# A Platform for Storing, Visualizing, and Interpreting Collections of Noisy Documents

Bart Lamiroy, Daniel Lopresti

# A Platform for Storing, Visualizing, and Interpreting Collections of Noisy Documents*

Bart Lamiroy†
Nancy Université – LORIA
Campus Scientifique, BP 239
54506 Vandoeuvre Cedex, France
Bart.Lamiroy@loria.fr

Daniel Lopresti
Computer Science and Engineering
Lehigh University
Bethlehem, PA 18015, USA
lopresti@cse.lehigh.edu

## Abstract

The goal of document image analysis is to produce interpretations that match those of a fluent and knowledgeable human when viewing the same input. Because computer vision techniques are not perfect, the text that results when processing scanned pages is frequently noisy. Building on previous work, we propose a new paradigm for handling the inevitable incomplete, partial, erroneous, or slightly orthogonal interpretations that commonly arise in document datasets. Starting from the observation that interpretations are dependent on application context or user viewpoint, we describe a platform now under development that is capable of managing multiple interpretations for a document and offers an unprecedented level of interaction so that users can freely build upon, extend, or correct existing interpretations. In this way, the system supports the creation of a continuously expanding and improving document analysis repository which can be used to support research in the field.

## 1 Introduction

The goal of document image analysis is to achieve *performance* using automated tools that is *comparable* to what a *careful* human expert would achieve, or at least to do *better* than *existing algorithms* on the same *task*.

Our use of terms like "performance," "comparable," and "better" indicate that there is an underlying notion of *quality* and therefore *measurement*. It suggests a controlled process that continually improves toward perfection. However, we also make mention of "careful" humans, "tasks," and "existing algorithms." While humans may believe themselves to be expert and careful when performing a task, there are situations where they unavoidably disagree [7, 17, 22, 2], meaning that, at best, quality and improvement are subjective notions. It also strongly suggests that, depending on the task, measurements will differ, advocating again for multiple ways of measuring overall performance.

On the other hand, shared reference benchmarks are essential in scientific domains where



†Bart Lamiroy is a Visiting Scientist in the Department of Computer Science and Engineering at Lehigh University, on an INRIA délégation with the Unité de Recherche Nancy – Grand Est.

reproducible experiments are vital to the peer review process. For instance, there have been numerous attempts to produce common datasets for problems which arise in document analysis [14, 25, 24]. It is important to note, however, that shared datasets are only a part of what is needed for performance evaluation, and since research in document analysis is often task-driven, specific interpretations of a dataset may exist. So whether the problem is invoice routing, building the semantic desktop, digital libraries, global intelligence, or document authentication, to name a few, the result tends to be application-specific, resulting in software solutions that integrate a complete pipeline of cascading methods and algorithms [14, 23]. This most certainly does not affect the intrinsic quality of the underlying research, but it does tend to generate isolated clusters of very focused problem definitions and experimental requirements. Crossing boundaries and agreeing on what kinds of tools, formats or measurements are the most useful is difficult and may, in fact, be impossible since the pursuit of goals may be prove orthogonal between domains.

In this paper we put forth a rather radical point of view: quality measurement of both human and automated document interpretations, ground-truths, and therefore performance measurements are so context dependent that it doesn't always make sense to consider them in an absolute reference frame where true and false would be universally agreed upon for a particular document and its interpretation. Instead, we are presenting a paradigm in which multiple interpretations and measurements coexist, and where measuring, comparing and interpreting require the presence of a well defined context. Taking this into account is a very different way of considering document analysis research and opens up a wide range of possible new research topics, provided the framework and tools for doing so are available. Section 3 describes such a platform, as it is currently under development as part of the DAE project at Lehigh University [11], and details its means of representing, comparing, and correcting data and interpretations in Section 4. Section 5 concludes with a discussion of open questions and ongoing work. Before that, the next section develops the different document models, abstractions and interpretations that need to be considered in order to make the rest of our work possible.

# 2 Contents, Abstractions and Interpretations

## 2.1 Vocabulary

In the introduction we mention "interpretations", "performance", "ground-truth" and other terms referring to what could be considered to be "true" or "false" in a context of document interpretation. Precisely defining all these terms is not very helpful and it would make this document unnecessarily verbose and long. However, in order to understand our work, and to capture the semantics of the used vocabulary, it is necessary to view these terms, and document interpretation in general, in the light of [5]. *Documents* are physical supports[1] that were created by an *author* to convey a *message* to a *reader*.

**Documents** exist and are unambiguous. They are mere physical entities and have an undisputed content value (pixel values on scanned documents, tags and fields in HTML documents, sampling and impulse values for audio recordings, ink molecules on a velum ...). These content values may

[1] One can argue about the term *physical support*. In our framework it might be a recorded audio message, as well as a twelfth century handwritten codex or a complex HTML or PDF document.

or may not be the result of the author's intent.

**The author's message** is embedded in the document and is a complex mix of syntactic representations, cultural context presuppositions, etc. The transcription of the message to the document is a noisy and imperfect process, and it can be generally assumed [5] that it is not reversible without meta knowledge.

**The reader's interpretation** is an attempt to retrieve whole or part of the author's message, based on the physical content of the document and a set of contextual assumptions[2].

This way of perceiving document interpretation sheds a new light on how to consider noisy documents since it not only covers the noise that affects the physical transcription process of the author's syntax on the document support, but it also covers the lack of contextual knowledge that affects the interpretation by the reader.

## 2.2 Comparing Interpretations

The definitions in the previous section insist that both the author and the reader operate in their own contextual frame. There is no guarantee that either of them, on the one side, or that two different readers, on the other side, share the same context. With this postulate, trying to determine which one of two interpretations is "better" becomes difficult.

Notwithstanding, it seems essential that in the context of experimental noisy document analysis, methods and algorithms are compared in order to evaluate scientific contributions. This is the reason for collections of evaluation documents to be annotated down to a fine level with the so-called "ground-truth" (*e.g.* the location and identity of every character represented in the document, in some cases, or even richer annotations, like the type size and typeface for each character in other cases). It would be a mistake to consider this ground-truth to be an absolute fact. Given the paradigm of the previous section, it is just an instance of one readers' interpretation. It is certainly not unique and may not cover the author's whole intent. It is merely a reflection of a specific reader's interpretation context.

Existing tools allow the user to indicate how he/she believes a document should be interpreted, but do little to help users understand differences in interpretations. Such differences might be called "errors" when there is a strong consensus about what constitutes the right answer. In many cases, however, there are legitimate differences of opinion [8, 15] by various readers of the document, and these may differ from the intention of the author (which is usually hard or impossible to determine, although sometimes we can get access to it [5]).

So, although standard document collections exist, their annotations or ground truth may be specific, recorded in pre-determined representations, incomplete or partially flawed for more generic contexts, while, on the other hand, there is a need to collect and manage annotations in ways that make it possible to construct more robust and general document analysis solutions (and therefore encompass broader contexts).

In the next sections of this paper, we seek to explore methodologies for storing, visualizing, and interpreting document collections that acknowledge ambiguity and integrate the fact that multiple interpretations are unavoidable. Our approach exhibits the following principles:

[2]There is no need for the user interpretation to actually be a tentative to retrieve part of the author's message. Interpretation can also consist in trying to recover part of the contextual assumptions of the author, or to try and retrieve information concerning the physical transcription and/or capture process.

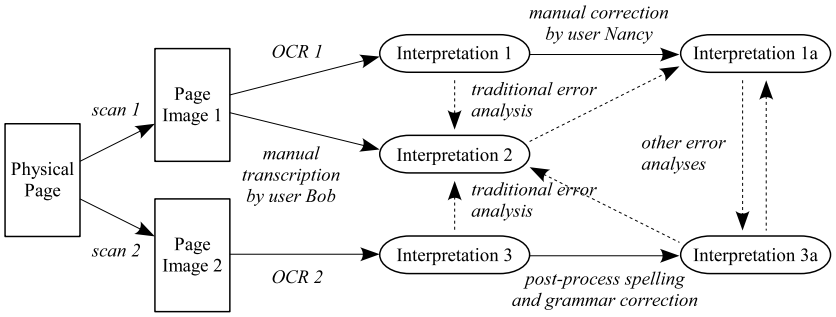


Figure 1: Alternate interpretations and relative error analyses (from [16]).

- allow that an interpretation for an entity on a page may be incompatible with a given context and not with another, assuming there may be more than one acceptable interpretation for a particular entity;
- support the interleaving of machine (automatic) and human (manual) interpretation steps that are intended to improve the quality of the document representation over time;
- facilitate the development of more accurate recognition algorithms by retaining and exploiting all of the user's interactions with the collection;
- help the collection as a whole to evolve to higher and higher levels of quality over time.

The next section briefly summarizes the features we believe should be present in a comprehensive system, drawing from the discussion in our earlier paper [16]. The concept of interpretation, which we defined previously as reader specific, plays a central role. An interpretation reflects the opinion of a reader of the document and, since opinions can vary, there may be no unique "correct" interpretation.[3]

[3]It should be understood that interpretations are not limited to simple transcriptions of text appearing on the page, but even in this case there could be differences of opinion.

### 2.3 Document Models

In order to support comparison of interpretations, there needs to be a set of basic entities in which to express document contents. These basic entities are generally agreed upon and consist of a hierarchy of regions that are labeled as pages, zones, text lines, words, and characters (see, *e.g.*, TRUEVIZ [13] and DOCLIB [9]). In other cases, entities may also be graphical components expressed with a visual vocabulary (see, *e.g.*, the QGAR toolkit [19], or [10] for more elaborate graphical document descriptions). Other relationships between entities should also be supported, including containment and reading order. Moreover, while sharing of document models is important for comparison of interpretations, they should not restrict users in their ways of interpreting documents. Users should be allowed to create their own label types and relationships without restriction to address the needs of specific applications.

Obviously, the successive pages that comprise a single document should be linked in sequential order. Some categories of documents naturally contain cross-page references (including this paper). In addition, it may be that different versions of a document are present in a

corpus, or multiple copies of the same version. Two photocopies of a page may lead to nearly identical images in the dataset, or they may differ in substantive ways (*e.g.*, one may contain handwritten annotations added after the original printing).

### 2.4 Alternative Interpretations

Interpretations for a page are created either by humans or by algorithms which have been designed to mimic human behavior on a certain class of inputs. In our model, it is natural to assume that a given page will have received multiple interpretations over time, originating from varying sources, and relating to varying contexts. Interpretations can be created from scratch (working directly from the page image), or they can build on previous interpretations (attempting to correct perceived errors). As suggested earlier in our discussion of document models, we can generally assume that information will be recorded at the character-level, word-level, sentence-level, paragraph-level, page-level, and document-level. The ability to compare interpretations is critical when it comes to quantifying how well an algorithm has done (*i.e.*, how closely it matches human performance). Figure 1 illustrates a range of possibilities.

We have suggested there may be no such thing as a "correct" interpretation, but when a page has multiple interpretations, which is the preferred one for comparison purposes? Here the notion of on-line *reputation* as practiced in Web 2.0 recommender systems may hold the key [18, 26, 20]. Researchers and algorithms already have informal reputations within the community. Extending this to the document interpretation data can provide a mechanism for deciding which annotations to trust.

The presence of differing interpretations provides a basis for defining what "noise" means in the context of document image analysis. We might take input noise to be any artifact that prevents a fluent reader – whether human or machine – from arriving at the interpretation of a document the author intended.[4] Noise in the input manifests itself as noise in the output. It is often unrealistic, however, to expect that we can recover the author's original intent. Hence, we prefer a more practical definition of noise as being the relative difference between the interpretations of two or more readers. If human readers arrive at similar interpretations which a particular computer algorithm cannot match, then we can conclude that there is noise in the output of the algorithm, but we should be careful about assuming there is noise in the input since it could be that we are simply dealing with a bad algorithm. If no computer algorithm can produce an interpretation similar to that of the humans, then we can say that the input document itself is noisy and that the problem is hard. An examination of this empirical approach to defining noise is one of the investigations made possible by the server architecture we describe next.

## 3 Supporting Platform

To begin studying how to support these goals, we have developed an operational platform that is publicly accessible [3] and capable of storing data, meta-data and interpretations as well as interaction software. The system, known as DAE (for "Document Analysis and Exploitation"), runs on a 12-core machine with 32 GB RAM and 48 TB disk space, and, as such, is a credible proof-of-concept prototype. It also stores complete provenance [11] (*i.e.* the full data and event pipeline that has contributed to the creation of an element in the data base) of all data generated through the platform in order to capture all available con-

[4] Here "fluent" refers both to language skills as well as to domain knowledge.

text information that might impact on measuring differences between interpretations.

The data model is based on the following principles:

- all data is typed; users can define new types;
- data can be attached to specific parts of a document image (but need not be),
- both data and algorithms are modeled; algorithms transform data from one type into data of another type;
- full provenance of all data transformations is recorded;

The DAE server has been implemented using both Oracle 11.2 and MySQL back-end database management systems. It is accessed by a web front-end that provides a Web 2.0-like interface and encapsulates SQL queries to the back-end. It also relies on an independent application server that is used for executing registered algorithms on the data.

A simplified representation of the data model is represented in Figure 2. It consists of three key elements: `algorithms`, `data_items` and `algorithm_runs`. The underlying reasoning is that data is transformed by `algorithms`. `data_items` are instances of data and are related to algorithms by explicit `algorithm_runs`. New `data_items` thus produced are stored in the database with the exact information of how they were obtained.

Pre-defined types of `data_items` are `files`, `page_images`, `page_elements` , `datasets` and `page_element_properties`. The first three are straightforward generic data types that fit into any document analysis schema:

`file` is a `data_item` corresponding to a file containing data pertaining to some specific problem, in any format. This allows users to plug into our framework in an unconstrained and direct manner, without having to convert individual data and file formats.

`page_image` is an image file representing a physical page at a given resolution and with a given image quality. It is perfectly possible to have multiple `page_images` representing the same physical page, as shown in Figure 1.

`page_element` is an area of a `page_image`, defined in as unconstrained a way as possible, either by a bounding box or a pixel map, or other representations by means of a specific `page_element_property`.

These elementary `data_items` can be further extended by user-defined `page_element_property` and grouped into `datasets`. Furthermore, every `data_item` can be annotated, commented, and rated through a Web 2.0 interface, as shown in Figure 3.

The focus is not just on the data itself. Data semantics come from the fact that they have been applied to, or are the results from, specific algorithms, and are therefore tasks and interpretations as mentioned in the introduction. Since all data is structured in the database, it becomes straightforward to query for "all image regions to which at least two OCR algorithms have been applied and for which the resulting interpretations differ", or "find all OCR algorithm results for **this** image patch". This greatly enhances the processes we have been describing in our previous work [16].

The advantages of our platform go even beyond and can be summarized as follows:

**Formats and Representations** are transparently handled by the system, since the user can define any format, naming or association convention within our system. Data can be associated with

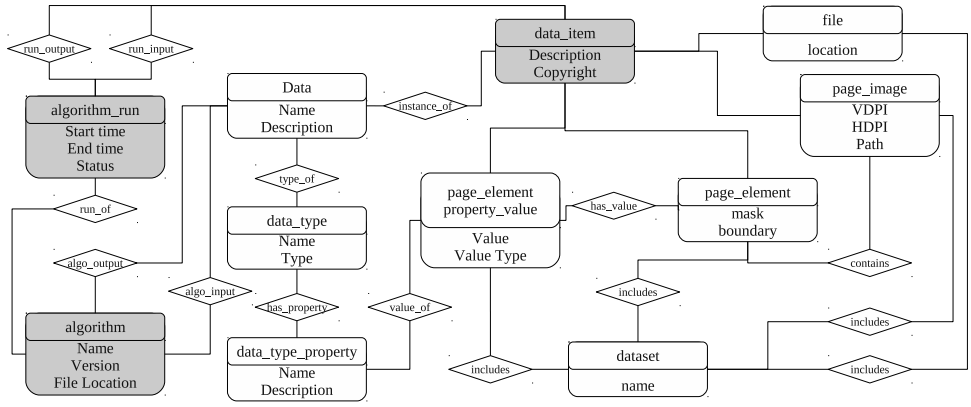


Figure 2: Simplified data model for the DAE web server (adapted from [11]; full version available on-line [4])

image regions, image regions can be of any shape and format, there is no restriction on uniqueness or redundancy, so multiple interpretations are not an issue. Furthermore, data can be conveniently grouped together in sets. These sets can in their turn be named and annotated as well. Data items do not need to belong exclusively to a single set, so new sets can be created by recomposition or combination of existing data sets.

**Storage and Access** architecture of the system makes it extremely easy to scale to higher demands, as both the storage and the computing infrastructures are conceived as physically separate entities. The current version already distributes some of its computing onto a high performance computing cluster for specific algorithms.

**Querying and Retrieval** is the great gain that the DAE platform offers. Because of its underlying data model and architecture, all data is accessible through SQL queries. The standard datasets that can be downloads from the platform are no longer monolithic `.zip` files, but potentially complex queries that generate new datasets on demand. Because of the degree of flexibility in annotating and supplementing existing data with meta-data, the potential uses are far beyond simple storage and retrieval of fixed data corpora.

**Interaction** with the data is integrated in the data model on the one hand (it represents algorithms, their inputs and outputs), but goes further by hosting algorithms that can be executed on the stored data, thus producing new meta-data and interpretations. Queries like finding all OCR results produced by a specified algorithm can either be used as an interpretation of a document, but can equally serve as a benchmarking element for comparison with competing OCR algorithms.

## 4 Uses and Interactions

The described DAE platform supports the kinds of interactions described in Section 2. Because of the flexibility of our data model, they all consist of similar interactions with the database.

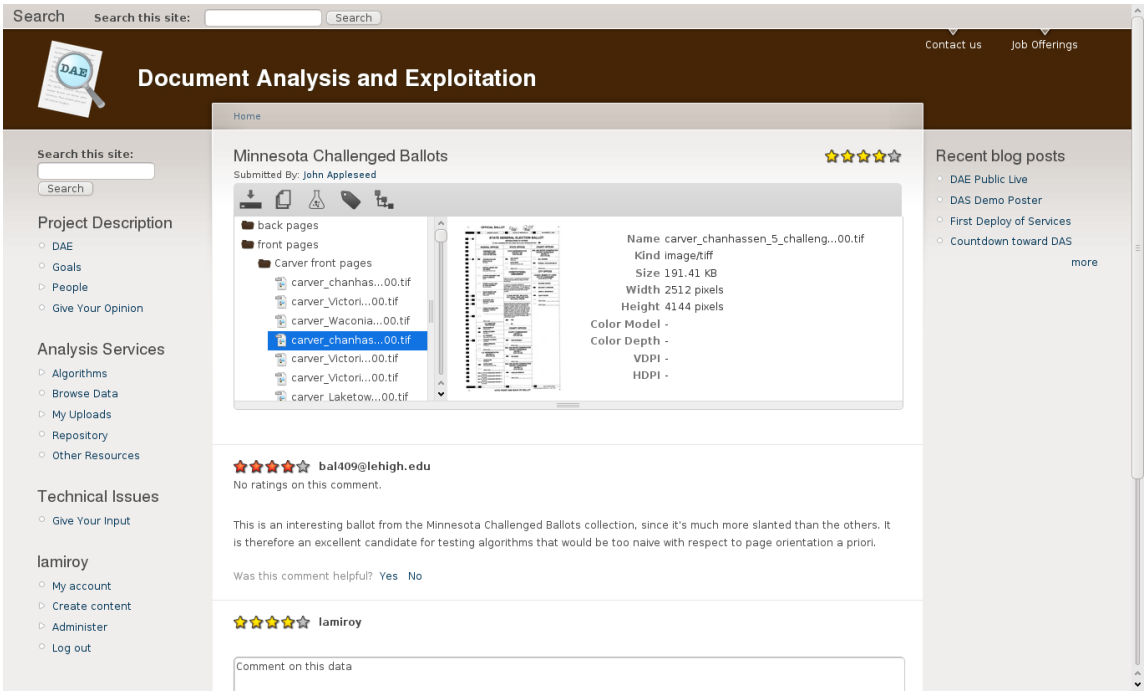


Figure 3: `http://dae.cse.lehigh.edu/DAE/?q=browse/dataitem/54246#47165` showing the on-line interface for annotating, commenting, and rating data.

### 4.1 Expressing Reading Order

Expressing reading order, as described earlier, is actually a restrictive instance of more general navigation support capacities. Navigation support can be defined as establishing directional links between regions of pages. Since page regions already exist in our data model by means of `page_elements`, we can define a new `page_element_property`. As shown in Figure 2, `page_element_properties` can be typed and are attached to any kind of `page_element`. We can therefore define a *link* property relating two `page_elements`.

Furthermore, the question of multiple versions of the same document are also handled in our system. Although not represented in the simplified data model in Figure 2, the complete data model [11] represents the notion of document as a set of "pages." Each page can be projected onto any number of `page_images`. `page_images` can thus be considered as single, stand-alone, document analysis objects in one context, or related to other images, known to represent the same physical object, but captured under other conditions.

A typical SQL query for retrieving other versions of a same `page_images` would consist in retrieving the master document of an image of interest, and then find all `page_images` generated from this document. For instance, if the image has an identifier `X` then we would get:

```
select PAGE_ID from GENERATES_PAGE_IMAGE
    where PAGE_IMAGE_ID = X;
```

to retrieve the master document ID, and calling it `M`,

```
select PAGE_IMAGE_ID from GENERATES_PAGE_IMAGE
    where PAGE_ID = M;
```

would yield all copies of the document.

### 4.2 Adding Interpretations

The wide range of interpretations mentioned in Section 2.4 can be handled in three ways.

1. The first one is the one we mentioned previously, by adding properties to `data_items`.

   For instance, during our OCR experiments, we have defined the following `datatype_properties`:

   **recognized text** represents the recognized textual information from a page element through some OCR algorithm,

   **number of characters** represents the number of characters that a page element has,

   **textline height** represents the height of a text line (in pixels).

   However, proceeding in this way would be equivalent to assigning a unique ground truth to an input. This is not what we are aiming at.

2. A more coherent way of adding interpretations is to include `algorithms` and provenance. This requires the same definition of `datatype_properties` as in the previous case, but these properties are not directly attached to `data_items`. Instead, they become the result of `algorithm_runs` that take `data_items` as input. The fundamental difference is that it is now possible to manage an large spectrum of similar annotations and interpretations of the same data. It even becomes possible to reuse and adapt interpretations to new contexts without loss of previous knowledge, as will be explained in Section 4.4.

   Furthermore, the semantics are now expressed by the explicit relation with an `algorithm` which can, in its turn, allow for a very detailed analysis of the results.

3. In a less formal, but still useful way, we have also mentioned that any `data_item` can be freely annotated and rated by users (as shown in Figure 3). This makes possible a user-friendly and user-centric interaction, beyond more formal and algorithm-oriented interactions. It can allow open discussion of alternate viewpoints before updating and modifying data in the database itself.

### 4.3 Comparing Interpretations

Given the tools and models described previously, comparing interpretations becomes a challenging goal. If interpretations share identical (syntactic) representations, this might be straightforward. It introduces also very interesting new perspectives when their syntactic representations differ and comparison requires operating on a more semantic level. This includes comparison of results from different algorithms addressing similar problems, combining and pipelining algorithms [12] to achieve similar results, or even different versions of the same algorithm. It is also possible to define *oracle* algorithms (*i.e.*, algorithms that do not actually correspond to executable code, but are, for instance, knowledgeable human experts. They are considered being flawless and always giving the exact correct result for a specific context). In that case, the result would be manually annotated datasets.

One of the main innovations in our approach is that the platform allows for modeling, storage, and execution of any kind of algorithm operating on data. Up to now, we implicitly assumed that these algorithms were document analysis-related, extracting interpreta-

tions from document images. But the platform can do more than this, since it can host and execute the evaluation algorithms that compare the final results of a given analysis pipeline.

The scenario depicted in Figure 1 gives rise, on our platform, to the following queries:[5]

- Retrieving Interpretation1 and Interpretation2 would be done through respectively

```
select DATA_ITEM_ID from
 ALGORITHM_RUN_OUTPUT, ALGORITHM_RUN_OF
where
 ALGORITHM_ID='OCR 1' and
 ALGORITHM_RUN_OUTPUT.ALGORITHM_RUN_ID =
     ALGORITHM_RUN_OF.ALGORITHM_RUN_ID;
```

  and an equivalent query replacing OCR1 by Bob's Transcription.

- Retrieving all interpretations compatible with a given evaluation metric (*i.e.* interpretation data that is in a required format type, for instance) requires that we check the `data_type` of the interpretations. The queries would therefore retrieve all `data_items` issuing from any `page_image` that is a representation of the given physical page and that have a `data_type` that is the same of the `Data` required as input for the evaluation algorithm.

### 4.4 Reusing Interpretations

Finally, reusing interpretations in new contexts becomes possible. It might however be necessary to adapt (or "correct") them. The need to create a new interpretation for existing data, usually arises from a change in context. In that case, one simply declares a new `algorithm` and uploads the interpretation data resulting from it. However, there can also be a need for a correction within an existing context, due to a misinterpretation of it, or a programming error, for instance. In that case, one contributes a new version of an existing algorithm and the platform has all the tools and information to automatically generate all new corresponding meta-data.

While new interpretation contexts often give rise to new algorithms which can be hosted and executed by the platform, manual annotations and interpretations are equally possible. As we have seen previously, manually annotated documents still require the registration of an *"oracle"* algorithm, such that provenance and dependencies are maintained.

[5]The SQL queries given in this paper are slightly simplified for ease of reading and hide some join operations with tables that are not essential for understanding the underlying principles.

# 5 Discussion

## 5.1 Paradigm Shift

Our ultimate goals for the research described in this paper distinguish it from past work on document ground-truthing and evaluation. We do not focus on establishing ground-truth or designing evaluation processes, but instead on providing a new paradigm showing how to handle access to multiple, contradictory, and incomplete interpretations. As such, it is not comparable to existing software libraries [19, 9], data formats or ground-truth creation algorithms [13, 21].

The paradigm shift that underpins our work is motivated by the fact that, while the above tools and solutions provide excellent means of sharing algorithm implementations and allow for building on previous work, they do not address the issue of comparing research results coming from different sources. They are merely there for people to build upon (which is, by the way, already quite a step in the right direction). Our work is to be seen in a broader context, integrating the latter. It builds on two major assumptions:

**The Need for Peer Assessment** of algorithms, methods and datasets is crucial for the research community. Because a lot of the research is conducted in focused or application-specific contexts, it is hard to effectively achieve real incremental research with fully and openly assessed and measured performances and cross-domain impact evaluations.

**Crowd-Power** is currently the most versatile and dynamic approach to peer assessment. Web 2.0 communities have shown their tremendous capacity to dynamically adapt to new information flows, and to have a selective evaluation of uncontrolled data. This model only works if two major conditions are met:

1. Unrestricted and open access to data for contribution, retrieval or extension (*e.g.* Wikipedia [1]).
2. The personal benefit perceived by the contributing individual increases with his level of contribution and outweighs the overhead of contributing to the system (*e.g.* Facebook [6]).

Our platform is exactly that: a comprehensive platform supporting la larger paradigm that will allow peer evaluation of algorithms and datasets through crowd cooperation.

Being able to capture and exploit all user interactions with a collection and its documents requires an approach to representing and relating the alternative interpretations described in this paper. Feeding this information back to improve the performance of document analysis algorithms then follows as a natural extension.

One of the critical points of this kind of paradigm is that it succeeds only in a collaborative and collective environment. In order to work, it needs to be widely adopted and used, but in order to be adopted, it must prove its usefulness (*cf.* the personal benefit mentioned previously). To help promote this, we are planning to populate our repository with widely used datasets, and create easy interfaces for upload and retrieval that do not require adopting a specific data format.

The infrastructure we have developed was designed for continuously evolving data and interpretations and will drive our research for a significant period of time, since it provides a new way of viewing experimentation, evaluation, and certification of scientific results in document analysis.

## 5.2 Further Work

**Widespread Adoption Needs Promotion** It is obvious that, although we believe in the

Web 2.0 viral spreading potential, our approach is not going to be adopted solely because it is a nice idea. The platform reported in this paper is fully operational and functional, but has not yet been openly promoted. In order be adopted by the community it needs a significant initial investment in promotion. At the first stages of adoption, it will need tuning to all the user needs we might not have initially envisioned.

This will be done through various ways:

1. making the full source code and documentation of the platform available under an open GNU-like license;

2. hosting a wide range of publicly available datasets, in collaboration with the IAPR TC-10 and TC-11 committees[6];

3. providing a flexible framework for hosting and running document analysis contests;

4. organizing demonstrations and tutorials at international events.

**"Eating our own Dog Food"** The dissemination initiatives mentioned in the previous section are lacking one essential element: algorithms. We are aware that the hosting and integration of any kind of algorithm is a very challenging task. Although there are mature technological solutions like virtual machine hosting as to overcome compatibility issues, or web-service infrastructures to take advantage of, if not cloud computing, at least distributed computing facilities, the engineering effort is very likely to be substantial (hence the interest of having the platform widely adopted, and thus supported, by the community).

In order to overcome the initial barrier, we are extensively using our own platform and currently providing access to and hosting or own algorithms and meta data, extending it continuously as our research advances and needs for comparison with other tools arise.

**Missing Features** Currently missing features of the platform are related to its user interface. The whole data model described in this paper, and detailed in [11, 4] provides for all the functionality that we have been mentioning. However, a great part of it requires substantial technical knowledge of the underlying architecture and implementation. This is an inconvenience for the average document analysis user. We are currently working on very low effort web based interaction tools that will ease the transition for a widespread adoption.

[6] The International Association for Pattern Recognition is an international association of non-profit, scientific or professional organizations concerned with pattern recognition, computer vision, and image processing in a broad sense. Its technical committees TC-10 and TC-11 are respectively concerned with graphics recognition and reading systems.

## 6 Acknowledgments

The DAE project and resulting platform is a collaborative effort hosted by the Computer Science and Engineering Department at Lehigh University and funded through a Congressional appropriation administered through DARPA IPTO via Raytheon BBN Technologies. The project currently involves the following members (in alphabetical order) Chang An, Sai Lu Mon Aung, Henry Baird, Michael Caffrey, Siyuan Chen, Brian Davison, Jeff Heflin, Hank Korth, Michael Kot, Bart Lamiroy, Daniel Lopresti, Dezhao Song, Pingping Xiu, Dawei Yin.

## References

[1] A. Bruns. *Blogs, Wikipedia, Second Life, and beyond: From production to produsage*. Peter Lang, 2008.

[2] A. Clavelli, D. Karatzas, and J. Lladós. A framework for the assessment of text extraction algorithms on complex colour images. In *Proceedings of the 8th IAPR International Workshop on Document Analysis Systems*, pages 19–26, Boston, MA, USA, 2010. ACM.

[3] Document Analysis and Exploitation (DAE) web server. http://dae.cse.lehigh.edu.

[4] DAE server entity-relationship model specification. http://dae.cse.lehigh.edu/Design/ER.pdf.

[5] U. Eco. *The limits of interpretation*. Indiana University Press, Bloomington :, 1990.

[6] N. Ellison, C. Steinfield, and C. Lampe. The benefits of Facebook" friends:" social capital and college students' use of online social network sites. *Journal of Computer-Mediated Communication*, 12(4):1143–1168, 2007.

[7] J. Hu, R. Kashi, D. Lopresti, G. Nagy, and G. Wilfong. Why table ground-truthing is hard. In *ICDAR01*, pages 129–133, 2001.

[8] J. Hu, R. Kashi, D. Lopresti, G. Nagy, and G. Wilfong. Why table ground-truthing is hard. In *Proceedings of the Sixth International Conference on Document Analysis and Recognition*, pages 129–133, Seattle, WA, September 2001.

[9] S. Jaeger, G. Zhu, D. Doermann, K. Chen, and S. Sampat. DOCLIB: a software library for document processing. In *Proceedings of Document Recognition and Retrieval XIII (IS&T/SPIE Electronic Imaging)*, volume 6067, pages 09.1–09.9, San Jose, CA, January 2006.

[10] S. K.C., B. Lamiroy, and J.-P. Ropers. Inductive logic programming for symbol recognition. In *10th International Conference on Document Analysis and Recognition*, pages 1330–1334, Los Alamitos, CA, USA, 26-29 2009. IEEE Computer Society.

[11] H. F. Korth, D. Song, and J. Heflin. Metadata for structured document datasets. In *Proceedings of the 8th IAPR International Workshop on Document Analysis Systems*, pages 547–550, Boston, MA, USA, 2010. ACM.

[12] B. Lamiroy and L. Najman. Scan-to-XML: Using Software Component Algebra for Intelligent Document Generation. In D. Blostein and Y.-B. Kwon, editors, *4th International Workshop on Graphics Recognition - Algorithms and Applications*, volume 2390 of *Lecture Notes in Computer Science*, pages 211–221, Kingston, Ontario, Canada, 2002. Springer-Verlag.

[13] C. H. Lee and T. Kanungo. The architecture of TrueViz: a groundTRUth/metadata editing and VIsualiZing toolkit. *Pattern Recognitino*, 36:811–825, March 2003.

[14] J. Liang, R. Rogers, R. M. Haralick, and I. T. Phillips. UW-ISL document image analysis toolbox: An experimental environment. In *Proceedings of the Fourth International Conference on Document Analysis and Recognition*, pages 984–988, Ulm, Germany, August 1997.

[15] D. Lopresti and G. Nagy. Issues in ground-truthing graphic documents. In *Proceedings of the Fourth IAPR International Workshop on Graphics Recognition*, pages 59–72, Kingston, Ontario, Canada, September 2001.

[16] D. Lopresti and G. Nagy. Tools for monitoring, visualizing, and refining collections

of noisy documents. In *AND '09: Proceedings of The Third Workshop on Analytics for Noisy Unstructured Text Data*, pages 9–16, New York, NY, USA, 2009. ACM.

[17] D. Lopresti, G. Nagy, and E. B. Smith. Document analysis issues in reading optical scan ballots. In *Proceedings of the 8th IAPR International Workshop on Document Analysis Systems*, pages 105–112, Boston, MA, USA, 2010. ACM.

[18] W. Raub and J. Weesie. Reputation and efficiency in social interactions: An example of network effects. *American Journal of Sociology*, 96(3):626–654, 1990.

[19] J. Rendek, G. Masini, P. Dosch, and K. Tombre. The search for genericity in graphics recognition applications: Design issues of the Qgar software system. In S. Marinai and A. Dengel, editors, *6th IAPR International Workshop on Document Analysis Systems*, volume 3163 of *Lecture Notes in Computer Science*, pages 366–377, Florence, Italy, 2004. Springer Verlag.

[20] J. Sabater and C. Sierra. Review on computational trust and reputation models. *Artificial Intelligence Review*, 24(1):33–60, 2005.

[21] E. Saund, J. Lin, and P. Sarkar. Pixlabeler: User interface for pixel-level labeling of elements in document images. In *10th International Conference on Document Analysis and Recognition*, pages 646 –650, 26-29 2009.

[22] E. H. B. Smith. An analysis of binarization ground truthing. In *Proceedings of the 8th IAPR International Workshop on Document Analysis Systems*, pages 27–34, Boston, MA, USA, 2010. ACM.

[23] Stefan Jaeger, Guangyu Zhu, David Doermann, Kevin Chen, and Summit Sampat. DOCLIB: a Software Library for Document Processing. In *International Conference on Document Recognition and Retrieval XIII*, pages 1–9. San Jose, CA, 2006.

[24] Tobacco800 data set. http://www.umiacs.umd.edu/~zhugy/Tobacco800.html.

[25] UNLV data set. http://www.isri.unlv.edu/ISRI/OCRtk.

[26] B. Yu and M. Singh. A social mechanism of reputation management in electronic communities. *Cooperative Information Agents IV-The Future of Information Agents in Cyberspace*, pages 355–393, 2000.