Metadata for Structured Document Datasets

Henry F. Korth
Dept. of Comp. Sci. and Eng.
Lehigh University
Bethlehem, PA 18015 USA
hfk@lehigh.edu

Dezhao Song
Dept. of Comp. Sci. and Eng.
Lehigh University
Bethlehem, PA 18015 USA
des308@lehigh.edu

Jeff Heflin
Dept. of Comp. Sci. and Eng.
Lehigh University
Bethlehem, PA 18015 USA
heflin@cse.lehigh.edu

ABSTRACT

In order for a large dataset of documents to be usable by document analysts, the dataset must be searchable on document features and on the results of prior analytic work. This paper describes a work-in-progress to develop such a document repository. We describe the types of data we plan to maintain regarding both the documents themselves and analyses performed on those documents. By storing the provenance of all metadata pertaining to documents, the repository will allow researchers to determine dependency relationships among document analyses. Our ultimate goal is to enable geographically separated teams of researchers to collaborate in large document analysis efforts.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; I.7.5 [**Document and Text Processing**]: Document Capture—*Document Analysis*

1. INTRODUCTION

Researchers in document analysis have in common with those in many other fields the need to create and maintain repositories of test data and results in a way that maximizes the possibilities of remote collaboration. Although the Web enables sharing of this sort, it fails to provide several essential features for effective data sharing:

1. There is no standard way for those publishing datasets to describe them. Keywords, combined with search engines, provide a less-than-ideal solution, for a variety of reasons, including semantic ambiguity.
2. There is no standard way to search analysis results. Blog-like collections of free-form text suffer from problems of semantic ambiguity.
3. There is no standard way to show how analytic results depend upon each other. Published results need to be linked to the specific version of any documents used and any prior analytic data used as input. These links must be robust to the publication of updated results.

At Lehigh University, we are in the process of designing a repository with the long-term goal of creating a globally accessible resource for storage of document datasets and analytic results pertaining thereto. This work is part of the larger Document Analysis and Exploitation Project at Lehigh whose goals include:

- Handwriting recognition.
- Multilingual techniques in document analysis.
- Information extraction (intelligence gathering) from documents.
- Systems issues in the organization and management of large document collections.
- Data mining applied to large document collections.

We envision the document repository enabling analyses over sets of documents:

- Identification of document metadata: inference of information about the document such as authorship, authenticity, topic, etc.
- Analysis across multiple documents: finding connections among documents, searching a document collection efficiently, and, more generally, making the value of a document collection grow superlinearly with the number of documents.

Our goal in this short paper is to describe our goals and the basic structure of our approach.

2. FOUNDATION OF OUR APPROACH

Although the origins of database systems were in applications with relatively short records over simple data domains (payroll, airline reservations, banking, etc.), database systems have a long history of adapting to the demands of new applications. It is these adaptions that provide a basis for our repository design.

The earliest extensions to database systems were to support computer-aided design, work that dates from the early 1980s[11] and spurred work in object-oriented databases[13]. More recently, a variety of scientific applications have spurred extensions to database systems, including bioinformatics[5], and astronomy[9, 17]. The recent Claremont Report[1] identifies a variety of current issues in database research, notably the first of five factors bringing change to the field:

Breadth of excitement about Big Data. In recent years, the number of communities working with large volumes of data has grown considerably, to include not only traditional enterprise applications and Web search, but also "e-science" efforts (in astronomy, biology, earth science, etc.), digital entertainment, natural language processing, social network analysis, ...

Our project is thus part of a long tradition in the database research community of reaching out to new applications, applying known results where feasible, and inventing new concepts, algorithms, and systems when necessary.

3. METADATA STRUCTURE

Although the scanned documents of a dataset are stored directly in a collection of files, the data about these datasets, that is, the *metadata* needs to be structured in a way that enables effective search. For these reason, we have designed a document-description database for metadata. There are several elements of our design: document structure, structural elements, algorithms, analytic results, and version relationships. We describe these informally below and online[1] in an entity-relationship diagram.[2] In the future, we anticipate adding support for metadata not related to the content itself, such as copyright status. Such data, combined with the relatively sophisticated security model in most SQL implementations, can allow broad sharing of a document database without creating opportunities for inadvertent copyright violation.

3.1 Document Structure

Although documents are often thought of as having a hierarchical structure, there are multiple overlapping hierarchies based on both physical and semantic considerations. A document can be viewed physically as a set of *pages*, each of which contains a collection of *page regions*. A page region may be an instance of a variety of complex content types (such as image, graph, mathematical formula, table, etc.) or a simpler type such as a line of text. Although certain types may be supported as "first class" types, it must be possible for analysts to include custom types without altering the database design. Page regions may themselves be subdivided further into regions to arbitrary depth. This view of a document is derived from the physical structure and placement of information on a page.

Alternatively, a document can be viewed semantically as a set of chapters or sections, subdivided into paragraphs, figures, tables, etc. Paragraphs, of course, may span physical pages. Figures and tables may be placed physically outside the textual boundaries of their enclosing paragraph or section (indeed they could all be at the end of the document). Elements of the document structure might overlap. This would be unlikely in a well designed and properly scanned document, but the metadata design must be able to accommodate apparent overlaps even if they result from errors.

[1] `www.cse.lehigh.edu/~korth/DAE-ER.pdf`

[2] See a database text such as [16] for details on entity-relationship design.

3.2 Structural Elements

Each element of a document structure has an associated set of properties. We assign each element a unique identifier and content type. Every element has certain content-type-independent properties, including physical storage size, physical boundary within the containing element (if any), position in reading order, etc. Content-type specific data are represented using a generalization hierarchy.[3] Space does not allow us to enumerate content types here. We envision that the hierarchy of types would expand over time.

3.3 Algorithms

The determination of document structure and the analysis of structural elements depends upon the algorithms used. For this reason, we allow algorithms to be registered in the database along with associated descriptive information. While some users may wish to keep source code and other proprietary details outside the database, we require each algorithm to have a unique identifier so that it can be determined exactly which algorithm was used in an analysis.

3.4 Analytic Results

Analytic results arise from the application of a set of algorithms to an input data set. The analyses could be as simple as running an OCR algorithm, inserting ground-truth data, or something more esoteric. The input data set could be a single document, a set of documents, or a specific set of page or subelements thereof that might be extracted from possibly several documents. Our database design associates analytic results with the precise input set, thereby allowing the identification of those results that might be invalidated by future updates to the input data. We also record any algorithmic parameters used in computing the results, so as to facilitate their reproduction and verification. The results themselves may be values for previously defined properties of certain document elements (for example, typeface, style, language, and so on), but they may also be previously undefined properties.

3.5 New Content and Result Types

Our full design includes many types of document-element content and many types of analytic-result data. However, our list cannot ever be complete. As new types (represented as "other") are inserted, the usefulness of the type specifications will slowly fade. To minimize this, we plan to integrate ways not only to insert new types into the system, but also to allow the user community to specify how various sets of terminology relate to each other. Semantic Web [3] research is likely to be useful here. One focus of this research is integrating information from various sources that use heterogeneous vocabularies. The first step is to express each vocabularly formally as an ontology consisting of classes and properties. These ontologies can evolve, be extended, and be related to other ontologies via logical axioms. One potential issue is that of "knowledge acquisition," i.e., from where do the ontologies come. A possible answer to allow vocabularies to evolve like folksonimies (the informal vocabularies that arise from social bookmarking), but then use algorithms to extract formal ontologies from this information [15].

[3] A generalization hierarchy is a tree or DAG-type structure in the entity-relationship data model [16] that is similar in spirit to a class hierarchy

4. PROVENANCE

Provenance refers to the data used to generate a result and the sources of those data. This is a more general and complex notion than that used in typical revision-control systems. Revsion-control systems record who did what to data (typically during document editing or software development). They do not record to what items the user may have referred in doing the revision. Provenance, on the other hand, involves identifying exactly which data served as input to the process. To be useful, provenance data needs to be as specific as possible, ideally including only a minimal complete set of dependencies. The issues we face here are not unique. Indeed it has been argued that future needs of most applications "will require building provenance technology into all computer systems of any importance."[7]

4.1 Why Provenance is Hard

It is well known [6] that if we consider not only *positive* data provenance, but also permit negation or aggregation, then provenance is a quite challenging problem. As a simple (non-document) example, suppose we have a table holding employee-id, department, and salary. An "analysis" that computes the average salary of the marketing department depends not only the those records pertaining to the employees of the marketing department, but also on the absence of any records pertaining to the marketing department beyond those used in computing the result. In this example, the result would remain valid in a new version of the database in which new employees were added to departments other than marketing. However, moving an employee to or from marketing, as well as either inserting or deleting a marketing employee would invalidate the result. Thus changing data not even used in an analysis could influence the provenance of that analysis. In its full generality, this problem is equivalent to the well-studied and computationally challenging database problem of *view maintenance* [4, 10].

4.2 User Provenance

Another aspect of provence is the ultimate source of data. As examples, note that analytic results from a highly reputable lab may be valued more highly than those from a high-school science-fair project; results generated from an algorithm later found to have a bug may lose some or all of their value, and so on. Web-based systems, including search engines and recommender systems, employ heuristics to rank data. However, in the scientific community, researchers may prefer to know the exact derivation sequence of data on which their work relies. This permits each researcher to draw her own decision about data provenance.

4.3 Versions

To store data provenance in our database, we need to maintain virtually all versions of data and algorithms and store "depends-on" information relating results to data and to algorithms. We envision that users will contribute their own provenance-evaluation algorithms that traverse the provenance graph to determine the trustworthiness of results. By representing provenance data directly in the database, we can enable highly general provenance searches that, for example, allow us to find all studies that might be improved based on a newly published result, or those impacted by a newly discovered bug.

The concept of version that we need here is distinct from the familiar notion of versions identified by a timestamp or a version number. Those familiar notions are based on a model of continuous development and improvement, along with the possibility of rolling back to a prior point in time. In our framework, a document many have multiple versions based on the type of scanner used, the resolution used, the particular physical hard copy used, etc. Some analyses might use more than one version of an input document. Thus, unlike a true version-management system, versions here are both dependent and independent at the same time.

5. MANAGING GROWTH

The storage requirement for scanned pages varies widely depending on the compressibility of the data on the page, but is on the order of megabytes per page [12, 18]. Metadata can easily grow to be larger than the documents they describe. While a document element's location may be constrained succinctly by the use of a bounding box, ultimately, the identification of which pixels describe a specific feature may require a bit-map *mask*.[4] Thus, a single collection of masks for a given document may have a size approaching that of the document itself. If experiments are run with several algorithms, each using several distinct parameter settings, it is easy to imagine metadata of a volume that ultimately dwarfs that of the base database of scanned documents.

5.1 Indexing and Searching

Rather than developing our own indexing and searching algorithms, we shall rely on the underlying database system to handle those tasks. Data items too large for a database record are stored either as files referenced in the database by a file name or are stored using the large-object feature of the database system (CLOBs and BLOBs). Such data items are indexed by their unique identifier and other properties, which are stored in the database. Using this approach, we can rely on the database system for automatic optimization of complex queries.

5.2 Distributing Data and the "Cloud"

Database systems include features to manage distributed and replicated data. It would not be necessary for our document repository to be hosted on a single machine, nor even a single site. The challenges of distribution come more from the human side than the technical side: the difficulty of maintaining effective administration and control of the system across independent institutions. These considerations lead us to believe that over the medium-to-long term, when our repository might be best shared among several sites, it may be worthwhile to take advantage of the emerging "cloud" services in which a third party hosts data and perhaps also provides the computing power to process those data. A detailed overview of the cloud computing model, its potential and its risks appears in [2].

For our repository, one of the great advantages to the cloud model is that it allows for a relatively simple distribution of storage and computing costs across the user community

[4]We note that since masks typically identify the binary condition of being inside or outside of a closed shape, they may be more amenable to compression techniques than gray-scale or color documents with complex features.

while providing commercial-grade uptime guarantees. A concern is overall cost, which remains uncertain as the market develops and evolves. We have a study currently underway using the present price lists of major cloud vendors (including Microsoft Azure, Amazon S3 and EC2, and Google) to identify when it is advantageous to use cloud storage version in-house servers. A prior study targeting astronomical data appears in [8]. Current work generally suggests a hybrid approach to the cloud in which high-traffic data are stored in-house, but even this "rule of thumb" depends on how the data are used in a computation. Since some providers do not charge for data accesses made by their own compute servers, an analysis that reads a high volume of data but returns a low volume of results may be cost-effective to run entirely in the cloud.

A lesser concern at present that may emerge later is the *eventual consistency* model of the cloud that allows users unknowingly to read slightly out-of-date data. The eventual-consistency model permits cloud providers to provide availability via replication without having to run a high-overhead protocol to ensure global serializability.[5] True consistency can be ensured, but at a price[14].

5.3 A Moving Target

We plan initially to build our system on an in-house server, duplicate some data in the cloud, and run experiments to validate the cost models we are developing. In doing so, we face a moving target, since cloud providers may change not only the dollar-cost values in their current pricing plans, but may change their entire pricing structure. The growing acceptance of cloud computing in commercial enterprises offers hope of both lower costs and stable pricing structures in the near term.

6. CONCLUSIONS

The document repository we have described here is in the design stage. We have explained some of the issues we face in creating a repository that contributes directly to collaborative document analysis research by storing not only documents and analytic results, but how they depend on each other. We have also discussed the challenges in making such a rich repository sharable in a cloud environment.

7. ACKNOWLEDGMENTS

This work is supported in part by DARPA via a subcontract from BBN.

8. REFERENCES

[1] R. Agrawal, A. Ailamaki, P. A. Bernstein, E. A. Brewer, M. J. Carey, S. Chaudhuri, A. Doan, D. Florescu, M. J. Franklin, H. Garcia-Molina, J. Gehrke, L. Gruenwald, L. M. Haas, A. Y. Halevy, J. M. Hellerstein, Y. E. Ioannidis, H. F. Korth, D. Kossmann, S. Madden, R. Magoulas, B. C. Ooi, T. O'Reilly, R. Ramakrishnan, S. Sarawagi, M. Stonebraker, A. S. Szalay, and G. Weikum. The Claremont report on database research. *Commun. ACM*, 52(6):56–65, 2009.

[5]the guarantee that a concurrent accesses are equivalent to a non-concurrent, serial sequence of accesses.

[2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. Above the clouds: A Berkeley view of cloud computing. Technical Report UCB/EECS-2009-28, EECS Department, University of California, Berkeley, Feb 2009.

[3] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, May 2001.

[4] J. A. Blakeley, N. Coburn, and P.-Å. Larson. Updating derived relations: Detecting irrelevant and autonomously computable updates. *ACM Trans. Database Syst.*, 14(3):369–400, 1989.

[5] P. Buneman, S. B. Davidson, K. Hart, G. C. Overton, and L. Wong. A data transformation system for biological data sources. In U. Dayal, P. M. D. Gray, and S. Nishio, editors, *VLDB'95, Proceedings of 21th Int'l Conf. on Very Large Data Bases, September 11-15, 1995, Zurich, Switzerland*, pages 158–169. Morgan Kaufmann, 1995.

[6] P. Buneman, S. Khanna, and W. C. Tan. Data provenance: Some basic issues. In S. Kapoor and S. Prasad, editors, *FSTTCS*, volume 1974 of *Lecture Notes in Comp. Sci.*, pages 87–93. Springer, 2000.

[7] J. Cheney, S. Chong, N. Foster, M. I. Seltzer, and S. Vansummeren. Provenance: a future history. In S. Arora and G. T. Leavens, editors, *OOPSLA Companion*, pages 957–964. ACM, 2009.

[8] E. Deelman, G. Singh, M. Livny, G. B. Berriman, and J. Good. The cost of doing science on the cloud: the montage example. In *SC*, page 50. IEEE/ACM, 2008.

[9] J. Gray and A. S. Szalay. Where the rubber meets the sky: Bridging the gap between databases and science. *CoRR*, abs/cs/0502011, 2005.

[10] A. Gupta and I. S. Mumick. Maintenance of materialized views: Problems, techniques, and applications. *IEEE Data Eng. Bull.*, 18(2):3–18, 1995.

[11] R. L. Haskin and R. A. Lorie. On extending the functions of a relational database system. In M. Schkolnick, editor, *SIGMOD Conference*, pages 207–212. ACM Press, 1982.

[12] S. J. and K. H. Mediateam document database II, a CD-ROM collection of document images, Univ. of Oulu, Finland, 1999.

[13] W. Kim. Object-oriented databases: Definition and research directions. *IEEE Trans. Knowl. Data Eng.*, 2(3):327–341, 1990.

[14] T. Kraska, M. Hentschel, G. Alonso, and D. Kossmann. Consistency rationing in the cloud: Pay only when it matters. *PVLDB*, 2(1):253–264, 2009.

[15] P. Mika. Ontologies are us: A unified model of social networks and semantics. *J. Web Semantics*, 5(1):5–15, 2007.

[16] A. Silberschatz, H. F. Korth, and S. Sudarshan. *Database System Concepts, 6th Edition*. McGraw-Hill Book Company, 2011.

[17] A. S. Szalay. The sloan digital sky survey and beyond. *SIGMOD Record*, 37(2):61–66, 2008.

[18] L. Todoran, M. Worring, and A. W. M. Smeulders. The UvA color document datatset. *International Journal of Document Analysis and Recognition*, 7(4):228–240, 2005.