NOAH: Creating Data Integration Pipelines over Continuously Extracted Web Data

Valerio Cetorelli
Università Roma Tre
valerio.cetorelli@uniroma3.it

Valter Crescenzi
Università Roma Tre
valter.crescenzi@uniroma3.it

Paolo Merialdo
Università Roma Tre
paolo.merialdo@uniroma3.it

Roger Voyat
Università Roma Tre
roger.voyat@uniroma3.it

ABSTRACT

We present NOAH, an ongoing research project aiming at developing a system for semi-automatically creating end-to-end Web data processing pipelines. The pipelines continuously extract and integrate information from multiple sites by leveraging the redundancy of the data published on the Web. The system is based on a novel hybrid human-machine learning approach in which the same type of questions can be interchangeably posed both to human crowd workers and to automatic responders based on machine learning (ML) models. Since the early stages of pipelines, crowd workers are engaged to guarantee the output data quality, and to collect training data, that are then used to progressively train and evaluate automatic responders. The latter are fully deployed into the data processing pipelines to scale the approach and to contain the crowdsourcing costs later. The combination of guaranteed quality and progressive reductions of costs of the pipelines generated by our system can improve the investments and development processes of many applications that build on the availability of such data processing pipelines.

CEUR-WS.org/Vol-2841/PIE+Q_3.pdf

1 INTRODUCTION AND MOTIVATION

The Web is the largest knowledge base ever built by humans. However, most of the data on the Web are not directly available to applications, unless complex data extraction and integration pipelines are set-up. Creating these pipelines to build structured knowledge bases and continuously maintain them in a cost effective way is still a challenging problem. Currently, most projects fulfill their data processing needs by means of case-by-case solutions that cannot be reused across projects.

This paper presents NOAH, a research project that aims at developing a system for creating and maintaining over time end-to-end data processing pipelines for continuously extracting and integrating Web data. NOAH is based on an hybrid human-machine learning approach, whose goal is to guarantee the quality of processed data by leveraging feedbacks provided by human crowd workers. Our approach can be classified in the realm of *Open Information Extraction* [31], because it aims at extracting and integrating information both at the instance (objects) and at the schema (attributes) levels into an internal knowledge base (IKB) that is created, populated and maintained for every domain. Indeed, if new sources are incrementally added to an already generated pipeline, the system is able to discover new entities and new attributes from the aforementioned sources.

© 2021 Copyright for this paper by its author(s). Published in the Workshop Proceedings of the EDBT/ICDT 2021 Joint Conference (March 23–26, 2021, Nicosia, Cyprus) on CEUR-WS.org. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0)

In order to contain the crowdsourcing costs, the proposed approach leverages two techniques. First, it exploits the inherent redundancy of Web sources to automatically find correct domain information: data published by several independent sources are more likely to be correct and can be easily discerned by noisy or non-relevant data [8, 15]. Secondly, it exploits the collected data to continuously train ML models. Those ML models are progressively introduced in the form of automatic responders that replace crowd workers [1, 30], and are continuously evaluated during each step of the data processing pipelines: only responders that become sufficiently reliable are fully deployed in the operations of the created pipelines.

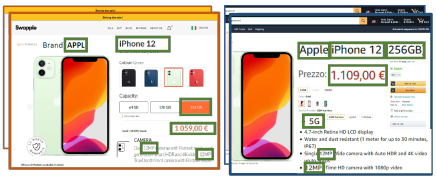


Figure 1: Web detail pages in the *Smartphone* domain.

Problem Description. Given a set of sources $\mathcal{S} = \{S_1, S_2 \ldots\}$ from the same domain (e.g., *Smartphones*); each source S_i is specified by means of n_i URLs of *detail* pages about domain objects (e.g., *IPhone 12*, *Mi 10T*). By detail page we mean a page reporting information about a particular object, the *topic entity* [29] of the page, on which it publishes values of several attributes. An example of detail pages from two sources, about the same *IPhone 12* domain object is shown in Figure 1 where the values of several attributes of interest such as Model, Memory, Price are highlighted.

A domain includes a set of objects $O = \{o_1, o_2, \ldots\}$ and a set of attributes $\mathcal{A} = \{A_1, A_2, \ldots\}$ which will be populated with data extracted from the pages of the sources belonging to that domain. New attributes and new objects of a domain can be discovered as new sources are considered part of the domain.

Each source publishes detail pages reporting the values of a subset of domain attributes, for a subset of domain objects. We use the terms *source attributes* or *source objects* when we want to denote the version of a domain attribute or object as published by a source, i.e., we are referring to the occurrences of attribute values about an object as published by a source. It is worth noticing that some domain attribute can be published, possibly with inconsistencies amongst the provided values, by several sources, e.g., Model, while other attributes, e.g., ReviewScore or Price, have values which are inherently source-specific.

In the following, we identify source objects by means of the URL of the detail page hosting its data, and we identify source

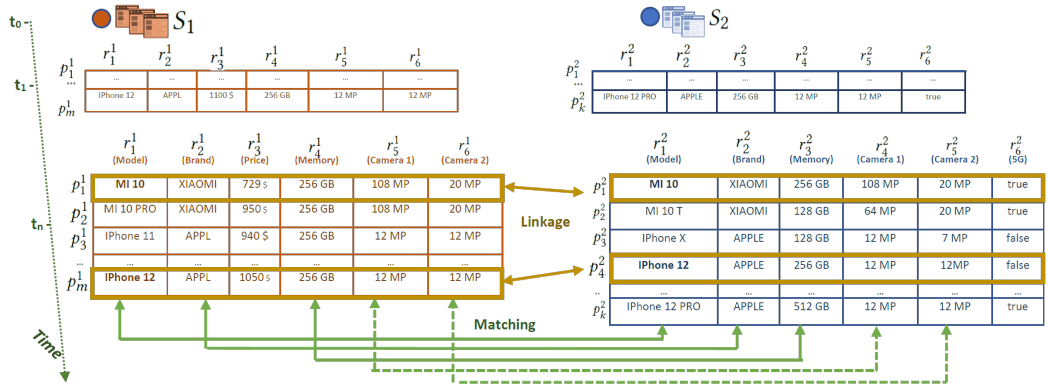


Figure 2: Running Example — The *Smartphones* domain includes 2 sources crawled at n instants. Over each source 6 correct extraction rules working on several detail pages are given: r^i_j ($j = 1, \ldots, 6$) denotes the j-th rule working on source S_i, each extracting the value of a source attribute from a detail page associated with a source object. For example, p^1_3 indicates the page about *IPhone 11* from source S_1 and rule r^1_2 extracts the *Model* from every page of the same source. At every time t, the values extracted from the two sources are conveniently depicted as organized in tables: each row of the table is associated with a detail page of the source, and each column is associated with an extraction rule around the same source. The set of domain attributes includes: *Model*, *Brand*, *Price*, *Memory*, *Camera 1*, *Camera 2*. Correct linkages can be represented as pairs of pages about the same domain objects: $\{(p^1_1, p^2_1), (p^1_m, p^2_4)\}$. Correct source attribute matches can be represented as pair of correct extraction rules: $\{(r^1_1, r^2_1), (r^1_2, r^2_2), (r^1_4, r^2_3), (r^1_5, r^2_4), (r^1_6, r^2_5)\}$.

attributes by means of a unique, within the domain, identifier of the extraction rule that is capable of locating its value from the detail page. By extraction rule we mean a function extracting at most one value from a detail page. It does not matter the formalism, e.g., XPath expressions, in which it will be specified.

Our goal is that of continuously extracting data of guaranteed level of *quality* from the detail pages composing to sources. The data are reorganized into an *Integrated Knowledge Graph* (IKB) while minimizing the overall *costs*. As a measure of data quality, we will use standard measures such as precision, recall, and F-measure over integrated data [23]. As a measure of the *cost*, the goal is that of minimizing the crowdsourcing costs [5, 27].

In IKB the following information will be available: (*linkages and matches*) how the source attributes and objects are respectively mapped to the domain attributes and objects; (*values provenance*) the source attribute values for every object in the domain.

The problem we want to solve is that of continuously creating $\mathcal{K}^t$, that is an IKB at every time t in which the snapshots of the detail pages from every source in a domain $\mathcal{D}$ are gathered. We illustrate the problem definition by means of a running example shown in Figure 2.

2 SCOPE, OPPORTUNITIES, CHALLENGES

Building and maintaining effective data processing pipelines over Web data is a challenging problem for several reasons. First, Web sources are autonomous and remote: they can unpredictably change and therefore break all the extraction rules created on previous versions of the same source to extract data. Second, the set up of an integration pipeline requires to solve many interrelated tasks, each of which has motivated flurry of research works, including: *sources discovery*, *data extraction*, *schema matching*, *record linkage*, *data fusion*, *data labeling*, and *data cleaning*. Each of these problems has been extensively studied over the last decades, with tens, if not hundreds in some cases, of well-recognized research works [6, 13, 19, 34, 39].

The focus of our research project covers the three problems that we believe are at the *core* of any Web data integration pipeline: *extraction*, *matching*, and *linkage*. It does not include, on one hand, the sources discovery problem, and the automatic synthesis of crawling programs; on the other hand, it does not include the data fusion problem.

Our solution can help several projects that need to set up and maintain over time Web data processing pipelines, but require a guaranteed quality of the pipelines' output data to be business meaningful.

Clearly, the amount of work outsourced to crowd workers to guarantee the quality level largely depends on the inherent characteristic of the domain: those containing static attributes that are largely redundant from source to source can dramatically simplify domain data detection, extraction and schema matching; an attribute working as a soft identifier across several sources can contribute significantly to reduce the cost of the record linkage task for a domain (i.e, books' ISBN).

Unfortunately, it turns out that many interesting domains (e.g., job postings, real estates, ...) do not exhibit such redundancy and the type of redundancy that the system has to exploit is at an *intensional level*, i.e., type and format of values, range of values, labels of extracted data. Generally speaking, separating domain data from other information become largely dependent on the context in which the attributes are proposed, and on the availability of human feedback to check the correctness of proposed hypotheses.

Redundancy as OpenIE Enabler

The redundancy plays a fundamental role in our system to keep the crowdsourcing costs at reasonable levels. Whenever redundancy of data across sources is properly detected and exploited, domain data can be discerned by other noisy or out-of-domain information. For example, WEIR [4] assumes that linkages between collection of pages from two sources are already known as part of the input, and then it exploits the redundancy of distinct and

independent sources that publish information about the same objects and attributes to automatically find correct extraction rules and schema matches.

NOAH aims at escalating to the largest possible extent the use of redundancy for extracting and integrating Web data as pioneered by WEIR. It will exploit at least the following forms for redundancy:

Intensional several sources publish the same domain attributes

Extensional several sources publish information about the same domain objects

Temporal a source publish data about the same domain objects and attributes over time

Intra-source a source can publish data about the same objects in pages of distinct type, e.g., a result page containing snippet of records with most relevant attributes plus link to detail pages containing all attributes [21]

At the same time, and with the help of human feedback, NOAH aims at overcoming WEIR's limitations by relaxing its rather strict underlying assumptions on the input domain: WEIR requires that enough intensional and extensional redundancy is available to discern all domain data from all other information.

WEIR and NOAH falls in the realm of the OpenIE approaches [3, 4, 16, 29, 33, 37]: unlike the ClosedIE approaches [18, 20, 25, 28] where the managed knowledge base does not grow in terms of subjects and predicates but only in terms of values, new schema information, e.g., new domain attributes, can be progressively discovered while populating the knowledge base with entities and values of schema already known.

There are two main differences between NOAH and other OpenIE [29] systems: first, we do not require a pre-populated Knowledge Base, as we start from an empty IKB and we populate it as new sources over the domain; second, we aim at *continuously* extracting and integrating data [11], as we believe that the temporal setting is important both for business reasons (many projects need continuous stream of data rather than snapshots), and for taking into the main problem definition the maintenance costs of the generated pipelines over time, costs that are largely neglected in many research proposals [29].

Despite many of the problems that need to be tackled to create our pipelines have already been extensively covered in the research literature, we believe that semi-automatizing the creation of Web data processing pipelines can be still considered a relevant problem [10].

We argue that if the costs and the guaranteed level of quality [17] are explicitly considered, many projects relying on data processing pipelines can be re-conducted into a much more controllable investment and validation process, and their overall feasibility can be significantly improved because many business projects are strongly and directly affected by the cost of creating and maintaining the underlying Web data processing pipelines.

Moreover, we believe that by posing to human and automatic responders the same type of queries, they become interchangeable enough to motivate the study of new deployment methodologies for Web data processing pipelines. The goal of such methodologies is to progressively lowering the crowdsourcing costs by means of machine-learning techniques while keeping under control the output quality level since the early stages of the deployed pipelines. Indeed, many development projects often experience unpredictable and erratic time-to-market (TTM) and return-on-investment (ROI) because, especially in the early stages, they adopted ML algorithms but lack the amount and quality of training data, and the validation, needed to guarantee the desired output quality.

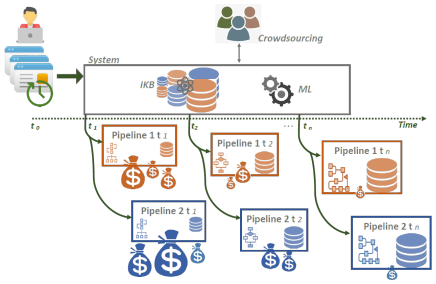


Figure 3: Overview of NOAH System & Pipelines created

3 NOAH SYSTEM AND PIPELINES

The NOAH system supports the semi-automatic generation of end-to-end Web data processing pipelines over several domains. Figure 3 shows how the system can generate and operate many pipelines at the same time, each having an IKB that is progressively and continuously populated with data coming from the sources of the domain on which it operates. Our system will interact with external systems by means of two major components: the *Crawler*, that continuously downloads snapshot of pages from every source with a frequency specified by a cron expression; and the *Crowd Manager*, that manages the interactions with a crowdsourcing platform.

During operations, NOAH will generate *pipeline queries* for the responders engaged by the crowdsourcing platform. The responders will contribute to solve the *system tasks* needed to set up and maintain new pipelines: for example, tasks are needed to select initial extraction rules over every domain source, select and label the source attributes, finding the linkages between source objects to a common mediated domain object, and matching the source attributes across several sources to a mediated domain attribute.

System Tasks

The main system tasks that need to be tackled to set up a NOAH pipeline are shown in Figure 4: *Page Linkage*, *Data Extraction*, *Schema Matching*, and *Object Linkage*.

Page Linkage aims at obtaining a first approximate top-k page linkages. Two pages have a linkage if they both publish data related to the same domain object.

Example 3.1 (Page Linkage). In Figure 2 we can see two possible page linkages at time t_n: $\{(p_1^1, p_1^2), (p_m^1, p_4^2)\}$. Their distances, i.e., 0.09 and 0.12, are shown at the top of Figure 5a.

Data Extraction aims at finding all the correct extraction rules. It generates all the possible extraction rules and discover the correct ones by exploiting the redundancy of published data across several independent sources [4] when available, while querying the responders [7] to confirm uncertain hypotheses.

Schema Matching aims at finding matches between extraction rules by exploiting an instance-based distance measure between source objects. The instance-based distance between two extraction rules assumes the availability of correct object linkages to align source objects related to the same domain object, as produced in output by the next system task: the distance is obtained

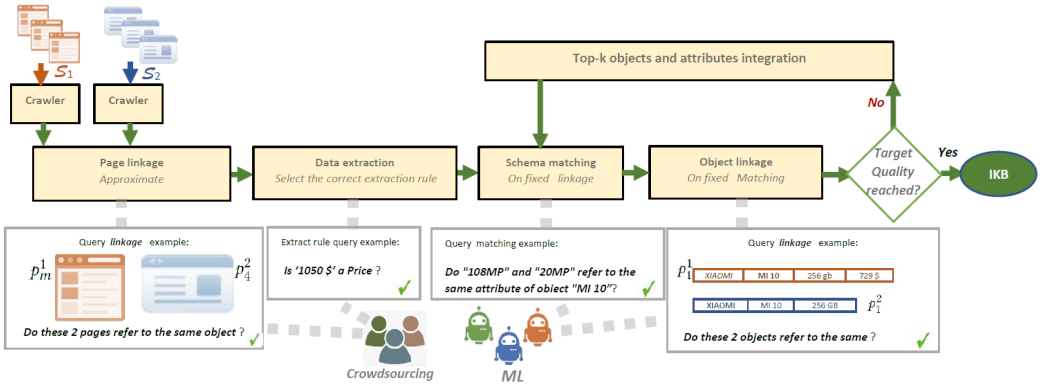


Figure 4: Running example (Pipeline example with queries): tasks provided by system and query generated for hybrid human-machine responders.

by averaging the distance between extracted values over all the aligned detail pages.

Example 3.2 (Schema Matching). Consider source S_1 and S_2 at time t_n and the set of page linkages $\{(p_1^1, p_1^2), (p_m^1, p_4^2)\}$ in Figure 2: possible matches are $\{(r_1^1, r_1^2), (r_4^1, r_3^2), (r_4^1, r_3^2)\}$. The pairwise attribute distances, i.e., 0.19 and 0.22, are shown at the top of Figure 5c.

Object Linkage aims at finding linkages between source objects by exploiting a pairwise attribute distance measure between source attributes. The pairwise attribute distance between two source objects assumes the availability of correct schema matches across the extraction rules to align source attributes related to the same domain attribute, as produced in output by the previous system task: the distance is obtained by averaging the distance between the two values over all matching attributes.

We name the linkage/matching loop of system tasks *Linkage/Matching Duality*; we further discuss it in Section 3.1.

Pipeline Queries

For every system task necessary to set up and maintain a pipeline, Noah tries to solve it by using a *human-in-the-loop* approach [9, 26]: unsupervised algorithms will generate most-likely hypothesis based on the available redundancy. These hypothesis are later confuted or validated by means of queries posed to responders, initially only human responders, and later, also by using automatic responders based on ML models that have been trained with the data collected while operating the Noah pipeline (see Section 4).

An example of the queries posed to the responders for every system task is shown in Figure 4: *Page Linkage*, *Data Extraction*, *Schema Matching* and *Object Linkage*.

Example 3.3 (Data Extraction Query). Figure 4 shows an example of query for Data Extraction tasks. The uncertainty of an extraction rule generated by wrapper inference can be validated by checking the extracted value on a detail page by means of a query such as: *"Is '1050$' a Price?"*, where Price is a candidate label for the extraction rule and *'1050 $'* is the extracted value.

Example 3.4 (Schema Matching Query). Figure 4 shows that schema matching tasks can be solved by means of queries confirming or refuting a single match: 'Do "108MP" and "20MP" refer to the same attribute of object "MI 10"?'. The template of the query to support a schema matching task has been filled up with values extracted from two pages of distinct sources, e.g., using extraction rules (r_5^1, r_4^2). These are two detail pages considered in a linkage, and "MI 10" is the name associated with the corresponding domain object.

Example 3.5 (Page Linkage Query). A query such as *'Do these two pages refer to the same object?'* posed to human responders in Figure 4 can validate or refute a page linkage (p_m^1, p_4^2). In order for the query to be as simple as possible [35], we can show the user a screenshot of the original pages.

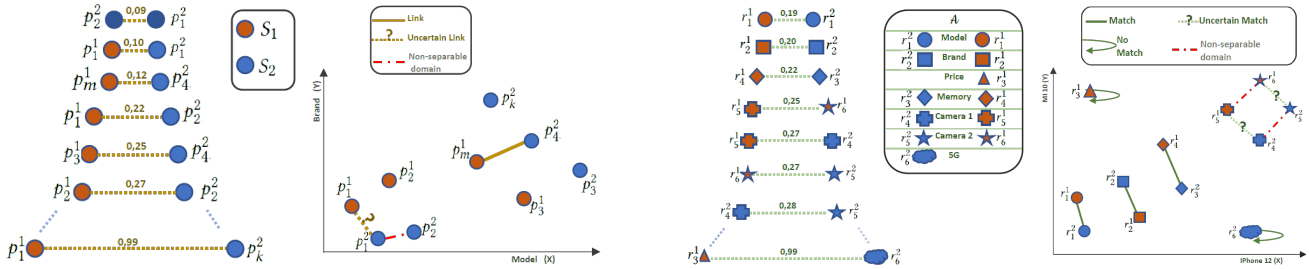
Example 3.6 (Object Linkage Query). Unlike the case of page linkage tasks above, here the query is posed directly on source objects with extracted values. A query such as *'Do these 2 objects refer to the same?'* posed to human responders in Figure 4 can validate or refute an object linkage (p_1^1, p_1^2). To make the query as simple as possible for an human responder, it is shown together with two records whose attributes have been already aligned by leveraging the results of a schema matching task.

The tremendous success of crowdsourcing [24] can be partially explained by saying that human supervision can represent the essential final ingredient to unmask those problems really hard to solve through automatic algorithms but that can be transformed into rather simple questions for human workers. However, it is well known that in practice, the availability and the accuracy of crowd workers, especially of unskilled ones, is strongly dependent on the way the questions are posed and rewarded [35]. One of the Noah goal is that of exploiting IKB, which is progressively built, also to make the crowdsourcing queries as simple as possible. For example, a query to check a record linkage exploits the schema matching already computed to make the two records easy to be visually compared.

3.1 Linkage / Matching Duality

Figure 4 shows that two important integration tasks operated by Noah pipelines, i.e., *Schema Matching* and *Object Linkage*, are part of a loop in which each one assumes the availability of the output of the other to solve its own task. *Page Linkage* is the system task outside the loop needed for its initial triggering.

We assume available two normalized distance functions providing a value between 0 and 1 when comparing two rules, and two source objects (records), respectively: the *instance-based distance* and the *pairwise attribute distance*. The former compares two rules over the values they extract from a set of detail pages which have been previously aligned, i.e., their linkages are fixed.



(a) Linkages Distances (b) Page Linkages over 2 Attributes (c) Matches Distances (d) Matches over 2 Linkages

Figure 5: Running example (Distance Similarity): 5a and 5c show distances in Pyramids; 5b and 5d expose relations in Cartesian Plane where 'Uncertainties' are due to the breaking of LC with Non-separable Domain

The latter compares two source objects over the values of some of their attributes which have been previously aligned, i.e., their matches are fixed.

Example 3.7 (Normalized Distance Functions). Instance-based distance: let (p_m^1, p_4^2) and (p_1^1, p_1^2) be two given correct linkages for the detail pages associated with *IPhone 12* and *MI 10* source object from source S_1 and S_2 as shown in Figure 5d. The distance between the rules (r_5^1, r_4^2) can be computed as follows: $d(r_5^1, r_4^2) = d(r_5^1(p_1^1), r_4^2(p_1^2)) + d(r_5^1(p_m^1), r_4^2(p_4^2)) = d(\text{'108MP'}, \text{'108MP'}) + d(\text{'12MP'}, \text{'14MP'}) = 2.9$. The normalized distance in the range $[0, 1]$ is 0.27.

Pairwise attribute distance: let (r_2^2, r_2^1) and (r_1^1, r_1^2) be two given correct matches for *Brand* and *Model* attributes (see Figure 5b). The distance between the two source objects about *MI 10 PRO* and *MI 10T* can be computed as follows: $d(o_2^1, o_2^2) = d(r_2^1(p_2^1), r_2^2(p_2^2)) + d(r_1^1(p_2^1), r_1^2(p_2^2)) = d(\text{'XIAOMI'}, \text{'XIAOMI'}) + d(\text{'MI 10 PRO'}, \text{'MI 10T'}) = 3.2$. The normalized distance in the range $[0, 1]$ is 0.27.

We revisit and propose an extension of two domain properties, called *Local Consistency* and *Separable Domain*, underlying the formal approach presented in WEIR [4] for solving the extraction and matching problem when the page linkage is given as input.

Our ambition is twofold: on the one side, we aim to extend that approach to cover the whole trio of extraction, matching and linkage problems at the core of Noah pipelines; on the other hand, we want to relax the underlying assumptions by mean of the feedback provided by human crowd workers, so making the approach adaptable to domains with more disparate characteristics that those originally covered in the WEIR project. Here we briefly recall the two properties and sketch how we plan to extend them.

Local Consistency (LC) In a source there cannot be two distinct source attributes that refer to the same domain attribute. The dual property that we additionally assume is that two distinct detail pages from the same source cannot publish data about the same domain object.

Separable Domain (SD) In a mapping composed of several extraction rules, each from a distinct source, and associated with the same domain attribute, the instance-based distances between the rules of the mapping are always smaller than distances with rules associated with a different domain attribute. For computing the instance-based distance, the object linkages are fixed and already known. The dual property that we additionally assume is that in a linkage composed of several source objects from distinct sources and related to the same domain object, the pairwise attribute distances are always smaller than distances with source objects associated with a different domain object. For computing the pairwise attribute distance, the source attribute matches are fixed and already known.

For domains in which such properties hold, the WEIR system is able to match the extraction rules and build their mappings into cluster of source attributes related to the same domain attribute by comparing all the similarity distances, while at the same time, it can separate the correct extraction rules from noisy ones. The idea is pretty simple and depicted in Figure 5: DS suggests to sort the set of all possible matches (pair of extraction rules) by an instance-based distance leveraging the alignment of detail pages (see Figure 5c). Those pairs are then processed in order of increasing distances: every pair of rules are merged in the same mapping as long as the addition of the rules will not lead to a violation of the LC property, i.e., two rules (source attributes) from the same source would end up being present in the same output mapping (see Figure 5d). For certain domains, with sufficiently overlapping sources, WEIR can automatically find the correct extraction rules and their matching with rules over other sources provided that the correct linkages between detail pages are known.

The dual algorithm will solve the problem of finding correct object linkages provided that correct schema matches between source attributes are given as depicted in Figure 5: DS suggests to sort the set of all possible linkages (pair of source objects) by a pairwise attribute distance (see Figure 5a). Those pairs are then processed in order of increasing distances: every pair of source objects are merged in the same linkage as long as the addition of the objects processed into an existing linkage will not lead to a violation of the LC property, i.e., two source objects from the same source would end up being present in the same output linkage (see Figure 5b).

This algorithm exploits the duality of the matching and linkage problems, in this setting, and it is at the core of integration engine for the Noah project. However, differently from WEIR, it does not halt the integration as soon as a LC violation is detected: rather, it generates pipeline queries to confirm the choice, and continue the processing of all pairs in increasing order of distances, until it is below a threshold over which no further matches/linkages are expected with meaningful distance functions.

Unfortunately, as also recognized in WEIR [4], some domains have sources and attributes with very similar but semantically different values (e.g., the resolution of the front/rear cameras in Figure 2). This situation easily lead to violation of the LC and SD assumptions, and finding the mappings is a challenging problem for many interesting domains.

Example 3.8 (Non-separable Domains for Schema Matching). In Figure 2, source S_1 and S_2 both have extraction rules ((r_5^1, r_6^1), and (r_4^2, r_5^2), respectively) with a low distance (Figure 5c) because camera resolutions (e.g., 1-front and 2-back) are typically within a small range of values expressed in megapixel (MP). In Figure 5d it is shown that the pair of rules (r_5^1, r_6^1) at distance 0.25 violates the LC and DS assumptions because their distance is smaller than the distance of (r_5^1, r_4^2) that is 0.27.

Actually, it is well known that the Record Linkage dual problem, is even much more challenging than the Schema Matching itself: the attributes containing the correct signals for considering two objects equivalent can change from object to object even within the same source (think at smartphones of different brands with different policies for naming the models and differentiating the features of each model). Assuming that every object in the domain does not lead to a separability violation is quite unrealistic, beside toy cases.

Example 3.9 (Non-separable Domains for Object Linkage). In Figure 5b the linkage (p_1^1, p_1^2) is uncertain due to the presence of p_2^2. The two values ('MI 10' vs 'MI 10T') extracted by rule r_1^2 from pages p_1^2 and p_2^2 differ by a single letter: the wrong linkage (p_1^2, p_2^2) violating the LC property has a pairwise attribute distance of only $0,09$ which is smaller than the distance of a the correct linkage (p_1^1, p_1^2), and therefore the domain is not separable.

We believe that the violations of LC and DS assumptions can be manually fixed and that they help to find the most informative pipeline queries that need to be posed to external responders, i.e., paid crowd workers, or suitably trained automatic responders.

By interleaving the dual linkage/matching algorithms in a loop in which external responders can contribute, as shown in Figure 4, each execution can contribute to improve the accuracy of the distance function used by the other task, either by improving the linkages used by the instance-based distance, or improving the matches used by the pairwise attribute distance.

Our vision is that with the precious help of crowdsourcing and a loop of interleaving linkage/matching operations, the desired target quality can be reached even in presence of non-separable domains: responders will be engaged to assess the quality of the output, and to repair the uncertain choices made by the integration algorithm. The linkages and matches confirmed by human feedback can be frozen and exploited in the following iterations, somehow progressively solving and hence removing from the domain the linkages or matches that made the domain inseparable.

4 RESEARCH DIRECTIONS

In the early stages of its life, the IKB $\mathcal{K}$ of a new Noah pipeline might be scarcely populated. As redundancy builds up over time with the addition of new sources to feed up the IKB, the accuracy of the extraction and integration process increases.

The absence of overlapping between objects and attributes published by a rather limited set of sources could limit the amount of available redundancy. In this situation, for operating the pipeline, Noah would end up generating a lot of queries supporting the system tasks. As an alternative solution, Noah supports the incremental addition of a source into an existing pipeline. A new source might contribute to lower the overall costs if it significantly overlaps with the sources already available for the domain [14]. On the contrary, to integrate new sources publishing new objects or new attributes, additional costs might be incurred to support the integration with existing IKB.

We are interested to study ML techniques that could decrease crowdsourcing costs even in absence of redundancy. The main research area is that of synthesizing automatic responders capable of answering the same type of pipeline queries that are normally posed to human responders for solving Noah tasks, with the goal of progressively replacing human responders [7] and scaling the approach up to many thousands of sources.

Unfortunately, state-of-the-art ML unsupervised techniques [40, 42] can be adapted to provide accurate and reliable answers to those queries only if enough training data have been collected. Indeed, fairness and bias, or simply misuse of machine learning algorithms, is a well-known problem in literature [12, 32] that affects many development projects, especially in the scenarios which are most commonly found in practice [38]: pre-trained ML models and/or enough training data are not available up-front, so that the ML models cannot be properly tuned and exhibit erratic and unpredictable performance [41].

Snorkel [36] is another project exploiting the idea of leveraging human work to train ML algorithms. However, it is based on the idea of engaging skilled workers in every step of the processing pipeline, while Noah aims at engaging non skilled workers to whom can be interchangeably posed queries in the same form as those posed to automatic responders. Several other projects such as qodco [2] and SEER [22] have made use of crowdsourcing by mainly focusing on the problem of selecting the correct extraction rules, while Noah applies the same query control methodology for all the tasks in the considered pipelines.

It is also well known that by using automatic responders not accurate enough, it might turn out to be more expensive engaging them than not using them at all, as additional human workers should be engaged only to offset their wrong answers [7].

We envision a system in which crowd workers are used for indirectly controlling the deployment of automatic responders, and the two types of responders are interchangeably engaged. Crowdsourcing workers contribute to collect domain data that are then used to train and evaluate automatic responders, before fully deploying them. Automatic responders will progressively replace crowd workers to scale the approach and to lower the operating costs, but only after enough evidence that their accuracy does not compromise the overall guaranteed output quality data. At regime, crowd workers will be minimally used only to keep monitoring the performance of automatic responders.

We have identified several novel research challenges:

- formalizing and proving the correctness of an algorithm that solves the full trio of extraction, matching and linkage tasks;
- creating and maintaining over time the continuous Web data processing pipelines at low costs, with guaranteed output quality;
- designing several independent automatic responders based on ML models that are capable of answering queries normally posed to crowd workers;
- effectively measuring the available redundancy in a domain;
- estimating from the characteristics of a domain the crowdsourcing costs necessary to obtain and maintain the desired output quality.

REFERENCES

[1] Tara S Behrend, David J Sharek, Adam W Meade, and Eric N Wiebe. 2011. The viability of crowdsourcing for survey research. *Behavior research methods* 43, 3 (2011), 800.

[2] Moria Bergman, Tova Milo, Slava Novgorodov, and Wang-Chiew Tan. 2015. Query-oriented data cleaning with oracles. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. 1199–1214.

[3] Nikita Bhutani, Yoshihiko Suhara, Wang-Chiew Tan, Alon Halevy, and Hosagrahar Visvesvaraya Jagadish. 2019. Open information extraction from question-answer pairs. *arXiv preprint arXiv:1903.00172* (2019).

[4] Mirko Bronzi, Valter Crescenzi, Paolo Merialdo, and Paolo Papotti. 2013. Extraction and integration of partially overlapping web sources. *Proceedings of the VLDB Endowment* 6, 10 (2013), 805–816.

[5] Valter Crescenzi, Alvaro AA Fernandes, Paolo Merialdo, and Norman W Paton. 2017. Crowdsourcing for data management. *Knowledge and Information Systems* 53, 1 (2017), 1–41.

[6] Valter Crescenzi, Giansalvatore Mecca, and Paolo Merialdo. 2002. RoadRunner: automatic data extraction from data-intensive web sites. In *Proceedings of the 2002 ACM SIGMOD international conference on Management of data*. 624–624.

[7] Valter Crescenzi, Paolo Merialdo, and Disheng Qiu. 2019. Hybrid Crowd-Machine Wrapper Inference. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 13, 5 (2019), 1–43.

[8] Nilesh Dalvi, Ashwin Machanavajjhala, and Bo Pang. 2012. An analysis of structured data on the web. *arXiv preprint arXiv:1203.6406* (2012).

[9] AnHai Doan. 2018. Human-in-the-Loop Data Analysis: A Personal Perspective. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics (HILDA'18)*. Association for Computing Machinery, New York, NY, USA, Article 1, 6 pages. https://doi.org/10.1145/3209900.3209913

[10] AnHai Doan, Adel Ardalan, Jeffrey R. Ballard, Sanjib Das, Yash Govind, Pradap Konda, Han Li, Erik Paulson, Paul Suganthan G. C., and Haojun Zhang. 2017. Toward a System Building Agenda for Data Integration. arXiv:cs.DB/1710.00027

[11] AnHai Doan, Alon Halevy, and Zachary Ives. 2012. *Principles of data integration*. Elsevier.

[12] Pedro Domingos. 2012. A Few Useful Things to Know about Machine Learning. *Commun. ACM* 55, 10 (Oct. 2012), 78–87. https://doi.org/10.1145/2347736.2347755

[13] Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. 2014. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 601–610.

[14] Xin Dong, Barna Saha, and Divesh Srivastava. 2012. Less is more: Selecting sources wisely for integration. *Proceedings of the VLDB Endowment* 6, 37–48.

[15] Xin Luna Dong and Divesh Srivastava. 2015. Big data integration. *Synthesis Lectures on Data Management* 7, 1 (2015), 1–198.

[16] Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the 2011 conference on empirical methods in natural language processing*. 1535–1545.

[17] Wenfei Fan and Floris Geerts. 2012. Foundations of data quality management. *Synthesis Lectures on Data Management* 4, 5 (2012), 1–217.

[18] Tim Furche, Georg Gottlob, Giovanni Grasso, Xiaonan Guo, Giorgio Orsi, Christian Schallhart, and Cheng Wang. 2014. DIADEM: thousands of websites to a single database. *Proceedings of the VLDB Endowment* 7, 14 (2014), 1845–1856.

[19] Chaitanya Gokhale, Sanjib Das, AnHai Doan, Jeffrey F Naughton, Narasimhan Rampalli, Jude Shavlik, and Xiaojin Zhu. 2014. Corleone: hands-off crowdsourcing for entity matching. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*. 601–612.

[20] Pankaj Gulhane, Amit Madaan, Rupesh Mehta, Jeyashankher Ramamirtham, Rajeev Rastogi, Sandeep Satpal, Srinivasan H Sengamedu, Ashwin Tengli, and Charu Tiwari. 2011. Web-scale information extraction with vertex. In *2011 IEEE 27th International Conference on Data Engineering*. IEEE, 1209–1220.

[21] Jinsong Guo, Valter Crescenzi, Tim Furche, Giovanni Grasso, and Georg Gottlob. 2019. RED: Redundancy-Driven Data Extraction from Result Pages?. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, Ling Liu, Ryen W. White, Amin Mantrach, Fabrizio Silvestri, Julian J. McAuley, Ricardo Baeza-Yates, and Leila Zia (Eds.). ACM, 605–615. https://doi.org/10.1145/3308558.3313529

[22] Maeda F Hanafi, Azza Abouzied, Laura Chiticariu, and Yunyao Li. 2017. Synthesizing extraction rules from user examples with seer. In *Proceedings of the 2017 ACM International Conference on Management of Data*. 1687–1690.

[23] Bernd Heinrich, Marcus Kaiser, and Mathias Klier. 2007. How to measure data quality? A metric-based approach. (2007).

[24] Jeff Howe. 2006. The rise of crowdsourcing. *Wired magazine* 14, 6 (2006), 1–4.

[25] Nicholas Kushmerick, Daniel S Weld, and Robert Doorenbos. 1997. *Wrapper induction for information extraction*. University of Washington Washington.

[26] Guoliang Li. 2017. Human-in-the-Loop Data Integration. *Proc. VLDB Endow.* 10, 12 (Aug. 2017), 2006–2017. https://doi.org/10.14778/3137765.3137833

[27] Guoliang Li, Yudian Zheng, Ju Fan, Jiannan Wang, and Reynold Cheng. 2017. Crowdsourced Data Management: Overview and Challenges. In *Proceedings of the 2017 ACM International Conference on Management of Data (SIGMOD '17)*. Association for Computing Machinery, New York, NY, USA, 1711–1716. https://doi.org/10.1145/3035918.3054776

[28] Colin Lockard, Xin Luna Dong, Arash Einolghozati, and Prashant Shiralkar. 2018. Ceres: Distantly supervised relation extraction from the semi-structured web. *arXiv preprint arXiv:1804.04635* (2018).

[29] Colin Lockard, Prashant Shiralkar, and Xin Luna Dong. 2019. Openceres: When open information extraction meets the semi-structured web. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 3047–3056.

[30] Adam Marcus and Aditya Parameswaran. 2015. Crowdsourced data management: Industry and academic perspectives. *Foundations and Trends in Databases* 6, 1-2 (2015), 1–161.

[31] Mausam Mausam. 2016. Open information extraction systems and downstream applications. In *Proceedings of the twenty-fifth international joint conference on artificial intelligence*. 4074–4077.

[32] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2019. A survey on bias and fairness in machine learning. *arXiv preprint arXiv:1908.09635* (2019).

[33] Christina Niklaus, Matthias Cetto, André Freitas, and Siegfried Handschuh. 2018. A survey on open information extraction. *arXiv preprint arXiv:1806.05599* (2018).

[34] Erhard Rahm and Philip Bernstein. 2001. A Survey of Approaches to Automatic Schema Matching. *VLDB J.* 10 (12 2001), 334–350. https://doi.org/10.1007/s007780100057

[35] Bahareh Rahmanian and Joseph G. Davis. 2014. User Interface Design for Crowdsourcing Systems. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces (AVI '14)*. Association for Computing Machinery, New York, NY, USA, 405–408. https://doi.org/10.1145/2598153.2602248

[36] Alexander J Ratner, Stephen H Bach, Henry R Ehrenberg, and Chris Ré. 2017. Snorkel: Fast training set generation for information extraction. In *Proceedings of the 2017 ACM international conference on management of data*. 1683–1686.

[37] Michael Schmitz, Stephen Soderland, Robert Bart, Oren Etzioni, et al. 2012. Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. 523–534.

[38] Zeyuan Shang, Emanuel Zgraggen, Benedetto Buratti, Ferdinand Kossmann, Philipp Eichmann, Yeounoh Chung, Carsten Binnig, Eli Upfal, and Tim Kraska. 2019. Democratizing Data Science through Interactive Curation of ML Pipelines. In *Proceedings of the 2019 International Conference on Management of Data (SIGMOD '19)*. Association for Computing Machinery, New York, NY, USA, 1171–1188. https://doi.org/10.1145/3299869.3319863

[39] Kai-Sheng Teong, Lay-Ki Soon, and Tin Tin Su. 2020. Schema-Agnostic Entity Matching using Pre-trained Language Models. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2241–2244.

[40] Sebastian Thrun and Lorien Pratt. 2012. *Learning to learn*. Springer Science & Business Media.

[41] Doris Xin, Litian Ma, Jialin Liu, Stephen Macke, Shuchen Song, and Aditya Parameswaran. 2018. Accelerating Human-in-the-Loop Machine Learning: Challenges and Opportunities. In *Proceedings of the Second Workshop on Data Management for End-To-End Machine Learning (DEEM'18)*. Association for Computing Machinery, New York, NY, USA, Article 9, 4 pages. https://doi.org/10.1145/3209889.3209897

[42] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)* 10, 2 (2019), 1–19.