Human-Machine Information Extraction Simulator for Biological Collections

Icaro Alzuru
CISE Department
University of Florida
Gainesville, USA
ialzuru@ufl.edu

Aditi Malladi
CISE Department
University of Florida
Gainesville, USA
aditi.malladi@ufl.edu

Andréa Matsunaga
ACIS Lab.
University of Florida
Gainesville, USA
ammatsun@acis.ufl.edu

Maurício Tsugawa
ACIS Lab.
University of Florida
Gainesville, USA
tsugawa@ece.ufl.edu

José A.B. Fortes
ACIS Lab.
University of Florida
Gainesville, USA
fortes@ufl.edu

***Abstract*—In the last decade, institutions from around the world have implemented initiatives for digitizing biological collections (biocollections) and sharing their information online. The transcription of the metadata from photographs of specimens' labels is performed through human-centered approaches (e.g., crowdsourcing) because fully automated Information Extraction (IE) methods still generate a significant number of errors. The integration of human and machine tasks has been proposed to accelerate the IE from the billions of specimens waiting to be digitized. Nevertheless, in order to conduct research and trying new techniques, IE practitioners need to prepare sets of images, crowdsourcing experiments, recruit volunteers, process the transcriptions, generate ground truth values, program automated methods, etc. These research resources and processes require time and effort to be developed and architected into a functional system. In this paper, we present a simulator intended to accelerate the ability to experiment with workflows for extracting Darwin Core (DC) terms from images of specimens. The so-called HuMaIN Simulator includes the engine, the human-machine IE workflows for three DC terms, the code of the automated IE methods, crowdsourced and ground truth transcriptions of the DC terms of three biocollections, and several experiments that exemplify its potential use. The simulator adds Human-in-the-loop capabilities, for iterative IE and research on optimal methods. Its practical design permits the quick definition, customization, and implementation of experimental IE scenarios.**

Keywords—Information extraction, simulator, human-machine, human-in-the-loop, crowdsourcing, optical character recognition, natural language processing

I. Introduction

The digitization of the information stored in biological collections (biocollections) has accelerated in the last decade [1]. Around the world, funding programs, like the Advancing Digitization of Biodiversity Collections (ADBC) [2] of the National Science Foundation, crowdsourcing initiatives for the transcription of the specimens' information, like DigiVol [3] of the Australian Museum, and worldwide engagement campaigns for the digitization of biocollections, like WeDigBio [4]; have made possible for the information from hundreds of millions of specimens to become available in online data repositories [5][6]. The potential use of this information is enormous and crucial to preserve Earth's biological heritage.

In general, digitization entails the conversion of a physical entity into a representation that can be processed by computers. The digitization of a biocollection includes the curation, cataloging (e.g., bar coding of specimens), imaging, information transcription, and post-processing of specimens and their related data. The information transcription and post-processing steps can be completed at the same time as the imaging process [7], but they are commonly performed in a posterior process to preserve the integrity of the specimen and to benefit from the utilization of volunteers or non-expert users for the transcription task [8]. In this paper, we use the term Information Extraction (IE) to refer to the process of identification and transcription of Darwin Core (DC) Terms [9] from the photos of the specimens, saving those values in a structured file or database. This IE from biocollections is the focus of this work.

Driven by the challenge of digitizing billions of specimens [10], the use of non-expert users for the transcription of information (crowdsourcing) [11] [12] has motivated studies on how to engage [13], evaluate [14], and efficiently use human work [15]. The complex characteristics of biocollections' images justify the utilization of volunteers to perform the transcription of the specimens' metadata. The text in these pictures may be written in different languages and styles (handwriting, typewriting, printed, and stamped text), using different typefaces and font sizes; it can be skewed, overlapped by objects, and have different background colors. The layout of their content does not follow any specific pattern. This variability causes Optical Character Recognition (OCR) engines, when applied to these images, to be prone to errors, which compromises Natural Language Processing (NLP) algorithms' ability to extract the correct DC values.

The progress made in Artificial Intelligence during the last decade has especially impacted OCR and NLP techniques. In particular, separated neural networks (previously used to recognize each of the characters of every font type) have been replaced by Long short-term memory (LSTM) networks, which have improved the character error rate [16] and have enabled general handwriting recognition models [17]. Despite this progress, OCR outputs still contain errors and human labor is needed to correct and complete the extracted values.

The HuMaIN project [18] was created with the objective of studying hybrid human-machine approaches for the efficient IE from biocollections. One of its proposals has been a workflow model called SELFIE (Self-aware IE) [19], which organizes the available IE methods in a cost-incremental order to minimize the amount of human work dedicated to crowdsourcing in IE projects. SELFIE tasks identify when the values extracted by automated methods are correct, preventing these values from being extracted by humans.

978-1-7281-0858-2/19/$31.00 © 2019 IEEE
978-1-7281-0858-2/19/$31.00 ©2019 IEEE

The implementation of SELFIE workflows and research projects on IE from biocollections by using either crowdsourcing, automated processes, or hybrid human-machine methods, requires effort, time, and resources to be designed and implemented. Studies involving crowdsourcing require datasets, platforms, and volunteers to perform the tasks. Automated IE processes require access to datasets and their ground truth values to measure quality. These requirements delay and prevent research studies. To date, limited semi-automated approaches have been implemented for the IE from biocollections.

In this paper, we present the HuMaIN Simulator, a tool for data scientists and biologists to easily conduct and experimentally validate research about human-machine IE from biocollections. The contributions of this study are:

- Simulator: The engine and its scripts permit to run and supervise the simulation process, to clone projects and workflows, to define and run sets of similar simulations, and to run Human-in-the-loop (HITL) workflows, enabling crowdsourced data being used to iteratively train machine tasks. The available metric and post-processing scripts permit to visualize the results through tables and graphs and compare the output from different simulations.
- Dataset: The dataset of the Augmenting-OCR Working Group of iDigBio [20] was extended to include the crowdsourced transcription of three DC Terms, a reviewed ground-truth version of the data, the OCR output of three engines (OCRopus [21], Tesseract [22], and the Google-cloud OCR (GC-OCR) [23]), and the output for several automated IE methods, which include the execution time, for the specimens of the three biocollections of the dataset. These data permit researchers to try different scenarios and compare the quality of their methods.
- Workflows: Hybrid IE workflows for three DC Terms (Event-date, Scientific-name, and Recorded-by) are made available with the simulator. Their automated components implement three different IE techniques that are applicable to most of the types of terms found in Darwin Core:
  - Regular expressions (for the Event-date term) or pattern matching of the named entity's values. See sections IV.A, IV.B, and IV.C.
  - Pre-built dictionaries (for the Scientific-name term) or gazetteers: a countable set of known values or alternatives. See sections IV.D and IV.E.
  - Unknown dictionary (for the Recorded-by term): a countable set of undetermined values. See section IV.F.

  Users can clone and use as their starting point these workflows and IE methods included with the simulator.
- Examples: Four experiments (see sections IV.B, IV.C, IV.E, and IV.F,) exemplify the use of the features and data of the simulator to test research scenarios.
- Open Source and Reproducibility: Experiments can be easily replicated. The code and data are openly accessible at https://github.com/acislab/HuMaIN_Simulator.

Researchers are encouraged to extend the simulator, use their own use cases, and sharing their data and code contributions.

II. Related Work

To the best of our knowledge, there is no simulator for IE from biocollections, nor a hybrid human-machine IE simulator, nor a HITL IE simulator. In the area of biocollections, current IE projects seem to focus on the engagement of volunteers [13], instead of the automation of IE methods.

In other areas, several simulators that use HITL have been implemented [24], but not for IE. In the last years, the research on HITL simulators has been pushed by the driver-less car industry [25] and the automation of car capabilities [26].

In the area of IE, the HITL approach has been utilized to keep an ontology up-to-date [27] or extracting relations from unstructured content [28], and in general, research projects have used human-annotated text to train machine learning models. Nevertheless, these studies and others in named-entity recognition [29] work with documents containing natural language, where parts of speech and other NLP technologies are the enablers of their methods. That is not the common case in the text found in biocollections' images, which are basically scattered values.

Several organizations, like iDigBio [5] and GBIF [30], provide online access to the images and DC terms of millions of specimens. Open data access and promotion of research on biodiversity are two of their goals. They provide access to two important resources for IE research: the specimens images and their correspondent transcribed DC terms. API interfaces allow to programmatically download these data. However, the human or machine IE processes which generate those DC terms, the associated data, and metrics are not shared.

In the HuMaIN Simulator, the emphasis is on improving and testing IE processes, the metrics used to evaluate the quality of the output, and the human-machine integration. These steps entail significant time and resource overheads in real life.

Mei et. al [31] provide a dataset for OCR post-processing evaluation. It is especially useful for OCR engines' quality comparison and line segmentation, but the utilized images are different from biocollections' images. Their images come from biodiversity books with pages of natural-language text and low graphical variability.

In [32], Dillen et. al share a dataset of labeled data from herbarium specimen images. Their dataset is limited to herbaria specimens. They provide the segmentation coordinates of the labels of the images, which may be useful for the localization of blocks of text but not for lines (used by OCR engines). In this previous study there are no data from crowdsourcing (which is the most common extraction method), full text transcriptions of the images (for OCR studies), OCR outputs, or customizable NLP scripts. The authors provide a diverse set of images and DC values from herbarium specimens, but research studies are not facilitated in any other way. Similar capabilities can be obtained by using the API functions of iDigBio or GBIF.

In the area of NLP, we are aware of GLUE [33], a benchmark and analysis platform for natural language understanding, which could potentially help the biodiversity community in their IE projects. This is a general-purpose test suite, which includes a convenient performance evaluation tool

for language models. However, images found in biocollections contain text that uses natural language very sparsely, basically in only two DC terms: locality and habitat. The rest of the DC terms may be more appropriate for extraction techniques that do not rely on natural-language features.

Human-machine simulators have been utilized in other areas. For example, in the usability analysis of Web sites [34], or in the accuracy analysis of information transmitted to nurses and physicians [35]. Nevertheless, these projects based their simulations on dynamic interfaces and the mimicking or acting of medical cases using real nurses and physicians; this is different from the simulations in HuMaIN, where we use real IE tasks' results, whenever necessary, to emulate the execution of human-machine IE workflows from biocollections' images.

The HuMaIN Simulator presents few capabilities when compared to general purpose workflow management systems like Pegasus [36] or Kepler [37], because it is specific to biocollections. However, our simulator includes data manipulation scripts, workflows, and capabilities, like HITL, that would need to be added by users to these general-purpose simulation systems.

III. The HuMAIN Simulator

The HuMaIN Simulator works by emulating the execution of the tasks. The simulated tasks have been previously executed. Their results and correspondent metrics are reused in the simulations. Not all the tasks must be simulated, the engine also permits the execution of tasks. However, for performance purposes, we recommend executing in a workflow only those tasks that are under study.

The HuMaIN Simulator permits to conduct different types of studies, for example:

- *Parameter Tuning*: IE tasks usually have multiple parameters. Users can find the optimal value of a parameter by varying its value through different simulations. This use case is illustrated in section IV.E.
- *Tasks Comparison*: Researchers may try different ways to perform one of the tasks of the workflow and evaluate the impact of every option in the output of the workflow. This scenario is exemplified in sections IV.B and IV.C.
- *Evaluation of IE Approaches*: The same DC term(s) can be extracted by different IE workflows. Their evaluation and comparison may be simplified by the re-utilization of the common tasks in the respective workflows.
- *HITL Workflows*: Crowdsourcing results can be used to iteratively improve an automated task. This approach is tested in section IV.F.

To simulate an IE workflow, the components shown in Figure 1 must be defined and made available.

Simulation Configuration

Every simulation is defined through an `xml` file that contains the values for the parameters of every task. This allows running the same workflow with different parameters by simply modifying their values in the configuration file.

Workflow Definition

A workflow is a set of tasks arranged in a specific order, which may potentially include tasks that run in parallel. In the HuMaIN Simulator, a workflow is specified using a `csv` file. Figure 2 shows an IE workflow that follows the SELFIE model: self-aware IE processes arranged in incremental-cost order [19]. In this case, the cost is the execution time. Nevertheless, the simulator can potentially be used to define human-only, machine-only, or other types of workflows. The format of each line in a workflow definition file follows the following pattern:

```
<task_name>, <list_of_prerequisite_tasks>
```

See section IV.A for a workflow definition example.

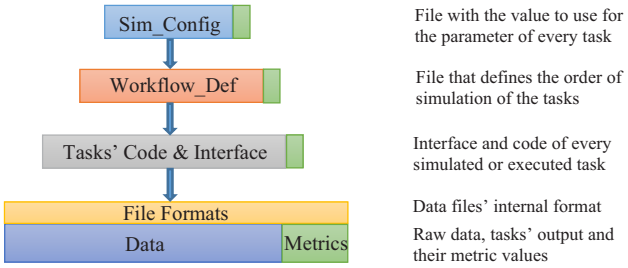

Figure 1. Necessary components for a Simulation.

Tasks' Code & Interface

Tasks are Python scripts that simulate or execute data-handling or IE methods. The simulation of a task is done by retrieving its precomputed results. The simulation script copies the results and metric values from the data repository to the `results` directory, where the output of the simulation is being saved. The simulation saves in a log the information about the process, enabling its debugging and supervision.

The interface (list of parameters) of all the tasks available to the workflows are defined in the file `tasks.xml` of the project. Section IV.A includes an excerpt of this file.

File Formats and Datatypes

The validation of the tasks' input and output values is one of the responsibilities of the simulator. For verification purposes, only a set of datatypes and file formats are accepted, which can be extended modifying the code. The simulator checks every parameter and validates the existence and format of the data in the indicated directories and files passed as parameters. The input and output datatypes and file formats understood by the HuMaIN Simulator are specified in the file `constants.py`, they are the following:

```
INPUT_TYPES = ['INT', 'FLOAT', 'STRING', 'JPG', 'TXT',
'TSV', 'D_JPG', 'D_TXT', 'D_AR']
OUTPUT_TYPES = ['O_JPG', 'O_TXT', 'O_TSV', 'O_D_AR',
'O_D_JPG', 'O_D_TXT']
```

The datatype `D_AR` indicates a directory with two sub-directories: `Accept` and `Reject`, utilized by self-aware tasks to separate the specimens for which a value was generated from the specimens that need to be processed by a higher cost IE process.

Data

The Data component represents the files with the values to be utilized by the tasks. Examples of data are crowdsourced transcriptions, images, OCR's output text, or cropped lines.

Metrics

The metrics correspond to the variables to be measured during the simulation of a workflow. Their values are included in a folder called `/metrics` located inside every data directory. If, for example, a workflow is going to use the execution time as a metric, every task must include a file with the per-specimen execution-time value in the `/metrics` folder.

The specification details of the tasks' interface, workflow-definition, and simulation-configuration files can be found in the wiki page of the HuMaIN Simulator [38]. Section IV.A includes excerpts of these files for the Event-date workflow.

The HuMaIN Simulator, including code and data, can be downloaded by cloning its GitHub repository [38]. It requires Python3 to run, and its installation consists in updating the PYTHONPATH environment variable and the BASE_DIR internal variable.

After downloading and configuring the environment variables, an IE experiment can be defined and executed by following four steps:

1) Project and Workflows: A project is a set of related workflows. Users can define a project and its workflows from scratch, with a text editor by following the specification rules of the simulator, or they can create a copy of an existing project (including workflows) using the script `create_project`, and adapt it to their convenience. The workflows and experiments presented in this paper, can be found in the project called `selfie`, at the simulator's repository.

2) Simulation Configuration: The simulation-configuration file specifies the values of the parameters to use in every task of the workflow. This configuration file can be defined from scratch or created from a copy of an existing file by using the `create_simulation` script. The metrics and post-processing scripts are specified in this file. Post-processing scripts permit to generate tables and graphs from the metrics' results. Two features of the simulator that facilitate IE experimentation are:

a) Groups of Simulations: This feature permits to create a single configuration file to run several simulations. One or more parameters among the group of simulations can be varied to perform, for example, Parameter Tuning studies. The script `create_sim_grp` can be used to automatically generate a configuration file for a Group of Simulations. The simulation engine undertands this type of configuration file.

b) Synthetic metric values: The metrics' values for a simulated task can be syntethically generated using the script called `gen_values`, which permits to generate random values following different types of statistical distributions.

3) Simulation: The simulations are run by using the script `run_simulation`, which arguments are the names of the project, workflow definition, and simulation configuration.

4) Results Verification: Besides the post-processing scripts specified in the configuration file, users can verify the correct simulation of the workflow by checking the log file generated by the HuMaIN Simulator, which registers the parameters and messages of every task.

IV. Experimental Setup and Results

In order to show the potential and usability of the HuMaIN Simulator the following experiments were conducted:

- Section A: The IE workflow for the Event-date DC Term is detailed. The included excerpts of the workflow definition, tasks interfaces, and simulation-configuration files teach how to run a simulation. The results for the quality and execution-time metrics are shown.
- Section B: This experiment studies how the quality of the OCR engine affects the final quality of the workflow. Three different OCR engines are used. This experiment exemplifies how to compare tasks and implement groups of simulations.
- Section C: The experiment shows how different crowds may affect the quality of the workflow's output. In this example, the human task is modified and studied.
- Section D: A workflow for the extraction of the Scientific-name term is presented. The results for the quality and execution-time metrics are shown.
- Section E: The Scientific-name workflow is used to exemplify how to tune an IE parameter. The parameter is the similarity threshold that decides when to accept or reject an extracted value. A group of simulations is utilized in this experiment.
- Section F: The HITL workflow for the extraction of the Recorded-by term is explained. The number of values extracted by human and machine methods are compared. The dynamics and characteristics of HITL workflows are illustrated in this example.

Additional data, workflows, and experiments are available in the GitHub repository of the simulator.

A. Event-date Workflow

Figure 2 shows the SELFIE workflow implemented for the extraction of the Event-date term, which is the date when the specimen is collected. This workflow was previously proposed by Alzuru et. al [19]. In this paper, the workflow was automated following the specification rules of the HuMaIN Simulator. The machine extraction task was extended to several OCR engines.

For testing purposes, 100 specimens were randomly selected from the Insects, Herbs, and Lichens biocollections included in the simulator. The data available for the OCRopus engine were used when emulating the OCR step. The output data for all the tasks on this subset of specimens are included in the simulator's repository. In the workflow, the transcription generated by the OCR is scanned by a script that uses regular expressions to extract the Event-date candidate values.

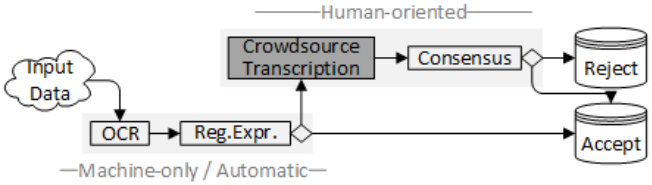

Figure 2. SELFIE workflow for the extraction of Event-date.

If no value is extracted by the regular expression script, the Event-date is asked to be transcribed by three different volunteers. A consensus algorithm uses majority voting and average similarity to decide, from the three transcriptions, the Event-date value for the specimen (see Figure 2).

Two metrics are used in the workflow: Execution Time (called here *Duration* for naming convenience) and Quality. Duration is the total sequential time of machine-processing and crowdsourcing effort required to generate the Event-date values. Quality is computed as the Damerau-Levenhstein [39] similarity of the extracted Event-date to the ground truth values.

The order of simulation of the workflow's tasks is defined in a `csv` file (`event_date.csv`), following the Simulator's syntax (see the *Workflow Definition* component in section III):

```
ocr_sim
ed_reg_expr_sim, ocr_sim
crowdsource_sim, ed_reg_expr_sim
consensus_sim, crowdsource_sim
```

Each of the workflow's tasks must have a correspondent Python script that simulates or executes this task. The parameters of all the tasks of a project are defined in the `tasks.xml` file. An excerpt of this file for the `selfie` project is:

```
…
<task name="ocr_sim">
    <parameter name="ocr_input_dir" type="D_TXT"></parameter>
    <parameter name="include" type="STRING"></parameter>
    <parameter name="specimens_list" type="TXT"></parameter>
    <parameter name="metric" type="STRING"></parameter>
    <parameter name="output_dir" type="O_D_TXT"></parameter>
</task>
…
<task name="crowdsource_sim">
    <parameter name="specimens" type="TXT"></parameter>
    <parameter name="crowd_data" type="TSV"></parameter>
    <parameter name="metric" type="STRING"></parameter>
    <parameter name="output_file" type="O_TSV"></parameter>
</task>
…
```

Once the tasks of the workflow have been defined, users must create a simulation-configuration file specifying the correspondent values for the tasks' parameters, the scripts to compute the final values of the metrics, and the post-processing scripts (graphs and tables generation). An excerpt of the simulation file for the Event-date workflow is the following:

```
<tasks>
  …
  <task name="ocr_sim">
    <parameter name="ocr_input_dir">…/ocropus</parameter>
    <parameter name="include">True</parameter>
    <parameter name="specimens_list">…/specimens.txt</parameter>
    <parameter name="metric">duration</parameter>
    <parameter name="output_dir">…/ocr_sim</parameter>
  </task>
  …
</tasks>
<metrics>
  …
  <script name="quality_measure.py">
    <parameter name="accepted_f">…/regexp/accepted.tsv</parameter>
    <parameter name="accepted_f">…/consen/accepted.tsv</parameter>
    <parameter name="ground_truth">…/event_date.tsv</parameter>
    <parameter name="output_file">…/ed/quality.csv</parameter>
  </script>
  …
</metrics>
<post-processing>
  …
  <script name="basic_stats.py">
    <parameter name="metric_file">…/ed/quality.csv</parameter>
    <parameter name="output_file">…/ed/quality_st.txt</parameter>
  </script>
  …
</post-processing>
```

The metric and post-processing scripts are optional and are automatically executed after the workflow simulation has finished. The metric scripts help to aggregate the values for the metric generated in each task. For the Figure 2 workflow, the simulation file was called `event_date_sim.xml`.

Once verified that the data and scripts are in place, the simulation is started by running the `run_simulation.py` script:

```
run_simulation.py -p selfie -w event_date -s event_date_sim
```

In this experiment, the simulation took 2 seconds. The log file saved the parameter's values and the processing order, among other information. The post-processing scripts generated two tables, one for the total duration (execution time) and another one for the average Damerau-Levenshtein similarity of the extracted values to the ground truth data (see Figure 3).

Total Duration (sec)

duration
7033.318756341099

Average Quality

quality
0.818523807823658

Figure 3. Execution time (duration) and Quality (Damerau-Levenshtein similarity to the ground-truth values) for the Event-date workflow.

If real life, the crowdsourcing tasks and the execution of the scripts would have taken about 7,000 seconds in extracting the Event-date values for these 100 images. The similarity of 0.82 to the ground truth data indicates that machine and/or human extraction methods in the workflow have misspelled or extracted wrong values. Figuring out the cause of the errors and improving those IE processes are two of the possible objectives of the researchers who use the HuMaIN Simulator.

Once the workflow has been implemented following the HuMaIN Simulator specifications, users can reuse the workflow and its tasks to perform IE studies in a fraction of the time it would take to implement them without the simulator.

B. OCR Engines Comparison.

This study illustrates how the selection of the OCR engine affects the execution time and quality of the hybrid IE workflow's output. Three OCR engines are considered: OCRopus, Tesseract, and the Google Cloud OCR (GC-OCR).

To test the Event-date workflow with the three OCR engines, a group of simulations was configured by running the `create_sim_grp.py` script. This script creates a simulation-configuration file which internally defines three simulations, one per OCR engine. The script was executed as follow:

```
create_sim_grp.py -p selfie -w event_date -s event_date_sim
-a ocr_sim -p ocr_input_dir -v …/ocropus -v …/tesseract -v
…/gc-ocr -o ed_sim_grp
```

The generated xml file can be customized. In this example, we specified in the post-processing section that two bar graphs for the comparison of the OCR engines should be generated. These graphs are shown in Figure 4.

The output quality of Tesseract and GC-OCR was slightly higher than the obtained by OCRopus, but the GC-OCR required less time because more Event-date values were recognized by the automated process (OCR + regular expression). Further metrics and analysis can be added to this comparison. However, the objective of the study is to show how easy is, by using the HuMaIN Simulator, to modify a task and perform IE studies.

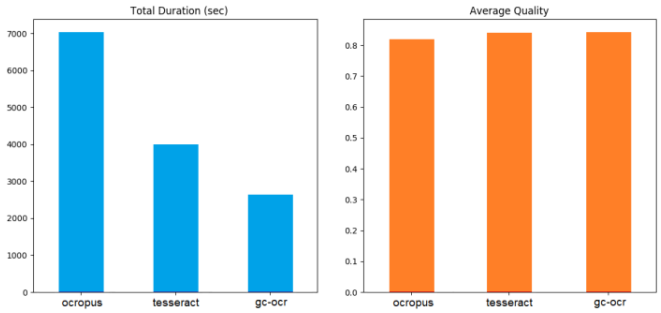

Figure 4. Duration (seconds) and Quality (similarity to the ground-truth values) using OCRopus, Tesseract, and GC-OCR in the Event-date workflow.

C. *Crowd comparison.*

In the last two simulations, we have utilized the transcription of the Event-date term performed by volunteers of the Zooniverse portal [40]. These valuable citizen scientists help for free in research projects by dedicating time and effort to complete repetitive tasks that are hard to do by machines. Zooniverse's users were asked to (voluntarily) read a tutorial before starting to transcribe the Event-date values from the images of the dataset.

In this study, we try the transcriptions generated by a different crowd and check how the execution time and quality of the Event-date workflow differs to the achieved by using the Zooniverse volunteers' data.

The new crowd are undergraduate students, paid at a rate of $10 per hour, who were instructed, in person, how to perform the transcription of the Event-date term. The consensus algorithm was applied to the data collected from these users to generate a final Event-date value for every specimen. These data were included in the simulator's dataset.

The simulation of the Event-date workflow was repeated for the new crowd's data. The comparison of the results obtained for both crowds is shown in Figure 5.

Total Duration (sec)

Zooniverse's Volunteers	Paid Students
7033.3	5960.4

Average Quality

Zooniverse's Volunteers	Paid Students
0.8185	0.8338

Figure 5. Duration (seconds) and Quality (Damerau-Levenshtein similarity) for the Event-date workflow, when using two different crowds.

The paid crowd accelerated the IE process by 15% and improved the result's quality by about 2%. This experiment and the previous one show how, after implementing in the HuMaIN Simulator an IE workflow, studies on the different tasks are highly simplified by reusing the same structure.

D. *Scientific-name Workflow.*

The SELFIE workflow proposed in [19] for the Scientific-name term was automated in the HuMaIN Simulator, see Figure 6. The workflow is composed by two automated self-aware processes and a human-centered IE process (crowdsourcing).

The simulator includes scientific name dictionaries for herbs, lichens, and insects. Because the subset of 100 images used to test this workflow includes specimens from the three collections, a dictionary that includes all their entries (more than 51 thousand) was added to the `dataset` directory.

Scanning and comparing every word extracted by the OCR to the 51K entries of the dictionary may be slow. That is the reason why the first self-aware process tries to identify scientific names inside the OCR output using suffixes commonly found in scientific names (like -iae or -anum). After identifying words and pair of words with these suffixes, they are compared to the entries of the dictionary. If this IE process does not extract the Scientific-name value, all the text is scanned and its words are compared to the entries in the dictionary. If this second method also fails to extract a high-confident value, three volunteers are asked to transcribe the scientific name of the specimen.

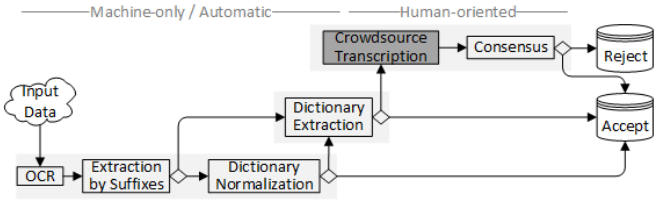

Figure 6. SELFIE workflow for the extraction of Scientific name.

The data collected in the real execution of all these IE tasks were saved in the simulator's `dataset` folder. The simulation tasks and the xml configuration files for the simulation were implemented. Using the GC-OCR as OCR engine, the Scientific-name workflow was run. The successfully extracted values were distributed as shown in Figure 7.

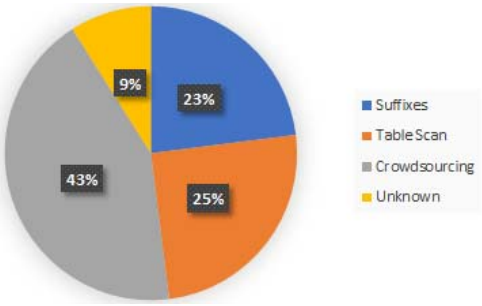

Figure 7. Percentage of Scientific-name values extracted by each of the IE processes or not extracted (Unknown).

The extracted values had an average similarity of 0.63 to the ground-truth values and, in average, every value took 78.5 seconds to be extracted. The simulation took about 3 seconds.

E. *Similarity Threshold for Scientific-name.*

The Dictionary Extraction algorithm used in the Scientific-name workflow accepts or rejects a candidate value based on its similarity to the dictionary's entries. This value was arbitrarily assumed as 0.85 in the previous section. In this study, we want

to know how this threshold affects the quality and number of values accepted during the second IE process of the workflow.

A group of simulations, like in section B, was created for running 11 simulations. They correspond to the values from 0.5 to 1.0 (with a step of 0.05) of the similarity threshold. The results obtained from these simulations are shown in Table I.

For threshold values less than 0.6, the quality of the workflow degraded. For a threshold equal or greater than 0.6, the number of accepted values and their quality remained basically constant. Probably the relatively short length of the compared strings prevents a smoother change in these metrics.

TABLE I. AVERAGE GENERAL QUALITY AND NUMBER OF SCIENTIFIC-NAMES VALUES EXTRACTED IN THE DICTIONARY EXTRACTION TASK FOR DIFFERENT VALUES OF THE SIMILARITY THRESHOLD.

Similarity Threshold	0.5	0.55	0.6 - 0.85	0.9 – 1.0
Number of Accepted Values	37	36	25	24
Avg. Similarity to Ground-truth	0.53	0.55	0.63	0.63

F. HITL Recorded-by Workflow.

The SELFIE workflow proposed in [19] for the extraction of the Recorded-by (Collector) term was converted to HITL and automated for the HuMaIN Simulator. This term is different to Scientific-name because there is not a pre-defined dictionary of collectors. Since biological collections tend to have a reduced number of collectors, humans can be used to transcribe the Recorded-by value of a limited number of specimens and a dictionary or list of collectors can be built with these values on-demand. Using the dictionary entries, the text in the remaining specimens can be scanned to search for the same collectors found before. These steps can repeat, expanding the dictionary in every iteration, until all the specimens have an extracted value or have been processed using crowdsourcing.

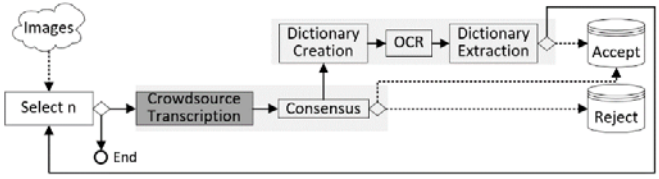

Figure 8. Human-in-the-loop workflow for the extraction of the Recorded-by (collector) term.

The HITL workflow for Recorded-by is found in Figure 8. A task randomly selects the specimens that are going to be processed by the human-centered tasks in every iteration. When there are no more specimens to be processed, the simulation of the workflow stops.

The dynamic of iterative workflows cannot be observed with few images. Therefore, for this study we utilized a different dataset of 10 biocollections and 14,233 specimens (images). The collections were processed in separated simulations.

Table II shows the number of images per collection, the iterations required by the workflow to process all of them, and the humans-accepted, machine-accepted, and rejected specimens, considering all the iterations. There are collections with very few collectors, which required just two or three iterations, while other collections required more iterations and crowdsourcing sessions to unveil all the possible collectors.

TABLE II. ABSOLUTE VALUES OF THE ITERATION PROCESS, PER COLLECTION.

Collection	1	2	3	4	5	6	7	8	9	10
# Images	739	2880	1041	1639	2152	704	901	954	1252	1971
# Iterations	5	11	2	3	6	1	5	6	4	5
Human Accepted	224	504	73	136	252	45	207	253	169	205
Machine Accepted	511	2359	967	1489	1897	654	685	686	1077	1743
Rejected	4	17	1	14	3	5	9	15	6	23

In average, about 20% of the Recorded-by values needed to be transcribed by humans, see Figure 9. Using HITL, the transcription of about 80% of the values could be automatically extracted by using the dictionary iteratively generated with the crowdsourced data. Leading to big savings in the number of humans required to perform the IE project.

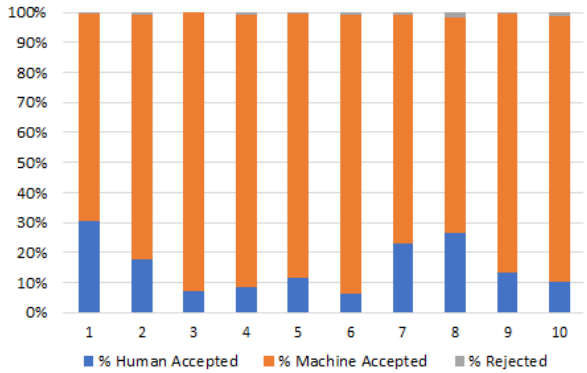

Figure 9. Per-collection distribution of the Recorded-by values accepted by crowdsourcing (humans), dictionary extraction (machines), or rejected.

CONCLUSIONS

The Information Extraction (IE) from photographs of biocollections specimens is a challenging process that needs to be accomplished with hybrid human-machine approaches because automated machine-only methods cannot provide, to date, an output quality as good as the humans can provide.

In order to advance in the research of the IE from biocollections, this study proposes a human-machine simulator for the extraction of the specimens' Darwin Core terms. The IE workflows can include executed and simulated tasks. The simulated tasks reuse the output of tasks previously executed.

The simulator permits to accelerate the experimental process by copying and reusing workflows, tasks, simulations, and data. Groups of simulations can be automatically generated by specifying different parameter values, while Human-in-the-loop capabilities allow running iterative simulations that incrementally improve automated tasks from the data generated by humans. Embedded graphical capabilities permit to generate tables, box plots, and bar graphs to easily visualize the results and compare different simulations.

After implementing a workflow in the HuMaIN Simulator, several experimental scenarios can be easily explored:

parameter tuning, tasks comparison, evaluation of IE approaches, and HITL workflows.

The process of definition of the components of a workflow was detailed, while three workflows and four experiments were presented to exemplify the research process and potentiality offered by the HuMaIN Simulator.

ACKNOWLEDGMENT

This work is supported in part by the National Science Foundation (NSF) grants No. ACI-1535086, DBI-1115210, DBI-1547229, and the AT&T Foundation. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF or the AT&T Foundation.

REFERENCES

[1] Nelson Gil and Ellis Shari, "The history and impact of digitization and digital data mobilization on biodiversity research," *Philos. Trans. R. Soc. B Biol. Sci.*, vol. 374, no. 1763, p. 20170391, Jan. 2019.

[2] National Science Foundation, "Advancing Digitization of Biodiversity Collections." [Online]. Available: https://www.nsf.gov/funding/pgm_summ.jsp?pims_id=503559. [Accessed: 23-Jun-2019].

[3] Australian Museum, "DIGIVOL." [Online]. Available: https://digivol.ala.org.au/. [Accessed: 23-Jun-2019].

[4] "Worldwide Engagement for Digitizing Biocollections (WeDigBio)." [Online]. Available: https://wedigbio.org/. [Accessed: 23-Jun-2019].

[5] "Integrated Digitized Biocollections (iDigBio)," *iDigBio*. [Online]. Available: https://www.idigbio.org/home. [Accessed: 23-Jun-2019].

[6] "Global Biodiversity Information Facility (GBIF)." [Online]. Available: https://www.gbif.org/. [Accessed: 23-Jun-2019].

[7] P. Sweeney *et al.*, "Large-scale digitization of herbarium specimens: Development and usage of an automated, high-throughput conveyor system," *Taxon*, vol. 67, Mar. 2018.

[8] G. Nelson *et al.*, "Digitization Workflows for Flat Sheets and Packets of Plants, Algae, and Fungi," *Appl. Plant Sci.*, vol. 3, no. 9, p. 1500065, Sep. 2015.

[9] Biodiversity Information Standards (TDWG), "Darwin Core quick reference guide." [Online]. Available: https://dwc.tdwg.org/terms. [Accessed: 29-Jun-2019].

[10] A. H. Ariño, "Approaches to estimating the universe of natural history collections data," *Biodivers. Inform.*, vol. 7, no. 2, Oct. 2010.

[11] P. Flemons and P. Berents, "Image based Digitisation of Entomology Collections: Leveraging volunteers to increase digitization capacity," *ZooKeys*, no. 209, pp. 203–217, Jul. 2012.

[12] E. R. Ellwood *et al.*, "Accelerating the Digitization of Biodiversity Research Specimens through Online Public Participation," *BioScience*, vol. 65, no. 4, pp. 383–396, Apr. 2015.

[13] E. R. Ellwood *et al.*, "Worldwide Engagement for Digitizing Biocollections (WeDigBio): The Biocollections Community's Citizen-Science Space on the Calendar," *BioScience*, vol. 68, no. 2, pp. 112–124, Feb. 2018.

[14] C. Qiu, A. Squicciarini, D. R. Khare, B. Carminati, and J. Caverlee, "CrowdEval: A Cost-Efficient Strategy to Evaluate Crowdsourced Worker's Reliability," in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, Richland, SC, 2018, pp. 1486–1494.

[15] A. Matsunaga, A. Mast, and J. A. B. Fortes, "Workforce-efficient Consensus in Crowdsourced Transcription of Biocollections Information," *Future Gener Comput Syst*, vol. 56, no. C, pp. 526–536, Mar. 2016.

[16] C. Reul, U. Springmann, C. Wick, and F. Puppe, "State of the Art Optical Character Recognition of 19th Century Fraktur Scripts using Open Source Engines," *ArXiv181003436 Cs*, Oct. 2018.

[17] R. R. Ingle, Y. Fujii, T. Deselaers, J. Baccash, and A. C. Popat, "A Scalable Handwritten Text Recognition System," *ArXiv190409150 Cs*, Apr. 2019.

[18] ACIS Lab, UF, "Human & Machine Intelligent Network (HuMaIN)." [Online]. Available: http://humain.acis.ufl.edu/. [Accessed: 27-Jun-2019].

[19] I. Alzuru, A. Matsunaga, M. Tsugawa, and J. A. B. Fortes, "SELFIE: Self-Aware Information Extraction from Digitized Biocollections," in *2017 IEEE 13th International Conference on e-Science (e-Science)*, 2017, pp. 69–78.

[20] "iDigBio Augmenting OCR Working Group & Hackathon," *GitHub*. [Online]. Available: https://github.com/idigbio-aocr. [Accessed: 27-Jun-2019].

[21] "OCRopy - Python-based tools for document analysis and OCR.," 29-Jun-2019. [Online]. Available: https://github.com/tmbdev/ocropy. [Accessed: 29-Jun-2019].

[22] "Tesseract Open Source OCR Engine," 30-Jun-2019. [Online]. Available: https://github.com/tesseract-ocr/tesseract. [Accessed: 29-Jun-2019].

[23] "Detect text in images," *Google Cloud*. [Online]. Available: https://cloud.google.com/vision/docs/ocr. [Accessed: 29-Jun-2019].

[24] L. Rothrock and S. Narayanan, Eds., *Human-in-the-Loop Simulations: Methods and Practice*. London: Springer-Verlag, 2011.

[25] H. Yang, X. Li, P. Li, and Y. Gao, "The Driver-in-the-Loop Simulation on Regenerative Braking Control of Four-Wheel Drive HEVs," in *Advances in Mechanical Design*, 2020, pp. 214–222.

[26] C. Allmacher, M. Dudczig, S. Knopp, and P. Klimant, "Virtual Reality for Virtual Commissioning of Automated Guided Vehicles," in *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, 2019, pp. 838–839.

[27] A. Alba, A. Coden, A. L. Gentile, D. Gruhl, P. Ristoski, and S. Welch, "Multi-lingual Concept Extraction with Linked Data and Human-in-the-Loop," in *Proceedings of the Knowledge Capture Conference*, New York, NY, USA, 2017, pp. 24:1–24:8.

[28] I. Lourentzou, A. Alba, A. Coden, A. L. Gentile, D. Gruhl, and S. Welch, "Mining Relations from Unstructured Content," in *Advances in Knowledge Discovery and Data Mining*, 2018, pp. 363–375.

[29] S. M. Yimam, S. Remus, A. Panchenko, A. Holzinger, and C. Biemann, "Entity-Centric Information Access with Human in the Loop for the Biomedical Domain.," in *BiomedicalNLP@ RANLP*, 2017, pp. 42–48.

[30] "Global Biodiversity Information Facility (GBIF)," 23-Jun-2019. [Online]. Available: https://www.gbif.org/. [Accessed: 23-Jun-2019].

[31] "MiBio: A dataset for OCR post-processing evaluation - ScienceDirect." [Online]. Available: https://www.sciencedirect.com/science/article/pii/S235234091830951X. [Accessed: 26-Jun-2019].

[32] M. Dillen *et al.*, "A benchmark dataset of herbarium specimen images with label data," *Biodivers. Data J.*, no. 7, Feb. 2019.

[33] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding," *ArXiv180407461 Cs*, Apr. 2018.

[34] E. H. Chi *et al.*, "The Bloodhound Project: Automating Discovery of Web Usability Issues Using the InfoScent™ Simulator," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA, 2003, pp. 505–512.

[35] Y. Bogenstätter, F. Tschan, N. K. Semmer, M. Spychiger, M. Breuer, and S. Marsch, "How Accurate Is Information Transmitted to Medical Professionals Joining a Medical Emergency? A Simulator Study," *Hum. Factors*, vol. 51, no. 2, pp. 115–125, Apr. 2009.

[36] E. Deelman *et al.*, "Pegasus, a workflow management system for science automation," *Future Gener. Comput. Syst.*, vol. 46, pp. 17–35, May 2015.

[37] "The Kepler Project — Kepler." [Online]. Available: https://kepler-project.org/. [Accessed: 09-Oct-2019].

[38] ACIS Lab., "Human-Machine Information Extraction Simulator for Biological Collections," *GitHub*, 27-Jun-2019. [Online]. Available: https://github.com/acislab/HuMaIN_Simulator. [Accessed: 27-Jun-2019].

[39] W. H.Gomaa and A. A. Fahmy, "A Survey of Text Similarity Approaches," *Int. J. Comput. Appl.*, vol. 68, no. 13, pp. 13–18, Apr. 2013.

[40] "Zooniverse." [Online]. Available: https://www.zooniverse.org/. [Accessed: 19-Jul-2019].