting, detecting and extracting tables from WAHIS PDF reports. The approach has involved automatically collecting WAHIS reports using Selenium as a tool and extracting tables using a heuristic rule-based solution.

This thesis has contained a background on the surrounding context of table extraction from WAHIS's web database. Moreover, it has provided a literature review that discussed potential methods that have been previously explored for table detection and extraction. These methods included rule-based, machine learning-based, layout-based, and content-based approaches. Additionally, this thesis has outlined the methodology used when developing the solution for extracting WAHIS reports. Lastly, this thesis has analysed the approach's results in a discussion and has demonstrated its value added to EPIWATCH.

6.1 Value Added to EPIWATCH

The integration of the automated WAHIS table extraction tool presents great value to the operations of EPIWATCH. Several notable benefits to EPIWATCH include enhanced data accessibility, streamlined information retrieval, reduced manual processing, improved scalability, and strategic implications for public health surveillance.

As the automated tool collects and processes WAHIS reports, manual efforts associated with reading through PDF reports and extracting data is substantially reduced. By incorporating this streamlined information retrieval system, EPIWATCH can focus resources on critical analysis and strategic responses. Timely insights made by EPIWATCH can have significant impacts on disease detection and prevention, thus reducing time costs involved with data collection can have significant impacts on responses to emerging health scenarios. Additionally, automated tools provide EPIWATCH with opportunities to scale and adapt to evolving data requirements. With growing numbers of disease cases from a large number of countries, meeting scalability requirements can allow EPIWATCH to perform on a global scale without compromising efficiency. Consequently, the rapid and accurate extraction of tabular data improves EPIWATCH's ability to advance public health surveillance and provides means of effective disease management.

Disease information collection is a critical part of EPIWATCH's operation. In surveying current disease trends, the solution can be used to detect and predict threatening disease behaviours. Thus, this solution provides significant value to EPIWATCH's operations by improving data accessibility, reducing manual processes, and increasing scalability. As a result, EPIWATCH can benefit from greater responsiveness to animal disease outbreaks fostering strategic advances in animal public health.

6.2 Areas of Future Work and Improvement

Although this thesis has presented a solution for extracting WAHIS-specific reports, there exist some areas for further improvement. Areas involving fine-tuning table detection parameters, integrating machine learning techniques, user interface controls, documentation, and regular maintenance can improve the solution's accessibility, adaptability, and effectiveness.

Currently, the solution is very limited to tables provided by WAHIS reports that are clearly defined as tables. Due to the complex construction and potentially unique presentation of tables, the solution fails to respond to identify tables from a wide domain of reports. As disease information is delivered from a variety of sources, collecting and processing data from a number of sources provides significant benefits to EPIWATCH. By improving table detection methods, and implementing machine learning techniques throughout the extraction process, the domain of reports that can be extracted using this solution can be widened.

Moreover, to improve usability and accessibility, future areas of work can involve incorporating user interface controls to help specify extraction targets within the WAHIS website. To narrow extraction results, incorporating a user-friendly method of extracting country-specific information, or data within a date range, can improve usability. As not all end users may have a technical background in computer science, providing a user interface can significantly expand the number of potential users while offering greater control.

Additionally, implementing regular maintenance to respond to changing WAHIS web features will ensure the solution remains relevant. As the web driver, Selenium navigates web pages by clicking on HTML and CSS elements, the solution must be maintained to effectively navigate the website.

Thus, areas for future improvement lie in expanding the domain of extractable reports, incorporating user interfaces for non-technical end users, and providing regular maintenance to respond to website changes.

Bibliography

Ariga, A. (2019) PY: Read tables in a PDF into dataframe, tabula. Available at: https://tabula-py.readthedocs.io/en/latest/ (Accessed: 21 November 2023).

[1] Colter, Z. et al. (2022) 'Tablext: A combined neural network and heuristic based table extractor', SSRN Electronic Journal [Preprint]. doi:10.2139/ssrn.4127694.

[2] Embley, D.W., Tao, C. and Liddle, S.W. (2005) 'Automating the extraction of data from HTML tables with unknown structure', Data &amp;amp; Knowledge Engineering, 54(1), pp. 3–28. doi:10.1016/j.datak.2004.10.004.

García, B. et al. (2021) 'Automated driver management for selenium WebDriver', Empirical Software Engineering, 26(5). doi:10.1007/s10664-021-09975-3.

Glez-Peña, D. et al. (2013) 'Web scraping technologies in an API world', Briefings in Bioinformatics, 15(5), pp. 788–797. doi:10.1093/bib/bbt026.

Kim, J. and Hwang, H. (2020) 'A rule-based method for table detection in website images', IEEE Access, 8, pp. 81022–81033. doi:10.1109/access.2020.2990901.

Mazrui, J. (2005) What's in a PDF? the challenges of the popular Portable document format, The American Foundation for the Blind. Available at: https://www.afb.org/aw/6/6/14571 (Accessed: 19 November 2023).

[3] Mikhailov, A. and Shigarov, A. (2021) 'Page layout analysis for refining table extraction from PDF documents', 2021 Ivannikov Ispras Open Conference (ISPRAS) [Preprint]. doi:10.1109/ispras53967.2021.00021.

[4] Oro, E. and Ruffolo, M. (2009) 'PDF-Trex: An approach for recognizing and extracting tables from PDF documents', 2009 10th International Conference on Document Analysis and Recognition [Preprint]. doi:10.1109/icdar.2009.12.

[5] Paliwal, S.S. et al. (2019) 'TableNet: Deep Learning Model for end-to-end table detection and tabular data extraction from scanned document images', 2019 International Conference on Document Analysis and Recognition (ICDAR) [Preprint]. doi:10.1109/icdar.2019.00029.

Pavlovskyt&amp;#279;, E. (2023) Scrapy vs. Beautiful Soup: A comparison of web scraping tools, Oxylabs. Available at: https://oxylabs.io/blog/scrapy-vs-beautifulsoup (Accessed: 19 November 2023).

Richardson, L. (2019) Beautiful Soup documentation¶, Beautiful Soup Documentation - Beautiful Soup 4.4.0 documentation. Available at: https://beautiful-soup-4.readthedocs.io/en/latest/ (Accessed: 19 November 2023).

[6] Schreiber, S. et al. (2017) 'DeepDeSRT: Deep Learning for detection and structure recognition of tables in document images', 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR) [Preprint]. doi:10.1109/icdar.2017.192.

Whitington, J. (2012) PDF explained. Sebastopol, CA: O'Reilly Media.

Appendix 1

A.1 Sample Extracted Table from WAHIS Report

GENERAL INFORMATION	Unnamed: 0	Unnamed: 1	Unnamed: 2	cattle	NEW 450	1	0	0.1	0.2	0.3	NEW OUTBREAKS	- -
GENERAL INFORMATION	ANIMAL TYPE	DISEASE CATEGORY	EVENT ID									
ZONE	TERRESTRIAL	Listed disease	5274									
DISEASE	CAUSAL AGENT	GENOTYPE / SEROTYPE /	START DATE									
		SUBTYPE	START DATE									
Bluetongue virus (Inf. with)	Bluetongue virus	3	2023/10/10									
REASON FOR NOTIFICATION	DATE OF LAST OCCURRENCE	CONFIRMATION DATE	EVENT STATUS									
Recurrence of an	2021/04/05	2023/10/12	On-going									

eradicated disease												
				(domestic)	TOTAL 450	1	0	0	0	0		
				sheep	NEW -	-	-	-	-	-		
				(domestic)	TOTAL 592	2	1	0	0	0		
				all species	NEW 450	1	0	0	0	0		
					TOTAL 1042	3	1	0	0	0		
											OB_127196 - 23-009-00008 - KLEVE	
	END DATE	DETAILED CHARACTERISATION									OUTBREAK REFERENCE START DATE	

	2023/11/06	-									23-009-00008 2023/10/27	
	THIRD ADMINISTRATIVE	EPIDEMIOLOGICAL UNIT									FIRST ADMINISTRATIVE DIVISION SECOND ADMINISTRATIVE	
	DIVISION										DIVISION	
	Kleve	Farm									Nordrhein-Westfalen Kleve	
	OUTBREAKS IN CLUSTER	Measuring unit									LOCATION Latitude, Longitude	
	-	Animal									Kleve 51.81 , 6.2	
												UPDATED OUTBREAKS
												OB_126858 - 23-009-00003 - GOCH
	DETAILED CHARACTERISATION											OUTBREAK REFERENCE START DATE END DATE
	-											23-009-00003 2023/10/19 2023/11/06
	EPIDEMIOLOGICAL UNIT											FIRST ADMINISTRATIVE DIVISION SECOND ADMINISTRATIVE THIRD ADMINISTRATIVE
												DIVISION DIVISION

	Farm											Nordrhein-Westfalen Kleve Goch
	Measuring unit											LOCATION Latitude, Longitude OUTBREAKS IN CLUSTER
	Animal											Goch 51.65 , 6.13 -
												(Approximate location)
												AFFECTED POPULATION DESCRIPTION
												-
	Vaccinated											Species Wildlife Susceptible Cases Deaths Killed and Slaughtered/ Killed for
												type Disposed of commercial use
	-											sheep NEW - - - - -
	0											(domestic) TOTAL 22 1 1 0 0
												METHOD OF DIAGNOSTIC
												Diagnostic test
												CONTROL MEASURES DIFFERENT FROM EVENT LEVEL
												MEASURES NOT IMPLEMENTED ADDITIONAL MEASURES
												- -

												OB_126469 - 23-009-00002 - KLEVE
	DETAILED CHARACTERISATION											OUTBREAK REFERENCE START DATE END DATE
	-											23-009-00002 2023/10/10 2023/11/06
	EPIDEMIOLOGICAL UNIT											FIRST ADMINISTRATIVE DIVISION SECOND ADMINISTRATIVE THIRD ADMINISTRATIVE
												DIVISION DIVISION
	Farm											Nordrhein-Westfalen Kleve Kleve
	Measuring unit											LOCATION Latitude, Longitude OUTBREAKS IN CLUSTER
	Animal											Kleve 51.86 , 6.06 -