A pipeline for tabular dataset formation from unstructured data provided by ACR Appropriateness Criteria guidelines

Anderson A. Eduardo[1], Rafael M. Loureiro[2], Adriano Tachibana[2], Pedro V. Netto[2], Tatiana F. de Almeida[3], André Pires[1]

[1]TMDA – Hospital Israelita Albert Einstein (HIAE)
01451-001 – São Paulo, SP – Brazil
[2]Relacionamentio Médico – Hospital Israelita Albert Einstein (HIAE)
05652-900 – São Paulo, SP – Brazil
[3]Departmento de Imagem – Hospital Israelita Albert Einstein (HIAE)
05652-900 – São Paulo, SP – Brazil

{act082, rafael.loureiro, adriano.tachibana, pedro.netto, tatiana.almeida, andre.dsantos}@einstein.br

Abstract. *Among the current data-centric technologies, clinical decision support systems (CDSS) figure out as one of the most promising for healthcare. Despite the technological advances facilitating its implementation, the maintenance of knowledge base for CDSS remains open to improvements. Here, we argue that the Appropriateness Criteria provided by ACR guidelines can be used as an open data-source that, combined with appropriate algorithms, can push forward basic research and technological developments regarding knowledge bases for CDSS. Therefore, we developed a pipeline capable of forming tabular datasets from ACR guidelines, stored in a web site in textual PDF files. We also experimentally demonstrate that the proposed pipeline successfully recovers the interested contents, and the best composition, in terms of its component algorithms, is discussed. Future research focused on algorithms flexibility in the face of PDF template updates could improve our work.*

Resumo. *Entre as tecnologias centradas em dados, os sistemas de suporte à decisão clínica (CDSS) figuram entre os mais promissores. Avanços tecnológicos facilitaram sua implementação, mas a manutenção da base de conhecimento para CDSS permanece aberta a melhorias. Aqui, defendemos as diretrizes de adequabilidade do ACR como fonte valiosa de dados abertos e que, se combinados com algoritmos apropriados, podem impulsionar a pesquisa com CDSS. Portanto, desenvolvemos um pipeline capaz de formar conjuntos de dados tabulares a partir das diretrizes do ACR, armazenados em website como arquivos PDF. Também demonstramos experimentalmente que esse pipeline recupera com sucesso os conteúdos de interesse e a melhor composição, em termos de seus algoritmos componentes, é discutida. Pesquisas futuras que focarem na flexibilidade do pipeline frente a atualizações de template dos PDFs contribuirão para o avanço deste trabalho.*

1. Introduction

In the last decade, the concept of data has become a central topic for a wide range of human activities. Concerns on how data is collected, its contents, where and how it is stored, among other issues, currently permeate even disparate areas of inquiry, from law to mathematics, passing through virtually all fields related to modern digital technology. For health sciences, the historical importance of data collection and its availability dramatically deepened with the dawn of data-centric paradigm [Towbin 2019].

Among current data-centric technologies in health sciences, decision support systems (and more specifically, clinical decision support systems - CDSS for short) figures out as one of the most promising concepts [Parsania and Jani 2015, Doyle *et al.* 2019, Sutton *et al.* 2020, Akturk 2021]. Practically speaking, a CDSS is implemented as a software, usually integrated with previously existing information systems [Sutton 2020]. Such software runs under the rotinetely use of a local system, displaying some type of signalization to the user when the prescription of clinical exams is detected. The signalization delivers suggestions of best practices, given patient information [Sutton *et al.* 2020].

One of the most critical steps in implementing a CDSS regards the knowledge base formation and maintenance [Kumar 2016, Sutton *et al.* 2020]. In the first deployed systems, this work was done manually and was highly labor-intensive. But in the recent years, very flexible and performatic data structures becomes ready to use, greatly facilitating the implementation of knowledge bases. Even so, the maintenance remains open to advances [Kumar 2016, Greenes *et al.* 2018].

Guidelines for professional practice presents a rich and effective source of knowledge for clinical decision support systems [Shiffman and Greenes 1994, Shiffman 1997, Greenes *et al.* 2018]. Specifically for radiology, the American College of Radiology (hereafter, ACR) maintains the Appropriateness Criteria guidelines, a web resource in which a team of experts provide a series of high-quality guidance for prescription of radiological exams. The ACR guidelines are presented in Portable Document Format (*i.e.*, PDF files), listed in a searchable webpage, organized by clinical specialities. The access is free and open, just navigate to the URL https://acsearch.acr.org/list. Such resource encompasses the very nature of a knowledge base for CDSS.

Naturally, the ACR guidelines should be considered a major information source for clinical decision support in radiology (and, in fact, it is). However, it is designed to human readers, being presented in unstructured, textual PDF files, not easily prone to machine reading. We argue that ACR guidelines are too valuable data-resource and computational algorithms and pipelines able to parse it to structured data could help to push forward both basic research and application developments regarding CDSS for radiology.

Thus, in the present work, we conceptualize a pipeline for processing ACR guidelines to a structured tabular dataset. We also provide a python implementation, as well as a benchmark experiment, where correctness and computational performance were assessed.

2. Background

Currently, research on the different aspects of data collection by computational means unfolded in several research programs, each one focused on slightly different aspects of

data collection, its levels of complexity and types of digital applications. Inevitably, the literature is huge, spanning different areas of computer science for decades. Examples are text mining, web mining, document understanding, and information retrieval, just for naming some of them [Zhang and Segall 2008, Zhang *et al.* 2015, Harman 2019, Kim *et al.* 2021, Baviskar *et al.* 2021]. Despite of theoretical and technical specificities, raw data must be collected (or recovered) at some step.

In the context of digital health science projects, Towbin (2019) points out that data collection, besides data analysis, is a core activity, especially in radiology. Data collection can be performed through different approaches [Towbin 2019, Roh *et al.* 2021]. In manual data collection, a few people (or even only one) conduct all the work. It also can be carried out in a distributed manner, by professionals in disparate organizations or geographical locations. This last approach is currently employed by a number of research teams in a world-wide scale [Towbin 2019]. The main advantage concerns that the collection itself probably could be done by any minimally trained person and, in the case of multi-institutional teams, data could be available in a short timeframe. The main disadvantage is related to data quality and heterogeneity, as raw data potentially has been collected using different protocols and equipment. Moreover, the manually collected data is naturally prone to human error [Barchard and Pace 2011, Towbin 2019].

Electronic data collection is the most used approach, as digital storage of large volumes of data has become cheaper over the last few decades and database management systems have advanced to become more palatable to users [Bellatreche et al. 2018, Towbin 2019, Roh *et al.* 2021]. The advantage of this approach regards the large volume and accessibility provided by current data-base applications. Also, such applications can be moderately automated, increasing the availability of relevant data. But specialized professionals are demanded to structure the data-base application, especially in cases of data ingestion from multiple sources [Towbin 2019].

Fully automated data collection, extraction and evaluation is currently possible, thanks to advances in artificial intelligence algorithms and computational power [Towbin 2019, Roh *et al.* 2021]. By this approach, disparate and unstructured data can be processed along with structured data. The final datasets can be presented in tabular data structures, convenient for data analytics and machine learning. This approach has tremendously impacted digital data collection [Towbin 2019, Roh *et al.* 2021]. The downside is that highly specialized professionals are demanded, and a longer timeframe is needed till the application for data collection is up to be used [Towbin 2019].

Several conceptual and practical developments have been done for electronic data collection and fully automated data collection [Towbin 2019, Roh *et al.* 2021, Yin *et al.* 2022]. Despite of that, we were not able to find in the literature any work that has specifically focused on the formation of structured datasets from the ACR guidelines. Being Appropriateness Criteria provided by ACR a valuable data-source, we proceeded with a first conceptual approach and computational implementation and experimentation for tabular dataset formation from such data-source.

3. The proposed pipeline

The PDF documents of ACR guidelines are relatively well-structured texts, presenting its main contents (which are clinical indications and its respective recommendations of

radiological exams) in tables and in visually distinguishable locations throughout the documents. Despite of that, a number of other tables and textual contents usually are presented, such as expert considerations, synthesis of available empirical data, and referential literature. Moreover, the interested contents usually appear in many different places, composing different document sections. An example can be viewed at Figure 1.

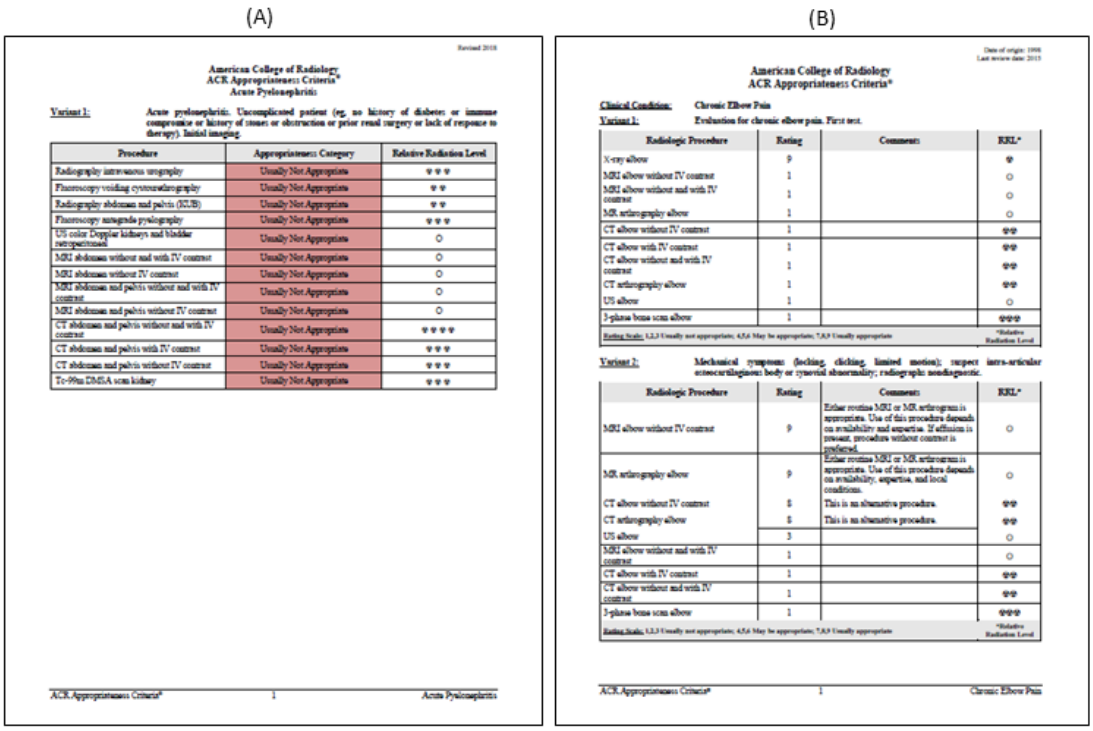

Figure 1. Examples of PDF files provided by ACR guidelines. Note that slightly different templates are used by ACR.

Here, our contents of interest comprises the *Variant* texts and the tables for *Appropriateness Criteria* (see Figure 1). Also, the filename presents the name of the guideline specific group, providing another piece of interested data for our purposes here. Thus, our main algorithm, the `DatasetBuilder` algorithm, is built upon sub-algorithms (also designed by us) able to map specific contents.

At the core of `DatasetBuilder` algorithm, the sub-algorithms `GetFileName`, `ExtractVariants`, `ExtractTables` and `BuildDataset` performs the main tasks independently. As the main algorithm iterates over the set of PDF filepaths provided by user, the `GetFileName` just manipulate such string in order to return the filename from an inputted filepath. The `ExtractVariants` algorithm is an `OCR`-based algorithm which converts each page in the PDF file into images, compute all text blocks coordinates throughout the document, performs the conversion from image to text (*i.e.*, the `OCR`-procedure itself) and, from the obtained set of strings, find *Variant* text blocks, returning it as its output. The `ExtractTables` algorithm is a wrapper for the `tabula-py` and `camelot` python libraries. Both libraries present the computational function `read_pdf()`, which parses file contents and returns the found tables. For a detailed

description of the algorithms underling these third party functions, please refer to the respective projects documentation (https://tabula-py.readthedocs.io/en/latest/index.html, for `tabula-py`; https://camelot-py.readthedocs.io/en/master/ for `camelot`). In a fourth step, the `BuildDataset` receives the output objects from the previous three algorithms and operates on them, in order to restructure such data structures into a single one (specifically, a matrix-like data structure). In tandem, these algorithms are able to extract the interested data from an ACR guideline PDF file, returning it as a single, structured tabular data for that file. Finally, the final output from `DatasetBuilder` is a dictionary, in which the keys are filenames and the values are tabular data structures bearing the interested contents for each respective file. A diagrammatic representation of our pipeline is shown in Figure 2.

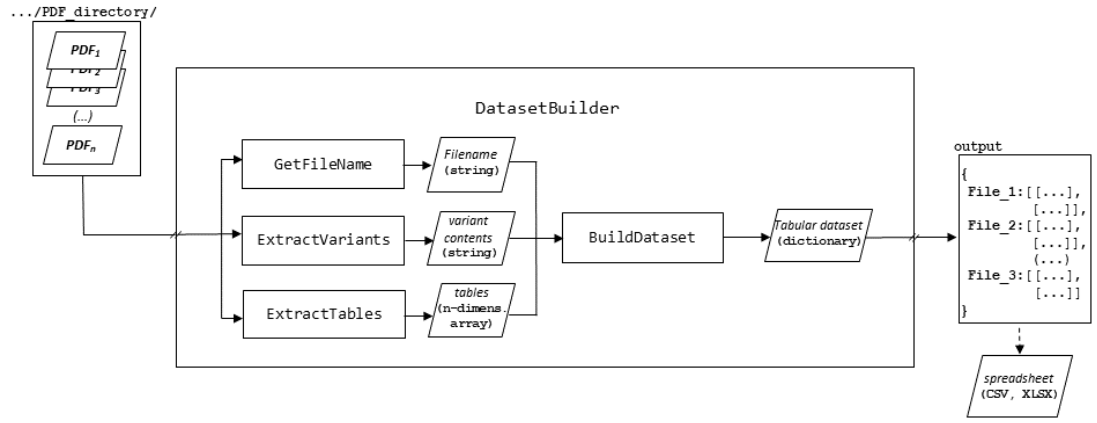

Figure 2. Diagrammatic representation of the proposed pipeline, showing the component algorithms and indication of data structures.

4. Experiment

In order to accomplish the evaluation of the proposed pipeline, our first step was to produce a reference dataset for benchmark. Thus, we randomly selected 10 PDF files at ACR Appropriateness Criteria website (https://acsearch.acr.org/list). The interested contents were manually parsed to a usual `CSV` file and located in a convenient directory in our project file system.

Our benchmark algorithm iterates over each column name found in the benchmark dataset. This is necessary to constrain our analysis focused on the interested contents, teasing apart possible failures regarding other ACR guidelines content, beyond the scope of our research. Future work should be carried out on this topic. For each column, all of its row contents were concatenated into a single textual data structure. Then, our benchmark algorithm performs the comparison of such aggregated column data by the means of comparing the `strings` for the dataset produced by the pipeline against the respective ones in the benchmark dataset. For the measurements of the level of match, the Levenshtein distance were used. Also, we compute a percentual error, by dividing the Levenshtein metric value by the total number of characters in the reference `string` (for

each column). The values were registered along with PDF filename, column name and number of characters in the reference `string` for that column.

To consistently evaluate the proposed pipeline, the benchmark algorithm was structurally encapsulated into an iterative algorithm tailored specifically for this task. This algorithm is responsible for running experiment replicates. These replicates were designed to make it possible to evaluate the time complexity of the proposed pipeline, the performance with different combinations of PDF files, the table extraction core methods (*i.e.*, `tabula-py` and `camelot`), and the stability of the computational implementation. All data referring to iterations were also registered along with the metric value, as described in the paragraph above. The Algorithm 1 (`BenchmarkExperiment`) provides the pseudocode representation for our benchmark algorithm (note that `RandomSelectFiles`, `ReadBechmarkData`, `DatasetBuilder`, `GetColumns`, `GetStringForColumn`, `Levenshtein`, `AgregateData` are auxiliary functions – for the full implementation, please refer to https://github.com/AndersonEduardo/pipeline_acr_guidelines).

```
-------------------------------------------------------------
Algorithm 1: The BenchmarkExperiment algorithm
-------------------------------------------------------------
n_iterations ← user input for the number of iterations
n_max ← user input for the max number of PDF files to be
        processed in the experiment
l ← string inputted by the user, informing the path to
    the directory containing the PDF files
a⃗ := [tabula-py, Camelot]
o := an empty tabular data structure
for each a ∈ a⃗ do
   for each n ∈{1, ..., n_max} do
        for each i ∈{1, ..., n_iterations} do
             p⃗ ← RandomSelectFiles(l, n)
             B ← ReadBechmarkData(p⃗)
             M ← DatasetBuilder(p⃗)
             c⃗ ← GetColumns(B)
             for each c ∈ c⃗ do
                  s_reference ← GetStringForColumn(B[c])
                  s_test ← GetStringForColumn(M[c])
                  d ← Levenshtein(s_reference, s_test)
                  o ← AgregateData(o, d, c, i, n, a)
             end for
        end for
  end for
end for
return o
```

All the implementations were conducted using `python` version `3.8.8`, in a personal computer with an `Intel® Core™ i5-10210U`, `1.60GHz-2.11GHz CPU` and `16GB RAM`. The data produced by the experiment was analyzed graphically, using `Matplotlib`

version `3.3.4` and `Seaborn` version `0.11.1`. All the implementation code is provided via a dockerized project, available at https://github.com/AndersonEduardo/pipeline_acr_guidelines.

5. Results and Discussion

Experimental results show that the proposed pipeline was able to recover up to 100% of our benchmark dataset for the columns *Relative Radiation Level* and *Category*. Also, a high performance was observed for the other columns, especially *Procedure* and *Appropriateness Category*. Strictly speaking, the lower performance was observed for *Subcategory* (Levenshtein distance of ~4), but it must be noted that the percentual error was <0.10% (Figure 3). By inspectioning the experiment outputs, we verify that just a small fraction of characters was not correctly captured by the pipeline in such top tier results.

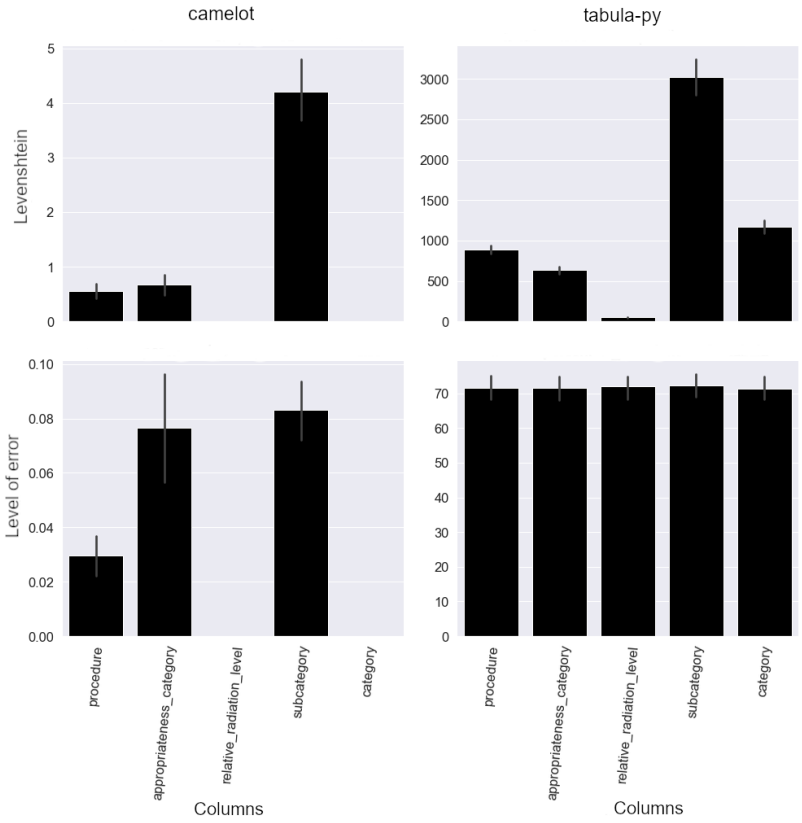

Figure 3. Experiment results for comparison between output datasets, produced using the proposed pipeline, and the benchmark dataset. All comparisons were performed in terms of whole column contents. The percentual error is computed dividing the Levenshtein value by the total number of characters in the reference dataset (details in the text). Note that y-axes are in different scales.

The core function employed in the `ExtractTables` algorithm strongly affected the pipeline output. The best performance was observed only when `camelot` is employed. Using `tabula-py`, the whole performance decays to critical levels, with the interested contents being only loosely recovered. In the worst cases, whole tables were not recovered, compromising the final pipeline output. In fact, `camelot` is built upon `tabula-py`, improving many of its algorithms. Despite of that, the performance results for `tabula-py` do not resembles the one observed for `camelot`, meaning that the relatedness between these python libraries is not translated in terms of similar performance for our pipeline.

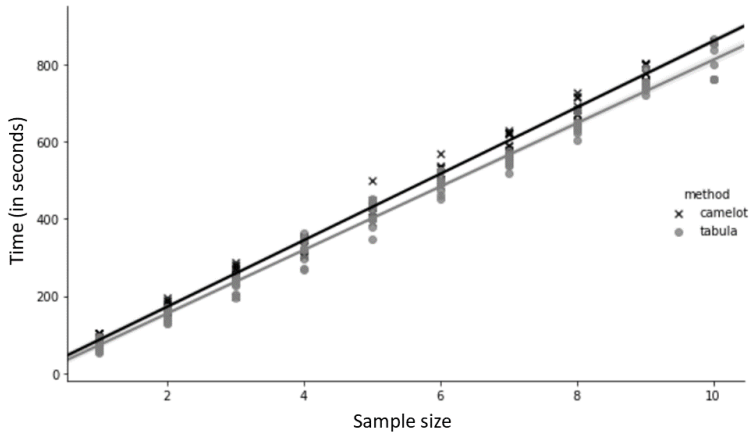

Figure 4. Results of time complexity for the proposed pipeline. The two core functions (*i.e.*, `tabula-py` and `camelot`) for table extraction are compared. A very similar, linear pattern is observed for both versions of the pipeline.

For the range of input files considered in our experiment, the empirical time complexity shows a linear pattern (Figure 4). Moreover, it was very similar for both `camelot` and `tabula-py`, being only sensitively lower for the second one. We attribute such observations to the fact that `tabula-py` loses some tables, thus it is prone to parse slightly less data from de PDF files. In other words, the better performance provided by `camelot` does not take additional time cost, in relation to `tabula-py`. Future work should be carried out on this topic, in order to explore a wider range of input-files number (ideally, in terms of hundreds of PDF files).

6. Conclusion

In this work, we proposed an experimental pipeline designed to form tabular datasets from online Appropriateness Criteria guidelines of ACR. Combining authoral and third-party algorithms (open source), our proposed pipeline relies on NLP and computer vision concepts and technics, being able to successfully parse PDF files from ACR guidelines. Taken together, experimental results shown that our approach recovered the contents of our benchmark dataset with a percentual error of <0.1%, when `camelot` is employed for

table extraction. Through the pipeline, the Appropriateness Criteria data from ACR becomes readily accessible for machine learning and data analytics studies. The python implementation is available, and it shows a good computational performance even in an ordinary personal computer.

Future work should focus on making the pipeline algorithms more flexible, in view of stability in the face of changes or updates to PDF templates by ACR.

References

Akturk, C. (2021). "Bibliometric analysis of clinical decision support sys- tems". In *Acta Informatica Pragensia* 10(1), pages 61–74. doi: 10. 18267/J.AIP.146.

Barchard, K. A. and Pace, L. A. (2011). "Preventing human error: The impact of data entry methods on data accuracy and statistical results". In *Computers in Human Behavior* 27(2011), pages 1834–1839. doi: 10.1016/j.chb.2011.04.004.

Baviskar, D. *et al.* (2021). "Efficient automated processing of the unstructured documents using Artificial Intelligence: A systematic literature review and future directions". In *IEEE Access* 9(2021), pages 72894–72936. doi: 10.1109/ACCESS.2021.3072900.

Bellatreche, L., Valduriez P. and Morzy T. (2018). "Advances in Databases and Information Systems". In *Information Systems Frontiers* 20(2018), pages 1–6. doi: 10.1007/s10796017-9819-2.

Doyle, D. *et al.* (2019). "Clinical decision support for high-cost imaging: A randomized clinical trial". In *Plos One* 14(3-2019), e0213373. doi: 10.1371/journal.pone.0213373.

Harman, D. (2019). "Information Retrieval: The Early Years". In *Foundations and Trends in Information Retrieval* 13(5), pages 425–577. doi: 10.1561/1500000065.

Geewook, K. *et al*. (2021). "Donut: Document Understanding Transformer without OCR". In *ArXiv* (Nov. 2021). doi: 10.48550/arxiv.2111.15664. url: http://arxiv.org/abs/2111.15664.

Greenes, R. A. *et al.* (2018). "Clinical decision support models and frameworks: Seeking to address research issues underlying implementation successes and failures". In *Journal of Biomedical Informatics* 78, pages 134–143. doi: https://doi.org/10.1016/J.JBI.2017.12.005Parsania, V. and Jani, N. (2015). "Reviewing and Modeling Clinical Decision Support System". *In International Journal of Technology and Science* 7 (Dec. 2015), pp. 15–17.

Roh, Y. *et al.* (2021). "A Survey on Data Collection for Machine Learning: A Big Data-AI Integration Perspective". In *IEEE Transactions on Knowledge and Data Engineering* 33(4), pages 1328–1347. doi: https://doi.org/10.1109/TKDE.2019.2946162

Shiffman, R. N. (1997). "Representation of Clinical Practice Guidelines in Conventional and Augmented Decision Tables". In *Journal of the American Medical Informatics Association* 4(5), pages 382–393. doi: https://doi.org/10.1136/jamia.1997.0040382

Shiffman, R. N. and Greenes, R. A. (1994). "Improving Clinical Guidelines with Logic and Decision-table Techniques". In *Medical Decision Making* 14(3), pages 245–254. doi: https://doi.org/10.1177/0272989X940140030Sutton, R. T. *et al.* (2020). "An

overview of clinical decision support systems: benefits, risks, and strategies for success". In *NPJ Digital Medicine* 3(1), pages 17-29. doi: 10.1038/s41746-020-0221-y.

Towbin, A. J. (2019). "Collecting Data to Facilitate Change". In *Journal of the American College of Radiology* 16(2019), pages 1248–1253. doi: 10.1016/j.jacr.2019.05.032.

Zhang, Q. and Segall, R. S. (2008). "Web mining: a survey of current research, techniques, and software". In *International Journal of Information Technology & Decision Making* 7(2008), pages 683–720. doi: 10.1142/S0219622008003150.9.

Yin, A. L. *et al.* (2022). "Comparing automated vs. manual data collection for COVID-specific medications from electronic health records". In *International Journal of Medical Informatics* 157, page 104622. doi: https://doi.org/10.1016/j.ijmedinf.2021.104622

Zhang, Y., Chen, M. and Liu, L. (2015). "A review on text mining". In *6th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, pages 681–685. doi: 10.1109/ICSESS.2015.7339149.10.