2000

An Architecture for Text Management in Organizations

Pankaj
Southern Illinois University Carbondale, pankaj@siu.edu

An Architecture for Text[1] Management in Organizations

Pankaj, Management Department, COBA, SIUC, Pankaj@siu.edu

Abstract

Most of the available on-line data/information in organizations that also is efficiently organized for storage and retrieval is numerical in nature. Along with numerical data/information, organizations also use a substantial amount of text-based data/information. With the advent of ecommerce and Intranets, more and more text-based information is now available on-line. While textual information can be a rich source of information to organizations, there are several issues regarding the efficient storage and retrieval of text-based data/information. This paper examines the issues with text storage and retrieval and proposes a high level architectural solution to overcome some issues. Many of the features in the proposed architecture are already implemented in various software solutions available today but in a fragmented fashion. The architecture emphasizes open standards to enable seamless sharing of text-based data/information in a networked environment.

Keywords: text management systems, information storage and retrieval, data management, document management.

Introduction

The popularity of the web has brought more and more text based information online. Estimates [1] say that 80% of the online information is textual while 20% is numerical. Among the text that is available online (Hypertext Markup Language (HTML) documents, plain texts etc.) about 90% is unstructured and only 10% is structured [2]. The amount of text stored online is constantly increasing due to the increasing popularity of Internet and Intranet as the medium of information exchange. Most of this text is unstructured. The decentralization of the web content management in organizations to individuals and departments means that there more unstructured text being put online, on a daily basis. Online text can serve as a valuable source of information, though collecting such information can often be tedious and time consuming. This paper is an example of the richness of the online text, as it primarily relies it as source of information.

Most formal organizational communications (internal communications and communications with external entities like partners, suppliers and customers) are in writing. Details from other modes of communication like conversations are also transcribed to text, e.g. transcripts of the conversations with the customers in a call center and minutes of the meetings. In most of the situations the information is stored in the text in an unstructured form (formats like HTML can be used to provide some structure to the information, they are primarily used to structure the presentation rather than structure the content). This text can provide valuable information. An analysis of the documents for meetings related to a topic can reveal a direction of thinking that all the participants have taken but which may not be apparent to the participants (a form of data mining). Similarly an analysis of the customer transcripts across various workstations in a call center may point to a product feature(s) that the customers like or dislike. There is often a lot of information to be gleaned from the on-line text/documents that can provide substantial benefits to an organization in terms of identifying market trends, solving problems etc. There is need for mechanisms that enable text to be stored and retrieved in an organized and efficient manner. This paper examines various issues in this area and proposes an application architecture that may be used to make the storage and retrieval of the on-line text, more efficient and organized.

The paper is organized into four sections. The second section examines some issues (and solutions where possible using existing technologies) associated with text storage, retrieval and management. The third section examines requirements for a text management system that are important for an organization operating in a global networked environment of Internet. It proposes an architecture for an integrated text management system for an enterprise. This architecture may be used as a blueprint for text management systems. The last section concludes the topic and talks about future directions in this area.

Key Issues in Text Storage and Retrieval

Text has several characteristics that distinguish it from numerical data, which is mostly stored with some

[1] In this paper though we distinguish between text and documents but we use them interchangeably. Documents may be treated as objects created in an OLE [object linking and embedding] compliant desktop environment. They may be composed of text, graphics and other objects in addition to pure text. Documents may imply formal written communication. While no statistics are presented here, text is the predominant part of most of the documents. Most of the documents may not embed objects and those that do embed objects do not go beyond simple graphics. So the two may be used interchangeably.

structure. This section examines some of the issues one faces when storing text. The issues can be divided into the broad categories of syntax and semantics. But it is expected that most of the issues would fall into both the categories, as the syntax will often influence the semantics and vice versa. Where possible, an issue may be classified into either category to separate a purely technical issue from a semantic/substantive issue.

Representation Formats

The issue of representation format may be treated as a syntax issue. There are a variety of different representation formats for text starting from the machine level to the application level. This is because that no single representation format is efficient for all purposes, e.g. the PDF format of Acrobat offers a much smaller sized document as compared to an MS Office document, which is useful for transmission on the Internet. But multiplicity of the formats means extra work in conversion between these representations and needs extra S/W (with their inherent incompatibilities) making sharing of text from different sources somewhat difficult. Running multiple S/W also increases the Total Cost of Ownership (TCO) of the computing equipment for an organization.

Representation formats may be distinguished at three levels. At the first or machine level the bit/byte level encoding may differ across machines. Having only two primary encoding schemes of ASCII and EBCDIC does not cause any big problems. At the operating system (OS) level each character is the encoded using a series of bytes. Thus a series of bytes is interpreted differently depending upon the language(s) supported by the operating system and also sometimes the brand/type of the operating system. The coming of more and more non-English users on the Internet and the availability of textual information in languages other than English make this an important issue [10]. E.g. a plain text document or an email typed in an operating system with Chinese language support will turn up as garbage on an operating system with no Chinese Support. In addition there are multiple encoding schemes available for each language. Support for multiple languages at the operating system level is needed (this is available in browsers but many times requires the user to manually change the encoding scheme). At the application level various applications running on the OS have their own representation formats. E.g. the various office suites all have different file formats.

The multiplicity in representations makes increases the complexity of documents sharing and requires multitude of S/W (e.g. plug-ins in browsers). Resolving multiplicity of representations may alleviate many of the problems and issues in sharing text. Currently sharing of text means that conversions need to be done at various levels. Conversion may happen from EBCDIC to ASCII, then from one application to another format (a significant move in this direction has already been made in the Microsoft Office 2000 suite, which supports HTML as a storage format), and then encoding scheme for the text is fixed. This causes overheads and increases TCO in ensuring universal accessibility for universally available text.

Structure

The issue of structure spans both syntax and semantics. Most of the processes that generate text create text that is not structured. Often only well designed processes that follow standards for text layout create text with structure and preserve it. This is mainly done through defined fields/headings to organize the text. E.g. to record the minutes of meeting one may use standard heading/section/field to record time/date of the meeting, attendees, list of issues discussed, resolutions, etc. But most of the text that is generated does not follow any structure apart from the most rudimentary form of topics and sub-topics. Further more the structure embedded in the document is often unique to the writer, process and the organization. Even with a structure in the document, different people may often interpret the text differently. A classic example is selection of the keywords for the central idea of the text (this may be used in searching). The difficulty is amply visible in the searches on the popular search engines. Often the same document is returned multiple times and if the search is not well formed then a major portion of the results from the search may not even be relevant.

The issue is also related to the representation. Most often the structure in the document may not be embedded into the representation format. This may especially be the case when either the S/W does not offer the capability or the capability though offered, is not used in the intended manner. E.g. on one extreme a simple text document created in notepad may be given a structure at a semantic level without any structure at a representation level while a text document created in MS word can be provided some structure using the style like ‘normal’, ‘heading’ etc. But most users use styles is for their presentation effects rather than their structuring capabilities. A proof of the previous statement is the paucity of users in a large IS project setting employing about 300 IS professionals including the author, who could generate automatic table of contents for a document using the styles in MS Word. So the structure in the text is often at the semantic level and not at the representation level. This makes it difficult for machines to glean meaning out of the text. Making sense of the structure that is at the semantic level needs Natural Language Processing (NLP), where the power of the available parsers is severely limited. When there is structure in the text at the representation level, often the quality of the

text that is fitted into structure may make the interpretation vague and complicated.

Another relevant factor is a significant amount of text available on-line as images of the paper texts. These images are being created in an ever increasing number as more and more organizations transfer the old documents from the paper medium to the computer storage. Here structure is primarily present at the semantic level in the captured images of the paper text. While Intelligent Character Recognition (ICR), Optical Character Recognition (OCR) and Pattern Recognition technologies can be employed to make the text in the image readable, imposition of the structure needs capabilities in NLP, which currently has limited capabilities to accomplish the task.

An opportunity for providing structure to the text at both the semantic and the representation level is offered by Standard Generalized Markup Language (SGML). SGML uses tags to provide structure to the text as done in HTML (HTML is a subset of SGML). SGML has been used extensively in publishing industry, but the SGML's inherent complexity makes it unfit for its common use in organizations. XML or eXtensible Markup Language (XML), a derivative of the SGML, offers promise. XML offers a viable alternative for SGML. It has 80% of the capabilities of the SGML while being only 20% as complex as SGML [3]. Although like HTML, XML can be used to define custom tags (XML may be used to define HTML itself) as compared to fixed tags of HTML and XML addresses the content as compared to HTML that addresses only presentation.

Providing structure to the text provides increased opportunities for use of the text by computers. Parsers can read a document with structure and extract relevant data. This extracted textual data can be stored in the databases, making searches and retrieval more efficient. The document/text may then be published using the data extracted from the database. Presentation of the document may also be manipulated (similar to the concept of Style sheets in XML). Applications like voice based sharing of textual information would also become easier and more effective.

Classification and Indexing

Classification and indexing is most important in any storage and retrieval. In the networked environment the relevant text could be residing on any of the accessible computers. The search for relevant text requires that the text be appropriately indexed and classified and that this classification and indexing be available to the search engines and algorithms. As the content management becomes more and more decentralized, who classifies and how becomes increasingly important. Objectivity is needed and issues here can be looked at in two parts.

The first is the need for a classification scheme that is universal and robust. A search in the brick and mortar library uses the classification and indexing system that is standardized and universal. Using this same classification system searches can be conducted in libraries anywhere in the world. In the online world there are no universal/ standard classification schemes. The classification and indexing schemes by portals like Yahoo have gained wide popularity and acceptance, but they are not standards. The problem is particularly compounded on an organizational intranet where a lack of proper classification/indexing scheme can hamper the use of the relevant textual information. Unlike the commercial efforts like Yahoo where there are substantial tangible (visible monetary) benefits of classifying and indexing information, such benefits are more intangible in an intranet. This coupled with the lack of standards for classification makes intra-company classification schemes a somewhat neglected area. The result is a classification and indexing that may vary across organizations making inter-organizations sharing complex.

The second problem is the process of classifying documents into categories of classification schema. The category to which the text should belong and keys for index is a matter of interpretation, which is best done by expert human beings (perhaps with an aptitude for taxonomy and linguistics). Classification and keywords may eventually turn out to be subjective/ad-hoc. Problems may arise due to the structure of categories like too many categories, too few categories and overlapping categories. Most problems may arise due to the interpretation of the text and the consequent category assignment and keyword selection. Having a set of rules of classification has its own problems. These rules cannot be rigid may need to be revised often. A loose set of rules may be needed when people are doing the assignment. It makes the process simpler [2] and more flexible but at the same time more subjective. Alternatively a more extensive set of rules for assignment may be defined that would increase the complexity of the task and restrict the task to real experts (which maybe in short supply). This may result in a much more accurate assignment or classification [6] and enable the use of computers.

Computers may also be used to come up with classification categories and rules based on a sample of documents [4]. These can be updated as more documents are processed. The problem here is again of the semantic interpretation of the text and limited capabilities of NLP. E.g. the context in which a word appears may determine its meaning; duty may be used in the sense of taxes as in customs duty, as an obligation and as a social responsibility to name a few instances. The use will determine the meaning of duty and the classification of the text containing 'duty' as the central idea [4]. In the same vane classification of the texts that are the images of

paper text would invariably need human experts till the ICR, OCR and NLP technologies become more mature.

Storage schemas

Numerical data often resides in some kind of database. The database offers multitude of advantages like scalable storage with fast and efficient retrieval. The storage schema for numerical data in databases always has an underlying semantic schema. Databases akin to that for numerical data do not exist for text. Semantic schemas for text within the document and across documents do not exist.

Most of the text currently exists as collection of files in several computers within the organization. A formal storage mechanism akin to a database does not exist in most organizations. Though many organizations have implemented document management systems to manage and store text, the system's use is mostly restricted to the text from processes that are supported in the document management systems. Documents of relevance that do not enter the document management system still suffer from the problems of ad-hoc organization and storage. A desirable solution for text storage in a networked environment would be a distributed text database with a distributed index or directory. The documents would be stored as individual files as it happens currently and would be accessed through the index or the directory. Storage of the images of text/documents may be done using Binary Large Objects (BLOBS), Object Oriented Databases (OODB), PDF files etc., which would be accessible through the document index/directory.

All the documents created and owned by a particular user would be maintained in the database owned by the user. The user would have the necessary authority and access to applications to completely manage this local database. The user may share his/her documents by making the local database part of a distributed database. Such schemes/facilities are already available through document management systems like Lotus Notes though interoperability between various systems is an issue.

Document Creation and Document Management Processes

Text/documents are created by users at all levels and with varying levels of computer skills. Given the vast subject coverage of the text, rules that structure the text and enable proper classification, become too complex and detailed to be handled by everyone with ease. In the absence of machine-based support for the tasks, most of the users are left to their own discretion to create a structure and do classification. Result is documents with no standard structure, heterogeneous classifications and other related problems.

The document management processes also lack maturity and rigor. The formal configuration management practices like baselines, marking revisions etc. are almost absent in a normal working scenario. Though these processes are a stringent requirement in a project situation, they are not even followed in entirety in the project situations also (experience of the author). As more and more text is created and put online, archival becomes an important issue. This is an area where the computers have helped enormously by transferring contents from old paper documents to the computer storage like optical disks, DVDs, CD ROMs etc. While archiving from the paper to the computer media has been successful, the archival process for online text is not so organized. It is not an uncommon sight to see people sitting with a bunch of floppy diskettes in an effort to locate an old document. There are immediate tangible benefits of archiving the paper text like increase in the life of the paper document etc. For the text that is on-line many times there are no immediate tangible benefits of archiving except freeing up disk space the cost of which is reducing day by day.

Documents creation and management is an activity performed by virtually everyone in the organization. It is therefore a requirement to inculcate good document creation and management practices. Since the same processes/practices are to be followed by different personnel with different level of skills, the processes have to be optimized for different users. Personnel training in these processes/practices is required. A standard set of document creation and management rules/practices can be enforced by embedding them into the tools used for document creation and management. A simple macro within an MS Word can force the author to fill in the details of the author and include it as part of the document, before the user can proceed to the creation of the document. Similarly the revision feature may be turned on so that the revisions may be marked when changes are made to the document. More than anything else, providing training to personnel on how to properly structure the document using the capabilities of the computer S/W being used by them to create the document, and establishing and communicating the guidelines for document management will provide the most immediate benefits. While the capabilities of the computers in the text-processing area are still evolving, humans can provide higher quality input data that augment the limited capabilities of the computers.

Towards an Architecture for Text Management: Requirements and Architecture

The issues discussed give an idea of desirable features/requirements for a system to manage texts. The features may be divided into system level, application level and user levels.

System Requirements

The system requirements comprises of four major technical requirements. The first one arises out of the distributed computing environment in which a text management application would be running. A directory service for resources (similar to the Novell Directory Services [8] or Microsoft Active Directory Services [9]) would be needed as a base on which the text management application would run. A resource directory service identifies all the resources/entities on the computer network. This will include users, computers, storage systems, routers etc. The directory service will aid in advertising the text resources on the computer network; provide security and access control; provide a level of abstraction to the programs, applications and users; and aid in other management functions. This directory service will also interface with directory services outside the organization and provide paths to text resources that have been made accessible by the external organizations. The distinct logon identity for the network provided by the resource directory can function as author and owner name for documents and control rights to various text resources. While the above functionality for text management is present in the current systems, it exists in a fragmented fashion and is not integrated into in the enterprise architecture. Also the functionality at the resource management level is not integrated with the applications. Integration of the resources management functionality can be done using resource directory service.

The second system requirement is of multilingual representation at the OS level. The OS should be able to support multiple languages at the level of the plain text. For each language supported there may be a need to support multiple encoding-schemes. This may imply some changes in the architecture of the operating systems. The best recourse in a short or immediate term may be a system level utility that runs on the top of the operating system that supports language other than the native operating system languages and automatically switches between languages. Several such utilities like NJWin (Chinese), Thai (ThaiMaster) etc. exist but support one language. Currently browsers support multilingual representation with multiple encoding. But the browsers are limited in functionality and don't support functions like editing and creation of text etc. The long-term solution may be implemented at the OS level as it may be may be the efficient way to handle the multilingual representation.

The third system level requirement comes from the need for common representation format at the application level. A desired solution may be to design a basic underlying representation, which would be at the core of all the applications. Individual applications can then build up extensions to this base representation to accommodate the strengths of the application. The presence of a universal core representation format would ensure that the text can be read by any application without loss of critical information and need for conversion and use of multitude of S/W. An analogue of the core representation format with extensions can be taken from JAVA. JAVA can be used with the native classes on any OS and extensions to JAVA classes could be made for specific OS (done by Microsoft and basis of dispute between Microsoft and Sun). Such core would assure universal representation and use of specific strengths of different applications. Application formats may be automatically split into a core sharable representation and extensions that may be application, OS, organization or process specific (here XML may be a good choice).

If a common representation format is infeasible then at least is a robust interchange format should be targeted. E.g. Open Database Connectivity (ODBC)[2] for databases Currently rich document format (RTF) provides portability between applications, but storage in RTF format has to be done explicitly and at times leads to some loss of information present in the original format. (XML can be used here [3] [5]).

The fourth requirement for a text management system is the storage schema that matches the capabilities of databases. For structured texts a traditional database may be used. Object oriented database (OODB) may be used for collection of unstructured text. But the complexity and overheads of running an OODB system on each desktop would complicate the simple scheme of things, which exist now. A distributed database schema discussed earlier may be used. This schema would need a subject oriented classification scheme and may be implemented as a distributed *document directory service* (DDS) working at the operating system level. The common structure (categories etc.) for this directory would be derived from a central server. The documents may be classified in the appropriate category at the time of creation of the document on the local directory. This classification scheme may be also serve as an extension current classification scheme on the OS (.exe, .pdf). Current OS classification is based on the application that creates the files. Classification of text may be done by the user and checked by a document directory application. The index/directory on each local computer may be consolidated at the department and enterprise levels to provide a consolidated directory for all enterprise documents. The searches may then be done based on the classification, defined indices and other fields in the structured documents. Management application suites (also available today) for the DDS can provide the management functionality index management etc.

[2] In ODBC the data from any ODBC compliant database can be read by another ODBC compliant database through the use ODBC connections (though drivers are needed for each of the database).

Application and User Requirements

Since the objective here is to propose an architecture, a comprehensive discussion on the application and user requirements is not presented here except for the emphasis on structure in the text. In general application and user requirements would relate to provision of text management services to the end users. The application and user requirements are discussed together they both effect each other. The application has to provide features that the user wants or needs. Not only should the application provide the features that the user wants, it may also provide features that the users may not perceive to be needs or likes, but are still beneficial. These features will ensure quality and standards. They will also move the user towards more mature business processes by incorporating best practices into the application. The conjecture here is that as computers and the work processes become more and more entwined, the desired process standards may be enforced using the S/W applications being employed by the users. E.g. most call center applications do not allow the call center representative to close the call record till the time he/she makes and entry into the call record about the customer disposition. In text creation that happens as part of a structured process with well-defined standards and guidelines, organizations can control text creation through forms and templates that have embedded structure. These forms and templates can be incorporated into text processing applications. Deviations from these forms and templates would be minimized. But not all documents are created as part of the standard processes and for such documents like contracts, letters etc.; standard templates that are more generic in nature may be defined and used. Features for forms and templates are already available in most of the word processing applications but their use is not common. As mentioned earlier the users need to be trained in the art of creating structured documents using the application capabilities. Also since not all users can be expected to create structured documents by following the standards and guidelines, text should be validated through human and computer experts.

Thus from a user and application perspective a text management system should provide mechanisms that are oriented towards validating documents from a structural perspective. Such mechanisms may also be implemented through various application modules (just like the help icon in MS Word which comes and says “it seems like you are typing a letter, would you like help”). A desirable class of applications would be applications with NLP capabilities that can be used in function such as translations, text-mining etc. NLP can also ease pressures on the document structuring and automate most of the tasks related to text management.

An Architecture for Text Management

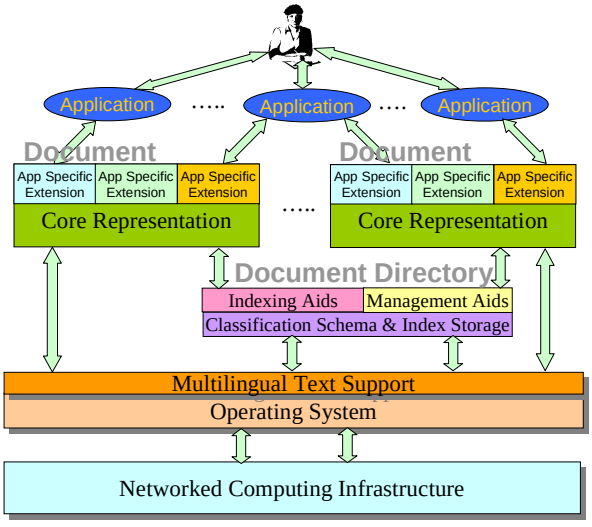

Figure 1: Architecture for Text Management

Figure 1: Architecture for Text Management shows the proposed architecture as a block diagram. A discussion of the available technologies available for each bock has been explored in some detail in the earlier discussions. We further look at an available technology for classifying documents called Northern Light [7], which has not been discussed earlier. Northern Light has the capabilities to read documents using text-mining technologies and classify them. It uses text/data mining technology to organize the information retrieved from text/documents into "folders" that have been created by librarians. There are four different types of folders (subject, document type, source, and language), approximately 20,000 broad hierarchical terms, and 200,000 to 300,000 additional terms. Individuals have created the index but the computer indexes the articles. Changes are made to the index itself when the computer system rejects an article that cannot be indexed. A person may then look at the article and makes a determination about whether or not to add new indexing terms. Northern Light is one of the examples of many text-processing computer applications that exist today among others mentioned. These computer applications lack integration into an overall scheme of enterprise level text management system. The architecture presented above may be used as a base to develop systems operating at enterprise level.

Conclusion

More and more text is becoming available online (at times exclusively) for reasons like paperless office, processes efficiency by cutting down on paper flow, making information available on-line in a universal fashion and so on. Not only has more information come on-line but the amount of text available has also increased substantially in amount (people often talk of the

information overload in the Internet era). The discussion here of the issues related to the text management point to some important considerations that need focus if organizations are to take full benefit of the textual data/information existing in the organizations today. The high level architecture/features presented here provides a direction for solutions to the issues.

Further work is needed in drawing up comprehensive details of the functionality for each of the components. A solution integrated from the existing S/W applications that provide the needed functionality (may be not the complete functionality) would be a good place to start the development of such a system. It is hypothesized that as technologies progress and more text comes online, the philosophy and the requirements for the management of text will develop further to become a topic of importance akin to storage of numerical data today. Also the various initiatives in the area of XML, natural language processing and text mining will provide the needed standards based technologies.

References

1. "Presentation by IBM Technology Watch", http://www.synthema.it/tewat/demo/pres/ntwprese.htm, (Current Apr 15, 2000).

2. Zorn Peggy, Emanoil Mary, Marshall Lucy and Panek Mary, "Meets the Web", *Online*, September/October 1999, pp. 17-28.

3. Dyck Timothy, "XML unleashes data, Cover Story: Red-hot technology builds bridges between enterprise apps by letting data move easily among them, regardless of original format", *PC Week Labs* November 22, 1999 9:00 AM ET, (http://www.zdnet.com/pcweek/stories/news/0,4153,2396831,00.html), (Current Apr 15, 2000).

4. Knight Kevin, "Mining Online Text", *Communications of the ACM*, Vol.42, No.11, Nov 1999, pp.58-61.

5. Rupley, Sebastian, "XML Spreads Out", *PC Week*, March 31, 1999 (http://www.zdnet.com/devhead/stories/articles/0,4413,2234894,00.html), (Current Apr 15, 2000).

6. Dagan Ido, "Automation of Information Access Tasks: Technological Trends and Opportunities", *Online*, May/June 1999, pp. 75-78.

7. "What we're all about " http://www.northernlight.com (Current Apr 15, 2000).

8. "NDS Corporate edition", http://www.novell.com, (Current Apr 15, 2000).

9. "Windows 2000 home page", http://www.microsoft.com, (Current Apr 15, 2000).

10. Hershman, Tania, "Tech Topples Tower of Babel", *Wired News*, 3:00 a.m. 11.Feb.2000 PST, http://www.wired.com/news/print/0,1294,34254,00.html, (Current Apr 15, 2000).