Tools for Monitoring, Visualizing, and Refining Collections of Noisy Documents

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

George Nagy
Department of Electrical, Computer, and Systems Engineering
Rensselaer Polytechnic Institute
Troy, NY 12180, USA
nagy@ecse.rpi.edu

ABSTRACT

Developing better systems for document image analysis requires understanding errors, their sources, and their effects. The interactions between various processing steps are complex, with details that can be obscured by the statistical methods that are employed in many cases. In this paper, we describe tools we are building to help the user view and understand the results of common document analysis procedures. Unlike existing platforms for ground-truthing page images, our system also allows users to visualize the results of automated error analyses. Recognition errors can be corrected interactively, with the effort to do so recorded as a measure that is useful in performance evaluation.

Beyond this functionality for exploring error behavior, we consider how such tools could be designed to improve the quality of collections of badly recognized documents incrementally as users interact with them on a regular basis. We conclude by discussing topics for future research.

Categories and Subject Descriptors

I.7.5 [**Document and Text Processing**]: Document Capture—*document analysis*

General Terms

algorithms, measurement, performance

1. INTRODUCTION

Collections of document images arise in a number of important applications, including digital libraries, newspaper "morgues," commercial and non-profit initiatives (*e.g.*, google books and The Million Book Project), the litigation discovery process (*e.g.*, the tobacco corpus), freedom of information disclosures and declassification activities by government agencies, electronic voting using scanned paper ballots, legacy patient medical records, and law enforcement, intelligence gathering, and military operations.

AND '09, July 23-24, 2009, Barcelona, Spain

Researchers developing technologies to support the analysis of such documents also typically build their own collections of scanned pages for the purposes of training and testing new algorithms. On occasion, these corpora are shared with the research community where they serve as a valuable (if sometimes overused) resource.

In both instances, additional information must be created to facilitate computer processing of the collection. When page images are brought online, a coded representation of the document is included along with the associated bitmaps. The requirements for what needs to be recorded depends on the intended application. To support end-user retrieval under a simple "bag-of-words" model, all that is necessary is the set of terms found on the page in no particular order (for retrieval purposes, the correct reading order may be indistinguishable from a random arrangement of the words). On the other hand, to support development of high quality optical character recognition methods, the pages in the collection may have to be annotated down to a very fine level, with the location and identity of every character represented in the so-called "ground-truth." Even richer annotations may be desirable in some cases, *e.g.*, an interpretation that includes the type size and typeface for each character. Translations from the original source language into other languages might be required. The amount of manual intervention needed depends not only on the quality of the input document, but also on the requirements on the quality of the output.

It is generally assumed that there is a single, unambiguous annotation in every case and that it is recorded correctly in the ground-truth. This is nearly always a tedious, time-consuming, and expensive proposition. The goal of document image analysis is to some day completely automate the task of creating such annotations, although as of today, some manual intervention is required in all but the simplest of cases, or the lowest of expectations. Ideally, this process will capture a perfect representation of the content and all associated meta-data. In practice, however, document analysis systems are "brittle" and a wide range of errors may arise. Some of these can severely impact intended uses of the acquired information. In practice, users must either tolerate a substantial amount of noise, or else forgo applications that are predicated on the assumption the collection is noise-free.

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 [7, 15].

In this paper, we describe ideas for new techniques for interacting with document images and collections that differ in important ways from traditional approaches. Imagine an important collection of pages that on the one hand is too large to be annotated completely by hand, but on the other cannot be successfully processed using fully automated techniques. The collection must be pressed into service immediately. We envision a paradigm where the quality of the collection can be improved over time as a result of users interacting with it, in the same spirit as online wikis which have proven to be surprisingly effective [21]. Some of these interactions may be conscious steps taken to correct errors or provide additional information. Others may be more subtle cues captured without the user being aware.

There are already a number of fine markup editors and suggestions for standardized file formats. It is not our intention to propose yet another such system to compete with those that already exist. Rather, we seek to explore new tools for monitoring, visualizing, and refining document collections that acknowledge the unavoidability of errors (and ambiguity) and that exhibit the following principles:

- Allow that the current interpretation for an entity on a page may be incorrect.
- Assume there may be more than one acceptable interpretation for a particular entity.
- Seek to help users understand where differences in interpretation have arisen and, in cases where they are regarded as errors, make it easy to correct them.
- 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.
- Optimize the task-directed performance of the overall human / machine system.

The remainder of this paper describes work in progress and is organized as follows. In Section 2, we briefly summarize related research. Section 3 presents some important models and abstractions designed to support the new paradigm. A prototype that is now under development is discussed in Section 4, and its error visualization techniques are described in Section 5. Section 6 concludes with a discussion of open questions and ongoing work.

2. RELATED WORK

Since the need to browse, edit, and manage documents is universal across so many fields, there is much past work that might be considered relevant. Here we focus on just a few representative efforts. In the area of architectures to support ground-truthing for document analysis research, we cite Pink Panther [23] and TRUEVIZ [10]. The DAFS standard [18] as well as "toolboxes" like Gobbledoc [17], UW-ISL [11], Gamera [3], and DOCLIB [8] with its GEDI markup editor also deserve mention.

The field of Natural Language Processing (NLP) has developed its own ground-truthing systems such as ATLAS [1]. Carefully curated datasets require tools and protocols that focus on achieving high levels of accuracy [19].

Early work on methods for classifying errors in optical character recognition (OCR) [5, 6], later extended to handle downstream text processing stages [12, 13, 14], form the basis for the analyses we describe later. Past discussions of performance metrics have informed our thinking [9, 16, 20].

The notion of exploiting user interactions with preliminary machine recognition results arises in pattern recognition research [22] and the CAVIAR paradigm [24], as well as in mixed initiative techniques for NLP corpus creation [2].

Recent developments in the World Wide Web, and in particular the concept of community-edited and maintained resources such as Wikipedia [21], can be considered a logical end when the models and tools we propose reach maturity.

The work we are about to describe differs from past attempts to create standard file formats and editor software. We view these as steps towards our ultimate goals which are to investigate the interpretation of documents by humans and machines, how these interpretations can be related with one another, ways in which interpretations are refined over time, and the connection between document recognition results and downstream applications.

3. MODELS AND ABSTRACTIONS

One focus of our research is to examine the ways in which documents and their contents are represented for various purposes, and the ways in which these representations interrelate and evolve over time as human users and algorithms interact to extract more and better information from the pages in the collection. Central to the discussion in the case of documents with textual content is the concept of an *interpretation*, which we define very broadly. While we normally think in terms of a verbatim transcript of the text on the page, an interpretation can also be an annotation, a gloss, a concordance, a translation, a bag of words, a critique, a summary, a rebuttal, or even just a simple citation.

In this section we survey features we believe should be incorporated in a comprehensive system. Whether or not certain functionality falls within the realm of feasibility for existing algorithms is a separate issue, since manual approaches are not only possible in every case, our philosophy places them on par with automated techniques. While we often aim for generality, a specific instance of a document should not require that all possible components be populated to be

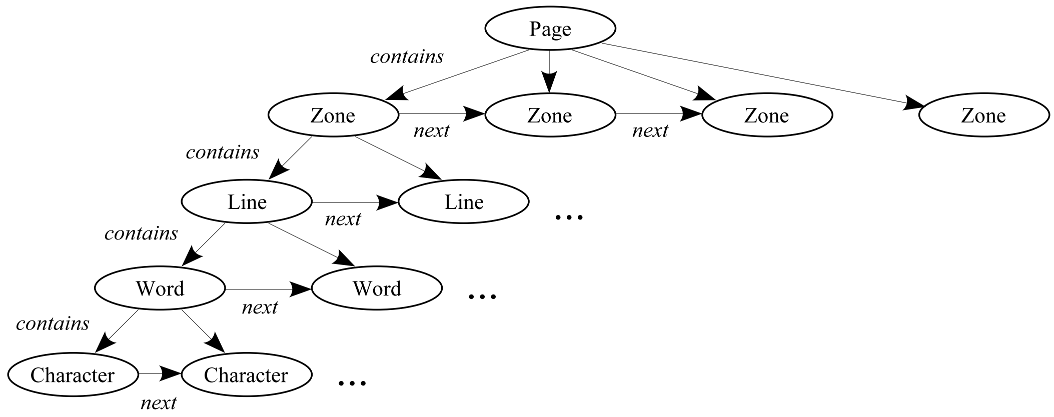

Figure 1: Document model (adapted from [10]).

considered "complete." Indeed, allowing for differing levels of quality and completeness in a collection is vital.

3.1 Document Models

As a result of earlier work on ground-truthing tools (*e.g.*, TRUEVIZ [10] and DOCLIB [8]), the basic entities needed to represent a document are well understood and generally consist of a hierarchy of regions that are labeled as pages, zones, text lines, words, and characters. In addition, several obvious kinds of relationships between entities should be supported, including reading order (next / previous) and containment (pages contain zones which contain text lines, etc.), as depicted in Figure 1. Beyond these pre-defined categories, there is a case to be made for supporting arbitrary, user-defined region labelings and relationships, in much the same way that online photo-sharing websites (*e.g.*, flickr) allow users to tag images using their own keywords.

Beyond the sort of mark-up functionality provided in standard ground-truthing tools for document image analysis, support must also be included for text tagging since this is required by NLP tools for sentence boundary detection, tokenization, part of speech labeling, named entity identification, etc. Rendering such tags graphically and making them editable is quite feasible [1].

3.2 Cross-Page Relationships

Much work in document analysis research takes an individual, page-at-a-time approach and ignores the fact that the page is part of a larger collection. Obviously, the pages that comprise a single document should be linked. Going beyond this, current digitization practice makes little use of the relationships built into documents, such as tables of contents, indices, captions, and footnotes. Human readers use this information extensively, and algorithms should, too. A useful system must encourage the creation, correction, and maintenance of such cross-page relationships.

Connections between pages and documents can either be explicitly defined (an algorithm or the user makes a direct connection) or implicitly defined (the connection arises as a result of some other action taken by the user). For example, if we have reason to believe that a user is reading through a multipage document based on her interactions with the interface, then the order that the pages are visited can confirm whether or not the computed page ordering is right; if not, we can reorder the pages based on our observations.

Often, multiple versions of the same document are available in a corpus, as in the case of successive editions, condensed versions, extracts, and translations. These should be exploited to construct glosses for modern documents, much as scholars now do for ancient manuscripts. Exact and near-duplicates may also be present, although this may be challenging to detect if the physical document was degraded badly between scans. Various sorts of cross-page relationships are shown in Figure 2.

3.3 Alternative Interpretations

The various entities on a page are located and interpreted by humans interacting with the system and by algorithms which have been invoked to process the page (or part of it). Over time, a given page may receive multiple, interleaved interpretations, some of which build on previous interpretations (correcting errors which have been found), and others of which start from scratch (working directly from the page image). Figure 3 illustrates a range of possibilities.

However we define interpretation in a particular application, one of its attributes must be the range of extracted data: character-level, word-level, sentence-level, page-level, document-level, corpus-level, world-level (inter-corpora).

A useful analogy comes from the field of software version control. Updating a computer program does not necessarily improve it, anymore than editing the interpretation of a document is guaranteed to raise its accuracy. The format and structure of representations are important, but so is the process by which representations are created. This observation is captured in for-profit and open source software factories, but not in the way ground-truth is currently maintained for document collections.

The quality of an interpretation should be measured relative to the intended application for the material which can, of course, range widely. Interpretations meant to benefit primarily human readers face different constraints than those intended for computer analysis. While the original author of

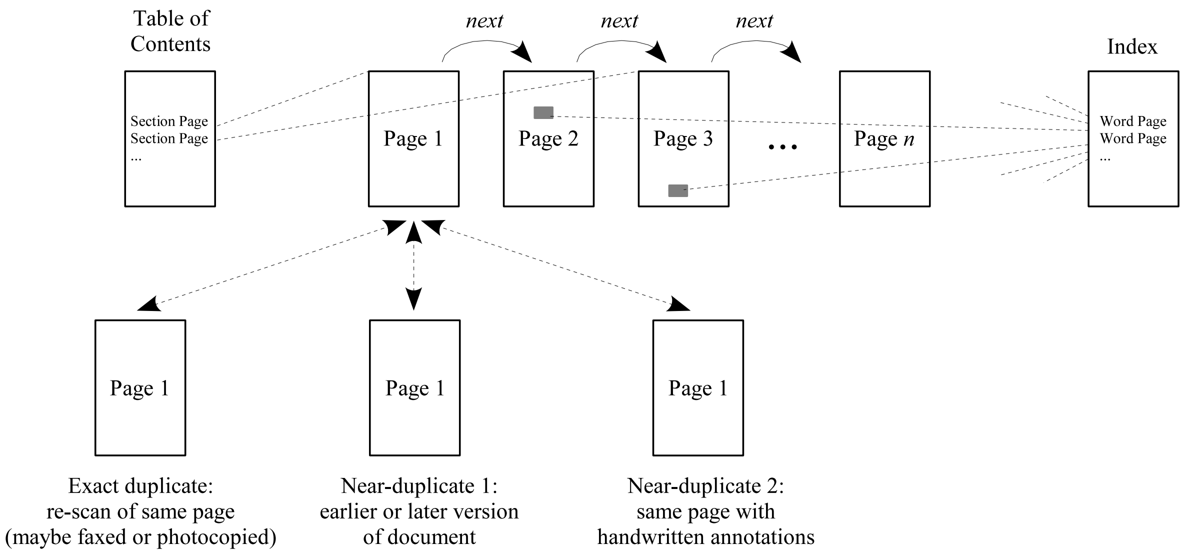

Figure 2: Cross-page relationships: consecutive pages, TOC, index, exact and near-duplicates.

a document can be viewed as a definitive voice, it is not clear whether looking backward is productive. More bluntly, errors that do not affect downstream processing are not really errors, whatever the author might think. For this reason, as well as because of the inherent ambiguity that is present in some cases, it is necessary to find ways to maintain and use multiple, alternate interpretations.

The fact that there is not necessarily a single canonical interpretation for a page raises questions about which representation to use for a particular purpose. This link between the corpus and its intended uses requires further exploration; a more active approach based on querying a document collection for representations that satisfy certain criteria that are pre-conditions for the downstream application is one potential solution. There may also be a place to incorporate notions of "trust" as are now being applied in retrieval research for the World Wide Web. Trust for human users and for machine algorithms can be calibrated, perhaps, by correlating performance when two or more interpretations are offered for the same page.

The result of interleaving machine and human interpretations will result in an interaction "history" for the page. Ways should be provided for navigating this history, including providing a form of "undo" so that users can engage in "what if" exercises. Histories can become tree-structured since users can go back to a previous point and then create a new alternate interpretation branch going forward.

3.4 Error Models and Analysis

As has been noted, errors under our model are nothing more than differences between two interpretations. This raises questions about how one performs error analysis since there is no "golden" truth to compare to. Two options are depicted in Figure 4:

1. Identify two nodes in the interpretation history for the page (recall Figure 3) and employ an automated technique to classify the differences when transforming from one to the other (*e.g.*, string edit distance if raw text is being compared [5, 6], or hierarchical edit distance if tagged text is the target [12, 13, 14]).
2. Edit a single interpretation to "correct" the errors, measuring the amount of work involved. Note that the editing could be a manual process performed by a human user (the traditional view) in which case we might count the number of editing steps or the total time, or it could be an algorithmic post-processing step that purports to correct errors made by a previous step (*e.g.*, running a spell checker that automatically corrects to words in its lexicon).

Error analyses serve two basic purposes: to facilitate the comparison of different approaches so we can choose the best option, and to identify the kinds of mistakes being made so that we can envision how to build more reliable methods. In this last context it is possible for an error analysis to be wrong. For example, simple string edit distance may take $m \rightarrow rn$, which is most often due to a single incorrect segmentation decision (when m is split into two pieces) and erroneously classify this as a pair of recognition errors, a substitution ($m \rightarrow r$) followed by an insertion ($\rightarrow n$). This can be confirmed through visual inspection of the character in question.

In the event a user disagrees with the way a particular error has been classified, she should be allowed to change it. Hence, not only are interpretations editable, error analyses which document the relationship between two interpretations are likewise editable. Both the documents in the collection and our understanding of the errors that have arisen become more accurate over time. Developing methods to measure this effect and to relate it to analyses of community-based editing on the World Wide Web is a topic for future research. We note, however, that the models are clearly not

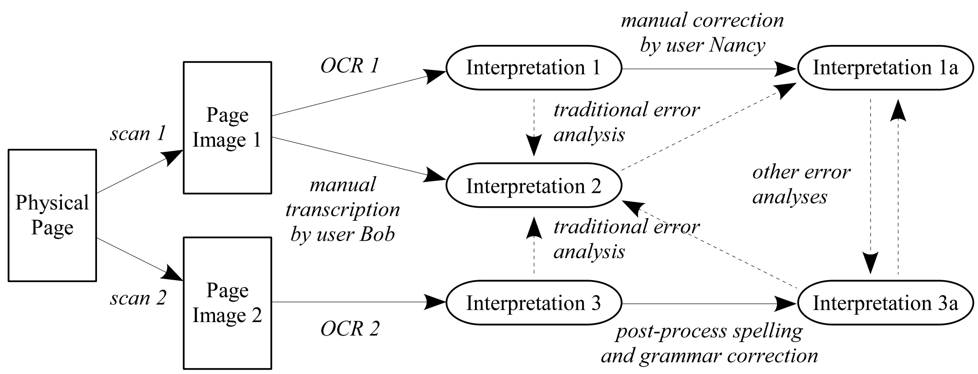

Figure 3: Alternate interpretations and relative error analyses.

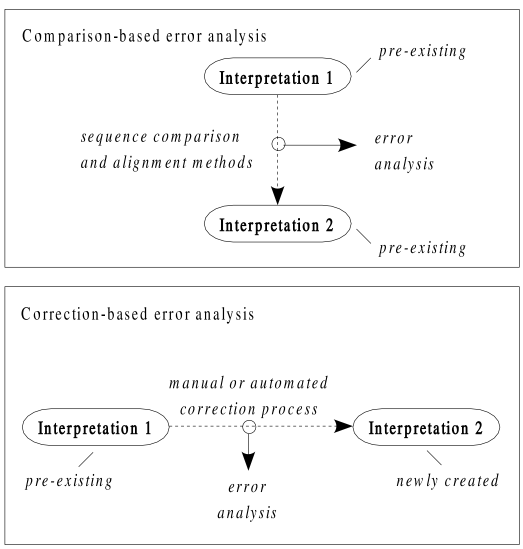

Figure 4: Comparison-based (top) and correction-based (bottom) error analyses.

identical because of our support for multiple interpretations.

4. PROTOTYPE

We have started to develop a prototype to implement some of the more significant functionality described in this paper. Our tool is coded in Tcl/Tk, a popular scripting language that is freely available across multiple platforms including MS Windows and Linux. In addition to browsing documents and their various interpretations, the ability to display and edit on-screen markup (rectangles, ovals, text labels, lines and arrows, and free-form scribbles) is provided. Also allowed is the editing and tagging of ASCII text.

The tool is designed to capture and timestamp all user interactions and save them into a log file. We shall use this data in later evaluations to determine the utility of various interaction techniques. When the user is editing an existing interpretation, our record of the character-by-character edits allows us to identify the errors that are implied and quantify the effort need to correct them. The record is detailed enough to allow replaying the editing session on the screen. A section of a log file is shown in Figure 6.

To facilitate editing, we plan to implement spell check, grammar check, and wild card search-and-replace features. User-settable lexicons are also under consideration. Anomalous word-frequency profile checks (like "spare" appearing too often in a document that has an above average number of occurrences of "spore") may be useful as well. Support for the logical markup of tables can be had by integrating a tool we built previously for that purpose [4].

Key principles we intend to follow throughout this work include keeping track of the interactions of multiple human and automated editors looking at the same documents, possibly in multiple versions. Subsequent users will have the results of earlier edits available in a convenient format (*i.e.*, consensual edits shown only once). Output will be guided by its usability in downstream processes, *e.g.*, translation, retrieval, question-answering. In other words, the target is already-working systems for which producing enough input is too costly. We intend the architecture of the overall system to be open access, much like Wikipedia. Ultimately, we wish to support web-based interaction tools in addition to the current standalone application.

5. ERROR VISUALIZATION TECHNIQUES

While a substantial amount of work has examined the issue of performance evaluation, there tends to be less attention devoted to helping users visualize errors and their impact. Techniques that are based on approximate string matching and related paradigms offer the benefit of being able to explain their measurements in terms of a well-defined underlying error model. Our past work in this area has included the ability to generate error traces which we have used for a variety of purposes [5, 6, 12, 13, 14]. In the new tool we are building, we have programmed interactive versions of these visualizations which users can employ to better understand the errors that have arisen as well as to correct errors that may have been misidentified.

Figure 7 gives a screen snapshot of our prototype showing a character-level error analysis of the OCR text for a zone when aligned to ground-truth which was entered manually. The failure of the OCR package (tesseract version 2.0) to detect the word "Living" is evident in the lower edit dis-

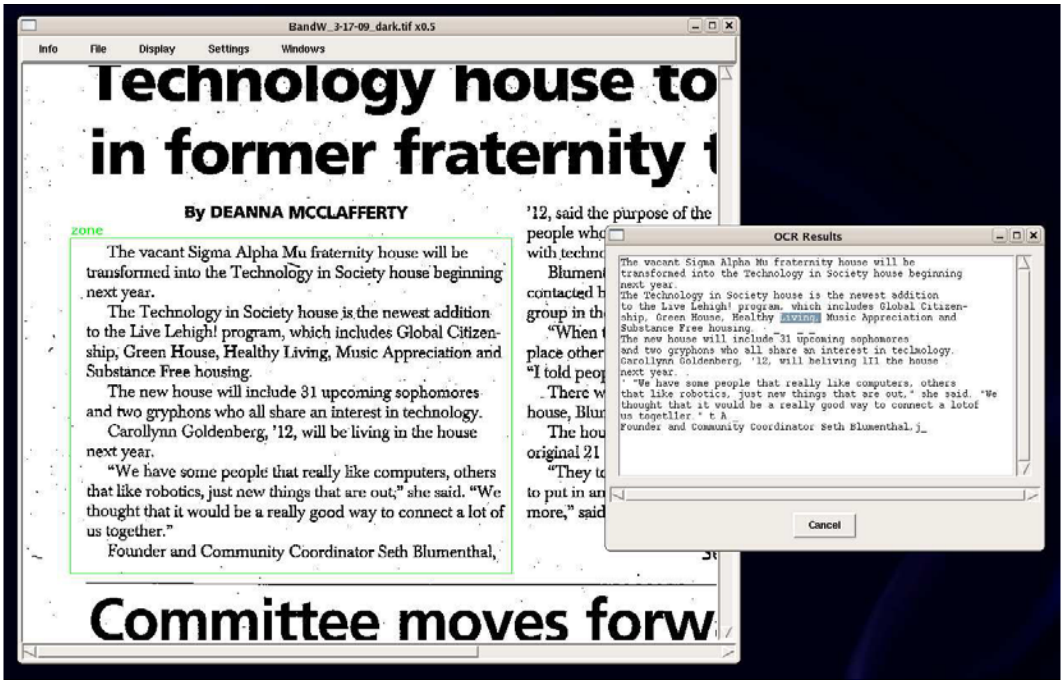

Figure 5: Screen snapshot of the prototype showing a page fragment, associated zone mark-up, and the editable textual contents for one zone (OCR results).

```
344748599 loadOCRResults
          Wed May 06 02:11:37 AM EDT 2009
21690 button_release insertion 1.52
22963 keypress BackSpace
23347 keypress BackSpace
28882 button_release selection 2.33 2.35
30498 keypress n
33840 button_release insertion 2.59
34809 keypress BackSpace
37176 button_release selection 3.0 3.4
37992 keypress BackSpace
40479 button_release insertion 3.22
41376 keypress BackSpace
...
43615 keypress BackSpace
44967 keypress period
51349 button_release selection 4.32 5.33
52126 button_release insertion 4.33
53390 keypress BackSpace
55885 button_release selection 4.34 4.35
55965 keypress space
58789 button_release insertion 4.55
59941 keypress BackSpace
67915 button_release insertion 6.27
69523 keypress Shift_L
69939 keypress L
70179 keypress i
70803 keypress v
...
```

Figure 6: Log file showing user activity in correcting OCR results (from Figure 5).

tance window (most likely this is due to the characters in the word touching and tesseract regarding the extended connected component as being non-text-like).

Figure 8 shows a more involved analysis of the output from OCR followed by two other common text processing stages (tokenization and part-of-speech tagging) using the techniques we presented at an earlier AND workshop [14].

Error analyses are a "first-class" datatype in our system. We plan to enhance our tool to generate confusion tables and other summary statistics to help the user. Ultimately, we wish to determine which kinds of visualizations are most useful for improving the performance of the system as well as for raising the quality of the document collection.

6. DISCUSSION

Despite the obvious similarities, our ultimate goals for the research described in this paper distinguish it from past work on document ground-truthing and evaluation. To be able to capture and exploit all user interactions with a collection and its documents requires new approaches for representing and relating alternative interpretations. Feeding this information back to improve the performance of document analysis algorithms then follows as a natural extension. Philosophically, this reflects the CAVIAR paradigm [24]: seek to optimize the performance of the human-machine system.

As our prototype tools reach a state of usability, we plan to begin conducting experiments. These will examine users' interactions with noisy documents to correct errors both directly and indirectly, as well as to find ways to help users develop an understanding of the errors that have arisen and what actions can be taken to prevent them.

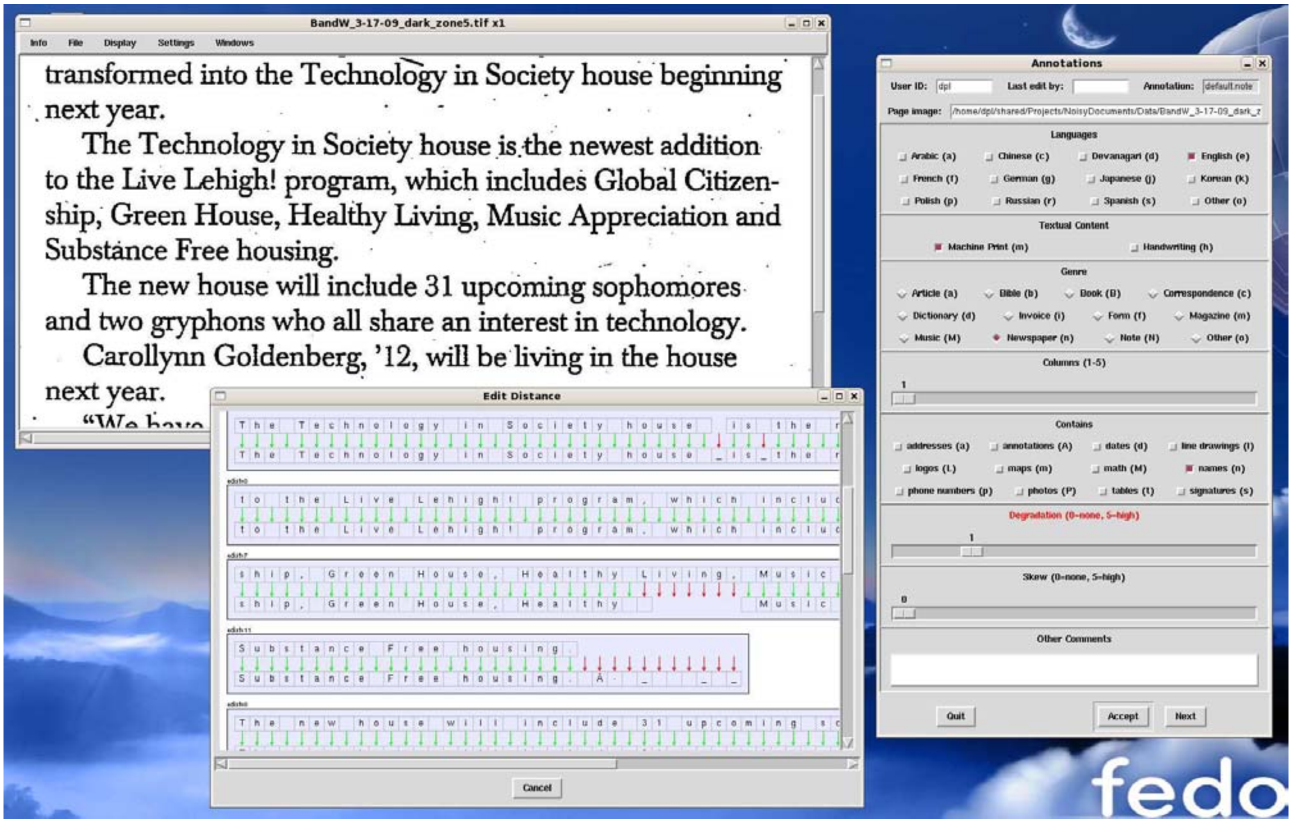

Figure 7: Screen snapshot of the alignments for a character-based error analysis using string edit distance.

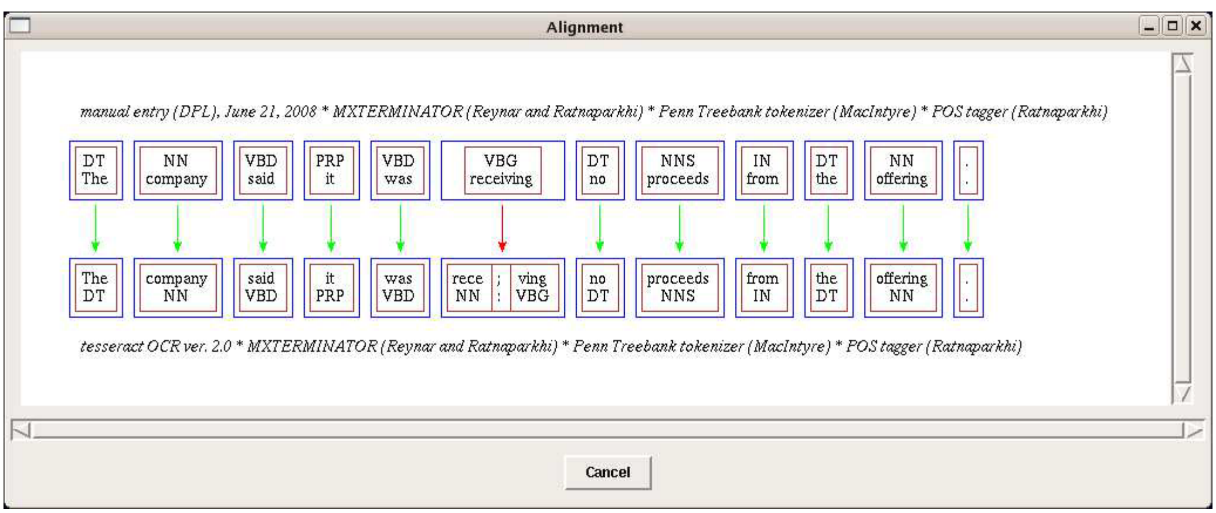

Figure 8: Screen snapshot of an alignment based on hierarchical edit distance for identifying errors across multiple processing stages (character recognition, tokenization, POS tagging).

7. ACKNOWLEDGMENTS

Daniel Lopresti acknowledges support from a DARPA IPTO grant administered by BBN Technologies. George Nagy acknowledges the support of the National Science Foundation under Grant #044114854.

8. REFERENCES

[1] S. Bird, D. Day, J. Garofolo, J. Henderson, C. Laprun, and M. Liberman. ATLAS: A flexible and extensible architecture for linguistic annotation. In *Proceedings of the Second International Conference on Language Resources and Evaluation*, pages 1699–1706, 2000.

[2] D. Day, J. Aberdeen, L. Hirschman, R. Kozierok, P. Robinson, and M. Vilain. Mixed-initiative development of language processing systems. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pages 348–355, Washington, DC, 1997.

[3] M. Droettboom, I. Fujinaga, K. MacMillan, G. S. Chouhury, T. DiLauro, M. Patton, and T. Anderson. Using the Gamera framework for the recognition of cultural heritage materials. In *Proceedings of the Second ACM/IEEE-CS Joint Conference on Digital libraries*, pages 11–17, Portland, Oregon, 2002.

[4] D. W. Embley, D. P. Lopresti, and G. Nagy. Notes on contemporary table recognition. In *Proceedings of the Seventh IAPR International Workshop on Document Analysis Systems*, pages 164–175, Nelson, New Zealand, February 2006.

[5] J. Esakov, D. P. Lopresti, and J. S. Sandberg. Classification and distribution of optical character recognition errors. In *Proceedings of Document Recognition I (IS&T/SPIE Electronic Imaging)*, volume 2181, pages 204–216, San Jose, CA, February 1994.

[6] J. Esakov, D. P. Lopresti, J. S. Sandberg, and J. Zhou. Issues in automatic OCR error classification. In *Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval*, pages 401–412, April 1994.

[7] 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.

[8] 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 O9.1–O9.9, San Jose, CA, January 2006.

[9] J. Kanai, T. A. Nartker, S. V. Rice, and G. Nagy. Performance metrics for document understanding systems. In *Proceedings of the Second International Conference on Document Analysis and Recognition*, pages 424–427, October 1993.

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

[11] 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.

[12] D. Lopresti. Performance evaluation for text processing of noisy inputs. In *Proceedings of the 20th Annual ACM Symposium on Applied Computing (Document Engineering Track)*, pages 759–763, Santa Fe, NM, March 2005.

[13] D. Lopresti. Measuring the impact of character recognition errors on downstream text analysis. In *Proceedings of Document Recognition and Retrieval XV (IS&T/SPIE Electronic Imaging)*, volume 6815, pages 0G.01–0G.11, San Jose, CA, January 2008.

[14] D. Lopresti. Optical character recognition errors and their effects on natural language processing. In *Proceedings of the Workshop on Analytics for Noisy Unstructured Text Data*, pages 9–16, Singapore, July 2008.

[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] G. Nagy. Document image analysis: Automated performance evaluation. In *Document Analysis Systems*, pages 137–156. World Scientific, Singapore, 1995.

[17] G. Nagy, S. Seth, and M. Viswanathan. A prototype document image analysis system for technical journals. *Computer*, 25(7):10–22, July 1992.

[18] D. J. Ross. DAFS: Document Attribute Format Specification. http://cool-palimpsest.stanford.edu/bytopic/imaging/std/dafsdrft.html.

[19] S. Strassel. Linguistic resources for Arabic handwriting recognition. In *Proceedings of the Second International Conference on Arabic Language Resources and Tools*, Cairo, Egypt, April 2009.

[20] L. Todoran, M. Worring, and A. W. M. Smeulders. Data groundtruth, complexity and evaluation measures for color document analysis. In D. Lopresti, J. Hu, and R. Kashi, editors, *Document Analysis Systems V*, volume 2423 of *Lecture Notes in Computer Science*, pages 643–649. Springer-Verlag, Berlin, Germany, 2002.

[21] D. Wilkinson and B. Huberman. Cooperation and quality in Wikipedia. In *Proceedings of the International Symposium on Wikis*, pages 157–164, Montréal, Canada, 2007.

[22] Y. Xu and G. Nagy. Prototype extraction and adaptive OCR. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(12):1280–1296, December 1999.

[23] B. A. Yanikoglu and L. Vincent. Pink Panther: A complete environment for ground-truthing and benchmarking document page segmentation. *Pattern Recognition*, 31(9):1191–1204, 1998.

[24] J. Zou. *Computer Assisted Visual InterActive Recognition*. PhD thesis, Rensselaer Polytechnic Institute, 2004.