Winter 2006

Template-Based Metadata Extraction for Heterogeneous Collection

Jienfeng Tang
Old Dominion University

TEMPLATE-BASED METADATA EXTRACTION FOR HETEROGENEOUS COLLECTION

by

Jianfeng Tang
B.E., July 1995, Beijing University of Aeronautics and Astronautics, China
M.S., July 1998, Institute of Computing Technology, China

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirement for the Degree of

DOCTOR OF PHILOSOPHY

COMPUTER SCIENCE

OLD DOMINION UNIVERSITY
December 2006

Approved b

Kurt Maly (Co-Director)

Mohammed Zubair (Co-Director)

Steven Zeil (Co-Director)

Frank C. Thames (Member)

C. Michael Overstreet (Member)

Ravi Mukkamala (Member)

ABSTRACT

TEMPLATE-BASED METADATA EXTRACTION FOR HETEROGENEOUS COLLECTION

Jianfeng Tang
Old Dominion University, 2006
Co-Directors of Advisory Committee: Dr. Kurt Maly
Dr. Mohammad Zubair
Dr. Steven Zeil

With the growth of the Internet and related tools, there has been a rapid growth of online resources. In particular, by using high-quality OCR (Optical Character Recognition) tools it has become easy to convert an existing corpus into digital form and make it available online. However, a number of organizations have legacy collections that lack metadata. The lack of metadata hampers not only the discovery and dispersion of these collections over the Web, but also their interoperability with other collections. Unfortunately, manual metadata creation is expensive and time-consuming for a large collection, and most existing automated metadata extraction approaches have focused on specific domains and homogeneous collections.

Developing an approach to extract metadata automatically from a large heterogeneous legacy collection poses a number of challenges. In particular, the following issues need to be addressed:

- Heterogeneity, i.e. how to achieve a high accuracy for a heterogeneous collection;
- Scaling, i.e. how to apply an automated metadata extraction approach to a very large collection;

- Evolution, i.e. how to process new documents added to a collection over time;
- Adaptability, i.e. how to apply an approach to a new document collection;
- Complexity, i.e. how many document features can be handled, and how complex the features should be.

In this dissertation, we propose a template-based metadata extraction approach to address these issues. The key idea of addressing the heterogeneity is to classify documents into equivalent groups so that each document group contains similar documents only. Next, for each document group we create a template that contains a set of rules to instruct a template engine how to extract metadata from documents in the group. Templates are written in an XML-based language and kept in separate files. Our approach of decoupling rules from programming codes and representing them in a XML format is easy to adapt to another collection with documents in different styles.

We developed our test bed by downloading about 10,000 documents from DTIC (Defense Technical Information Center) document collection that consists of scanned versions of documents in PDF (Portable Document Format) format. We have evaluated our approach on the test bed consisting of documents from DTIC collection, and our results are encouraging. We have also demonstrated how the extracted metadata can be utilized to integrate our test bed with an interoperable digital library framework based on OAI (Open Archives Initiative).

ACKNOWLEDGMENTS

This dissertation would not have been possible without the support and encouragement of my committee members and many other people. My sincere appreciation and gratitude goes to my advisors, Dr. Kurt Maly, Dr. Mohammed Zubair, and Dr. Steven Zeil for their support, encouragement, and guidance through the entire research. I would also thank them for their time and patience in proofreading of hundreds of thesis drafts. I thank Dr. Frank Thames, Dr. Michael Overstreet, and Dr. Ravi Mukkamala, the members of my committee, for their thorough review of this dissertation and for their valuable feedback.

I am grateful to the faculty, staff, and colleagues at the computer science department of Old Dominion University for their help. I am extremely grateful to Phyllis Woods for her kindness. I also would like to thank Ali Azhar, Naveen Kumar Ratkal, Paul Flynn, Li Zhou, and other members of the Digital Library Research Group at Old Dominion University for their helpful discussions and contributions to this research project.

I cannot end without thanking my family for their encouragement and love. Many thanks are due to my wife Yan Lin for her understanding and support. I would like to give special thanks to my son Andy. His smiles have helped me go through the most difficulty times and finish this dissertation. It is to them that I dedicate this work.

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

LIST OF TABLES

LIST OF TABLES (continued)

LIST OF FIGURES

LIST OF FIGURES (continued)

LIST OF FIGURES (continued)

LIST OF FIGURES (continued)

CHAPTER I

INTRODUCTION

1.1. Motivation

With the growth of the Internet and related tools, there has been a rapid growth of online resources. In particular, with high-quality OCR tools it has become easy to convert an existing corpus into digital form and make it available online. However, lack of metadata available for these resources hampers their discovery and dispersion over the Web.

First, using metadata can help resource discovery. For example, with metadata, a computer scientist might search for the papers written by Kurt Maly since 2003. With full-text searching, resources with these characteristics may be mixed with other irrelevant resources such as the resources about Kurt Maly. According to Doane's estimation [24], a company's use of metadata in its intranet may save about $8,200 per employee by reducing employee time for searching, verifying, and organizing the files.

Second, using metadata such as Dublin Core [27] can make collections interoperable with the help of OAI-PMH (Open Archive Initiatives Protocols for Metadata Harvesting), a framework based on metadata harvesting [47]. In the OAI-PMH framework, a repository interoperates with other components in the framework through supporting the same protocol and using at least Dublin Core metadata format. OAI-PMH specification defines these kinds of repositories as data providers. Data providers accept

The journal model for this dissertation is the IEEE Transactions on Information Theory.

OAI-PMH requests and provide metadata through a network. Besides data providers, OAI-PMH framework contains another kind of participants - service providers. A service provider harvests metadata from data providers and provides value-added services. For example, a service provider Arc [53] harvests metadata from OAI compliant repositories and renders search service on the harvested metadata.

Realizing the benefits of metadata, most modern digital libraries support processes for acquisition of metadata as part of the publication process. However, metadata does not exist for legacy collections that mostly have the form of scanned images either in PDF (Portable Document Format) format or some image format such as TIFF (Tagged Image File Format). There are a number of good commercial tools for scanning and applying OCR (Optical Character Recognition) to generate an electronic version of a document. Nevertheless, there is a lack of good tools that can take an electronic version of a scanned document and extract the metadata from the document. The process of creating metadata manually is expensive and time-consuming for a large collection. According to Rosenfeld's presentation in the DCMI 2003 workshop [21], it would take about 60 employee-years to create metadata for 1 million documents. The costs for manual metadata creation make a great case for the automated metadata extraction tools.

1.2. Problem Statements

This dissertation addresses the problem of how to extract metadata automatically from a large heterogeneous legacy collection. As we described previously, using metadata helps resource discovery and makes a collection interoperable with help of OAI-PMH. However, manual metadata creation is very expensive for a large collection.

Even though some existing approaches [9], [41], [42] addressed how to extract metadata from documents automatically, they mainly focused on specific domains or specific document types. Extracting metadata from a large heterogeneous collection with high accuracy is still a challenge.

In this dissertation, we mainly address the following issues:

- Heterogeneity, i.e. how to achieve a high accuracy for a heterogeneous collection;
- Scaling, i.e. how to apply an automated metadata extraction approach to a very large collection;
- Evolution, i.e. how to process documents added to a collection over time;
- Adaptability, i.e. how to apply an approach to a new document collection;
- Complexity, i.e. how many document features can be handled, and how complex the features should be.

1.3. Approach

In this dissertation, we propose a template-based metadata extraction approach to address the issues mentioned above. According to this approach, documents from a heterogeneous collection are first classified into document groups based on their similarity. For each document group we develop a template, or a set of rules, to instruct our metadata extraction engine how to extract metadata from the documents in this document group. In this the rest of this section, we shall discuss specifically how our template-based approach address the heterogeneity issue, the scaling issue, the evolution issue, the adaptability issue and the complexity issue.

To address the heterogeneity issue, our template-based approach classifies documents into document groups and makes each document group contain similar documents. In this way, a heterogeneous collection has actually been transformed into several homogenous collections. Furthermore, by using different templates, our approach processes documents from various document groups with different sets of rules.

Our template-based approach addresses the scaling issue by developing algorithms to classify documents into groups based on their similarity. Our code should process most documents for a large collection with much smaller number of groups.

Existing rule-based approaches [9], [41], [42] hardcode the rules to extract metadata from documents. In these approaches, changing the rules requires recompiling their programs. This makes them difficult to use for different collections. To address the adaptability issue, we develop a rule language and create a rule engine to understand the rules written in this language. In this way, rules in a template can be modified without changing our program. To extract metadata from documents in different document classes, our engine loads different templates at running time and process the documents accordingly.

For some collection, new kinds of documents may be added over time. Our template-based approach addresses the evolution issue by creating a new group for a new kind of documents. When a new document is coming, it will be checked against all the existing groups. If it belongs to one of the existing groups, our template engine will load the template associated with this group and process this new document accordingly. If it does not belong to any existing document group, a new group and a new template will be created for it.

Our template-based approach addresses the complexity issue by developing our own rule language. Because the templates in our approach need to be created manually, it is important to make the templates easy to develop. In our approach, we develop our own rule language so that we have the flexibility to create our own features. This will simplify the task of creating a template. For example, in our approach, we can define a feature named "dateformat" for any date format, such as "January 05, 2006", "11/20/2005", etc. Hence, users can simply use feature "dateformat" instead of creating a complex regular expression for any date format.

As a part of our template-based approach, we also address how to locate a document page with metadata information. We do not limit our approach to extract metadata from title pages or first page. Our template-based approach extracts metadata from a page with metadata information regardless whether the page is the first page or not.

1.4. Objectives

The main objective of this research is to automate the task of extracting metadata from a large legacy collection. The legacy collection we focus on is downloaded from the DTIC (Defense Technical Information Center) [26], which is responsible for the acquisition, storage, retrieval, dissemination, utilization, and enhancement of scientific and technical information for research and development managers, scientists, engineers, senior planners and others. Our downloaded DTIC collection consists of about 10,000 documents in PDF format.

I need mention that not all PDF documents are searchable. Actually, Adobe supports four forms of PDF for paper-based document: "PDF Image Only", "PDF

Searchable Image Exact", "PDF Searchable Image Compact", and "PDF Formatted Text and Graphics" [1]. "PDF Image Only" files contain images in PDF wrapper. They are not searchable because they do not contain text. "PDF Searchable Image Exact" and "PDF Searchable Image Compact" uses two layers: a layer to store image information and a layer to store text information. "PDF Formatted Text and Graphics", also known as "PDF Normal", contains text and graphics in one layer.

Our downloaded PDF documents are in either "PDF Image Only" or "PDF Formatted Text and Graphics". For simplicity, in the rest of our dissertation, we will call them "Image PDF" and "Text PDF" respectively. Please also note that even though we focused on documents in PDF format, our approach can be also applied to a collection of documents in other formats or even documents in print as long as these documents can be scanned or converted to PDF format.

In summary, we have the following objectives:

- To develop a flexible and adaptable approach for extracting metadata from physical collections, with the focus on the DTIC collections;
- To develop an efficient approach to classify documents into document groups;
- To integrate the techniques and tools developed for DTIC test bed into an interoperable digital library framework.

1.5. Organization of the Dissertation

The rest of this dissertation is organized as follows:

Chapter 2 - Background: In Chapter 2, we will present the background and related works in area of document classification and metadata extraction.

Chapter 3 – Template-based approach for metadata extraction: In chapter 3, we will describe our template-based approach for metadata extraction in detail. We will also discuss motivations and open research questions.

Chapter 4 – Document classification: In chapter 4, we will present the document classification algorithms used in our approach. In this chapter, we will also describe how to locate a page with metadata information in a document.

Chapter 5 – System implementation: In chapter 5, we will show the details of system implementation. In this chapter, we will present the overall architecture of converting a legacy collection to an interoperable repository and the details about document feature set, rule language, and rule engine.

Chapter 6 - Experimental results: In chapter 6, we will describe the experiments we conducted to address the issues of heterogeneity, scaling, evolution and complexity.

Chapter 7 - Conclusions and future work: Finally, in chapter 7, we will summarize the contributions of our research as well as the issues we addressed. In this chapter, we will also provide directions for the future work.

CHAPTER II

BACKGROUND

This chapter gives summarizes research activities in the area of extracting metadata from documents automatically and other related areas. We will introduce document classification in section 2.1 and metadata extraction in section 2.2.

2.1. Document Classification

One approach to solving the metadata extraction problem for a heterogeneous collection is to partition the collection into a set of homogeneous collections first and then solve the extraction problem for each homogeneous collection. Document classification is used to create equivalence groups of similar documents. Few researchers have addressed the problem of how to find the page(s) that will be used to differentiate the documents. We will address this problem in Chapter 4.

Existing approaches to classify documents (assuming that one has the page containing the metadata isolated) into equivalence groups include one that uses a document model based upon page layout structure [17], [31], [51]. X. Hao et al. [31] segmented documents into blocks and encoded the hierarchy layout structures into tree structures called "L-S trees". They divided a page into structured and unstructured parts. A structured part was further divided into static and dynamic parts. For documents of the same document type, a static part has fixed location and terms with the same meanings. For example, memo documents might contain the special terms "From" and "To". A dynamic part is related to a static part. In a form, a static part may be a field name and a dynamic part is the field value. The document classification in this approach is sample-

based. A knowledgeable user would tag the blocks of some samples either static or dynamic. A new document was classified into a document type if it had a similar L-S tree with a sample of this document type. X. Hao et al. [31] experimented with 100 documents and showed that only 10% of the memos and 20% of letters and 25% of journal papers were needed in a sample base in order to achieve 90% accuracy. X. Li et al. [51] represented document pages as directed weighted graphs. In a directed weighted graph for a given document page, they used a vertex for each block in the page and a directed edge for the adjacent relation between two blocks. They used the Levenshtein distance [3] between directed weight graphs to measure page similarity. X. Li et al. [51] did not report numerical results for any experiments in their paper. F. Cesarini [17] encoded a document's cover page into an MXY-Tree and used it for document classification. As an extension of XY-Tree [58], an MXY-Tree recursively cuts a page into blocks by separators (e.g. lines) as well as white spaces. A good feature of these approaches is that they are not sensitive to the absolute position of blocks and the absolute spaces among blocks, because these approaches mainly model the relative relationships among the blocks. Therefore, they are suitable for document pages like the samples shown in Fig. 1, i.e. they contain blocks with different absolute locations but similar relative relations. However, these approaches are sensitive to block identification, i.e. block boundary detection. Fig. 2 shows two samples with bad block boundary detections when using Scansoft's Omnipage OCR tool. For these two similar samples, the OCR tool generated two different structures. Using an approach such as MXY-Tree will fail to catch the similarity of these two documents, because their MXY-Trees are quite

different. For example, the top part of the left sample cannot be vertically cut into two parts, but the top part of the right sample can.

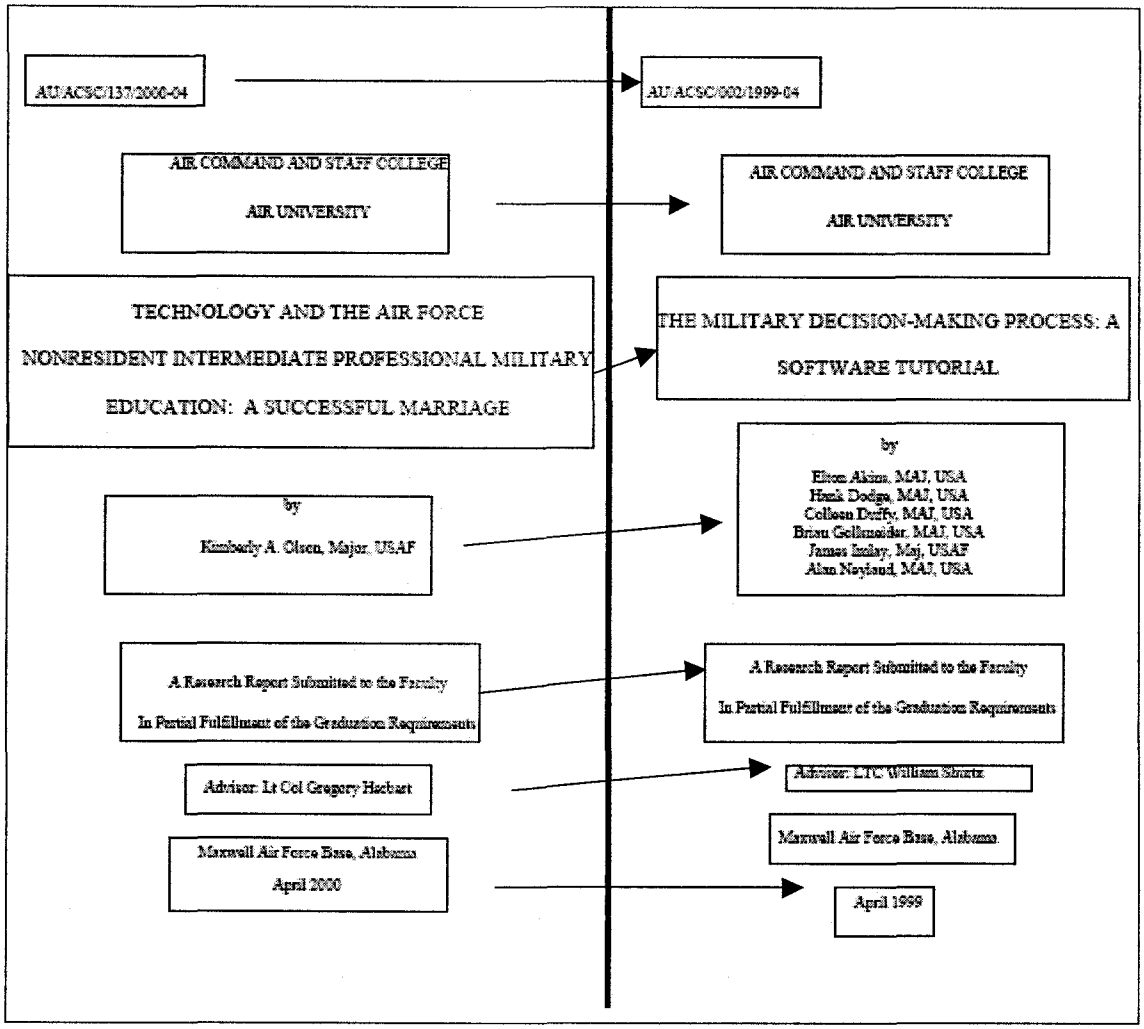


Fig. 1. Samples with similar structure

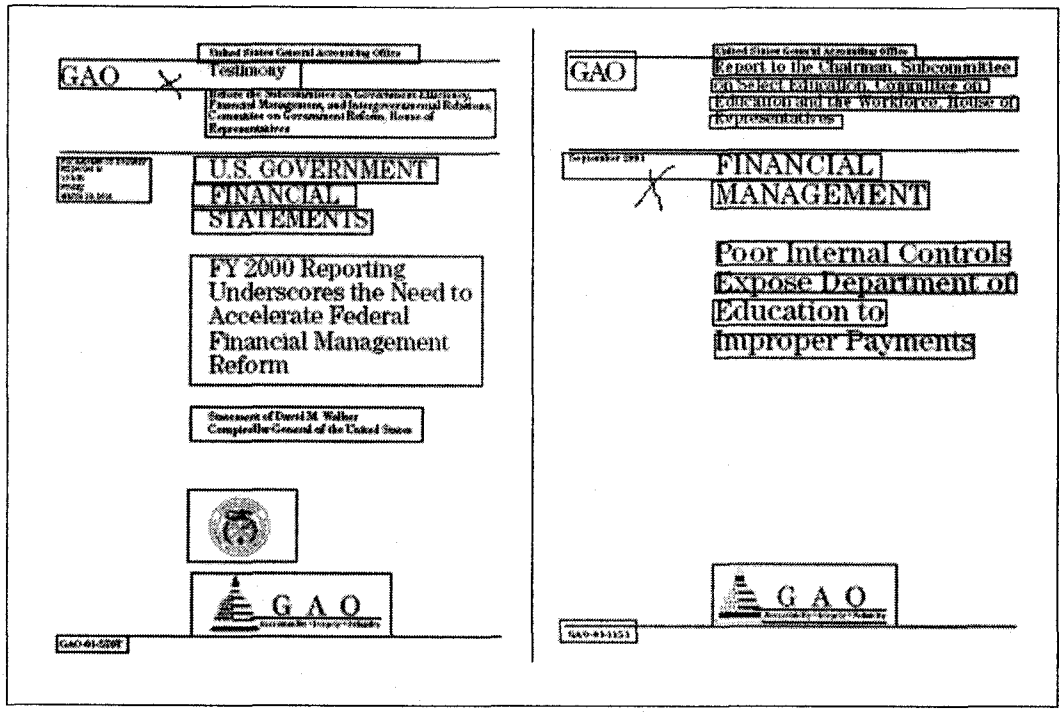


Fig. 2. Samples of bad block boundary detection

J. Hu et al. [36] used another way to measure page layout similarity. They partitioned a page into an *m* by *n* grid in which each cell was either a text cell (more than half of it was overlapped by a text block) or white space cell. With partitions by absolute positions, this approach measures page similarity by a block's absolute position. This approach is not as sensitive to block boundary detection. However, this approach is sensitive to absolute position. This will cause problem when pages with the similar style but with blocks of different sizes (e.g. a page with one author block may be different from a page with 10 author blocks). Fig. 3 demonstrates the limitation of this approach. In Fig. 3, a black cell represents a text cell, and a white cell represents a white space cell. Even though the two pages are similar, the similarity measured by J. Hu's approach is zero.

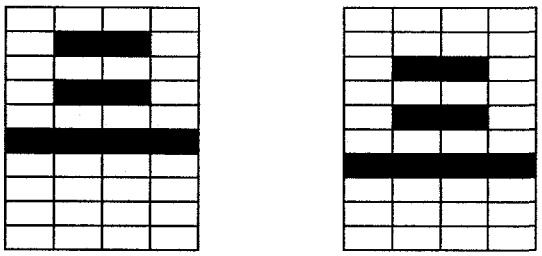


Fig. 3. Problem with absolute position match

2.2. Metadata Extraction

In this dissertation, metadata extraction has the following meaning. Metadata refers to information about a document that is used to catalogue a document and later to allow users to search and locate it. It is commonly clustered on one or more pages; examples include title, creators, affiliation, publisher, language, ID number, and date. Extraction refers to the process of automatically locating the pages that contain metadata, extracting the metadata and tagging them as the appropriate type. We classify approaches to build a metadata extraction system into: rule-based approaches and machine-learning approaches.

2.2.1. Rule-based Approach

The steps of building a rule-based metadata extraction system are typically as follows: first, some experts examine samples of the document collection and define rules for metadata extraction; then, software developers implement these rules either as part of an expert system or as part of an ad hoc rule engine. The accuracy, inventiveness, and appropriateness of the rules that experts defined play a critical role in building a system with high accuracy.

Most metadata extraction systems proposed so far are rule-based systems. The rules are mainly based on visual clues of the target documents and are typically confined to a set of similar documents. D. Bergmark [9] used a heuristic system for text PDF files from ACM. S. Klink, A. Dengel, and T. Kieninger [43] described a system to extract metadata from text PDF files by using a manually created rule base. J. Kim, D.X. Le, and G.R. Thoma [41] proposed a method to use rules to extract information from document images. XMLCities' XMLCapture Suite [74] provided a graphic interface for users to define the rules on the fly before they process a specific document.

These systems can be implemented straightforwardly. However, they usually lack adaptability. Because rules are defined and threshold values are chosen arbitrarily, many rules that work with one data set may not work with another data set. Adapting a rule-based system to different data sets is difficult. More often than not, it requires building another system from scratch.

2.2.2. Machine Learning Approach

We will use the definition of machine learning given by Dietterich in the article "Machine Learning": "Machine Learning is the study of methods for programming computers to learn" [22]. We also use the following terms defined by Dietterich:

- A classifier, a program to assign a class to an object;
- A labeled example (or sample), a pair of an object and its associated class;

Machine learning tasks can be classified into two categories: Empirical learning and Analytical learning. Empirical learning requires external inputs while analytical learning does not need external inputs. Based on whether the input data are labeled samples or not, Empirical learning can be further classified into supervised learning and

unsupervised learning tasks. A supervised learning task is one that analyzes a given set of objects with class labels while an unsupervised learning task is one that analyzes a given set of objects without class labels [22].

Machine learning methods used in metadata extraction usually belong to the supervised learning category. The two most commonly used machine learning methods for metadata extraction are: Hidden Markov Models (HMM) and Support Vector Machines (SVM). HMM is a machine learning technology to model sequential data (a document is represented as a sequence of tokens). SVM is usually used to build classifiers from labeled samples.

2.2.2.1. Hidden Markov Model (HMM)

HMM, which was introduced by Baum in late 60s, is a probabilistic technique for the study of time series events [63]. HMMs have been widely used in gene and speech recognition. The following definition taken from [72] is a concise introduction to this area of research: "The Hidden Markov Model is a finite set of *states*, each of which is associated with a (generally multidimensional) probability distribution. Transitions among the states are governed by a set of probabilities called *transition probabilities.* In a particular state an outcome or *observation* can be generated, according to the associated probability distribution. It is only the outcome, not the state visible to an external observer and therefore states are "hidden" to the outside; hence the name Hidden Markov Model."

We use the simple example that was given in "A Tutorial on Hidden Markov Models" [28] to illustrate the concept of HMMs. Assume a person sits in a closed room and produces a series of output symbols that may be either Tail or Head. At each event he

tosses one of three coins and depending on whether it shows head or tail, he writes the corresponding output symbol. We want to guess the sequence of tossed coins for a given sequence of tossing results. The tossing result is affected by:

1. Individual biasing of each coin; for example, if coin 3 has higher probability to produce heads than the other two coins and the other two have equal probabilities for heads and tails, we expect to see more heads in the tossing result;
2. The order of tossing coins; for example, supposing that the person inside the room never tosses coin 3 again once he tosses coin 1 or coin 2, we will be expected to see that the number of heads will be almost equal to the number of tails if he starts with coin 1 or coin 2;
3. The starting coin; if he starts with coin 3, we expect to see more heads.

In the other words, if we have information about individual biasing of each coin, the probabilities of transiting from one coin to another, and the probabilities of starting from each coin, we may increase the probability of our guessing right as to what coins were tossed at what time.

HMM is a finite state automaton to model scenarios like the above example. In this model, a sequence of symbols is produced by state transitions. It starts in one state, transits from that state to another, and emits a symbol in each state. The transition from state to state is probabilistic. At each state, symbol emission is probabilistic too. For HMM, the underlying states cannot be observed, i.e. they are hidden. For example, in our case, we do not know which coin is tossed. A HMM consists of:

- A set of hidden states; i.e., it is {toss of coin 1, toss of coin 2, toss of coin 3} in the above example;
- A set of observation symbols; i.e., it is {Head, Tail} in the above example.
- The initial state probability distribution; it is a vector of probabilities of starting in a state, i.e., the probabilities of starting with coin 1, coin 2, and coin 3 respectively in the above example;
- The state transition probability distribution; it is a matrix of the probabilities of transiting from one state to another, i.e., the probabilities of transiting from one coin to another in the above example.
- The observation symbol probability distribution; it is a matrix of the probabilities of observing a symbol at each state, i.e., the probabilities of observing a "Head" and "Tails" for coin 1, coin 2, and coin 3.

HMM for metadata extraction can use each metadata element as a hidden state and employ the unique words in documents as observation symbols. Its state transition probability distribution and the observation symbol probability distribution can be estimated by the tagged samples. For example, the probability of transiting from "title" to "creator" can be computed by dividing the number of transitions from "title" to "creator" into the total number of transitions from "title"

Given an HMM model and all its parameters, the problem is to find the most probable sequence of hidden states (metadata elements) for a given document or any sequence of words and extract the symbols (or words) associated with these states (metadata element). The process of determining the most probable sequence of hidden states for a given sequence of observation symbols can be solved by exhaustively

computing the probabilities for all possible sequences. More efficiently, it can be solved by the Viterbi algorithm [28].

A simple HMM example is shown in Fig. 1. Here we want to use the HMM to extract 'title' from documents. The hidden states are "title" and "other". For simplicity, let us assume that each document is a sequence of tokens or words. A token can take on three possible values: "A", "B", and "C". The task of extracting title from a document is to find the class "title" or "other" for each word in a given sequence, i.e. a document. Assume that we have already trained the HMM and determined all its parameters through providing it with a sufficient number of documents each tagged to its elements. For example, one input might be: C=title, B=title, B=other. After learning from the samples, the HMM estimated that the probability of starting with "title" is 0.9 and the probability of starting with "other" is 0.1. Other learned probabilities in this example are shown in Fig. 4.

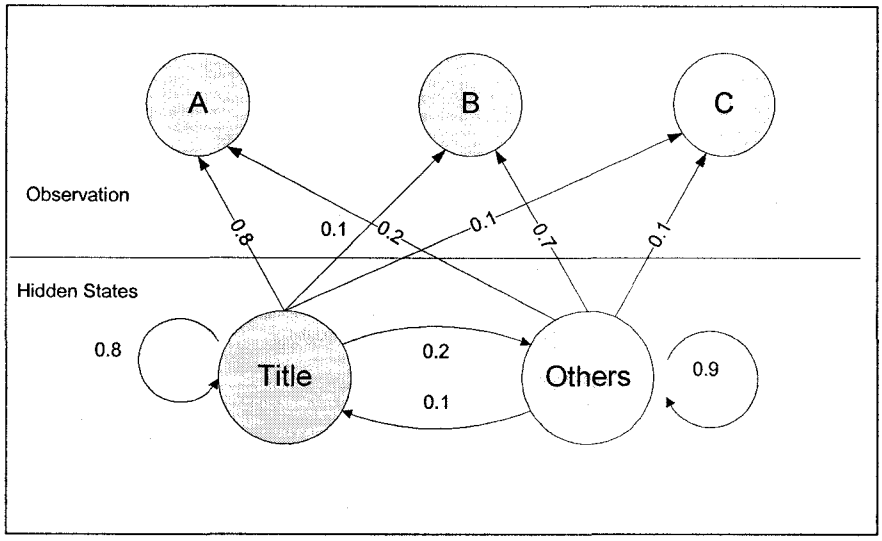


Fig. 4. HMM sample

In Fig. 4, we add a line in the middle to separate the hidden states from the observation symbols. Below the line are the hidden states, i.e. "title" and "other". Above the line are observation symbols, i.e. "A", "B", and "C". We use arrows to indicate the state transitions or the symbol emissions from a state. The number on each arrow shows the probability of transiting from one hidden state to another or the probability of emitting one symbol from a hidden state. For example, according to Fig. 4, the probability of a transition from "title" to "title" is 0.8. The probability of transition from "title" to "other" is 0.2. In "title" state, the probability of observing "A" is 0.8.

We can now use the HMM to extract metadata (hidden states) for a document (a sequence of observation symbols). Fig. 5 shows all possible sequences of metadata elements for a document "AACB".

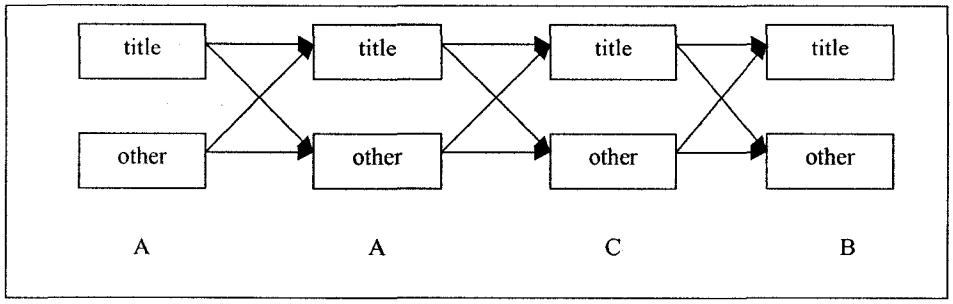


Fig. 5. Possible sequences of metadata elements for "ABC"

The probability of observing "AACB" for a hidden state sequence is:

$$P(AACB|S_1S_2S_3\ S_4)=P(S_1)*P(A|\ S_1)*P(S_1\text{-}>\ S_2)*\ P(A|\ S_2)\ *P(S_2\text{-}>\ S_3)*\ P(C|\ S_3)$$

$$*P(S_3\text{-}>\ S_4)*\ P(B|\ S_4)$$

Where $P(Si)$ is the probability of starting with state Si, $P(O_j|\ S_i)$ is the probability of observing symbol O_j at state S_i, and $P(S_i\text{-}>\ S_j)$ is the probability of transiting from state S_i to state S_j. For simplicity, we use "T" for "title" and "O" for "other" in the following equations:

$$P(AACB \mid TTTT) = (0.9 * 0.8) * (0.8 * 0.8) * (0.8 * 0.1) * (0.8 * 0.1)$$
$$P(AACB \mid TTTO) = (0.9 * 0.8) * (0.8 * 0.8) * (0.8 * 0.1) * (0.2 * 0.7)$$
$$P(AACB \mid TTOT) = (0.9 * 0.8) * (0.8 * 0.8) * (0.2 * 0.1) * (0.1 * 0.1)$$
$$P(AACB \mid TTOO) = (0.9 * 0.8) * (0.8 * 0.8) * (0.2 * 0.1) * (0.9 * 0.7)$$
$$P(AACB \mid TOTT) = (0.9 * 0.8) * (0.2 * 0.2) * (0.1 * 0.1) * (0.8 * 0.1)$$
$$P(AACB \mid TOTO) = (0.9 * 0.8) * (0.2 * 0.2) * (0.1 * 0.1) * (0.2 * 0.7)$$
$$P(AACB \mid TOOT) = (0.9 * 0.8) * (0.2 * 0.2) * (0.9 * 0.1) * (0.1 * 0.1)$$
$$P(AACB \mid TOOO) = (0.9 * 0.8) * (0.2 * 0.2) * (0.9 * 0.7) * (0.9 * 0.1)$$
$$P(AACB \mid OTTT) = (0.9 * 0.8) * (0.8 * 0.8) * (0.8 * 0.1) * (0.8 * 0.1)$$
$$P(AACB \mid OTTO) = (0.9 * 0.8) * (0.8 * 0.8) * (0.8 * 0.1) * (0.2 * 0.7)$$
$$P(AACB \mid OTOT) = (0.9 * 0.8) * (0.8 * 0.8) * (0.2 * 0.1) * (0.1 * 0.1)$$
$$P(AACB \mid OTOO) = (0.9 * 0.8) * (0.8 * 0.8) * (0.2 * 0.1) * (0.9 * 0.7)$$
$$P(AACB \mid OOTT) = (0.9 * 0.8) * (0.2 * 0.2) * (0.1 * 0.1) * (0.8 * 0.1)$$
$$P(AACB \mid OOTO) = (0.9 * 0.8) * (0.2 * 0.2) * (0.1 * 0.1) * (0.2 * 0.7)$$
$$P(AACB \mid OOOT) = (0.9 * 0.8) * (0.2 * 0.2) * (0.9 * 0.1) * (0.1 * 0.1)$$
$$P(AACB \mid OOOO) = (0.9 * 0.8) * (0.2 * 0.2) * (0.9 * 0.7) * (0.9 * 0.1)$$

Because *P(AACB|TTTO)* has the largest value, the most probable sequence of metadata elements for "*AABC*" is "*TTTO*". Therefore, we obtain that "AAB" is a title and "C" is "Other".

HMMs are particularly useful for detecting patterns of sequences of metadata elements. For example, if in a particular class of documents each document starts with a report number which is almost always followed by a title which in turn is followed by authors, we can train an HMM to discover these elements. All the information we have used so far in the examples is called 'textual'. Information such as 'a symbol is in the top half of a page' is much more difficult to incorporate into HMMs as is textual information.

Another problem with an HMM is that it requires many training data because it assumes that the probabilities learned from data set are the actual probabilities.

Inadequate training data may break this assumption and produce results that are not reliable.

2.2.2.2. Support Vector Machine (SVM)

Based on statistical methods, SVM is widely used in pattern recognition areas such as face detection, handwritten character recognition, and gene classification [14]. T. Joachims [39] successfully applied it to text categorization. Recently H. Hui et al. have used it to extract metadata from document [30].

SVM is a statistical model and was proposed by V. N. Vapnik [71]. For metadata extraction a useful characteristic of SVM is that it can be applied to solve problems with very large feature (an attribute of a document such as for example 'the number of lines it contains') sets. The basic idea of Support Vector Machine is to find an optimal hyperplane to separate two classes with the largest margin from pre-classified data. After this optimal-separation hyperplane is determined, it can be used for placing data into two classes based on which side of the hyperplane they are located.

Fig. 6 shows a simple example. In this example, the task is to determine whether a text string in a document is a title or not. Each document consists of lines of text with each line having the attributes (line number, font size). Table 1 shows an example of a document with its line attributes.

TABLE 1
SAMPLE DOCUMENT WITH LINE ATTRIBUTES

This is the title	(1,16)
And this is	(2,12)
The text of	(3,12)
The document	(4,12)

Each *plus* symbol or *minus* symbol in the figure is one sample that has been tagged with the correct attributes. The plus symbols indicate that the sample is a title and the minus symbols indicate the sample is not a title. In the example in Table 1, (1,16) would be a plus and (3,12) would be a minus point. As we can see, there are many hyperplanes to separate the two kinds of symbols in Fig. 6. To build an SVM from these training data means to find an optimal hyperplane that separates two classes with maximum margin. After the separation hyperplane is determined, it can be used to classify new data based on which side they are located. In this example, the data located on the left side of the separation hyperplane are a title and the points located on the right side do not represent a title.

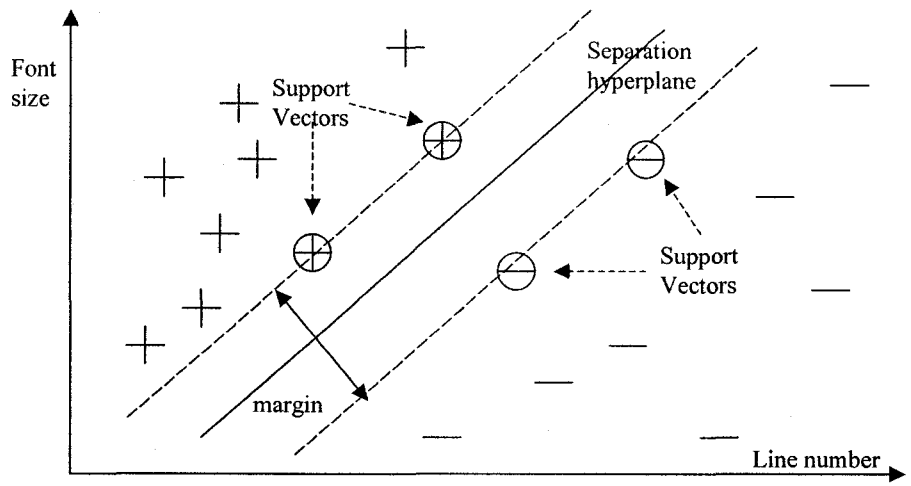


Fig. 6. SVM in two-dimension space

The points with the smallest distance to the optimal separation hyperplane are called "*support vectors*". As can be seen in Fig. 6, the location of the optimal separation hyperplane depends only on the support vectors and no other data points.

Finding the optimal hyperplane for a linear separable data set can be solved as a constrained optimization problem. The mathematical notions and equations in this section are taken from "A Tutorial on Support Vector Machines for Pattern Recognition" [14]. Labeled samples can be presented as $\{x_i, y_i\}, i = 1..l$, $y_i \in \{+1,-1\}, x_i \in R^d$, where l is the number of samples, and d is the dimension of the feature set. For the example shown in Fig. 6, d equals two, and x_i is a vector of the line number and the font size, such as (1,16). For any positive sample, y_i=+1, and for any negative samples, y_i=-1. Given a training set, there are many hyperplanes $w \bullet x + b = 0$ to separate the positive samples from the negative ones. Here, x_i, w, x, b are vectors, and $w \bullet x$ is the inner production of w and x. All training data satisfy:

$$x_i \bullet w + b \geq 1 \quad \text{for } y_i = +1 \tag{1}$$

$$x_i \bullet w + b \leq -1 \quad \text{for } y_i = -1 \tag{2}$$

The inequalities (1) and (2) can be combined into:

$$y_i(x_i \bullet w + b) - 1 \geq 0 \quad \forall i \tag{3}$$

SVM is used to find a hyperplane with maximum margin. The margin, as shown in Fig. 6, is $\frac{2}{\| w \|}$, which is the distance between the hyperplane $x \bullet w + b = 1$ and the hyperplane $x \bullet w + b = -1$. Hence the problem is to minimize $||w||^2$ with constraints (3).

This problem can be translated to a Lagrangian formulation by introducing positive Lagrange multipliers [10] $\alpha_i, i = 1..l$, giving:

$$L \equiv \frac{1}{2} \| w \|^2 - \sum_1^l \alpha_i y_i (x_i \bullet w + b) + \sum_1^l \alpha_i \tag{4}$$

Minimizing L subject to constrains $\frac{\partial L}{\partial \alpha_i} = 0$ can be solved by maximizing L subject to constrains $\frac{\partial L}{\partial w} = 0$ and $\frac{\partial L}{\partial b} = 0$. We will use our example shown in Table 1 to illustrate how to get *w* and *b* for an SVM. In our example, *l*=4, i.e. we have four training samples:

$$x_1 = [1\ 1\ 6], x_2 = [2\ 1\ 2], x_3 = [3\ 1\ 2], x_4 = [4\ 1\ 2]$$

We use $\begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$ for *w*, and $[x_{i1} x_{i2}]$ for x_i, then

$$L = \frac{1}{2}(w_1^2 + w_2^2) - \sum_1^4 \alpha_i y_i([x_{i1}x_{i2}] \bullet \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} + b) + \sum_1^4 \alpha_i$$

$$= \frac{1}{2}(w_1^2 + w_2^2) - (\alpha_1 \times 1 \times (w_1 + 16w_2 + b))$$
$$-(\alpha_2 \times (-1) \times (2w_1 + 12w_2 + b))$$
$$-(\alpha_3 \times (-1) \times (3w_1 + 12w_2 + b))$$
$$-(\alpha_4 \times (-1) \times (4w_1 + 12w_2 + b)) + \sum_1^4 \alpha_i$$

According to Karush-Kuhn-Tucker conditions [40], we get

$$\frac{\partial L}{\partial w_1} = 0,\ \frac{\partial L}{\partial w_2} = 0,\ \frac{\partial L}{\partial b} = 0, \text{ and } \alpha_i(y_i(x_i \bullet w + b) - 1) = 0$$

Therefore, we get the following equations and inequities:

$$w_1 - \alpha_1 + 2\alpha_2 + 3\alpha_3 + 4\alpha_4 = 0 \quad (5)$$

$$w_2 - 16\alpha_1 + 12\alpha_2 + 12\alpha_3 + 12\alpha_4 = 0 \quad (6)$$

$$-\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 = 0 \quad (7)$$

$$\alpha_1(w_1 + 16w_2 + b - 1) = 0 \quad (8)$$

$$\alpha_2(-2w_1 - 12w_2 - b - 1) = 0 \quad (9)$$

$$\alpha_3(-3w_1 - 12w_2 - b - 1) = 0 \quad (10)$$

$$\alpha_4(-4w_1 - 12w_2 - b - 1) = 0 \quad (11)$$

$$w_1 + 16w_2 + b - 1 \geq 0 \quad (12)$$

$$-2w_1 - 12w_2 - b - 1 \geq 0 \quad (13)$$

$$-3w_1 - 12w_2 - b - 1 \geq 0 \quad (14)$$

$$-4w_1 - 12w_2 - b - 1 \geq 0 \quad (15)$$

Solve the equations (5)-(11) with restrictions of inequities (13)-(15), we can get

$$w_1 = -\frac{2}{17}, w_2 = \frac{8}{17}, b = -\frac{109}{17}, \alpha_1 = \frac{2}{17}, \alpha_2 = \frac{2}{17}, \alpha_3 = 0, \alpha_4 = 0$$

We have described how to find an optimal hyperplane for linear separable data. Detailed information about how to handle non-separable data can be found in "A Tutorial on Support Vector Machines for Pattern Recognition" [14]. Considering nonlinear data, SVM can map them into another space and processes them in similar way to linear data. Fig. 7 shows a map from nonlinear separable data to another space, where mapped data are linear separable.

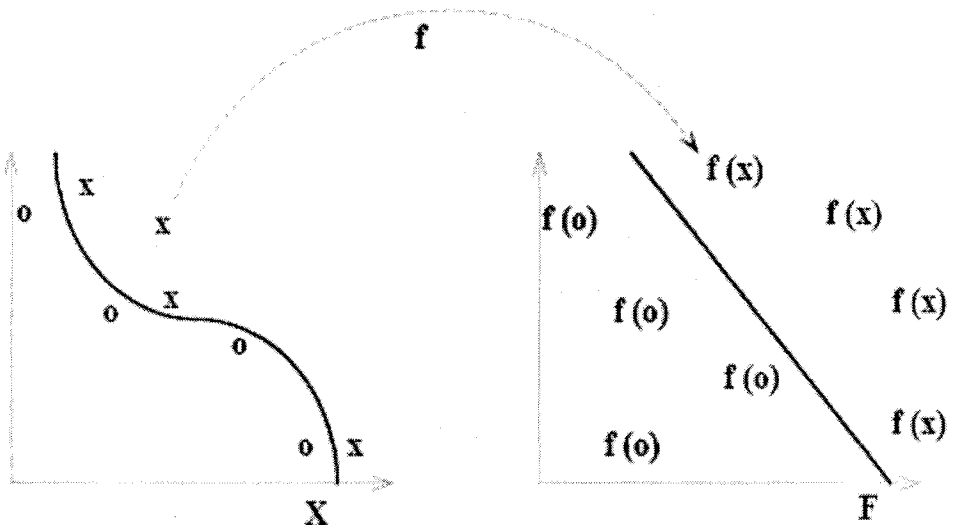


Fig. 7. Map nonlinear data to another space[1]

For the L in equation (4), from $\frac{\partial L}{\partial w} = 0$, we can get

[1] This figure is copied from http://www.support-vector.net/tutorial.html

$$w = \sum_{1}^{l} \alpha_i y_i x_i \tag{16}$$

Substitute w in the $w \bullet x + b$ with equation (16), we can get the determination function:

$$\sum_{1}^{l} \alpha_i y_i x_i \bullet x + b \tag{17}$$

From $\frac{\partial L}{\partial b} = 0$, we can get

$$\sum_{1}^{l} \alpha_i y_i = 0 \tag{18}$$

Substitute equation (16) and equation (18) into equation (4), we can get

$$L \equiv \sum_{1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \bullet x_j \tag{19}$$

Both the determination function (equation 17) and the equation (19) for training are in the form of inner products of data. Suppose the mapping function is Φ, we can find the optimal hyperplane in the mapped space, and use function $\sum_{1}^{l} \alpha_i y_i \phi(x)_i \bullet \phi(x) + b$ for classification. If there is a function $K(x_i, x) = \phi(x_i) \bullet \phi(x)$, we do not need to know what Φ is. Function $K(x_i, x)$ is called "kernel function". Detailed information about "kernel function" can be found in "A Tutorial on Support Vector Machines for Pattern Recognition" [14].

A Support Vector Machine can be used for metadata extraction. Instead of extracting information for each metadata element from documents, it decides whether each token in a document belongs to the class = metadata element. In this way, metadata extraction task is converted to an information classification task.

Basic SVM classification works only for one class: a token belongs to a class or it does not. However, for metadata extraction, we usually extract information for more than one metadata element. For example, Dublin Core metadata set contains 15 metadata elements. Hence, we need to extend the basic SVMs to multi-class SVMs. There are two main approaches:

- "One vs. All" approach trains one SVM for each class by using the data for this class as positive samples and the rest as negative samples. For every input, each SVM determines whether this input belongs to its associate class with a confidence score. Finally, this input is assigned to the class with the highest score.
- "One vs. One" approach trains a classifier for each pair by using one class as positive samples and the other as negative samples. For every input, each classifier makes a vote between two classes. The input is assigned the class with the largest number of votes.

As with other statistical learning models, SVM has to be well trained in advance which means that many labeled samples are required.

SVM can be used for metadata extraction because a metadata extraction task can be converted to a classification problem. For example, extracting a title from a document can be achieved by classifying each part of a document to see whether it is a title or not. Metadata extraction as a whole can be solved by a multi-class SVM with treating every metadata element as a class.

CHAPTER III

TEMPLATE-BASED APPROACH FOR METADATA EXTRACTION

As we described in the background section, rule-based metadata extraction approach has its advantages. It can be implemented directly without taking time to train models from samples. It is usually simple and has good performance for a homogeneous collection. However, for a large heterogeneous collection, humanly defining a set of rules to cover all situations in advance is an extremely time-consuming task. Furthermore, it is possible that some new kinds of documents will be added to the collection later. This makes it difficult to define a rule set in advance. The state of the art in automatic metadata extraction based on machine learning is limited too. Individual methods, such as SVM and HMM, work well with homogenous collections of documents in specific domains. It is a time-consuming task to prepare the training data set and to train the model to achieve high accuracy for a collection of a very heterogeneous nature. In addition, when a new kind of documents is added to the collection, it usually requires rebuilding the model.

To work with a heterogeneous collection, we propose a template-based approach that classifies documents into groups, creates a template, i.e. a set of rules, for each group, and extracts metadata from documents accordingly. We believe that one feasible way to handle a large heterogeneous collection is to classify documents into groups based on document *similarity* so that each group becomes a homogeneous sub-collection. We define a *metadata page* as a document page with richness in metadata. In this dissertation, we define *document similarity* as the similarity of metadata pages. In other

words, we say two documents are similar if they have at least one similar metadata page, which present a similar set of metadata fields in similar style (font size, location, layout, etc.). Fig. 8 shows two similar metadata pages. Their common metadata fields are identified with arrows.

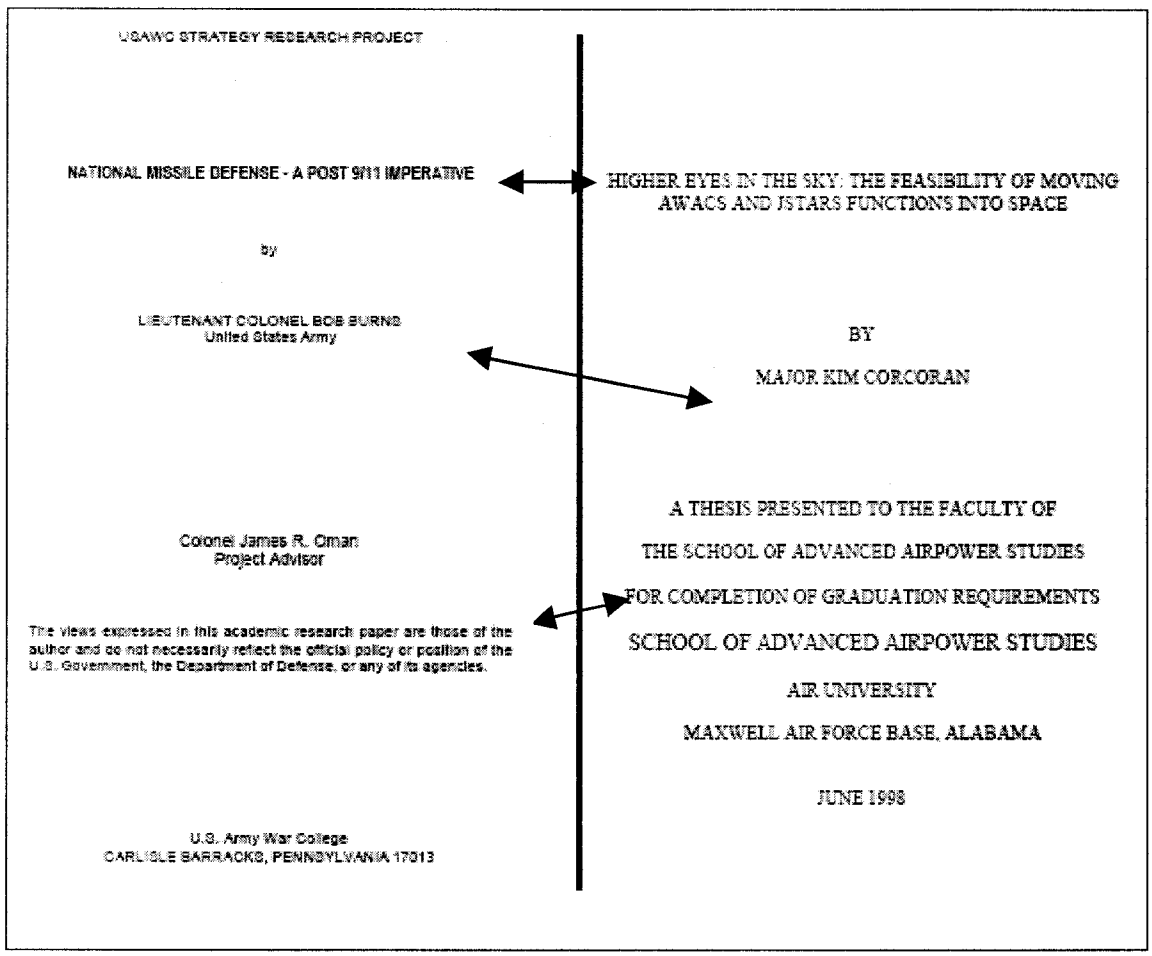


Fig. 8. Two similar metadata pages

It is worth noting that a document may have more than one metadata page. In our current implementation, we extract metadata from one metadata page only. A possible refinement is to extract metadata from multiple metadata pages and integrate metadata

from different pages together. Accordingly, we classify each document into one group based on only one of its metadata page. More precisely, in our approach, given a document, we first locate one of its metadata pages; then we classify the page into a group; after this, we extract metadata from the page by use the template associated with the group. In our template-based approach, a template is written in an XML-based language, which we will describe in the Chapter 5. For simplicity, template samples given in this chapter will be described in English.

3.1. Template-based Approach

A typical process of our template-based approach for metadata extraction is shown in Fig. 9. First we apply OCR to these documents to convert them into a certain format; then, we classify these documents into groups; after that, we manually create a template for each group to indicate how to extract metadata from documents in the group. For example, we may use a template like "the text in the largest font size is a title, the text located below a title but above a text line in date format is a creator" for one document group, and use another template like "the first line is a report identifier, the second line is a date, the text in font size 14 is a title" for another group. Finally, we run our metadata extraction engine to process them by using their corresponding templates and store their metadata into a database or into files.

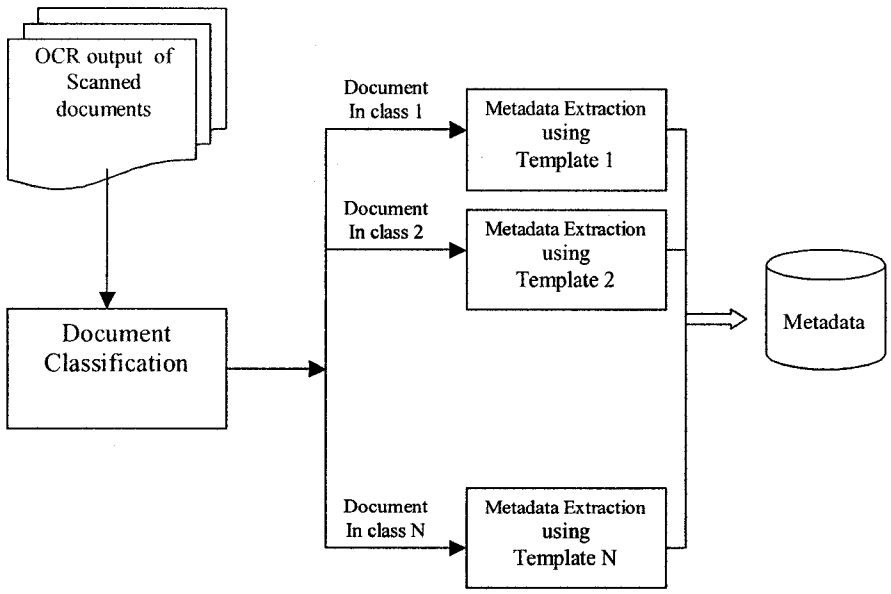


Fig. 9. Template-based metadata extraction

Our template-based approach has advantages over machine-learning approaches for extracting metadata from a large heterogeneous collection. First, our template-based approach is a rule-based system. Therefore, it can be implemented straightforward. It saves time not only for creating samples, but also for training a model. It is worthy of noting that creating samples to train a model for a heterogeneous collection is a time-consuming task. Second, machine-learning approaches require rebuilding the model if a new kind of documents is added to the collection. Our template-based approach solves this problem by creating a new group and a new template for a new kind of documents.

Even though our template-based approach is a rule-based approach, it differentiates itself from existing rule-based approaches with its features. First, our template-based approach has better adaptability. Unlike the simple rule-based systems that hard-coded their rules, our approach decouples the rules from metadata extraction code and stores

them in files. Our approach makes it possible to work for a different collection with little or without changing the metadata extraction code. Second, our template-based approach simplifies the task of rule creation. Unlike the complex rule-based systems that created a large set of rules to cover all possible situations in advance, our approach defines rules for documents in a certain document group only. In addition, our template-based approach can reduce rule errors. In our approach, it is possible for us to creating a template with simple rules and simple logic combination of these rules. This is because of that we create a template for one document group only, and the documents in one group are similar. Our approach reduces the probability of having bugs in a template because a template in our approach is simpler than a template for a whole collection. Furthermore, since templates are loaded at runtime in our approach, we can apply a template to some samples first, check the results, and refine the template without modifying metadata extraction code. In this way, we can correct errors before applying it to a large document set.

We use two samples in Fig. 10 to illustrate that our template-based approach simplifies rule creation. Considering the sample on the left, we can crate a template like "the text in the largest font size in the page belongs to an organization field; the text after the organization filed is a title ..." For the sample on the right, we can create a template like "the text in the largest font size in the page is a title; the text after the title field is an organization ..." In these two templates, to extract metadata fields "title" and "organization", we just need to use two kind of features: relative font size (largest font size) and relative location (e.g. "the text after the title field"). However, we cannot just use these features to create one template for these two samples. In the sample on the

right, the text in the largest font size is the title, and the title is located above the organization field, while in the sample on the left, the text in the largest font size is the organization, and the title is located below the organization. The inconsistence use of layout feature, such as relative font size and the relative location of metadata fields, increase the complexity of template creation.

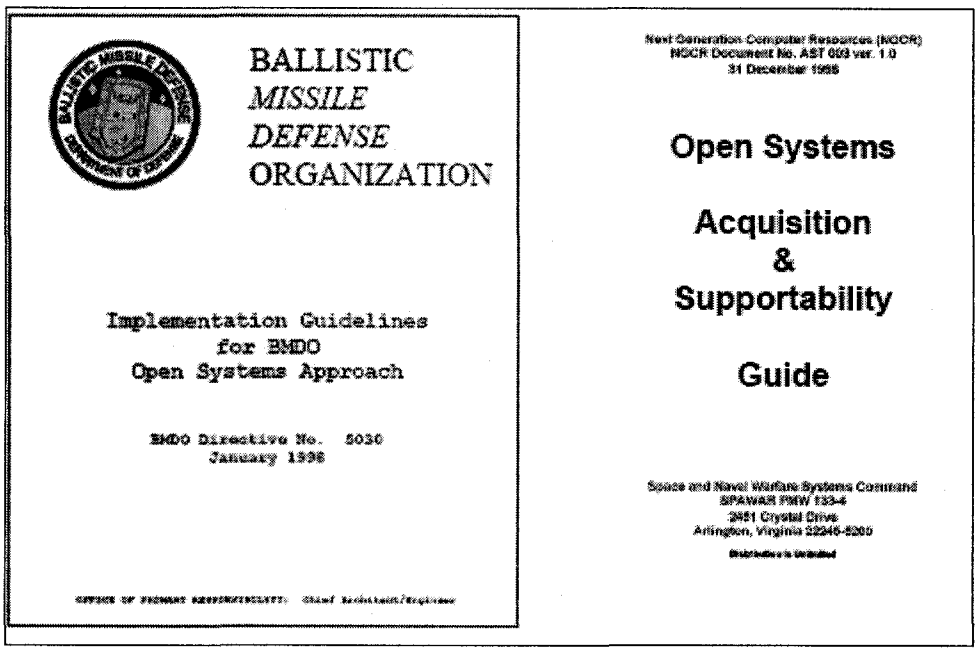


Fig. 10. Document samples with different styles

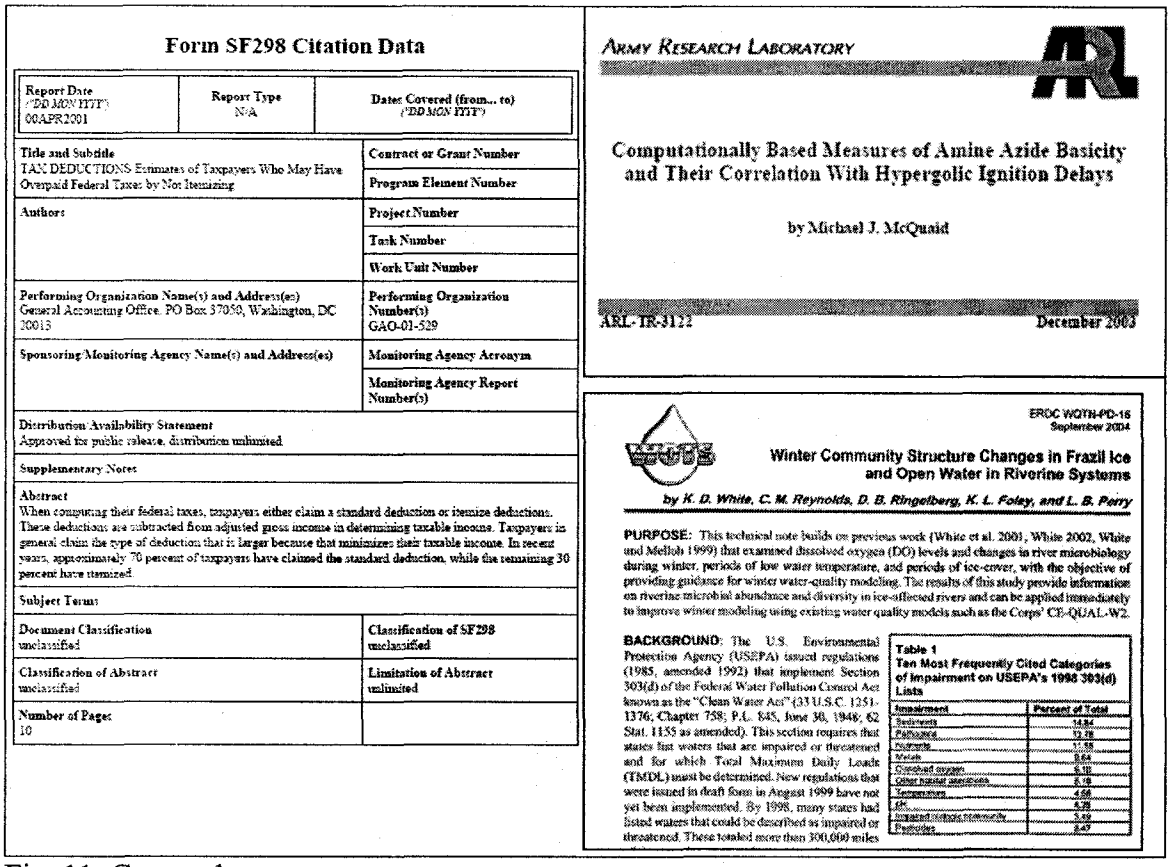
3.2. Template Types

In section 3.2, we show the advantages of our template-based approach. In practice, there are many ways to write a template. In this section, we will discuss what kind of templates should be used in our approach. This includes how specific a template should be and what kind of information it should contain.

3.2.1. General Template vs. Specific Template

As shown in Fig. 9, our template-based metadata approach first classifies documents into groups. For each group, we create a template for extracting metadata from documents in this group. According to our definition, a template is a set of rules instructing how to extract metadata for documents in a document group. Therefore, templates can be classified into a general template and a specific template based on what kind of groups they work with. A general template is a template for coarse classified document groups and a specific template is a template for fine-grained classified document groups. The definitions of coarse classified document groups and fine-grained classified document groups are relative.

In the context of DTIC report documents, document pages can be classified into some coarse groups, such as a cover page, a title page, and a table of contents. Fig. 11 shows three samples from three coarse document groups: a form page, a cover page, and a title page. For a coarse document group, sometimes, we can create a general template for it. For example, most title pages usually contain a title, several authors and their affiliations, and an abstract. It is possible to create a template like " a sentence in largest font size, without words from the organization database, and on the top half of the page is a title; a paragraph below 'ABSTRACT' is an abstract; authors and their affiliations are located between the title and the abstract." In practice, a template for a title page may be more complex. A general template can be used for extracting metadata fields without or with just a few variations. For example, a date just has several formats. It is possible to use a set of general rules to extract date information by matching these date formats.



Form SF298 Citation Data

Report Date (DD MON YYY) 00APR2001	Report Type N/A	Dates Covered (from... to) (DD MON YYY)
Title and Subtitle: TAX DEDUCTIONS Estimates of Taxpayers Who May Have Overpaid Federal Taxes by Not Itemizing		Contract or Grant Number
		Program Element Number
Authors		Project Number
		Task Number
		Work Unit Number
Performing Organization Name(s) and Address(es): General Accounting Office, PO Box 37050, Washington, DC 20013		Performing Organization Number(s): GAO-01-529
Sponsoring/Monitoring Agency Name(s) and Address(es)		Monitoring Agency Acronym
		Monitoring Agency Report Number(s)
Distribution/Availability Statement: Approved for public release, distribution unlimited		
Supplementary Notes		
Abstract: When computing their federal taxes, taxpayers either claim a standard deduction or itemize deductions. These deductions are subtracted from adjusted gross income in determining taxable income. Taxpayers in general claim the type of deduction that is larger because that minimizes their taxable income. In recent years, approximately 70 percent of taxpayers have claimed the standard deduction, while the remaining 30 percent have itemized.		
Subject Terms		
Document Classification: unclassified		Classification of SF298: unclassified
Classification of Abstract: unclassified		Limitation of Abstract: unlimited
Number of Pages: 10		

ARMY RESEARCH LABORATORY

Computationally Based Measures of Amine Azide Basicity and Their Correlation With Hypergolic Ignition Delays

by Michael J. McQuaid

ARL-TR-3121 December 2003

ERDC WQTH-PD-15 September 2004

Winter Community Structure Changes in Frazil Ice and Open Water in Riverine Systems

by K. D. White, C. M. Reynolds, D. B. Ringelberg, K. L. Foley, and L. B. Perry

PURPOSE: This technical note builds on previous work (White et al. 2001, White 2002, White and Melloh 1999) that examined dissolved oxygen (DO) levels and changes in river microbiology during winter, periods of low water temperature, and periods of ice-cover, with the objective of providing guidance for winter water-quality modeling. The results of this study provide information on riverine microbial abundance and diversity in ice-affected rivers and can be applied immediately to improve winter modeling using existing water quality models such as the Corps' CE-QUAL-W2.

BACKGROUND: The U.S. Environmental Protection Agency (USEPA) issued regulations (1985, amended 1992) that implement Section 303(d) of the Federal Water Pollution Control Act known as the "Clean Water Act" (33 U.S.C. 1251-1376; Chapter 758; P.L. 845, June 30, 1948; 62 Stat. 1155 as amended). This section requires that states list waters that are impaired or threatened and for which Total Maximum Daily Loads (TMDL) must be determined. New regulations that were issued in draft form in August 1999 have not yet been implemented. By 1998, many states had listed waters that could be described as impaired or threatened. These totaled more than 300,000 miles

Table 1
Ten Most Frequently Cited Categories of Impairment on USEPA's 1998 303(d) Lists

Impairment	Percent of Total
Sediments	14.84
Pathogens	12.78
Nutrients	11.58
Metals	9.64
Dissolved oxygen	8.18
Other habitat alterations	8.18
Temperature	4.66
pH	4.26
Impaired biotic community	3.16
Pesticides	2.47

Fig. 11. Coarse document groups

Document pages in each coarse classified document group may use different styles. For example, in some title pages, the authors are put together while in some other title pages authors are separated by their affiliations. Hence, according to their styles, these document pages can be classified into more specific groups that we call "fine-grained classified document groups". Fig. 12 shows samples from three fine-grained document page groups. These three samples belong to the same coarse document page group – cover page. However, based on their different styles, the samples can be divided into different fine-grained groups.

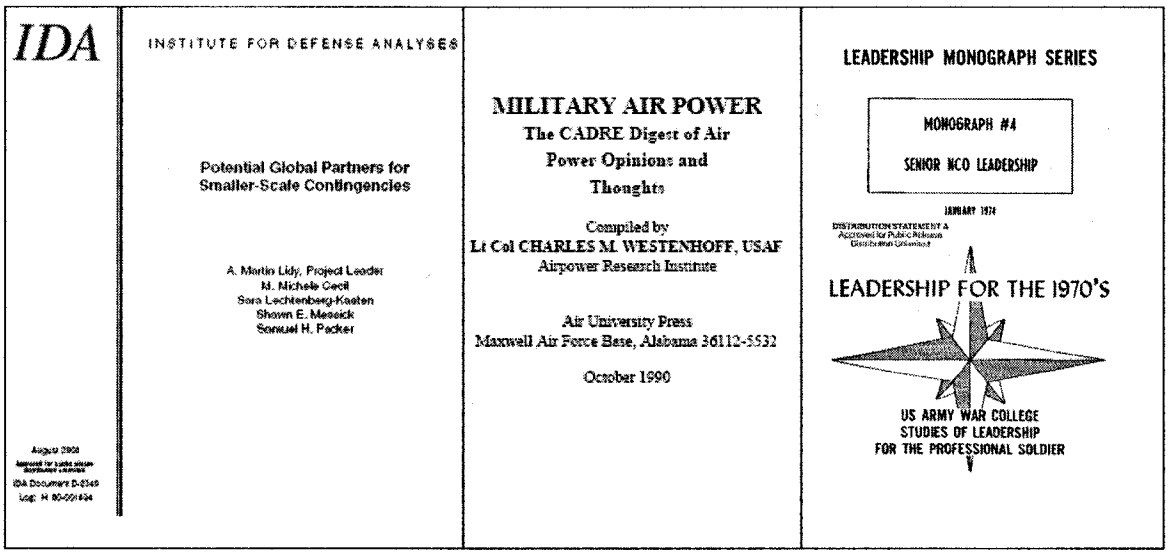


Fig. 12. Fine-grained document groups

For a fine-grained document group, we can create a specific template for it. For example, for the sample in the middle of Fig. 12, we can create a template like "the first line is the report identifier; the second line is the report date; the next text block with larger font size is the title; below the title and above a picture is a set of authors and their affiliation; after the picture is the organization." We call a template for a fine-grained document group a "specific template".

In our current implementation, we first classify metadata pages into two coarse groups, and then classify them into fine-grained groups. We create a template for each fine-grained group. A possible refinement is to add a general template for each coarse group so that we can apply one general template to a metadata page.

3.2.2. Pure Template vs. Integrate Template

In the previous section, we classified templates into two categories based on what kind of document groups they work with. A general template works with a coarse classified document group and a specific template works with a fine-grained classified

document. In this section, we classify templates in another way. A template is a set of rules to instruct metadata extraction engine how to process document pages in a document group. Therefore, it should contain instructions about how to extract metadata. The issue is whether a template should contain instructions for classification, i.e. how to check whether a document group belongs to this group or not. Based on what kind of information is defined in a template, we classify a template into two categories: an integrated template and a pure template.

An integrated template contains instructions both for classification and for metadata extraction. Integrated templates provide some knowledge and simplify the classification process. Given an integrated template, in order to determine whether the document belongs to a certain group, the classification module just needs to check whether a document page matches the template or not. This can increase the classification accuracy. However, in this way, when a new document comes, we need to match it with defined templates one by one. This may cause performance problem if there are too many groups. Furthermore, if a document page does not match any template, it cannot be processed until a new template has been defined.

A pure template contains instructions only for metadata extraction. By using pure templates, we can decouple the classification module from metadata extraction module. Documents are classified into groups and put at different locations (e.g. different folders) by a separate classification module. The metadata extraction module assumes that documents at one location (e.g. a folder) belongs to one group, and applies a pure template directly. Therefore, it is possible to use different classification module for different collections. This approach is desirable in at least two scenarios. First, for some

collections, documents may have already been classified. In this case, we can create a pure template for each group and apply the templates without any classification module. Second, some collection may be classified easily with its specific features. For example, documents may be classified based on their publication organizations if it is known that documents from one organization used the same styles. In this case, we can develop a specific classification module to put documents into groups based on their organization names, and the apply pure templates directly

3.3. Open Research Questions

In previous sections, we described our template-based approach for metadata extraction, its motivation, and different ways to write templates. In this section, we will discuss the issues we want to address.

- Heterogeneity, i.e. how to achieve a high accuracy for a heterogeneous collection;
- Scaling, i.e. how to apply an automated metadata extraction approach to a very large collection;
- Evolution, i.e. how to process a new kind of documents that are added to a collection over time;
- Adaptability, i.e. how to apply an approach to a new document collection;
- Complexity, i.e. how many document features can be handled, and how complex the features should be.

Heterogeneity issue is about how to achieve high metadata extraction accuracy for a heterogeneous collection. To apply a machine-learning approach to a collection with very heterogeneity nature, it is very time-consuming to prepare the training set and train

the model to achieve high accuracy. It is also difficult to apply existing rule-based approach to for a heterogeneous collection because creating a rule set to cover all situations in advance is extremely expensive and time-consuming or even impossible. In this dissertation, we address the heterogeneity issue by applying our template-based approach.

Scaling issue is about how to apply an approach for metadata extraction to a large collection. The performance issue may not be important for an approach to work with a small collection. However, it is very important when an approach works with a very large collection. Assume that checking whether a text line again a rule takes one millisecond, we have 1000 rules, and we have 10,000 documents with average of 1000 lines. It will take 10,000,000 seconds or about 115 days to check each line against each rule. In our template-based approach, with the help of document classification, a small set of rules instead of the whole rule set are applied to a document. Furthermore, because each document group contains similar documents, processing documents in a group only requires a very small number of rules.

Evolution issue may occur when new kinds of documents are added to a collection over time. The change of documents requires changing the rules for metadata extraction accordingly. Rule-based approaches that hardcode the rules have problems. For this kind of approaches, in order to change the rules, they need change their metadata extraction code and may require recompiling the code. Some rule-based approaches decouple the rules from metadata extraction code but use one set of rules for processing all documents in a collection. This kind of approaches also have problems with processing new kind of document, because it is time-consuming for them to find which

rules should change and make sure that the changes do not affect the processing of old documents. Existing machine learning approaches for metadata extraction also have the problem with a dynamic collection. A machine learning approach needs to train its model again in order to process new kind of document. Furthermore, there is potential lag time during which accuracy decays until sufficient training instances acquired.

Adaptability issue is about how to apply an approach to a different collection. Rule-based approaches tend to have difficulties to adapt to a different. Rule-based approaches that hardcode the rules have adaptability problem. Sometime, the efforts to adapt them to a different collection are even almost the same as the efforts to build another system from scratch. Rule-based approaches that use one rule base for all documents in a collection also have problems. Changing a large rule base to work for a different collection is usually expensive and time-consuming. Our template-based address the adaptability issue in two ways. First, it decouples the rules from metadata extraction code so that users can change the rules without changing the code. Second, it classifies documents into groups and allows users to create a template for a group. Therefore, rule creation is simpler.

Complexity issue is to address how complex a template is required in order to achieve desirable accuracy while save human effort as much as possible. A simple template is easy to create. However, it requires classifying documents into more fine-grained groups. Therefore, more groups will be generated. A complex template can be used for a general group. Therefore, the number of groups will be less. However, it requires more time to create a template. Which approach saves more human efforts, a simple template approach or a complex template approach?

CHAPTER IV

DOCUMENT CLASSIFICATION

As we have already seen, our template-based approach for metadata extraction first classifies the documents into groups, and then writes a template for each group to specify how to extract metadata from the documents in this group. In this chapter, we will describe our document classification approach. In this research, we classify documents into groups based on the similarity between their metadata pages. We divide metadata pages into two coarse groups: structured metadata pages and unstructured metadata pages. A *structured metadata page* is a metadata page in which almost all metadata fields can be identified by a set of fixed labels. Any metadata page that is not a structured metadata page is *an unstructured metadata page.* We use different approaches to classify documents from these two coarse groups into fine-grained groups. In the rest of this chapter, for simplicity, we will user the term "group" for the term "fine-grained groups", and use the term "category" for the term "coarse group". A new term "block" will be used in this chapter. In this dissertation, unless we specify its meaning explicitly, a "block" in a page has the similar meaning to the "region" defined in [18], i.e. blocks are "split by means of cuts along separators (e.g. lines)" and "cuts along white spaces" [18]. We use the term "block" instead of the term "region" because Scansoft Omnipage 14 pro Office used the text "region" as an element in its XML format.

The rest of this chapter is organized as follows: in section 4.1, we will present an overview of document classification for metadata extraction; in section 4.2 and section 4.3, we will describe how to locate and classify a structured metadata page and an

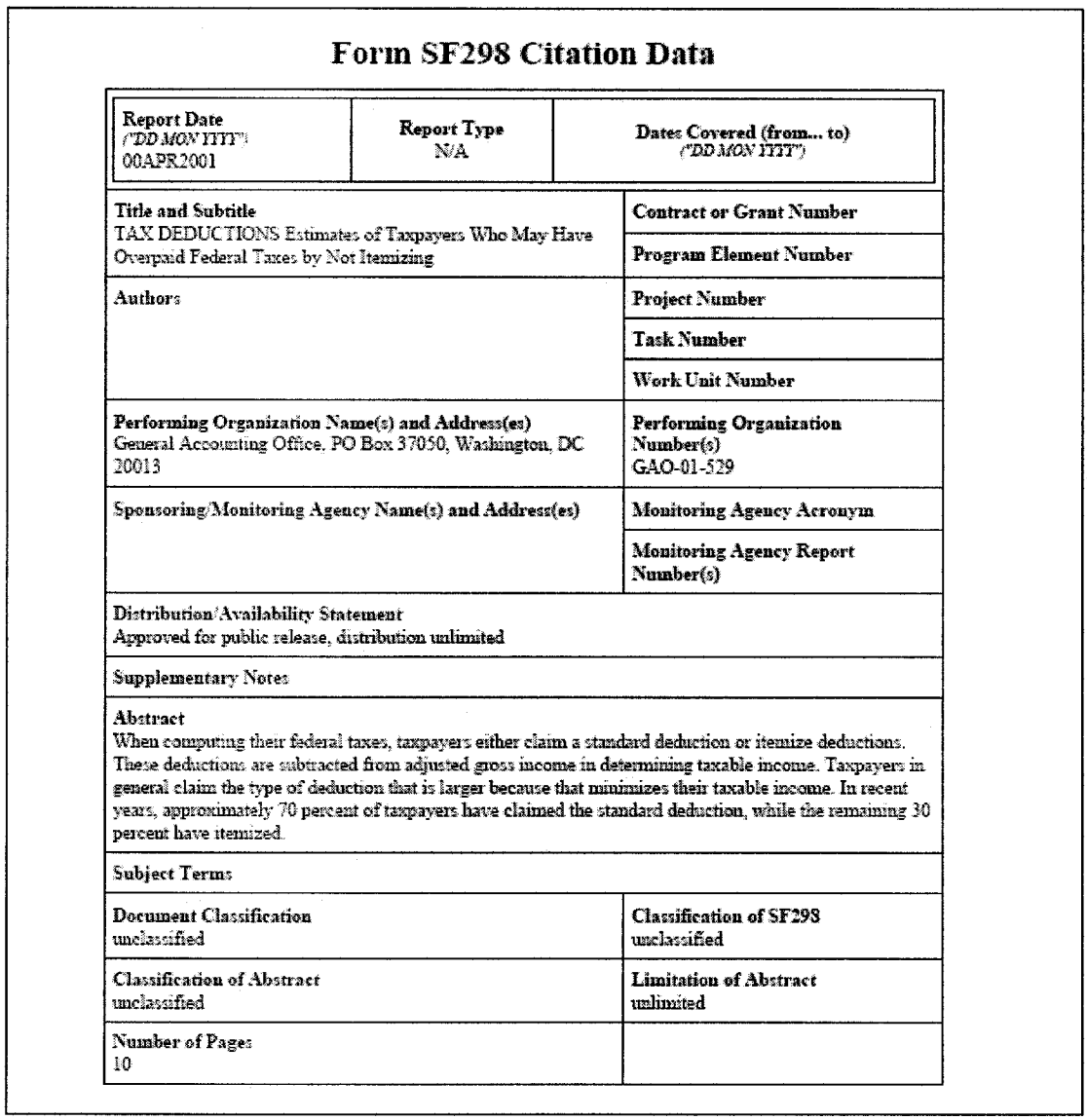
unstructured metadata page respectively; in section 4.4, we will give a summary of this chapter

4.1. Document Classification for Metadata Extraction

In our research, the objective of classifying documents into groups is to ease the task of metadata extraction for a heterogeneous collection. Documents are classified into groups based on the similarity of their metadata pages so that we can develop a simple template to extract metadata from documents in a group. We define two kinds of similarity for document classification in our research: *visual similarity* and *content similarity*. The *visual similarity* is the similarity of the geometrical arrangement of blocks (both text and graphics) on the metadata page as well as the typographic features of the text. Some examples of the typographic features are font size, text alignment, text height, and line spacing. The *content similarity* is the similarity of the occurrences of special labels (e.g. "ABSTRACT", "Title", and "Subject"), the occurrences of special text patterns (e.g. "three letters followed by nine digits"), the occurrences of the words from special databases (e.g., a word from a dictionary of English last names) in the text, and the statistical features of the text (e.g. a text with more than 50% letters in upper case).

In this our research, the task of document classification includes how to find metadata pages from documents and how to classify metadata pages into groups. The characteristics of the metadata pages may affect how to locate metadata pages and how to classify documents into groups. For metadata pages that use fixed labels to organize most of the information on the page, it is possible for us to identify such metadata pages and classify them into groups by their label sets. For metadata pages that use few fixed labels or do not use fixed labels at all, using fixed labels only may not be sufficient to locate

them and classify them into groups. In this research, we divide metadata pages into two categories: structured metadata pages and unstructured metadata pages. Based on the different characteristics of the metadata pages from different categories, we use different strategies to locate and classify them into groups.



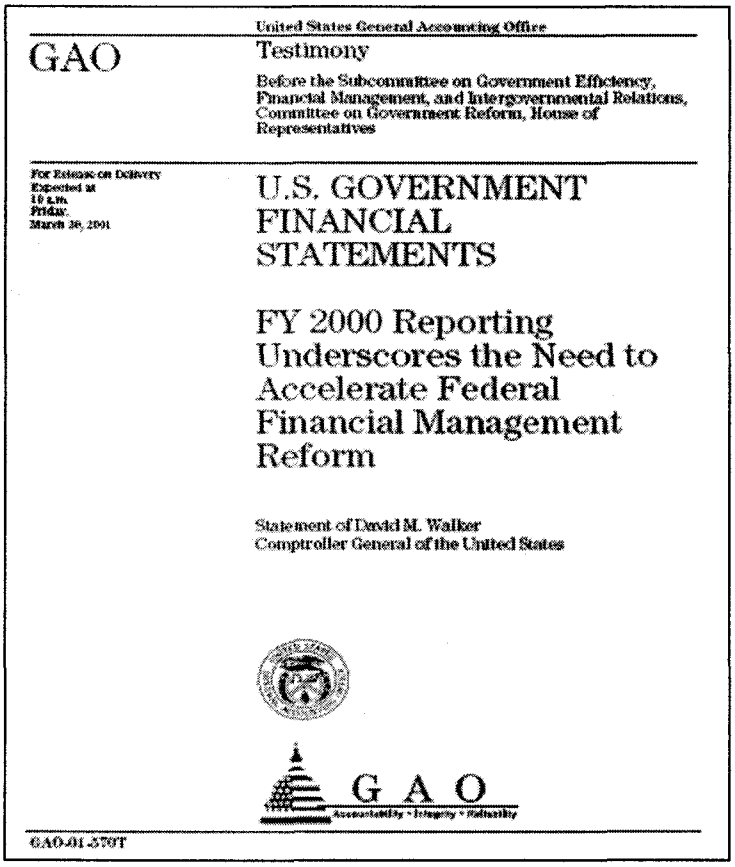
Form SF298 Citation Data

Report Date ("DD MON YYYY") 00APR2001	Report Type N/A	Dates Covered (from... to) ("DD MON YYYY")

Title and Subtitle TAX DEDUCTIONS Estimates of Taxpayers Who May Have Overpaid Federal Taxes by Not Itemizing	**Contract or Grant Number**
	Program Element Number
Authors	**Project Number**
	Task Number
	Work Unit Number
Performing Organization Name(s) and Address(es) General Accounting Office, PO Box 37050, Washington, DC 20013	**Performing Organization Number(s)** GAO-01-529
Sponsoring/Monitoring Agency Name(s) and Address(es)	**Monitoring Agency Acronym**
	Monitoring Agency Report Number(s)
Distribution/Availability Statement Approved for public release, distribution unlimited	
Supplementary Notes	
Abstract When computing their federal taxes, taxpayers either claim a standard deduction or itemize deductions. These deductions are subtracted from adjusted gross income in determining taxable income. Taxpayers in general claim the type of deduction that is larger because that minimizes their taxable income. In recent years, approximately 70 percent of taxpayers have claimed the standard deduction, while the remaining 30 percent have itemized.	
Subject Terms	
Document Classification unclassified	**Classification of SF298** unclassified
Classification of Abstract unclassified	**Limitation of Abstract** unlimited
Number of Pages 10	

Fig. 13. A structured metadata page

A structured metadata page uses a set of labels to identify most of its metadata fields. Fig. 13. shows one structured metadata page sample. This document page uses one label (e.g. "Report Date") to indicate the location of each metadata field. Our strategy of processing documents that contain structured metadata pages is to define the label sets in advance and to classify the documents into groups based on these label sets.



United States General Accounting Office

GAO

Testimony

Before the Subcommittee on Government Efficiency, Financial Management, and Intergovernmental Relations, Committee on Government Reform, House of Representatives

For Release on Delivery
Expected at
10 a.m.
Friday,
March 30, 2001

U.S. GOVERNMENT FINANCIAL STATEMENTS

FY 2000 Reporting Underscores the Need to Accelerate Federal Financial Management Reform

Statement of David M. Walker
Comptroller General of the United States

G A O

GAO-01-579T

Fig. 14. An unstructured metadata page

An unstructured metadata page does not use labels for most of its metadata fields. The identification of some metadata fields in an unstructured metadata page relies on the arrangement of the components on this page, the typographic features (e.g. the text in the largest font size), as well as the content of the text (e.g. the text starting with a month name). Fig. 14 shows one unstructured metadata page.

In our current implementation, we create a set of rules based on statistics features (e.g. the number of words, the number of lines, the fonts used, and occurrence of person names, etc.) to locate an unstructured metadata page. An assumption here is that statistically an unstructured metadata page tends to be different from a page that is not a metadata page (e.g. few words, few lines, large fonts, etc.). A possible alternative is to use statistical machine-learning techniques.

We provide two methods to classify documents into groups based on the similarity of their unstructured metadata pages. The first method is to classify documents into groups with the pre-defined knowledge of their unstructured metadata pages. This method is similar to our approach of classifying structured metadata pages, since both of them are based on pre-defined knowledge. However, their pre-defined knowledge is different. The pre-defined knowledge here is not limited to the features of the text. Instead, it includes the features of the blocks in a page, the relationship among these blocks, the sample pages, and the similarity threshold value based a certain method to compute the similarity of the page. Basically this kind of knowledge specifies a set of criteria for each known group so that only the pages meets these requirements are classified in this group. With this method, it is possible that some unstructured page may not be classified into any group since it does not meet the requirements of any group.

However, the knowledge is extensible and the knowledge is a configuration file, which is loaded at running time. Therefore, this method can be applied to a collection incrementally. With increasing the knowledge, more and more unstructured metadata pages can be resolved. The details about this method will be described in Section 4.3.1.

The other method is to classify unstructured metadata pages into groups without prior knowledge. It computes the similarity between an unstructured metadata page and the representative page of each existing group. It classifies the page into the group with the largest similarity if the similarity is larger than the pre-defined threshold. If the largest similarity is smaller than the pre-defined threshold, it will generate a new group and assign the page as the representative page of this new group. This method will classify every document into a group.

As we described in section 2.1, there are some existing approaches ([17], [31], [36], and [51]) of classifying documents into equivalence groups. Our approach is different from them in two aspects. First, our approach addresses issue of how to locate a metadata page in a document. It first locates metadata pages from documents, and then classifies the documents into groups based on the similarity of their metadata pages. Existing approaches ([17], [31], [36], and [51]) of document classification did not address issue of locating a metadata page in a document. They either assumed the first page of a document is a metadata page, or assumed the input is a document page instead of a document. This makes them not suitable for processing documents whose metadata pages cannot be identified simply by the page number. Second, our approach divides metadata pages into two categories: structured metadata pages and unstructured metadata pages, and uses different strategies to process them accordingly. For documents with structured

metadata pages, our strategy is to locate the structured metadata pages and classify them into groups with the label sets that are defined in advance. For documents with unstructured metadata pages, we provide two methods. One method is to classify documents with prior knowledge, and the other method is to classify documents without prior knowledge.

In the rest of this chapter, we will describe how to classify documents with structured metadata pages and how to classify documents with unstructured metadata pages respectively. In our research, since we focus on extracting metadata from one metadata page, a document is classified into a group based on one metadata page. Hence, classifying documents into groups is to classify metadata pages into groups.

4.2. Structured Metadata Page Location and Classification

A structured metadata page uses a set of fixed labels to identify their metadata fields. Our strategy is to use the label sets to locate the structured metadata pages and put documents whose structured metadata pages have the same label set into one group. In our approach, a metadata field is extracted based on the locations of its label and its neighbors. For example, in Fig. 15, the "Field 2" can be extracted as "the text located below Label 2, above Label 4, on the right of field 1 and on the left of label 3". An assumption here is that the metadata fields on structured metadata pages with the same label set are arranged in the same way. A possible refinement is to include information about the relative locations of the labels in the templates.

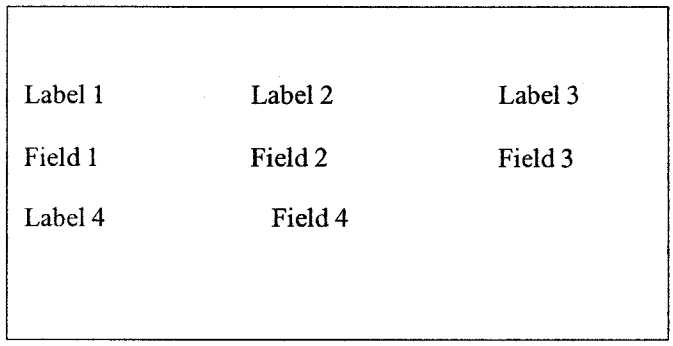
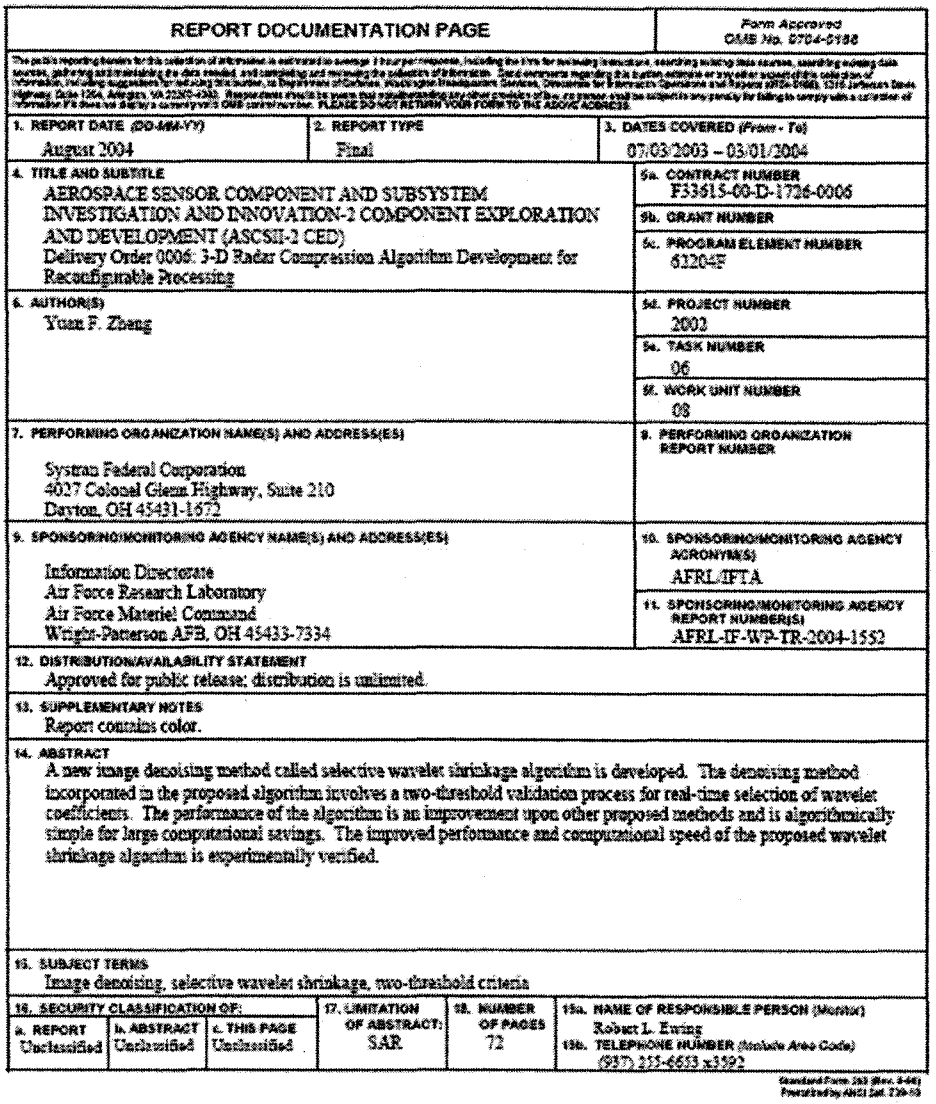


Fig. 15. Metadata extraction with label locations

To use our approach with a collection, we first get the knowledge about the label sets used by the structured metadata pages in this collection. Such knowledge may not be easy to obtain for a large collection. In this case, we randomly select a relative small document set from the collection so that we can check the documents one by one to get the label sets used by this document set. Then we write a template for each label set. A page matches a template if the page contains the text that is same to the label set defined in the template. A structured metadata page may be located in a document based on its content, i.e. a page is a structured metadata page if it matches at least one template. Depends on which template the structured metadata page matches, its associated document can be classified into a group accordingly. It is possible that one page may match more than one template if the label set specified in one template is a subset of that specified in another template. In this case, the matched template with the largest label set will be chosen. Fig. 16 shows another structured metadata page. The structured metadata pages in Fig. 16 and Fig. 13 use different sets of labels.



REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. REPORT DATE (DD-MM-YY)	2. REPORT TYPE	3. DATES COVERED (From - To)
August 2004	Final	07/03/2003 – 03/01/2004

4. TITLE AND SUBTITLE
AEROSPACE SENSOR COMPONENT AND SUBSYSTEM INVESTIGATION AND INNOVATION-2 COMPONENT EXPLORATION AND DEVELOPMENT (ASCSII-2 CED)
Delivery Order 0006: 3-D Radar Compression Algorithm Development for Reconfigurable Processing

5a. CONTRACT NUMBER: F33615-00-D-1726-0006
5b. GRANT NUMBER:
5c. PROGRAM ELEMENT NUMBER: 62204F

6. AUTHOR(S): Yuan F. Zheng

5d. PROJECT NUMBER: 2002
5e. TASK NUMBER: 06
5f. WORK UNIT NUMBER: 08

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Systran Federal Corporation
4027 Colonel Glenn Highway, Suite 210
Dayton, OH 45431-1672

8. PERFORMING ORGANIZATION REPORT NUMBER:

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Information Directorate
Air Force Research Laboratory
Air Force Materiel Command
Wright-Patterson AFB, OH 45433-7334

10. SPONSORING/MONITORING AGENCY ACRONYM(S): AFRL/IFTA

11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S): AFRL-IF-WP-TR-2004-1552

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
Report contains color.

14. ABSTRACT
A new image denoising method called selective wavelet shrinkage algorithm is developed. The denoising method incorporated in the proposed algorithm involves a two-threshold validation process for real-time selection of wavelet coefficients. The performance of the algorithm is an improvement upon other proposed methods and is algorithmically simple for large computational savings. The improved performance and computational speed of the proposed wavelet shrinkage algorithm is experimentally verified.

15. SUBJECT TERMS
Image denoising, selective wavelet shrinkage, two-threshold criteria

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT:	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON (Monitor)
a. REPORT	b. ABSTRACT	c. THIS PAGE			Robert L. Ewing
Unclassified	Unclassified	Unclassified	SAR	72	19b. TELEPHONE NUMBER (Include Area Code) (937) 255-6633 x3592

Standard Form 298 (Rev. 8-98) Prescribed by ANSI Std. Z39-18

Fig. 16. Another structured metadata page sample

4.2.1 Structured Metadata Page Model

In this research, we divide the components of a structured metadata page into three parts: a caption, field names, and field values. A *caption* is a fixed label associated with a structured metadata page instead of a metadata field, e.g. the label "REPORT DOCUMENTATION PAGE" in Fig. 16. A *field name* (e.g. the label "REPORT TYPE"

in Fig. 16) is a fixed label to identify a metadata field on a structured metadata page. A *field value* (e.g. "Final") is the value of a metadata field.

4.2.2 Template of Structured Metadata Page

Structured metadata pages are located from a document by looking for pages that contain text same to one of the pre-defined label sets. A document may have more than one structured metadata page, and one structured metadata page may match more than one templates. In this dissertation, we classify the document into a group based on the matched templates with the largest defined label set. A part of our template language schema for structured metadata pages is shown in Fig. 17.

```
<xs:complexType name="OneForm">
  <xs:sequence>
    <xs:element name="match" minOccurs="0" maxOccurs="unbounded" type="StrMatch"/>
    <xs:element name="fixed" minOccurs="0" maxOccurs="unbounded" type="Fixed" />
    <xs:element name="extracted" minOccurs="0" maxOccurs="unbounded" type="Extracted" />
    <xs:element name="exclude" minOccurs="0" maxOccurs="unbounded" type="xs:string" />
  </xs:sequence>
  <xs:attribute name="max" type="xs:int" />
</xs:complexType>
<xs:complexType name="StrMatch">
  <xs:sequence>
    <xs:element name="line" minOccurs="0" maxOccurs="unbounded" type="xs:string" />
  </xs:sequence>
  <xs:attribute name="max" type="xs:int" />
</xs:complexType>
<xs:complexType name="Fixed">
  <xs:sequence>
      <xs:element name="field" minOccurs="0" maxOccurs="unbounded" type="Field"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="Field">
  <xs:sequence>
    <xs:element name="line" minOccurs="0" maxOccurs="unbounded" type="xs:string"/>
  </xs:sequence>
  <xs:attribute name="num" type="xs:string" />
  <xs:attribute name="optional" type="xs:string" />
</xs:complexType>
```

Fig. 17. Structured metadata page template schema

The entire XML schema is available in Appendix A. In a template, an element "match" is used to specify the value of a structured metadata page's caption. To match a template, any page should contain the text specified by the element "match". The element "match" has an attribute "max" and a child element "line". The attribute "max" is used to improve the performance by limiting the candidates for the pre-defined caption. Its value indicates how many lines in the top of a page are candidates for the caption. The value of the attribute "max" should be positive except the special value –1, which stands for "all lines". For example, if the caption of a structured metadata page from a group is always located within the first 5 lines, the value of the attribute "max" can be set to 5 so that only the first 5 lines on each page will be checked. The child element "line" is used to specify the text of the caption. The element "fixed" is for specifying the value of the field names used in a structured metadata page. It contains a sequence of the element "field", which specifies one fixed label for one field. The follows are the list of the children of the "field" element:

- The attribute "num": specifies the identifier of the field that makes it possible to define more than one label for one field;
- The element "line": specifies the fixed label of the field name;
- The attribute "optional": its value is a text string with two characters. The text between these two characters can be ignored during the process of matching a text in a page with the label specified by the element "field". For example, if the value of the attribute "optional" is the text string "()". The text string "Abstract (maximum 200 words)" will match the specified label "Abstract", since the text between "(" and ")" can be ignored.

A part of a template sample is shown in Fig. 18. It defines the caption and field names for a structured metadata page group, e.g. the caption should be "REPORT DOCUMENTATION PAGE", and the caption is located within the first five lines on a page.

```
....
<formForm max="-1">
  <match max="5">
    <line>REPORT DOCUMENTATION PAGE</line>
  </match>
  <fixed>
   <field num="1"><line>1. AGENCY USE ONLY (Leave blank)</line></field>
   <field num="1"><line>1. AGENCY USE ONLY</line></field>
   <field num="2"><line>2. REPORT DATE</line></field>
   <field num="3"><line>3. REPORT TYPE AND DATES COVERED</line></field>
   <field num="4"><line>4. TITLE AND SUBTITLE</line></field>
   <field num="5"><line>5. FUNDING NUMBERS</line></field>
   <field num="6"><line>6. AUTHOR(S)</line></field>
   <field num="7"><line>7. PERFORMING ORGANIZATION NAME(S) AND
ADDRESS(ES)</line></field>
    <field num="8"><line>8. PERFORMING ORGANIZATION REPORT NUMBER</line></field>
    <field num="9"><line>9. SPONSORING / MONITORING AGENCY NAME(S) AND
ADDRESS(ES)</line></field>
    <field num="10"><line>10. SPONSORING / MONITORING AGENCY REPORT
NUMBER</line></field>
   <field num="11"><line>11. SUPPLEMENTARY NOTES</line></field>
   <field num="12a"><line>12a. DISTRIBUTION / AVALILABILITY
STATEMENT</line></field>
   <field num="12b"><line>12b. DISTRIBUTION CODE</line></field>
   <field num="13"><line>13. ABSTRACT (Maximum 200 Words)</line></field>
   <field num="13"><line>ABSTRACT (Maximum 200 Words)</line></field>
   <field num="13"><line>13. ABSTRACT</line></field>
   <field num="14"><line>14. SUBJECT TERMS</line></field>
   <field num="15"><line>15. NUMBER OF PAGES</line></field>
   <field num="16"><line>16. PRICE CODE</line></field>
   <field num="17"><line>17. SECURITY CLASSIFICATION OF REPORT</line></field>
   <field num="18"><line>18. SECURITY CLASSIFICATION OF THIS PAGE</line></field>
   <field num="19"><line>19. SECURITY CLASSIFICATION OF ABSTRACT</line></field>
   <field num="20"><line>20. LIMITATION OF ABSTRACT</line></field>
</fixed>
.....
```

Fig. 18. A Template Sample for Structured metadata page

More than one "field" with the same identifier can be used to define the labels for one field. In that case, a string text match any of these defined fixed labels is the field name. For example, the first two lines starting with "<field" in Fig. 18 have the same identifier "1". They define that the field name of the field "1" should be either "1. AGENCY USE ONLY (Leave blank)" or "1. AGENCY USE ONLY".

4.2.3. Classification with Imperfect Input

The process of classifying one document into a group based on its structured metadata page is shown in Fig. 19. It loads all templates that define the fixed label sets, and tries to match all pages with all templates. If one page contains the text same to the label set specified in a template, the template is added to the candidate set. If the final candidate set is not empty, the document is classified into the group associated with the candidate with the largest label set.

```
Load all templates
For each page
{
For each template
        {
        If the page contains the text same to the label set described in the template
        {
            the matched template is added to the candidate set
        }
}
If the candidate set is empty
{
            The document is unresolved
}
Else
{
        Classify the document into the group associated with the template with the largest label
        set
}
```

Fig. 19. Structured metadata page classification

As shown in Fig. 19, if the candidate set is empty, the document cannot be resolved. There are two reasons. The first one is that the document does not have a structured metadata page. The other reason is that the structured metadata page in the document cannot be identified with current knowledge. To process these unresolved documents, we can either add more templates or simply leave them to be processed by using other methods.

Fig. 19 shows the process of classifying a document into a group by its structured metadata page in an ideal situation, where there are no OCR errors, and each field name has been identified correctly. In this ideal situation, a page matches a template if the page contains the text same to the label set specified in the template. In practice, we have to handle imperfect information during matching a page with a template.

The first issue is how to handle OCR errors. The OCR errors pose challenge to match text in a document with pre-defined fixed labels. For example, for some documents in our test bed, the OCR result of the text "REPORT DOCUMENTATION PAGE" is "REPORT DOCUMENTA110N PAGE". It is not desirable that a structured metadata page cannot be classified into its group due to minor OCR errors. To make our approach works with documents that may have minor OCR errors, we apply Levenshtein distance [3] in our string match algorithm. Levenshtein distance, also known as "edit distance", is a way to measure the similarity between two strings. The Levenshtein distance of two strings is the smallest number of single-character insertions, deletions, and substitutions required to change one string to another [3]. Instead of matching strings exactly, we consider two strings are matched if their edit distance is smaller than a certain threshold value. This brings up another issue, i.e. how to choose the threshold value of

the edit distance between two strings. Using a fixed threshold value is undesirable since different field names are different in length. The threshold value for strings with 200 characters should be different from the value for string with 10 characters. In addition, the percentage of matched words of two text strings also provides clues about how similar they are. Hence, we propose an algorithm to determine the threshold value of the Levenshtein distance between two strings dynamically based on their lengths and the percentage of matched words. Our algorithm of string match with Levenshtein distance is shown in Fig. 20. We also call our algorithm "similar match".

```
//String match with Edit distance
// return true if matched
SimiMatch (String s1, String s2)
{
        distance=EditDistance (s1, s2);
        wc= the larger of the number of words of s1 and s2;
        len= the larger length of s1 and s2;
        len= len/10;
        threshold= max (wc, len);

        //allows 1 error per word or 1 error per 10 characters
        if (distance < threshold) return true;

        wc2= number of words matched in s1 and s2;

        // increase the threshold if 75% words are matched
        if(wc2 > wc*0.75){
                if(distance<threshold*1.5) return true;
        }

        return false;
}
```

Fig. 20. String match with edit distance

The second issue is how to handle the damaged labels of the field names and the caption. For example, stamps or handwritings may damage some field labels or the

caption in a document page in print. That means sometimes we cannot locate all fields by the fixed label set defined in a template even though we apply our similar match algorithm. To address this issue, our approach matches a document page with a template with similarity. In our approach, a document page is considered a candidate of a certain structured metadata page if it matched some of the labels. For each candidate, a confidence score is assigned to each candidate based on how well they are matched. The candidate with the largest confidence score is chosen as the structured metadata page. In this way, the match of a document page with a template does not require all fixed labels to be matched. In our implementation, a document page is considered a candidate if one of the following holds:

1) Its caption is exactly matched with pre-defined caption;
2) Its caption and more than 5 field names are matched with our similar string match algorithm (i.e. strings are matched if their edit distance is smaller than the threshold value);
3) More than 10 field names are matched with our similar string match algorithm.

The confidence score of a candidate is computed by the following equation:

$$\frac{match\_f + partial\_f}{total\_f}$$

Where

match_f is the number of fields whose field names are exactly matched with pre-defined labels;

partial_f is the number of fields whose field names are matched with the edit distance smaller than the threshold value;

total_f is the total number of the fields as defined in the template.

A possible refinement could be to assign different weights to *match_f* and *partial_f* even though we assign the same weight to them in our current implementation,

The third issue is the granularity of matching. The OCR tool *Scansoft Omnipage Office 14 pro* produces a document in a hierarchical structure: document – page – zone/ region – paragraph – line – word – character. We can match a field name at the paragraph, line, word, or character level. In practice, we choose to match the field names on the line level, since the OCR tool sometimes has problems with determining the paragraph boundaries correctly, and the algorithm for matching a field name on the word level or the character level tends to be more complicated. To work on the line level, we need to handle two issues:

- Partial line field name: a field name is a part of one line in the OCR output;
- Multi-line field name: a field name goes beyond one line.

We solve the partial line field name problem by checking whether a sub string of a line is a field name. Matching field names that may go beyond one line is a challenge. First, the lines in an OCR output may not occur in the same order as on the original page. Developing an algorithm to reorder the text to guarantee that the lines in the OCR output have the same order as the lines in the original page is complicated and will be a future refinement. In our current approach, we handle this problem during matching a field name (i.e. we do not assume that the next line in an OCR output is the next line that appears on a page).

Second, a multi-line field name may be split at different locations. Table 2 shows three field name samples, which have different appearances. A specified field name "8. PERFORMING ORGANIZATION REPORT NUMBER" may be split into more than one line at various points. The sample 1 in Table 2 shows two variations. Even though in both variations the field names are split into two lines, they are separated at different places. One is separated at the end of the word "REPORT" while the other is separated at the end of the word "ORGANIZATION".

TABLE 2
SAMPLE OF DIFFERENT APPEARANCES OF FIELD NAMES

Samples	Same string with different appearance
Sample 1	8. PERFORMING ORGANIZATION REPORT NUMBER
	8. PERFORMING ORGANIZATION REPORT NUMBER
Sample 2	9. SPONSORING/MONITORING AGENCY NAME AND ADDRESS
	9. SPONSORING/MONITORING AGENCY NAME AND ADDRESS
Sample3	17. LIMITATION OF ABSTRACT
	17. LIMITATION OF ABSTRACT

To address the multi-line field name problem, we match a field name part by part. The detail of our multi-line field match algorithm is shown in Fig. 21.

```
MultiLineFieldMatch (String fieldname, List lines){
        Result=null;
        Left=fieldname;
        For each( Line in lines){
                Len=length(Left);
                If(Len==0) return Result;
                If (Line equals Left){
                   Result=Result + Line;
                   return Result;
                }
                Else If (Left starts with Line){
                    If (Line is the next line of PreLine){ *
                          Result=Result + Line;
                          Left=Left-Line;
                          PreLine=Line;
                    }
                }

        Len=length(Left);
        If(Len==0) return Result;
        Else return null;
}
```

* line B is the next line of line A if all the following are true:

- The vertical distance between A and B is less the large height of A and B
- There is no other lines between A and B
- B locates below of A
- There is an overlap if projecting A and B on X axis

Fig. 21. Algorithm for matching field name

To improve the performance of our algorithm, we reorder the text by sorting lines by their coordinates as follows:

- If two lines have different y-coordinates, a line with smaller y-coordinate will occur first;
- If two lines have the same y-coordinate, the line with smaller x-coordinate will occur first.

With this implementation, lines that appear closely on the original page tend to be close in the OCR output.

4.3. Unstructured Metadata Page Location and Classification

An unstructured metadata page lacks fixed labels for most of its metadata fields. In this dissertation, the locating of unstructured metadata pages mainly relies on statistical features (e.g. a page contains less than 10 lines), the arrangement of components on this page (e.g. a page with a lot of spaces), and the typographic features (e.g. large line height).

4.3.1 Unstructured Metadata Page Location

To locate an unstructured metadata page in a document, we use rules to describe the characteristics of the unstructured metadata page that we are interested in. The process of locating unstructured metadata pages in documents involves the following steps:

1) Analyze the characteristics of the unstructured metadata pages;
2) Write rules to define the characteristics;
3) Locate structured metadata pages in documents by using defined rules.

We will use cover pages in our DTIC collection as a sample to illustrate the process involved in locating an unstructured metadata page in a document. A *cover page* precedes the start of the document body and consists of metadata. It usually contains information about title, publisher, authors and affiliations, etc. Fig. 22 shows two cover page samples in our DTIC collection.

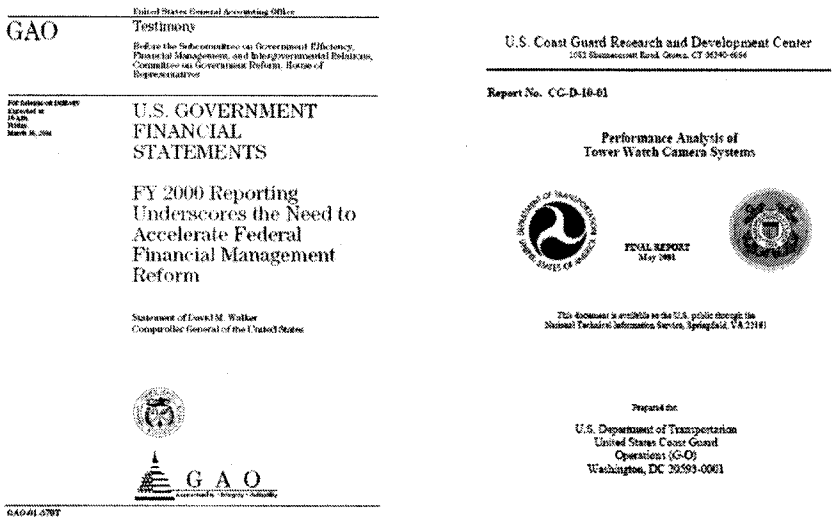


Fig. 22. Cover page samples

As the first step, we analyzed the cover pages in our test bed, and found that they have the following properties:

- A cover page in our test bed is usually one of the first five pages;
- A cover page in our test bed usually contains fewer lines; therefore it is possible to set a threshold value so that any page with the number of lines larger than the threshold can be removed from the candidates of the cover page;
- A cover page in our test bed usually contains fewer words; therefore it is possible to set a threshold value so that any page with the number of words larger than the threshold can be removed from the candidates of the cover page;
- A cover page in our test bed usually contains more than three blocks;

- The layout of a cover page in our test bed is usually balanced. Authors rarely put all information in a small area and keep all other places blank. They tend to put information among the top, middle and bottom of a page. For example, each part (the top, the middle, or the bottom) of a cover page has something (either image or text) in our samples shown in Fig. 22;
- A cover page in our test bed contains a few lines that contain numbers (address, date, etc.);
- A cover page in our test bed contains few lines ending with a digit;
- A large number of lines in a cover page contains text in a title case only;

Accordingly, we wrote a set of rules. In our current implementation, any page that meets all the following rules is identified as a cover page:

a. Page number <= 5;

b. Number of blocks >=3;

c. Balanced;

d. Number of lines <= 30;

e. Number of letters <= 700;

f. Number of words <=200;

g. Number of lines that contains digits <=9

h. Number of lines that ends with digits <=4

i. Average number of words per line <=10

j. More than 50% of lines whose texts are in title case

4.3.2. Unstructured Metadata Page Classification

After unstructured metadata pages have been located, they can be classified into groups based on their visual and content similarity. We use two approaches to classify unstructured metadata pages into groups: knowledge-based classification and classification without prior knowledge.

4.3.2.1. Knowledge-based Classification

In this approach, we define a set of page formats in advance, and classify a document into a group based on which page format is matched. A page format includes the information about the features of the blocks, the relationship among the blocks, the sample pages, and the threshold valued of the similarity. Basically, a page format consists of a set of criteria (i.e. features of blocks, block relations, similarity threshold values). Only the pages that meet all these requirements are classified into the group associated by this page format. We provide several ways to measure the similarity of a page and a sample page. A pair of the threshold value and how to measure the similarity serves as one requirement for any page below to a certain group. For different page formats, we can specify different ways to measure the similarity with different threshold values. For example, for one page format, we may think the components (text or graphic blocks) of a page are important. Therefore, we can measure the similarity based on the components on the pages. For another page format, the font sizes may be used to distinguish pages in its group from other pages. Then we can measure the similarity based on the font sizes. We can also use multiple ways to measure the similarity.

In this approach, an unstructured metadata page is classified into a group if it matches at least one of these pre-defined formats. If a page fails to match any defined

formats, it will be assigned to a special group called "unknown". An unstructured metadata page matches a pre-defined format if it meeting all the criteria specified in the page format. A page format is defined in XML format. The elements of the XML schema for a page format are shown in Fig. 23.

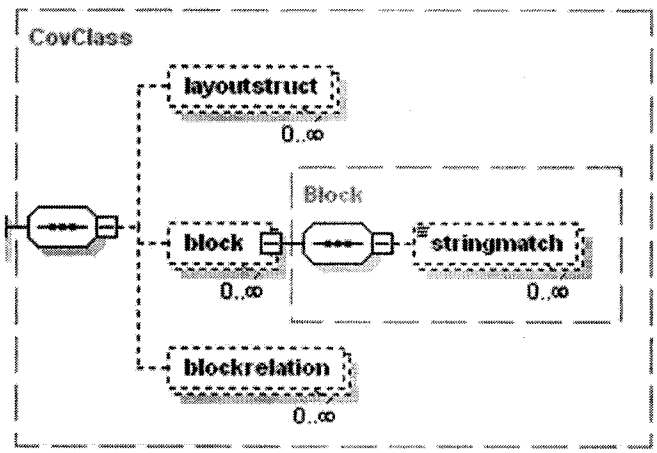


Fig. 23. XML schema for defining document page

The entire XML schema is included in Appendix B. In this schema, an element "covclass" is used for each page format. The element "covclass" consists of:

- Attribute "name" that allows users to specify a name for each page format;
- Element "layoutstruct" that is used to specify which method to compute the similarity between a page and a sample page and the threshold value of the similarity;
- Element "block" that is used to specify the features of a block;

- Element "blockrelation" that is used to define the relationships between two blocks.

The element "covclass" can have any number of the elements "layout", "block", and "blockrelation". In the rest of this subsection, we will introduce these elements in detail.

The element "layoutstruct" is used to describe a page by reference to a sample. All document pages in a document group should be similar to this sample. The element "layoutstruct" has three attributes: "compare", "min", and "type". The "compare" attribute is used to define the file name of the sample. The attribute "min" is used to specify the minimum value of the similarity between a document page and the sample in a document group. In the other words, if the similarity of an unstructured metadata page in a document and the sample page is less than the minimum value, the document does not belong to the document group associated to this sample. The value of the attribute "min" is a real number between zero and one. The attribute "type" defines the similarity measurement between a page and the sample. Its value can be "blocktype", "bin", or "graphmatch". These values stand for three different ways to compute the similarity of two pages.

- "blocktype": a page is converted to an MXY Tree [36] (a horizontal cut has higher priority than a vertical cut so that each page has a unique MXY Tree). Then a sequence of block types ("g" for a graphic block or "t" for a text block) is extracted from the MXY tree. The similarity of two pages is based on the edit distance [3] between their block type sequences. Given

two sequences s1 and s2, the similarity

$$Sim(s1,s2) = 1 - \frac{editdis\tan ce(s1,s2)}{\max(length(s1),length(s2))};$$

- "bin": a page is cut into 100*200 bins in equal size. We use similar concepts to [17]. A bin can be a graphic bin, a text bin, or a white space bin. A bin is a graphic bin if more than half of the bin is overlapped by graphics. A bin is a text bin if more than half of the bin is overlapped by text. If a bin is neither a text bin nor a graphic bin, it is a white space bin. If bins in corresponding positions on two pages are of the same kind, we consider that a "hit". We then compute the similarity between two pages as $\frac{numberofhits}{100*200}$;
- "graphmatch": the graphic block list in a page is extracted from its MXY-Tree [36]. Our OCR tool uses a rectangle to hold any graphic block. For a graphic block b, (b.x1, b1.y1) is the coordinate of its top-left point, and (b.x2, b.y2) is the coordinate of its bottom-right point. Given two graphic blocks b1 and b2, they are matched if all the following criteria are held:
  - $|b1.x1 - b2.x1| \le thresholdx$;
  - $|b1.x2 - b2.x2| \le thresholdx$;
  - $|b1.y1 - b2.y1| \le thresholdy$;
  - $|b1.y2 - b2.y2| \le thresholdy$;

  In our implementation,

$$thresholdx = \frac{PageWidth}{20}$$

$$thresholdy = \frac{PageHeight}{10}$$

Given two $L_1 = b_1, b_2 \ldots b_m$, and $L2 = a_1, a_2 \ldots a_n$, where a_i and b_i are graphic blocks. These two lists are matched if:

- m=n;
- $\forall i$ from 1 to m, a_i and b_i are matched.

Two pages are considered "graphmatch" if their graphic block lists are matched.

The element "block" specifies the features for an individual block. The element "block" has several attributes and contains one element "stringmatch". The attributes of the element "block" are listed in Table 3.

The element "stringmatch" is a child of the element "block". It specifies what kind of the text strings are in a block. The value of the element "stringmatch" (i.e. the string between <stringmatch> and </stringmatch>) is the target text string (i.e. what the block should contain). The element "stringmatch" has three attributes. The attribute "case" indicates whether matching the target text string is case sensitive or not. The attribute "loc" specify the location of the target text string. In our current implementation, its value is either "equal" or "beginwith". The value "equal" means that the text in the block equals to the target text string. And the value "beginwith" indicates that the text in the block starts with the target text string.

TABLE 3
THE ATTRIBUTE LIST OF ELEMENT BLOCK

Attributes	Value	Description
Name	A string	The identifier of the block.
Align	"left", "right", or "center"	The alignment of the text in the block.
Xsize	"long" or "short"	The relative width of the block. A block is a long block if its width is larger than half of the page width.
Loc	"equal" or "startwith"	The text in the block equals or starts with the defined text
Allupcase	"true" or "false"	Whether the text in this block consists of letters in upper case only.
Firstupcase	"true" or "false"	Whether all the first letters of words are capitalized

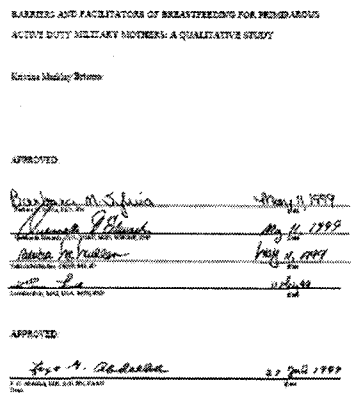
The element "blockrelation" defines the relative relationship between defined blocks. It contains four attributes, which are shown in Table 4. For example, *<blockrelation begin= "b1" end= "b2" relation= "top" adjacent= "true"/>* means that the block *b1* is located above the block *b2* and they are neighbors.

TABLE 4
THE ATTRIBUTE LIST OF ELEMENT BLOCKRELATION

Attributes	Value	Description
Begin	A string	The identifier of the block one
End	A string	The identifier of the block two
Relation	"top", "below", "left", or "right"	The relative relationship between block one and block two.
Adjacent	"true" or "false"	Whether block one is adjacent to block two or not in a defined relationship.

Table 5 gives a sample of a page format. It defines three blocks and two relations among them.

TABLE 5
SAMPLE PAGE FORMAT

Unstructured metadata page	Page Format Definition
	<covclass name= "approval page"> <block num= "title" align= "left" allupcase= "true" xsize= "long" loc= "top" /> <block num= "author" align= "left" xsize= "short" /> <block num= "label" align= "left" > <stringmatch case= "true" loc= "equal" distance= "1"> APPROVED: </stringmatch> </block> <blockrelation begin= "title" end= "author" relation= "top" adjacent= "true" /> <blockrelation begin= "author" end= "label" relation= "top" adjacent= "true" /> </covclass>

The information specified in Table 5 includes:

- A block "title" with the following characteristics: text in this block is left-aligned, letters in this block are all capitalized letters, the block is a long block (i.e. the width of the block is larger than the width of the page), and the block is located on the top of the rest of the page;
- A block "author" with the following characteristics: text in the block is left-aligned, and the block is a short block;
- A block "label" with the following characteristics: the text in the block is left-aligned, and the edit distance [3] between the text in the block and the string "APPROVED:" is equal to or less than 1;
- The block "title" is on the top of the block "author", and they are located adjacently;

- The block "author" is on the top of the block "label", and they are located adjacently.

According to this definition, any unstructured metadata page that has such three blocks belongs to the group "approval page".

4.3.2.2. Classification without Prior Knowledge

In previous section, we introduced the approach of classifying an unstructured metadata page: classification with prior knowledge. In that approach, we classify a document into a group based on pre-defined page formats. In this section, we will introduce another approach: classification without prior knowledge. In this approach, we classify an unstructured metadata page into a group based on the similarity of the page and the representative pages of each group. The steps to classify a new unstructured metadata page are as follows:

1. Load the representative pages from all existing groups;
2. For each group, compute the similarity of the page and the representative page of the group, and record the maximum similarity;
3. If the maximum similarity is larger than the threshold value, the page is classified into the corresponding group;
4. If the maximum similarity is smaller than the threshold value, a new group is created. The page is classified into this new group and is assigned as the representative page of the group.

Fig. 24 shows our current algorithm to measure the similarity of two unstructured metadata page without prior knowledge.

```
PageSimilarity ( Page p1, Page p2)
{
        Sim=0;

        Bsim=blockTypeSimilarity(p1, p2);
        If(Bsim<0.75)
                return 0;

        If (graphMatch(p1,p2))
                Sim=Sim+0.5;
        Sim=Sim+binSimilarity(p1,p2,100,200);
        If (Sim < 0.7)
                Return 0;

        Xsim=xsizeSimilarity(p1,p2);
        If (Xsim<0.75)
                Return 0;
        Return Sim;
}
```

Fig. 24. Unstructured metadata page similarity

In our implementation, we integrate several methods to measure the similarity between two pages. Most of them have already been introduced in the "knowledge-based classification" section. The "blockTypeSimilarity", "graphMatch", "binSimilarity" in Fig. 24 refer to the "blocktype", "graphmatch", and "bin" methods in the "knowledge-based classification" section respectively. The "xsizeSimilarity" is based on the relative block sizes. A page is converted to an MXY Tree [36] (a horizontal cut has higher priority than a vertical cut so that each page has a unique MXY Tree). A block is an either a long block or a short block based on whether its width is larger than the half of the page width or not. Then a sequence of block widths ("L" for a long block or "S" for a short block) is extracted from the MXY tree. The similarity of two pages is based on the edit distance [3] between their block type sequences. Given two sequences s1 and s2, the similarity is computed by the following equation:

$$Sim(s1, s2) = 1 - \frac{editdis\,\tan ce(s1, s2)}{\max(length(s1), length(s2))};$$

In the algorithm shown in Fig. 24, we also choose the threshold values based on our experience. The classification with prior knowledge and the classification without prior knowledge share some common underlying methods to measure the similarity between two pages. However, in the classification with prior knowledge approach the threshold values and the choices of the methods are specified in the page formats, while in the classification without prior knowledge the threshold values and the methods are fixed in the document classification code.

4.4. Summary and Discussion

In this chapter, we described our document classification approaches. The document classification in our research includes two subtasks: the locating of the metadata pages, and the classifying metadata pages into groups. We first divide metadata pages into two categories: structured metadata pages and unstructured metadata pages. For structured metadata pages, we define the label sets in advance. Then we locate the metadata pages and classify them into groups based these pre-defined label sets. For unstructured metadata pages, we first write rules to locate the metadata pages based on their statistical features. Then we use two approaches to classify them into groups. The first one is the classification with prior knowledge and the second one is the classification without prior knowledge.

For a collection, we have flexibility to choose an approach or a set of approaches to classify documents into groups. To work with our DTIC collection, we first define label sets, locate structured metadata pages, and classify them into groups. For the documents that are not resolved by this approach, we develop rules to locate the

unstructured metadata pages, and classify these unstructured metadata pages into groups with or without prior knowledge.

We provide a set of methods to compute the similarity of two unstructured metadata pages. However, how to measure the similarity of two unstructured metadata pages without prior knowledge is worthy of further research. A possible refinement is to develop an algorithm to measure the similarity based on the edit distance of two trees. How to measure the edit distance of two document pages is still an issue. In our current approach of classification without prior knowledge, we use the first metadata page that added to a group as the representative page, and an unstructured metadata page is classified into a group based on the similarity of the page and the representative pages. A further refinement is to adjust the representative page of a group after a new unstructured document page is added to the group.

CHAPTER V

SYSTEM IMPLEMENTATION

5.1. Overall Architecture of Automated Batch Processing

The overall architecture that converts a legacy collection into an interoperable digital library framework is shown in Fig. 25. The main steps to build an interoperable Digital Library out of a physical collection are as follows:

- **Scan and OCR**: Commercial OCR software is used to scan the documents.
- **Extract Metadata:** Extract metadata by using rules and machine-learning techniques. The extracted metadata are stored in a local database. In order to support the Dublin Core metadata schema, it may be necessary to map extracted metadata to the Dublin Core format.
- **Build an OAI layer:** To make the digital collection interoperable, we implement an OAI data provider layer to make it OAI-compliant. The OAI layer accepts all OAI requests, gets the information from the database and encodes metadata into XML format as responses.

In addition, we also implement a search engine for local search. Users can search the metadata and access the original documents. With different XSL (eXtensible Stylesheet Language), the original documents can be presented differently for users of different devices, such as Web browsers and PDAs (Personal Digital Assistant).

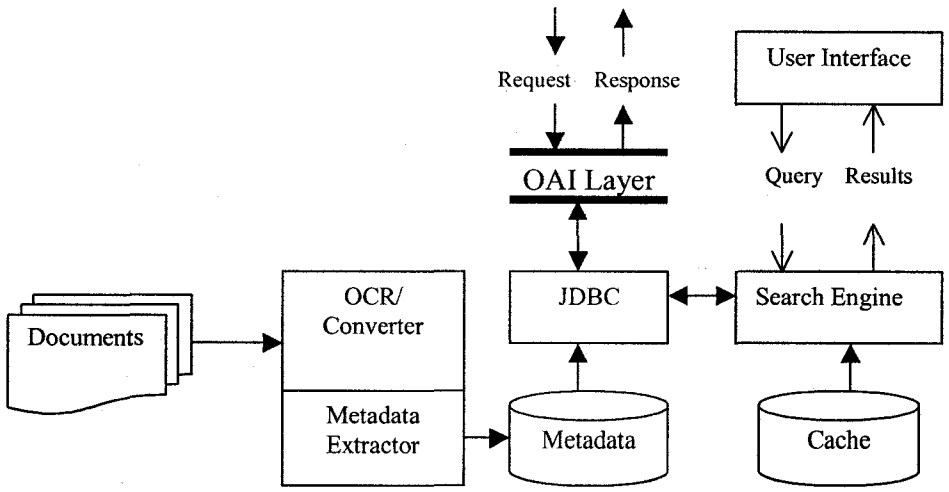


Fig. 25. System architecture

5.2. Scan and OCR

As shown in Fig. 25, the first step of building an interoperable Digital Library from a physical collection is converting documents into searchable electronic documents. This can be done by using a commercial OCR tool.

There are many OCR tools available. To fit into our architecture, an OCR tool with the following features is desirable in addition to high recognition accuracy:

- Documents already in electronic format as well as support for scanned documents: a physical collection may contain two kinds of legacy documents: documents available as files and scanned documents. An OCR tool should support both kinds of documents as input. In other words, it needs to support input from scanners as well as input from file folders. Particularly, it should support PDF file format, because many scanned documents exist in PDF format.

- XML output support: since we will do further work on OCR output, we should choose a format that is easy to be processed. XML format is a good choice as the output format from the OCR tool.
- Automated process support: our system aims to support a dynamically increasing collection. New documents may continue to be added to a collection. An OCR tool used in our system should be able to process a batch of new documents automatically.

Based on the requirements listed above, we chose ScanSoft Omnipage pro 14 Office software. ScanSoft Omnipage pro 14 Office claims more than 99% accuracy for 119 languages. It supports inputs from scanners, file folders, and even files from over a network. For XML output, it supports two formats: their own schema as well as standard WordprocessingML [57]. In addition, it can automatically process new added documents with its "watch folder" feature.

Even though ScanSoft Omnipage pro 14 Office has high accuracy, it is not error free. In addition, we found other limitations during the experiments with our test bed:

- Results of automated image/text separation, table area identification, and equation identification are not satisfactory yet;
- Occasionally, it does not recognize the text on a page and puts an empty page element in the output XML file;
- Some layout features in its XML output are not reliable. For example, it may assign left align feature to some central or even right aligned text;
- Sometimes, it has a problem of determining the paragraph boundary and produces more paragraphs.

These limitations pose additional challenges to our system.

5.3. Metadata Extraction

The most important part in our system is the metadata extraction module. Our objective is to extract metadata from various documents. To achieve high accuracy, we use the template-based approach described in Chapter 3.

Our template-based approach is a rule-based system, and its objective is to make our system work for different document types. Instead of using a rule set that can handle any document in the entire collection, we first classify the documents into groups based on similarity and then create a template for each group. We extract metadata from a document based on the rules defined in its template.

With our OCR tool "Omnipage pro 14 Office", a document is processed and saved in XML format with hierarchy structure, i.e., document-page-zone/region-column-paragraph-line-word-character. The details of the hierarchy structure will be described later in the "Engine" section. Then the metadata page of this document is located. The issue of locating a metadata page among a document has already been addressed in Chapter 4. After that, our engine extracts metadata from the metadata page of the document. Since the OCR results on the paragraph level are unreliable, our engine currently works with the line level information. In the other words, the metadata page is converted to a vector of lines. The engine loads the rules from the corresponding template and applies these rules to all the lines.

The main components in the implementation of our template-based approach are the features that we use for metadata extraction, the language that we use for the template

definition, and the template engine to extract metadata according to the defined template. In the rest of this section, we will introduce these components one by one.

5.3.1. Features

In our template-based approach we extract metadata from documents based on the rules defined in templates. Each rule defines how to extract metadata based on some features of the documents. The issues are what kinds of features are available for metadata extraction, and what kinds of features we should use. We will discuss these two issues in the following subsection.

5.3.1.1. Basic Document Features

An author can choose many features to render document metadata. Generally, the document features can be divided into two categories: layout features and textual features. Layout features are the features describing an object's physical appearance on a page, for example:

- Boldness, i.e., whether text is in bold font or not;
- Font size, i.e., the font size used in text, e.g. font size 12, font size 14, etc;
- Alignment, i.e. whether text is left, right, central, or adjusted alignment;
- Geometric location, for example, a block starting with coordinates (0, 0) and ending with coordinates (100, 200);
- Geometric relation, for example, a block located directly below another block.

Textual features are used to describe whether a line contains some special words or special patterns of characters, for example:

- Special words, for example, a string starting with "abstract";

- Special patterns, for example, a string with regular expression "[1-2][0-9][0-9][0-9]";
- Statistical features, for example, a string with more than 20 words, a string with more than 100 letters, and a string with more than 50% letters in upper case;
- Knowledge features, for example, a string containing a first name from a name dictionary.

5.3.1.2. Complex Features

As we described above, there are many features available for metadata extraction. However, sometimes, using these basic features directly to define a rule is difficult and may require special knowledge. For example, in order to define a rule to check whether a string agrees with a "name format" or not, a user may have to write a complex regular expression since there are a lot of name formats. Table 6 lists some name formats. A user without the knowledge of all possible name format variations will find it difficult to write such a rule. Furthermore, it is not easy for a user to notice a bug in such a regular expression. Fixing a bug is even more difficult.

TABLE 6
SAMPLES OF NAME FORMATS

Name Format	Example
First-Name Last-Name	Jianfeng Tang
First-Initial Last-Name	J. Tang
First-Name Middle-Initial Last-Name	George W. Bush

To address these limitations, we introduce more complex features that are built on top of the basic features. We call this kind of features the advanced features. The goal of

this approach is to make template writing simple. Furthermore, using the advanced features makes a template short and improves its readability. We will use an example to illustrate the benefits of using advanced features. For example, sometimes, users are interested in whether a string starts with a month or not, but the name of the month is not relevant. We can define an advanced feature "*beginwithmonth*" for this, so that users do not need to enumerate the possible month names, such as "January", "February", and "June". Some of the advanced features we have created are listed below:

- *Beginwithmonth*, i.e., whether a string starts with a month name, such as "January", and their variations, such as "Jan";
- *Dateformat*, i.e., whether a string is in a date format; some data formats are "dd month yyyy" "month dd, yyyy" or "month yyyy", where "month" means a month name or its variation, such as "Jan ", "September", etc.;
- *Nameformat*, i.e., whether a string is in a name format; some name formats are "firstname lastname", "firstinitial lastname", "firstname middleinitial lastname", "lastname, firstname", etc. A name format can also include a title prefix, such as "Mr." and "Dr.", or a suffix, such as "Jr.".

5.3.2. Language

In the previous sub-section, we described the feature set used in our template-based approach for metadata extraction. Since templates in our approach are created manually by not necessarily technical experts but rather library staff members, we need to keep templates as simple as possible. We also want to make our templates easy to read and understand. Therefore, we introduced complex features. Besides the types of features we use, we need to address what type of language we should use to describe the rules that

make up a template. There is a trade-off, a simple language may have limitations on writing rules and a complex language may be difficult to use. In this sub-section, we will discuss our rule language in details.

Existing languages, such as Prolog [62] and CLIPS [20], can be used for defining rules. However, these languages have been designed for application developers to create expert systems. It may be difficult for users we expect to write templates to create rules for metadata extraction in these languages. To illustrate this, we use Prolog to define a rule, "A line with the largest font size on the top half of a page is a title." The Prolog code is shown in Fig. 26.

```
text(Line,X)        :- Line=line(X,_,_,_,_,_,_).
fontSize(Line,X) :- Line=line(_,X,_,_,_,_,_).
top(Line,X)         :- Line=line(_,_,_,X,_,_,_).
% largest fontsize in area
largerThanAllLines(_, []).
largerThanAllLines(Line1,[Line2|Rest]) :-
fontSize(Line1,Size1), fontSize(Line2,Size2), Size1 >= Size2,
largerThanAllLines(Line1, Rest).
largestLine(Line) :- getHalf(LineSet), member(Line, LineSet),
largerThanAllLines(Line, LineSet).
getHalf(M) :- document(LineSet), topHalf(LineSet,M).
topHalf([],[]).
topHalf([L|R],[L|R2]) :- top(L,X), X =< 500, topHalf(R,R2).
topHalf([L|R],R2) :-  top(L,X), X > 500,topHalf(R,R2).
%rule: a line with largest fontsize on top half is a title
titleLine(Line) :- largestLine(Line).
```

Fig. 26. A Prolog sample

In our template-based approach, we use our own template language to write a template. One advantage is that we can use any advanced feature as long as we implement it in our engine. For example, we can define a rule like "title :-

largeststrfontsize(0,0.5)" for what the Prolog code in Fig. 26 defined, if we implemented the "largeststrfontsize" in our engine as "A line with the largest font size". The other advantage of using our own language is that we have the flexibility to extend the feature set. An alternative to our approach is to build advanced features in an existing language (e.g. Prolog) and implement an interface on its engine so that advanced feature can be used in a template. The advantage of using our own language is that we have fully control on the syntax of template language.

Our template language is XML based. The schema of our currently implemented language is shown in Fig. 27. The root element of a template is the element "structdef". Under it, each metadata field is defined by an element "meta". The element "meta" has three attributes: "name", "min", and "max". The attribute "name" specifies the name of its corresponding metadata field (e.g. "title", "creator", etc.). The attributes "min" and "max" specify the minimum and maximum occurrences of the metadata field. Each element "meta" has two children: the element "begin" and the element "end". They define how to locate the starting point and end point for the metadata field on a page respectively. The starting/end point can be determined either by matching a special string or looking for a line with specified features.

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
 <xs:element name="structdef">
   <xs:complexType>
     <xs:sequence minOccurs="0" maxOccurs="unbounded">
      <xs:element name="meta">
        <xs:complexType>
         <xs:sequence>
           <xs:element name="begin">
            <xs:complexType>
              <xs:sequence>
                  <xs:element ref="stringmatch" minOccurs="0" />
               </xs:sequence>
             </xs:complexType>
           </xs:element>
          <xs:element name="end">
            <xs:complexType>
                <xs:sequence>
                     <xs:element ref="stringmatch" minOccurs="0" />
                 </xs:sequence>
              </xs:complexType>
            </xs:element>
         </xs:sequence>
        <xs:attribute name="name" type="xs:string" use="required" />
       <xs:attribute name="min" type="xs:int" />
        <xs:attribute name="max" type="xs:positiveInteger" />
     </xs:complexType>
     </xs:element>
  </xs:sequence>
  </xs:complexType>
  </xs:element>

<xs:element name="stringmatch">
<xs:complexType>
   <xs:attribute name="case" type="xs:string" use="required" />
  <xs:attribute name="loc" type="xs:string" use="required" />
 </xs:complexType>
</xs:element>
</xs:schema>
```

Fig. 27. Template Schema

The list of the currently implemented features is shown in Table 7. Either the element "begin" or the element "end" has an attribute, i.e. "inclusive". The attribute "inclusive" can have three values:

- "before": the line before the matched point;

- "after": the line after the matched point;
- "current": the line with the matched point.

TABLE 7
FEATURE LIST

Feature	Meaning
largersize	Return the position of the first line whose font size is larger than the font size of the current line (Lines less than 10 characters are ignored.).
sizechange (x)	Return the position of the first line whose font size is different from the current font size and the difference is larger than x.
featurechange	Return the position of the first line whose feature is different from the feature of the current line. That means one of the following is true: • Its font size is different from that of the current line; • Its boldness information is different from that of the current line (i.e. one is bold and the other is not bold); • All the letters in either it or the current line (but not both) is capitalized; • The letter in each word of either it or the current line (but not both) is capitalized.
largeststrsize (x,y)	Return the position of the first line whose font size is the largest among the lines with the relative position between x and y, where x and y are relative position on a page. They a float number between 0 and 1. The value 0 means the first line, and the value 1 stands for the last line. To overcome OCR errors, only a section with normal string is considered at the time. A string is a normal string if it matches all: • Its length is larger than 11; • It has more than one words; • Average word length is between 4 and 13; • Percentage of letters is larger than 0.8.
layoutchange	Return the position of the first line if it meets one of the following criteria: • Its font size is different from that of the current line; • Its boldness information is different from that of the current line (i.e. one is bold and the other is not bold).
boldchange	Return the position of the first line whose boldness information is different from that of the current line.
beginwithmonth	Return the position of the first line starting with a month name (e.g. "January") or a month name abbreviation (e.g. "Jan").
dateformat(*format*)	Return the position of the first line that is in the date format specified by the parameter "format". Currently only "month yyyy" and "dd month yyyy" are supported, where the "month" is a month name or a month name abbreviation, "dd" is a date, and "yyyy" is a year.

TABLE 7 (continued)

Feature	Meaning
dateformat	Return the position of the first line that is in one of the following date formats: "dd month yyyy", "month dd, yyyy" or "month yyyy", where the "month" is a month name or a month name abbreviation, "dd" is a date, and "yyyy" is a year.
nameformat	Return the position of the first line that agrees with a name format.
!nameformat	Return the position of the first line that does not agree with a name format.
size=x	Return the position of the first line whose font size is x, where x is an integer.
size (x,y)	Return the position of the first line whose font size is between x and y, where x and y are integers.
onesection	It means current metadata field is exact one line.
Other metadata	Use other metadata field to locate the starting point or end point of the current metadata field, e.g., the metadata field "creator" is after the metadata field "title".
subtitle	Return the position of the first line with one of the following features: all the letters are capitalized and the number of words is less than 4, or every word (but the special words "a", "of", "for", "the", "one", "in", and "to") starts with a capitalized letter.
begin	The first line on the page
end	The last line on the page

The element "stringmatch" define how to find a line matched with the specified text string. It has two attributes: the attribute "loc" and the attribute "case". The value of the attribute "loc" can be either "beginwith" or "equal". The former indicates to look for a line starting with the specified text string. And the latter indicates to look for a line same to the specified text string. The value of the attribute "case" can be either "yes" or "no" depending on whether the string match is case sensitive or not.

A part of a template is shown in Fig. 28. The following are what it defines:

- The "title" metadata field starts with the first line of the page and ends before the line starting with either the text string "THESIS", "DISSERTATION" or "D I S S E R T A T I O N";
- The "creator" metadata field starts after the last line that starts with either the text string "THESIS", "DISSERTATION" or "D I S S E R T A T I O

N"; It ends before the line starting with either the text string "AFIT" or "A F I T";

- The "identifier" metadata field is the line after the line starting with either the text string "AFIT" or "A F I T".

```
<?xml version="1.0" ?>
<structdef>
        <meta name= "title" min="1" max="1">
                <begin inclusive="current">begin</begin>
                <end inclusive="before">
                        <stringmatch case="yes" loc="beginwith">
                                THESIS|DISSERTATION|D I S S E R T A T I O N
                        </stringmatch>
                </end>
        </meta>
        <meta name= "creator" min="1">
                <begin inclusive="after">
                        <stringmatch case="yes" loc="beginwith">
                                THESIS|DISSERTATION|D I S S E R T A T I O N
                        </stringmatch>
                </begin>
                <end inclusive="before">
                        <stringmatch case="yes" loc="beginwith">AFIT|A F I T</stringmatch>
                </end>
        </meta>
        <meta name= "identifier" min="0" max="1">
                <begin inclusive="after">
                        <stringmatch case="yes" loc="beginwith">AFIT|A F I T</stringmatch>
                 </begin>
                <end inclusive="current">onesection</end>
        </meta>
.....
```

Fig. 28. Template sample (partial)

5.3.3. Engine

We have already discussed the feature set and the template language used in our template-based metadata extraction approach. In this section, we will discuss the implementation of our rule engine.

The rule engine is software that can parse the rules written in the language and take actions accordingly. Our rule engine is responsible for understanding the rules written in our template language and extracting metadata from documents.

We implement the rule engine in Java. This makes our template engine platform independent of the operating system. The inputs of our template engine are document pages and a template. Both are in XML format. The outputs are files containing metadata elements in XML format. The output will also be put into a database through JDBC (Java Database Connectivity) calls.

As shown in Fig. 29, the template engine mainly consists of three components: the XML Parser, the Data Preprocessor and the Metadata Extraction modules. We will introduce these three parts in the rest of this section respectively.

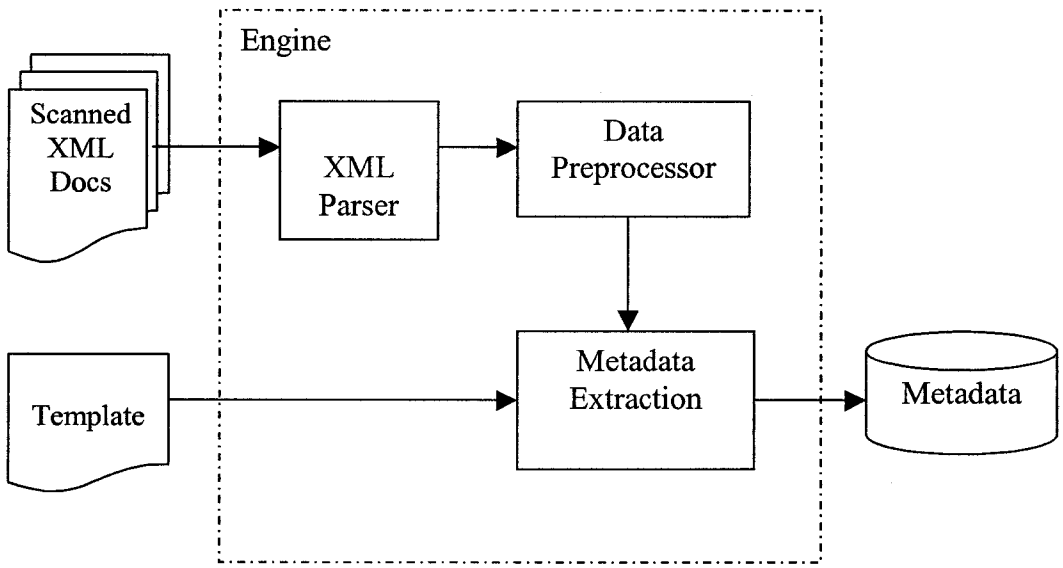


Fig. 29. Template engine

The XML Parser parses the document pages given in XML format, which are generated by commercial OCR tools. In our implementation, we choose ScanSoft Omnipage pro 14 Office as our OCR tool. ScanSoft Omnipage pro 14 Office uses its own XML schema named SSDOC. SSDOC schema represents a document in a hierarchical structure shown in Fig. 30.

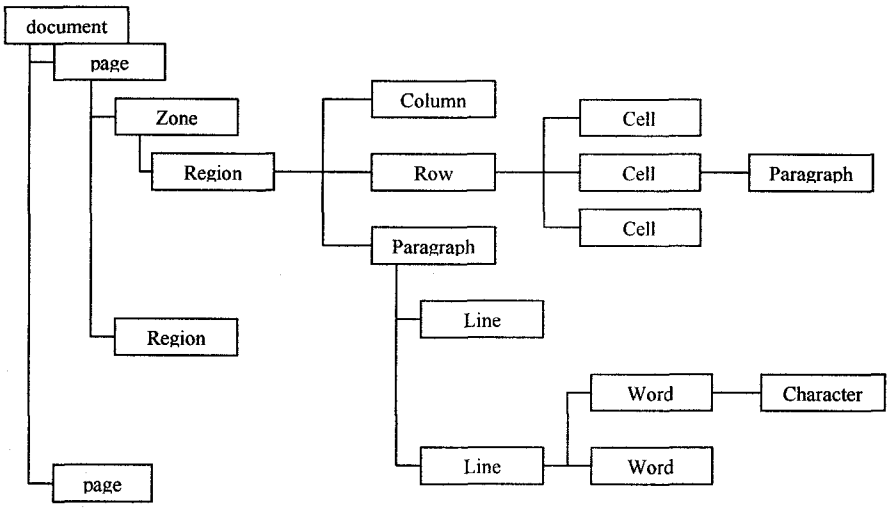


Fig. 30. SSDOC structure

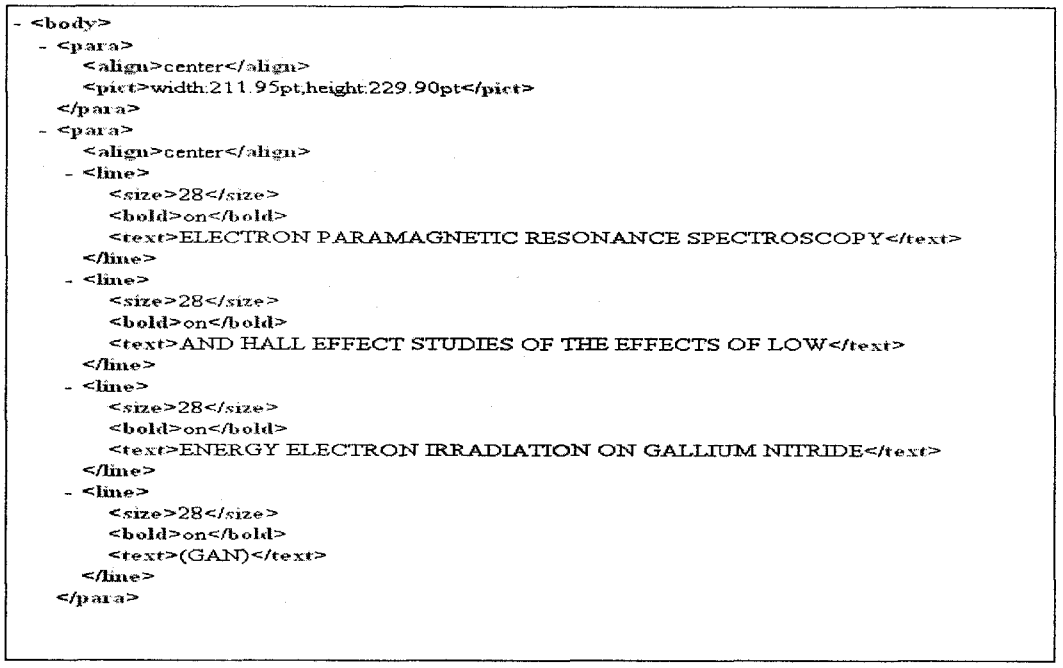
From Fig. 30, we can see that a document consists of pages, a page consists of zones and/or regions, a region consists of columns, rows or paragraphs, a paragraph consists of lines, a line consists of words, and a word consists of characters. Most elements in the SSDOC schema have attributes for layout features. A part of an SSDOC XML sample is shown in Fig. 31. In this schema, the same element may occur at different levels. For example, a region can be a child of a page or a child of zone. A

paragraph can be a child a region or a child of a cell. The XML Parser parses the pages according to this schema and stores the resulting trees into an internal data structure.

```
<?xml version="1.0"?>
<!--XML document generated using OCR technology from ScanSoft, Inc.-->
<document ssdoc-vers="SSDOC1.0" ocr-vers="OmniPage Pro 14" xmlns="x-
schema:http://www.scansoft.com/omnipage/xml/ssdoc-schema2.xml">
<page width="12240" height="15840" x-res="300" y-res="300" bpp="1" orientation="0" skew="0"
filename="C:\testbed\OCR\firstlast\ADA420816.pdf" language="0">
<region reg-type="graphic"><rc l="1991" t="1440" r="10269" b="6259"/></region>
<region reg-type="horizontal"><rc l="1991" t="6259" r="10269" b="14400"/>
<paragraph para-type="text" align="centered" left-indent="0" right-indent="0" start-indent="0" line-spacing="312">
<ln baseline="7248" ff="Times New Roman" fs="1400" char-attr="bold">
<wd l="2045" t="7061" r="3605" b="7258">ELECTRON</wd>
<wd l="3686" t="7061" r="6024" b="7258">PARAMAGNETIC</wd>
<wd l="6106" t="7061" r="7848" b="7258">RESONANCE</wd>
<wd l="7944" t="7061" r="10200" b="7258">SPECTROSCOPY</wd>
</ln>
```

Fig. 31. An SSDOC XML sample

The Data Preprocessor is responsible for cleaning the parsed data. As we have already introduced, our current engine works at the line level since the high level information generated by our OCR tool is unreliable. In the other words, for our current engine, we do not need all the information encoded in a SSDOC XML document. Therefore, we use the Data Processor to filter out any irrelevant information. A part of the cleaned XML file is shown in Fig. 32.



```
- <body>
  - <para>
      <align>center</align>
      <pict>width:211.95pt,height:229.90pt</pict>
    </para>
  - <para>
      <align>center</align>
    - <line>
        <size>28</size>
        <bold>on</bold>
        <text>ELECTRON PARAMAGNETIC RESONANCE SPECTROSCOPY</text>
      </line>
    - <line>
        <size>28</size>
        <bold>on</bold>
        <text>AND HALL EFFECT STUDIES OF THE EFFECTS OF LOW</text>
      </line>
    - <line>
        <size>28</size>
        <bold>on</bold>
        <text>ENERGY ELECTRON IRRADIATION ON GALLIUM NITRIDE</text>
      </line>
    - <line>
        <size>28</size>
        <bold>on</bold>
        <text>(GAN)</text>
      </line>
    </para>
```

Fig. 32. Cleaned XML sample (partial)

In previous paragraphs, we described the XML Parser and the Data Preprocessor. These two parts prepare data for the Metadata Extraction module. To extract metadata from documents, the Metadata Extraction module first loads the corresponding template and then parses the template to get the instructions about how to extract metadata. Finally, the Metadata Extraction module puts tags to the elements in the input data and presents the results in XML format. For example, for the instructions "The "title" metadata field starts with the first line of the page and ends before the line starting with either the text string "THESIS", "DISSERTATION" or "D I S S E R T A T I O N" (see Fig. 33 for the rule defined in our template), the Metadata Extraction module will work as follows:

- Marks the first line as the starting point of the metadata field "title";

- Locates the first line starting with the text string either the text string "THESIS", "DISSERTATION" or "D I S S E R T A T I O N", and marks the location before this line as the end point.
- Any text located between the starting point and the end point is a part of the value of the "title" metadata field.

```
<meta name= "title" min="1" max="1">
        <begin inclusive="current">begin</begin>
        <end inclusive="before">
                <stringmatch case="yes" loc="beginwith">
                        THESIS|DISSERTATION|D I S S E R T A T I O N
                </stringmatch>
        </end>
</meta>
```

Fig. 33. Template sample (one filed)

An output sample file is shown in Fig. 34. It includes "title", "creator", "identifier", "contributor", and "Rights" metadata fields.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <paper>
  - <metadata>
      <title>ELECTRON PARAMAGNETIC RESONANCE SPECTROSCOPY AND
        HALL EFFECT STUDIES OF THE EFFECTS OF LOW ENERGY ELECTRON
        IRRADIATION ON GALLIUM NITRIDE (GAN)</title>
      <creator>Kevin D. Greene, Major, USAF</creator>
      <identifier>AFIT/DSP/ENP/03-02</identifier>
      <contributor>DEPARTMENT OF THE AIR FORCE AIR UNIVERSITY AIR
        FORCE INSTITUTE OF TECHNOLOGY Wright-Patterson Air Force
        Base, Ohio</contributor>
      <right>APPROVED FOR PUBLIC RELEASE; DISTRIBUTION
        UNLIMITED</right>
    </metadata>
  </paper>
```

Fig. 34. Output metadata sample

5.4. Build OAI Layer

As shown in Fig. 25, we built an OAI layer to make the collection interoperable. The OAI layer is an implementation of OAI-PMH protocol to accept OAI requests from the network and provide OAI responses accordingly. OAI-PMH is a protocol developed by Open Archive Initiative to provide interoperability among heterogeneous network accessible repositories. OAI requests are sent by HTTP protocol, and OAI responses are encoded into XML format.

A guideline of implementing OAI-PMH is available on OAI website [46]. Minimum requirements of building an OAI compliant repository include:

- Dublin Core (DC) [27] metadata schema support: DC schema is the required metadata schema for OAI-PMH. In other words, every OAI-compliant repository has to support one common metadata schema – DC schema. An OAI-compliant repository can store DC metadata directly or convert native metadata to DC metadata instantly.
- An HTTP server to understand HTTP OAI requests. Six OAI requests have to be supported. They are GetRecord, Identify, ListIdentifiers, ListRecords, ListMetadataFormats, and ListSets [47].

In addition, a repository with a large collection usually implements a control mechanism to allow a harvester to retrieve a large number of records as a sequence of requests for smaller numbers of records. The purpose of this mechanism is to allow a data provider to manage its load and spread out large requests.

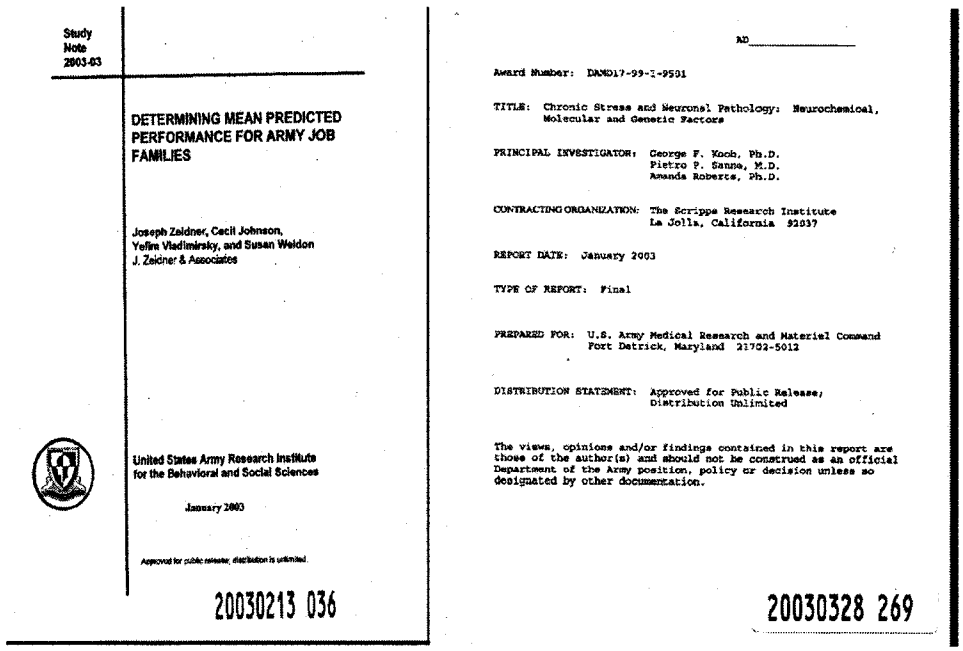
We leverage our earlier work on Arc [53] and Archon [54] to implement the OAI-PMH and search service. In our system, as soon as metadata for a document is extracted and put in our database, we apply a cross-walk program to create a Dublin Core metadata record. An index program is invoked on a periodic basis and thereafter it becomes available for searching and for inclusion in OAI-PMH requests.

CHAPTER VI

EXPERIMENTAL RESULTS

6.1. Test Bed

We used the Scientific and Technical Information Network (STINET) collection available on DTIC's website [26] to build our test bed. The STINET collection contains more than 118K technical reports in PDF format, and is heterogeneous, having documents from different organizations and with different metadata pages. Two metadata page samples are shown in Fig. 35.



Study Note 2003-03

DETERMINING MEAN PREDICTED PERFORMANCE FOR ARMY JOB FAMILIES

Joseph Zeidner, Cecil Johnson, Yefim Vladimirsky, and Susan Weldon
J. Zeidner & Associates

United States Army Research Institute for the Behavioral and Social Sciences

January 2003

Approved for public release; distribution is unlimited.

20030213 036

AD_____

Award Number: DAMD17-99-1-9501

TITLE: Chronic Stress and Neuronal Pathology: Neurochemical, Molecular and Genetic Factors

PRINCIPAL INVESTIGATOR: George F. Koob, Ph.D.
Pietro P. Sanna, M.D.
Amanda Roberts, Ph.D.

CONTRACTING ORGANIZATION: The Scripps Research Institute
La Jolla, California 92037

REPORT DATE: January 2003

TYPE OF REPORT: Final

PREPARED FOR: U.S. Army Medical Research and Materiel Command
Fort Detrick, Maryland 21702-5012

DISTRIBUTION STATEMENT: Approved for Public Release;
Distribution Unlimited

The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision unless so designated by other documentation.

20030328 269

Fig. 35. Two Metadata Page Samples from STINET Collection

There are two kinds of PDF files in STINET collections, text PDF files and image PDF files. We used 10,000 PDF documents from STINET collection for out test bed. Our test bed has been built as follows: first, we downloaded the 10,000 PDF documents from STINET website; then, we extracted the first five and last five pages from these PDF files; finally, we used Scansoft's Omnipage 14 pro Office to OCR them into XML format.

6.2. Evaluation

We need to evaluate the metadata extraction results. In this evaluation, we use the recall and precision metrics. The general definition of recall and precision is:

$$\text{Recall}=\frac{\text{Correct Answers}}{\text{Total Possible Answers}}$$

$$\text{Precision}=\frac{\text{Correct Answers}}{\text{Answers Produced}}$$

Following [6], we adapt the general definition of precision and recall to the metadata extraction as:

$$\text{Precision (P)} = \frac{\text{Number of data correctly extracted}}{\text{Number of data produced}}$$

$$\text{Recall (R)} = \frac{\text{Number of data correctly extracted}}{\text{Total Number of possible data}}$$

We also use F-Measure to evaluate our results. The definition of F-Measure is:

$$F - Measure = \frac{(\beta^2 + 1) \times P \times R}{\beta^2 \times P + R}$$

Note for F-Measure to give equal weights to recall and precision, we use $\beta = 1$. Essentially the F-measure gives harmonic mean of recall and precision.

In the rest of this sub section, we will define the correctness of extracted data. In our experiments, an extracted data can be completely correct, partially correct, or incorrect. Given a metadata M shown in a metadata page and its extracted value E, E is completely correct if E matches M. E is partially correct if it is not completely correct and one of the following is true by ignoring minor OCR errors:

1) M matches E;
2) M is a sub string of E and any metadata other than M is not a sub string of E;
3) E contains at least one line of M and for any other metadata T, P(M,E) ≥ P(T, E), where for a metadata X

$$P(X,E) = \frac{The\ number\ of\ lines\ of\ X\ that\ are\ a\ part\ of\ E}{The\ total\ number\ of\ lines\ of\ X}$$

An OCR error is minor if the edit distance between the original string *S* and the string after OCR *O* is less than one tenth of the smaller length of *O* and *S*. Some partial correct samples are shown in Table 8. E is incorrect if it is neither completely correct nor partially correct.

There are several motivations to introduce the concept of partial correctness. From Table 8, we can see that in these samples even though these extracted values do not match the orginal values, some can be cleaned by post-processing and some are related to OCR errors instead of our metadata extraction algorithm. For some metadata field such as "Title", a part of the value can also be useful for information retrieval.

TABLE 8
PARTIALLY CORRECT SAMPLES

Original value	Extracted value
CONTROLLED SUBSTANCES EXPORT REFORM ACT OF 2005	CONTROLLED SUBSTANCES EXPORT REFORM ACT OF
CODIFICATION OF TITLE 46, UNITED STATES CODE, "SHIPPING", AS POSITIVE LAW	CODIFICATION OF TITLE 46, UNITED STATES CODE, ??SHIPPING??, AS POSITIVE LAW
Mr. SENSENBRENNER	Mr. SENSENBRENNER, from the Committee on the Judiciary
JULY 14, 2005	JULY 14, 2005.?Referred to the House Calendar and ordered to be printed

6.3. Results by Issues

Our template-based approach has addressed the following issues: heterogeneity, scaling, evolution, adaptability, and complexity issues. In this section, we will organize our experimental results related to these issues.

6.3.1. Heterogeneity

Our template-based approach addresses the issue of heterogeneity by classifying documents into groups and creating a template for each document group. A template contains information about what kind of metadata fields we need to extract and how to extract them. In this section, we will show the results of applying our template-based approach to a heterogeneous document set, and compare the results with SVM approach, which is shown to be superior to other machine learning techniques such as HMM for metadata extraction [30].

6.3.1.1. Experiments with Template-based Approach

We selected 100 documents from our DTIC collection without looking their metadata pages. Then we manually classified these 100 documents into groups. After that, we created a template for each group. Finally, we applied our template-based

approach to extract metadata from these documents. For clarification, we arbitrarily gave each group a name. The group names were "sf298_1", "sf298_2", "generic", "thesis", "letter", "issuedby", "usawc", "afrl", "arl", "edgewood", "nps", "usnce", "afit", and "text". A list of these documents with their unique identifiers is available in Appendix C.1.

This data set of 100 documents is heterogeneous. The heterogeneity is not only in presenting the metadata fields on a metadata page but also in the richness of the metadata fields. For example, metadata pages in the group "sf298_1" or "sf298_2" have more than 20 metadata fields while metadata pages in the group "arl" contain less than 6 metadata fields. Our template-based approach has the flexibility to specify which metadata fields to be extracted. In our experiments, we defined three metadata fields, i.e. "date", "title", and "creator" as the core metadata fields. In the other words, as long as these metadata fields are presented in a metadata page, we would try to extract them. The reasons to choose these three metadata fields as core metadata fields are:

- According to [29], metadata fields "title", "author" (i.e. "creator"), and "subject" are the basic metadata fields for resource discovery;
- We removed the metadata field "subject" from the mandatory set because in our data set few metadata pages contain "subject" information;
- We added the metadata field "date" since we believe that the date information is important for resource discovery and retrieval.

We evaluated the overall results of these three metadata fields for all these 100 documents. Every document in this data set has the metadata field "title" and the metadata field "creator". 88 out of these 100 documents have the metadata field "date".

Table 9 shows the overall metadata extraction results of all 100 documents. The column "#doc" shows the number of documents contain each metadata. The column "#c", "p", and "in" show the numbers of extracted metadata that were completely correct, partially correct, and incorrect respectively. The numbers in the column "recall", "precision", and "F-measure" were computed based on the numbers in the column "#doc", "#c", "#p", and "#in". The column "compl" shows the completely correct results and the column "partial" shows the partially correct results. When we computed the completely correct results, we took the number of extracted data that were completely corrected as the number of data correctly extracted. When we computed the partially correct results, we took the number of extracted data that were completely or partially correct as the number of data correctly extracted. We will follow this naming convention in the rest of this chapter.

TABLE 9
METADATA EXTRACTION RESULTS OF DTIC DOCUMENTS
(CORE METADATA)

Field	#doc	#c	#p	#in	recall		precision		F-measure	
					compl	partial	compl	partial	compl	partial
Date	88	73	9	0	82.95%	93.18%	89.02%	100%	85.88%	96.47%
Title	100	90	8	0	90.00%	98.00%	91.84%	100%	90.91%	98.99%
Creator	100	84	14	0	84.00%	98.00%	85.71%	100%	84.85%	98.99%

From Table 9, we can also see that we got desirable completely correct results for the field "date" and "creator", and high accuracy for the field "title". We got high accuracy partially correct results for all the three fields, and all precision numbers are 100%. There are two reasons for these promising results. First, in our approach,

documents are classified into groups, and each group contains similar documents. In this way, a heterogeneous collection has been converted to several homogeneous sub-collections. The second reason is that in our approach we use different templates for different groups. This means that we can use different features to extract the same metadata field for different groups. Table 10 shows the different rules that we used to locate the starting points of the metadata field "title" for documents from different groups.

TABLE 10
DIFFERENT RULES FOR EXTRACTING FIELD "TITLE"

Group	Rule (partial)	Explain
Generic	Largeststrsize(0,0.5)	A line with largest font size in the region (0, 0.5), i.e. first half page.
Thesis	Largeststrsize(0,0.3)	A line with largest font size in the region (0, 0.3)
Usawc	type	A line after the field "type"
Afrl	date	A line after the field "date"
Nps	<stringmatch case="yes" loc="beginwith"> THESIS </stringmatch>	A line after the text string "THESIS"

The gaps between the completely correct results and the partially correct results indicate that our template-based approach still have spaces to improve. The gap of the filed "date" is mainly due to OCR errors. For example, a string "May 2003" in one metadata page was recognized as the string "May 2 003". This kind of OCR errors can be fixed to some extend by post-processing the extracted results. There are two reasons for the gap of the field "title". The first reason is the OCR errors. Secondly, in some metadata page only a part of a title was extracted due to the different features used for

different parts of the title. The relatively low results of the field "creator" are mainly because that our current implementation has the limitation to extract a metadata that occurs in multiple places.

The metadata extraction results of individual groups with the templates that we used for individual groups are available in Appendix C. We extracted additional fields besides the three core metadata fields to demonstrate the flexibility of our template-based approach to extract different set of metadata fields from different groups, and to address the limitation of our current implementation.

From the templates shown in Appendix C.3, we can see our template-based approach has flexibility to extract different metadata set from different groups. For example, for the group "sf298_1" we can extract about 20 fields, while for the group "arl" we extracted only four fields. The metadata sets can be different both in the number of fields and in the field names. It is worthy of noting that it is not necessary to extract all information from a metadata page by using our template-based approach. Our template-based approach has the flexibility to determine which metadata fields to be extracted. For example, the metadata field "Rights" was not extracted from the metadata pages in the group "arl" even though these metadata pages contain the field "Rights".

Table 11 shows the metadata extraction results for the DC Metadata fields other than our three core metadata fields. We mapped the field "sponsor" in the group "sf298_1" and "sf298_1" to the DC field "Contributor", and mapped the field name "abstract" to "Description" when we compiled the results in Table 11.

TABLE 11
METADATA EXTRACTION RESULTS OF DTIC DOCUMENTS
(OTHER DC METADATA)

Field	#doc	#c	#p	#in	recall		precision		F-measure	
					compl	partial	compl	partial	compl	partial
Type	77	39	4	3	50.65%	55.84%	84.78%	93.48%	63.41%	69.92%
Rights	96	26	1	0	27.08%	28.13%	96.30%	100%	42.28%	43.90%
Publisher	51	12	2	0	23.53%	27.45%	85.71%	100%	36.92%	43.08%
Identifier	60	24	0	1	40.00%	40.00%	96.00%	96.00%	56.47%	56.47%
Contributor	61	42	1	0	68.85%	70.49%	97.67%	100%	80.77%	82.69%
Coverage	7	3	0	3	42.86%	42.86%	50.00%	50.00%	46.15%	46.15%
Subject	5	5	0	0	100%	100%	100%	100%	100%	100%
Description	4	4	0	2	100%	100%	66.67%	66.67%	80.00%	80.00%
Relation	0	0	0	0	N/A	N/A	N/A	N/A	N/A	N/A
Language	0	0	0	0	N/A	N/A	N/A	N/A	N/A	N/A
Format	0	0	0	0	N/A	N/A	N/A	N/A	N/A	N/A

From Table 11, we can see that we got desirable precision for most fields except the field "Coverage" and the field "Description". The three incorrect extracted data for the field "Coverage" are from the group "sf298_1". The metadata field "typecoverage" is used in the metadata pages in the group "sf298_1" for both the metadata field "Type" and metadata field "Coverage". For the same reason, we got three incorrect extracted data for the field "Type". The reason of the low precision of the field "Description" is that some metadata pages in group "sf298_1" or "sf298_2" contain text other than the real abstract in the abstract field. We got low recall for most fields partially because we did not try to extract them in all templates. Our approach has the flexibility to choose which metadata set to be extracted for each group. Since our test bed used controlled vocabulary for the values of the field "Rights", we can improve the recall of the field "Rights" with little effect on the precision by searching text strings. The recall of the field "Type" and field "Identifier" can also be improved with matching special text strings or special text patterns with little affect on the precision. Extracting the field "Publisher" may affect the

its precision for some group where different styles were used. A possible refinement is to add some knowledge bases such as organization names, states, etc.

From the metadata extraction results of individual groups in Appendix C.4, we can see that for a few groups we low recall/precision that was computed based on completely correct extracted metadata for the field "title", "creator", or "date", while its corresponding number of partially correct metadata are high. Table 12 gives the reasons for each core metadata field that we failed to get desirable recall or precision.

TABLE 12
REASONS OF THE LOW NUMBER OF COMPLETELY CORRECT EXTRACTED METADATA

Group	Field	Reasons
generic	creator	2 out of 7 partially correct creators contain their affiliation information and 5 are just a part of creators.
Thesis	date	Due to the limitation to recognize a date with only year information. 2 out of 3 dates contain only year information.
issuedby	title	Due to OCR errors
edgewood	creator	Only a part of creators were extracted due to the limitation to extract metadata that occurs in multiple places.
Text	date	Mainly because of OCR errors. For example, "May 2003" was reconginzed as "May 2 003". One space was added between "2" and "003".

In the completed correct results, we also got low recall/precision for some other fields. The main reasons for these low recall/precision are listed below.

1) First, OCR errors can affect the metadata extraction results, especially the results that were extracted based on pre-defined labels. The OCR errors affect the results in several ways:

   - OCR errors in the extracted data will affect the reall/precision directly, e.g. the field "title" in the group "issuedby".

- We will fail to extract metadata a field if we fail to locate its related field names due to OCR errors, e.g. the field "subject" of some metadata pages in the group "sf298_1".
- Some irrelevant information will not excluded for some field due to OCR error, e.g. the "cls_report" field of some metadata pages in the group "sf298_1").

2) Damaged page: a metadata page was damaged or in bad quality. For example, the low precision of the field "distribution" (i.e. field 12a) of the group "sf298_1" is because the stamp on the page (i.e. "DISTRIBUTION STATEMENT A ...") was extracted as a part of the value of the field.
3) Failure to separate a field label from a field value: this is for the structured metadata page. Our current code for extracting metadata from a structured metadata page assumed that the value and the label of this field are not in the same line or there is a significant space the label and the value if they are in the same line. However, this assumption is not always true. For example, on some metadata pages in the group "sf298_1", a part of the value of field "date" occurred in the same line of the label of this field. This is a limitation of our current implementation.
4) Incomplete feature: for example, the low recall of the field "creator" in the group "generic" (shown in Table 26) are partially because some name formats are not covered by our currently implemented feature "nameformat". Another example is that our currently implemented feature "dateformat" does not cover the date format with year information only.

This is mainly reason of the low recall for the field "date" in the group "thesis" (shown in Table 27);

5) Multiple occurrences of a metadata field: our current implemented metadata extraction code has a problem with extracting a metadata field that occurs multiple times. For example, it has a problem with extracting multiple creators that are separated by other metadata fields. This is partially reason of the low recall of the field "creator" in the group "generic" (shown in Table 27).

6.3.1.2. Experiments with SVM

As long as the documents use the same metadata set, SVM can train a classifier for each metadata field, and classify the line into one metadata field group. In this sense, SVM can work with a heterogeneous collection with the same metadata set. The objective of the experiments in this subsection is to compare the accuracy of the SVM approach and our template-based approach.

We developed code to extract metadata by using SVM approach and applied it to the same data set. The SVM approach that we implemented is similar to the approach described in [30]. There are a few SVM tools available for classifying data with SVM. We used free software LIBSVM [19] in our experiment. LIBSVM supports multi-class SVM. The LIBSVM package can be downloaded from http://www.csie.ntu.edu.tw/~cjlin/libsvm. We used a feature set that is similar to the one used in [30] and converted data in our data set to the LIBSVM required format. The feature set consists of line specific features as well as word specific features. The line specific features that we used is same to the one used in [30]. It includes:

- The number of words in the line;
- The position of the line, i.e., line number;
- The percentage of the dictionary words in the line;
- The percentage of the non-dictionary words in the line;
- The percentage of the dictionary words with first letter capitalized in the line;
- The percentage of the non-dictionary words with first letter capitalized in the line;
- The percentage of the numbers in the line.

The word specific features that we used are shown in Table 13. As suggested by[44], we linearly scale the value range of each feature to [0,1] to avoid that features with large numeric value ranges dominating the features with small value range. As in [30], we used 75% of the data for training and the rest for testing.

TABLE 13
WORD SPECIFIC FEATURES USED IN SVM EXPERIMENTS

Feature	Explanation
:email:	Match regular expression: [A-Za-z]([a-zA-Z_0-9])*@([A-Za-z0-9_])+(\.[A-Za-z0-9_])+
:url:	Match regular expression: http://(\.)*\s
:pubnum:	a word or bigram in the publication word list
:country:	A country name
:state:	A state name of United states or a province name of Canada
:city:	A city name in the United states or Canada
:keyword:	Keyword, key word, keywords, etc.
:singleCap:	A single capitalized letter such as T
:abstract:	abstract
:intro:	Introduction, introductions, etc.
:phone:	Tel, fax, telephone
:month:	A month name or its abbreviation
:postcode:	Abbreviation of the state name such as IL
:mayName:	A word in the name list. Our name list was generated from the CERN collection of Archon[7].
:affi:	A word in the affiliation word list, e.g. "University"
:addr:	A word in the address word list, e.g. "street"
:degree:	A word in the degree word list, e.g. "master"
:prep:	At, in, of
:notenum:	A word in the note word list
:DictWord:	Small case dictionary word in the English word list [13]
:NonDictWord:	Small case non dictionary word
:Cap1DictWord:	A dictionary word with the first letter capitalized
:Dig[x]:	A number with x digits, where x is an integer
"The word-specific feature considers text orthographic properties, e.g., BU-cs-93 is converted to :CapWord2-LowerWord2-Digs2:" [30]	

It worth noting that the results from our SVM approach are based on lines. The results are computed by using one line as a datum in the equations in the section 6.2. A datum is correct if it was classified into the same group as it was tagged. Since we tagged the XML files, all OCR errors were ignored. There is also no partial correctness concept in SVM results evaluation. A line is either correctly or incorrectly classified. We re-evaluated the results from our template-based approach in the same way as we evaluated the SVM results to make them comparable. Table 14 compares our template-based approach with SVM approach on metadata extraction performance for the field "date", "title", and "creator".

TABLE 14
COMPARISON ON THE METADATA EXTRACTION RESULTS BY USING TEMPLATE-BASED APPROACH AND SVM

Field	Recall		Precision		F-measure	
	Template-based	SVM	Template-based	SVM	Template-based	SVM
Date	91.01%	96.00%	100%	100%	95.29%	98.00%
Title	96.09%	69.01%	99.10%	74.24%	97.57%	72.00%
Creator	91.43%	81.82%	92.75%	61.36%	92.09%	70.00%

For the field "title" and the field "creator" our template-based approach got significantly better results than SVM approach. One main reason is that our template-based approach divided a heterogeneous collection into several homogeneous sub-collections (i.e. groups) and used a template specific to each individual group. One interesting result is that SVM approach got a little better result for the field "date" than our template-based approach. This is partly because of our currently implemented feature "dateformat" does not include the date format that contains only the year information (e.g. "1994").

6.3.1.3. Summary

Our template-based approach handles the issue of heterogeneity by dividing a heterogeneous collection into several homogeneous sub collections and creating a template specific to each sub collection. We applied our template-based approach to a heterogeneous data set and got desirable results. We also compared the results from our template-based approach with the results from an SVM approach. With our template-based approach we got significantly better results for the field "title" and "creator", and with the SVM approach we got slightly better results for the field "date".

Our current implementation could still be improved. The code for metadata extraction from structured metadata pages is sensitive to OCR errors in the pre-defined labels. For example, "PERFORMING ORGANIZATION..." in one metadata page was recognized as "gPqEgRFr~ MINCE ORGANIZATION ...". With this OCR error, our code failed to locate the corresponding label in the metadata page. One possible refinement is to rebuild a label if its surrounding labels are located successfully. The code for metadata extraction from unstructured metadata pages has incomplete feature problem. It also has a problem with extracting a metadata field that has multiple occurrences. One possible refinement for the latter is to extend our current template engine so that we can use a loop in extracting a metadata if it has multiple occurrences. Another limitation of our current implementation is that we used only two rules for extracting each metadata. Using multiple rules will be a future refinement.

6.3.2. Scaling

The issue of scaling is how to apply an approach to a large collection. In our approach, we first classify documents into groups. Then we create a template for each document group to instruct our engine how to extract metadata for documents in a group. To work with a large collection, our approach should be able to process most documents with a small number of groups. The objectives of the experiment in this section is to see how many groups are needed in order to process most of the documents in a large collection and whether the number of groups increases much slower than the number of document pages.

In our current implementation for our DTIC collection, we first detect documents with known structured metadata pages, and then classify the rest based on their

unstructured metadata pages. Using this approach, most of documents in our DTIC collection were classified into groups by structured metadata pages. To address the scaling issue more generally, we classified documents into groups without knowledge specific to a collection.

We first detected cover pages from our DTIC collection with the rules that have been described in the section 4.3.1. About 7413 cover pages were detected. Then we applied our classification code without prior knowledge (the classification algorithm has been described in the section 4.3.2.4) to these 7413 cover pages. We applied our classification code to sets with different numbers of cover pages. In our experiments, we applied it to the sets with 200, 400, 800, 1200, 2000, 3000, 4000, 5000, 6000, and 7413 cover pages respectively. The cover pages in each set are randomly selected from the 7413 cover pages. We recorded the number of groups that were generated by our classification code for each set. In the experiments with a set with a small number of cover pages (i.e. 200 and 400), we repeated the process four times and used the average number of groups as the results. The classification results are shown in Fig. 36. It shows that a small number of groups are required for processing most documents in a relatively large collection.

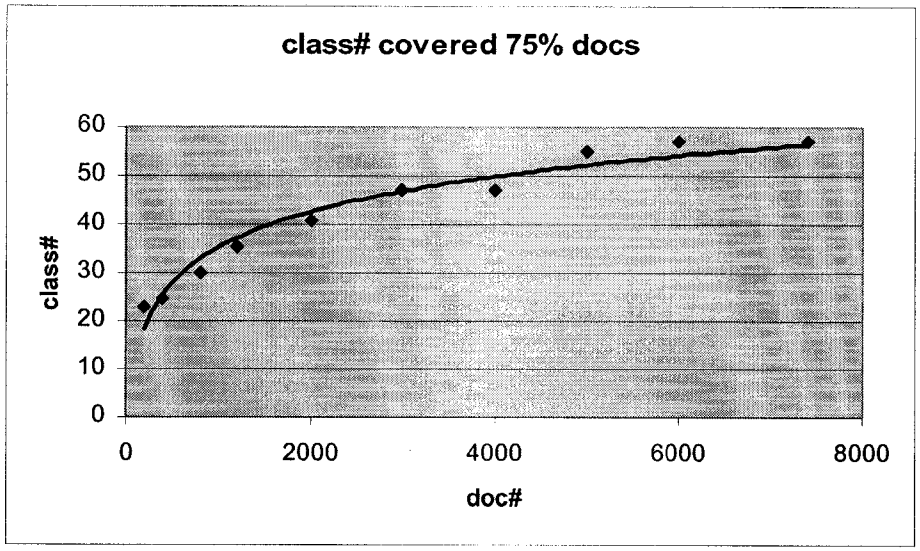


Fig. 36. Document classification result

To estimate the relations between the number of groups and the number of cover pages, we compared our classification results with several big-Os. The results are shown in Table 15. We observed that the function in Fig. 36 is a slow growing function, appearing to grow faster than *O(log N)* and slower than *O(sqrt(N))* where "*N*" is the number of cover pages.

During this experiment, our classification is based on unstructured metadata pages (i.e. cover pages) without prior knowledge. With the prior knowledge of structured metadata pages and some common unstructured pages, a portion of documents can be classified by their structured metadata pages or by their unstructured metadata pages with knowledge. In this experiment, some unnecessary singleton groups were generated. This is partially because of errors in the block boundary detection, which we described in Chapter 2.

TABLE 15
GROWING ESTIMATION

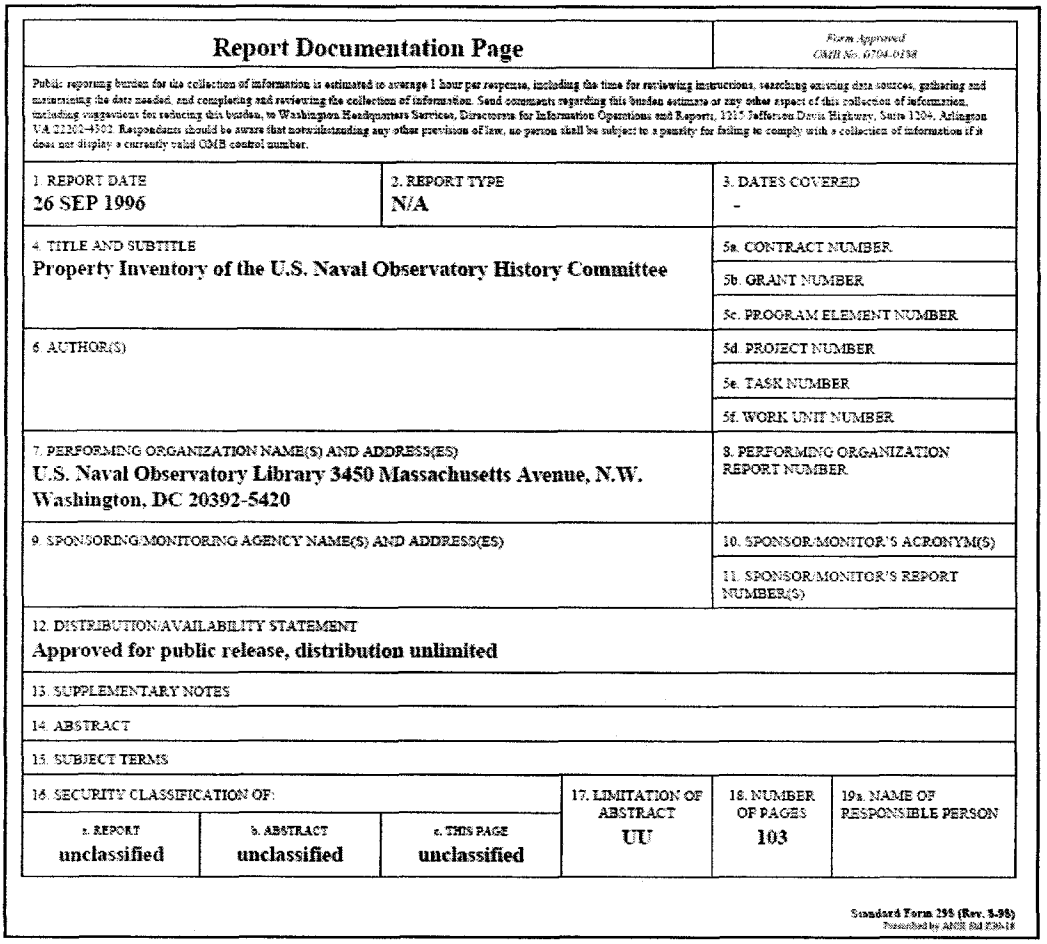
Doc#	Group#	O(1)	O(logN)	O(N)	O(NlogN)	$O(N^2)$	$O(N^{1/4})$	$O(N^{1/2})$
200	22.75	22.75	9.886877	0.11375	0.049434	0.000569	6.049562	1.608668
400	24.5	24.5	9.415617	0.06125	0.023539	0.000153	5.478367	1.225
800	30	30	10.33382	0.0375	0.012917	4.69E-05	5.640905	1.06066
1200	35.25	35.25	11.44785	0.029375	0.00954	2.45E-05	5.989131	1.01758
2000	41	41	12.42037	0.0205	0.00621	1.03E-05	6.13093	0.916788
3000	47	47	13.51693	0.015667	0.004506	5.22E-06	6.350641	0.858099
4000	47	47	13.04809	0.01175	0.003262	2.94E-06	5.909937	0.743135
5000	55	55	14.869	0.011	0.002974	2.2E-06	6.540639	0.777817
6000	57	57	15.08674	0.0095	0.002514	1.58E-06	6.47645	0.735867
7413	57	57	14.72871	0.007689	0.001987	1.04E-06	6.142941	0.66203
Goodness of fit:		0.313573	0.170557	1.044222	1.268398	2.17269	0.056026	0.297695

6.3.3. Evolution

The issue of evolution addresses how an approach can process new kinds of documents that are added to a collection over time. For a new kind of documents come, our template-based approach will create a new group and a new template for these documents. After that, document classification module will determine whether a document is old type or this new type. The old type documents will be processed as before, and the new type documents will be processed with the new template. DTIC provided us the date information of almost all documents in our DTIC collection. This enabled us to emulate a collection where new documents are added over time.

6.3.3.1. Experiments with Structured Metadata Page

In this subsection, we will use an example to demonstrate how our template-based approach processes new type structured metadata pages. Fig. 37 displays a structured metadata page existed before 1997 in our collection. Its template is available in the Appendix C.2 (see the template for the group "sf298_2").



Report Documentation Page

Form Approved OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
26 SEP 1996	N/A	-

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Property Inventory of the U.S. Naval Observatory History Committee	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) U.S. Naval Observatory Library 3450 Massachusetts Avenue, N.W. Washington, DC 20392-5420	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

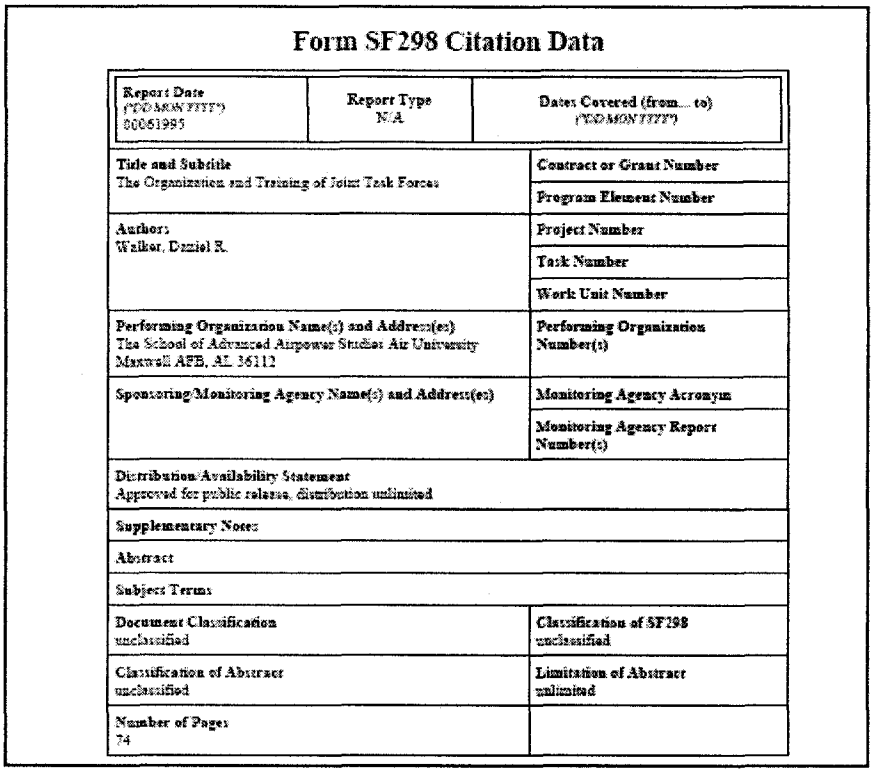
15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	unclassified	unclassified	UU	103	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

Fig. 37. A metadata page existed before 1997

Fig. 38 shows a type of structured metadata pages, which first appeared in 1997 in our collection. As shown in Fig. 37 and Fig. 38 respectively, these two metadata pages are different in the number of fields and the label set.



Form SF298 Citation Data

Report Date (DD MON YYYY) 00061995	Report Type N/A	Dates Covered (from... to) (DD MON YYYY)

Title and Subtitle The Organization and Training of Joint Task Forces	**Contract or Grant Number**
	Program Element Number
Author: Walker, Daniel R.	**Project Number**
	Task Number
	Work Unit Number
Performing Organization Name(s) and Address(es) The School of Advanced Airpower Studies Air University Maxwell AFB, AL 36112	**Performing Organization Number(s)**
Sponsoring/Monitoring Agency Name(s) and Address(es)	**Monitoring Agency Acronym**
	Monitoring Agency Report Number(s)
Distribution/Availability Statement Approved for public release, distribution unlimited	
Supplementary Notes	
Abstract	
Subject Terms	
Document Classification unclassified	**Classification of SF298** unclassified
Classification of Abstract unclassified	**Limitation of Abstract** unlimited
Number of Pages 74	

Fig. 38. A metadata page appeared in 1997

A new template, which is shown in Fig. 39, was created for processing the new type metadata page. A new group was created too. After that, the metadata pages of this new type would be classified into the new group and be processed with the new template.

```
<template>
<form max="-1">
<match max="5">
        <line>Form SF298 Citation Data</line>
</match>
<fixed>
        <field num="rd"><line>Report Date ("DD MON YYYY")</line></field>
        <field num="rd"><line>Report Date</line></field>
        <field num="rt"><line>Report Type</line></field>
        <field num="dc"><line>Dates Covered (from... to) ("DD MON YYYY")</line></field>
        <field num="dc"><line>Dates Covered (from... to)</line></field>
        <field num="dc"><line>Dates Covered ("DD MON YYYY")</line></field>
        <field num="dc"><line>Dates Covered</line></field>
        <field num="ts"><line>Title and Subtitle</line></field>
        <field num="cgn"><line>Contract or Grant Number</line></field>
        <field num="pen"><line>Program Element Number</line></field>
        <field num="pn"><line>Project Number</line></field>
        <field num="tn"><line>Task Number</line></field>
        <field num="wun"><line>Work Unit Number</line></field>
        <field num="a"><line>Authors</line></field>
        <field num="pona"><line>Performing Organization Name(s) and
Address(es)</line></field>
```

Fig. 39. New template (partial)

Table 16 displays the five groups of the structured metadata pages, and the year that they first appeared in our collection.

TABLE 16
THE EVOLUTION OF THE STRUCTURED METADATA PAGES

Group Name	Year
Sf298_2	1942
Sf298_1	1963
Sf298_4	1977
Sf298_3	1997
control	1997

We created a template for each group, and extracted metadata from our collection with these templates. We manually checked about 264 documents, and the results are presented in Table 17. Our template-based approach shows good results to handle the structured metadata pages of new documents that were added to our collection over time.

TABLE 17
METADATA EXTRACTION RESULTS OF STRUCTURED METADATA PAGES

Group Name	Precision	Recall	F-Measure
Sf298_1	95%	97%	96%
Sf298_2	92%	98%	95%
Sf298_3	93%	100%	96%
Sf298_4	100%	100%	100%
Citation_1	100%	100%	100%

6.3.3.1. Evolution Experiments

The objective of the experiments in this subsection to see how often we need to create a new group yearly in our collection. We used all documents before 1/1/2000 as the historical documents, and added the documents after 2000 into the collection by years. First, we classified the historical data into groups based on their metadata pages. After that, we added the documents into the collection year by year, classified the newly added documents into groups, and recorded how many new groups were created in each year. The experimental results are shown in Table 18.

In our experiments, the structured metadata pages were located with their fixed labels as we described in section 4.2, and the cover pages (unstructured metadata pages) were detected by the rules described at the end of the section 4.3.1. In Table 18, the column "Doc#" shows the number of documents that we have, the column "The number

of added groups" shows how many groups created in each year, and the column "The number of new groups per documents" shows the ratio of the number of the groups created to the number of documents added in each year. The results in Table 18 indicates that only a small number of groups need to created each year to process our DTIC collection.

TABLE 18
EVOLUTION EXPERIMENT RESULTS

Year	Doc#	The number of new groups	The number of new groups per document
Historical	733	44	6.00%
2000	367	2	0.54%
2001	1242	7	0.56%
2002	861	3	0.35%
2003	1844	21	1.14%
2004	4237	62	1.46%
2005	35	0	0.00%

We manually checked the groups that were created in 2000, 2001, and 2002. The results are promising. Every group contains only one type of metadata pages. The two groups were created in 2000 are presented in Fig. 40. Our code for classification of unstructured metadata pages can automatically detect and create a new group without human intervention. However, 2 out of these 12 groups should be merged. The classification algorithm for unstructured metadata pages could be refined to reduce unnecessary groups. The code for the classification of unstructured metadata pages

sometimes generated more groups than necessary. For example, the 7 news groups were created in 2000 in our experiment. However, in fact, they should be 6 groups.

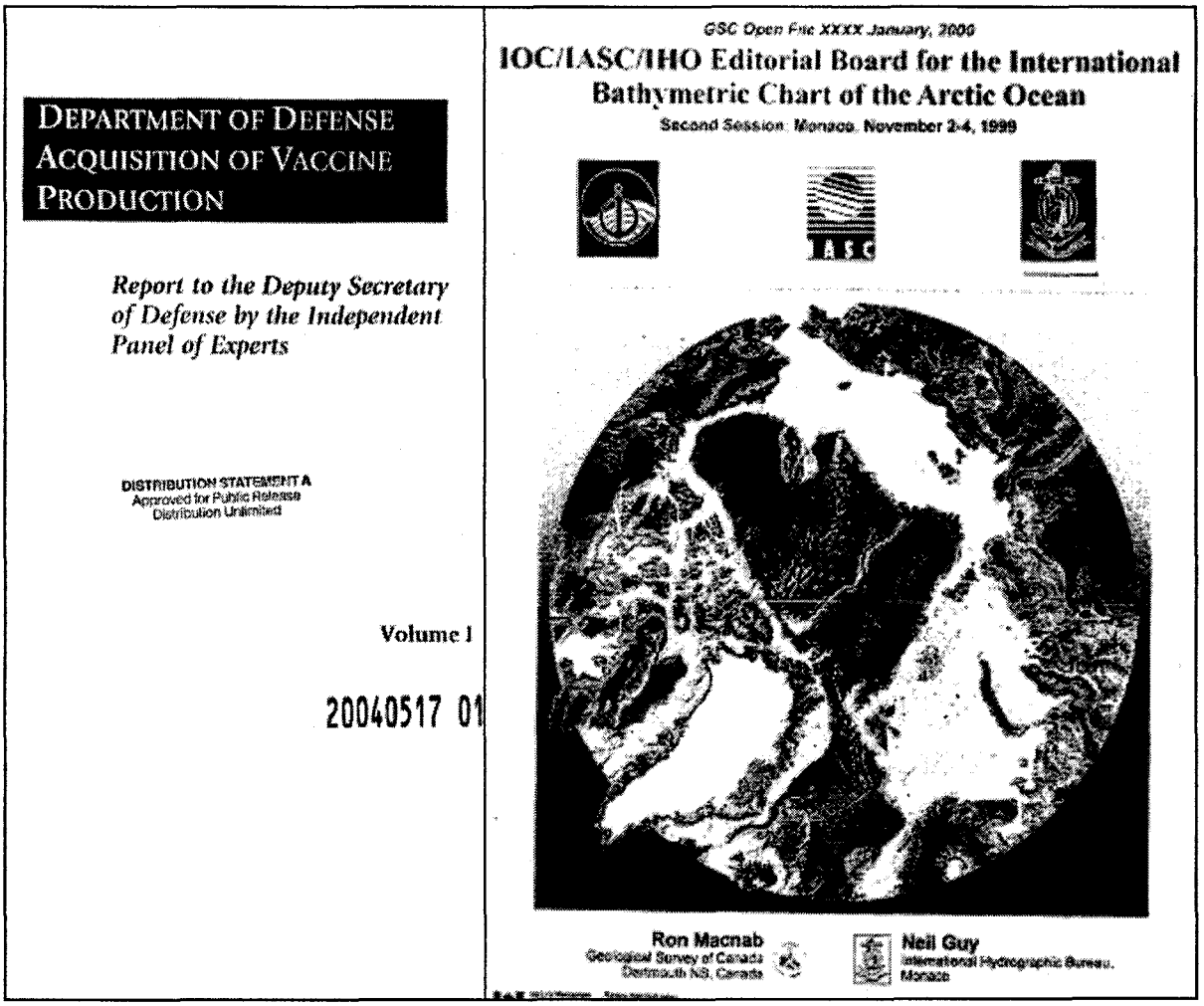


Fig. 40. Two metadata page groups added in 2000

6.3.4. Adaptability

Even though our template-based approach has been implemented to work with DTIC documents, it is possible to adapt our approach to another collection. In order to show the adaptability of our template-based approach, we applied it to a sample

collection of GPO (U.S. Government Printing Office) documents [69]. This collection contains 103 documents. The list of identifiers of this data set is available at Appendix D.1. Based on their metadata pages, we classified them into four groups: "GPOForm", "GPONonForm", "Congress Report", and "Public Law". The group names were chosen arbitrarily. Fig. 41, Fig. 42, Fig. 43, and Fig. 44 show sample metadata pages from these four groups respectively.

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
DOT/FAA/AR-05/29		

4. Title and Subtitle	5. Report Date
INSPECTION DEVELOPMENT FOR NICKEL BILLET—ENGINE TITANIUM CONSORTIUM PHASE II	September 2005
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
Mike Keller[1], Thadd Patton[1], Andrei Degtyar[2], Jeff Umbach[2], Waled Hassan[3], Andy Kinney[3], Ron Roberts[4], Frank Margetan[4], and Lisa Brasche[4]	

9. Performing Organization Name and Address		10. Work Unit No. (TRAIS)
[1]General Electric Company Cincinnati, Ohio 45215	[2]Honeywell Engines, Systems, & Services Phoenix, AZ	
[3]Pratt & Whitney East Hartford, CT	[4]Iowa State University Ames, IA	11. Contract or Grant No. DTFA0398FIA029

12. Sponsoring Agency Name and Address	13. Type of Report and Period Covered
U.S. Department of Transportation Federal Aviation Administration Office of Aviation Research Washington, DC 20591	Final Report
	14. Sponsoring Agency Code ANE-110

15. Supplementary Notes

The FAA William J. Hughes Technical Center Technical Monitor was Cu Nguyen.

16. Abstract

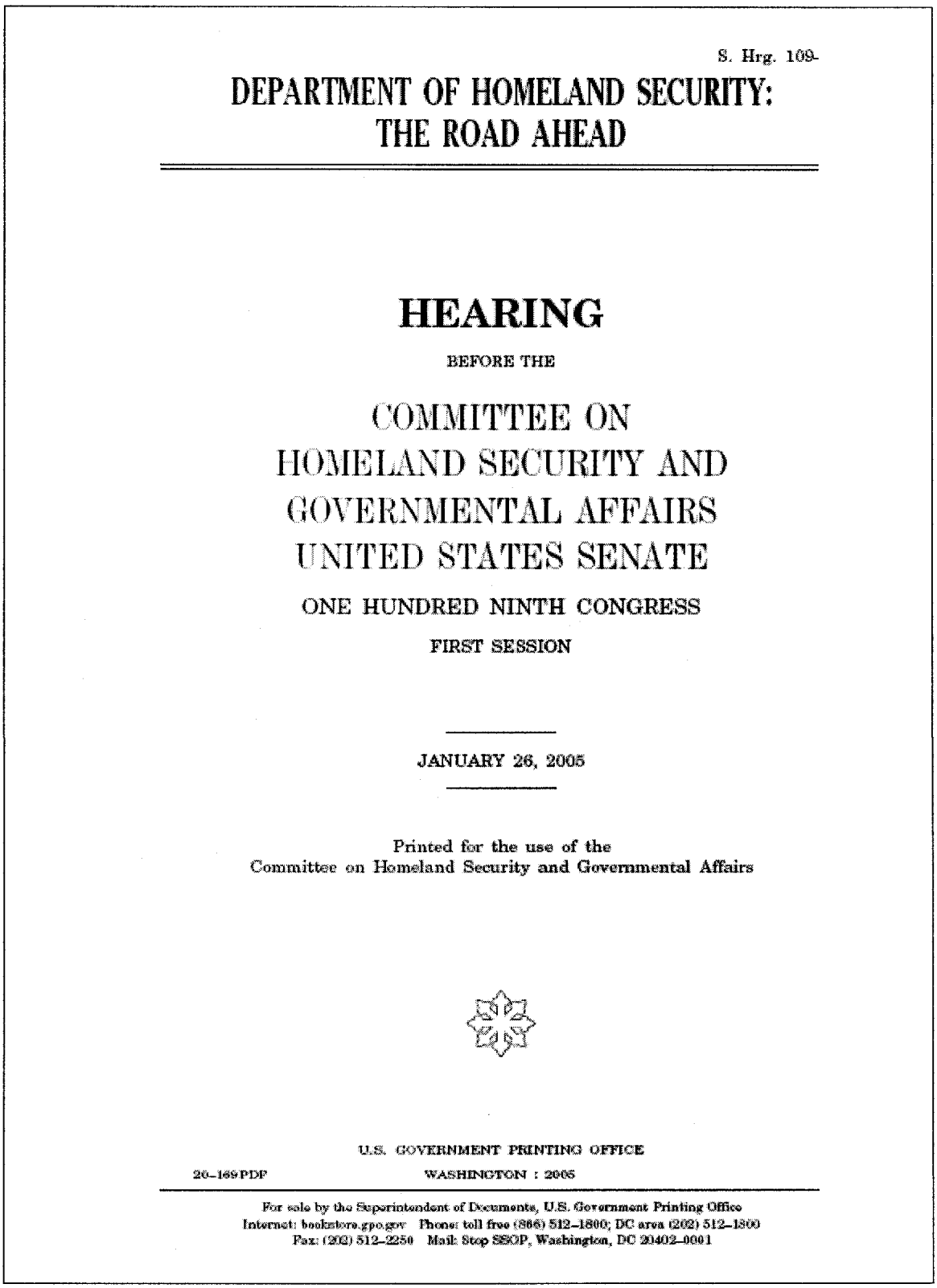
The Engine Titanium Consortium (ETC) is comprised of Iowa State University; General Electric; Pratt & Whitney; and Honeywell Engines, Systems & Services, and its objective was to develop improved inspection methods for critical jet engine components. The ETC Phase I program focused on improved inspection of titanium (Ti) billet using zoned inspection approaches. In the current work reported here, the multizone inspection procedure and transducers used for Ti were applied to typical nickel billet, namely Inconel 718 and Waspaloy. The program goal was achieved for 5" and 10"diameter billet.

17. Key Words	18. Distribution Statement
Nickel billet, Ultrasonic inspection	This document is available to the public through the National Technical Information Service (NTIS), Springfield, Virginia 22161.

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	119	

Form DOT F1700.7 (8-72) Reproduction of completed page authorized

Fig. 41. Metadata page sample of group "GPOForm"



S. Hrg. 109-

DEPARTMENT OF HOMELAND SECURITY: THE ROAD AHEAD

HEARING

BEFORE THE

COMMITTEE ON HOMELAND SECURITY AND GOVERNMENTAL AFFAIRS UNITED STATES SENATE

ONE HUNDRED NINTH CONGRESS

FIRST SESSION

JANUARY 26, 2005

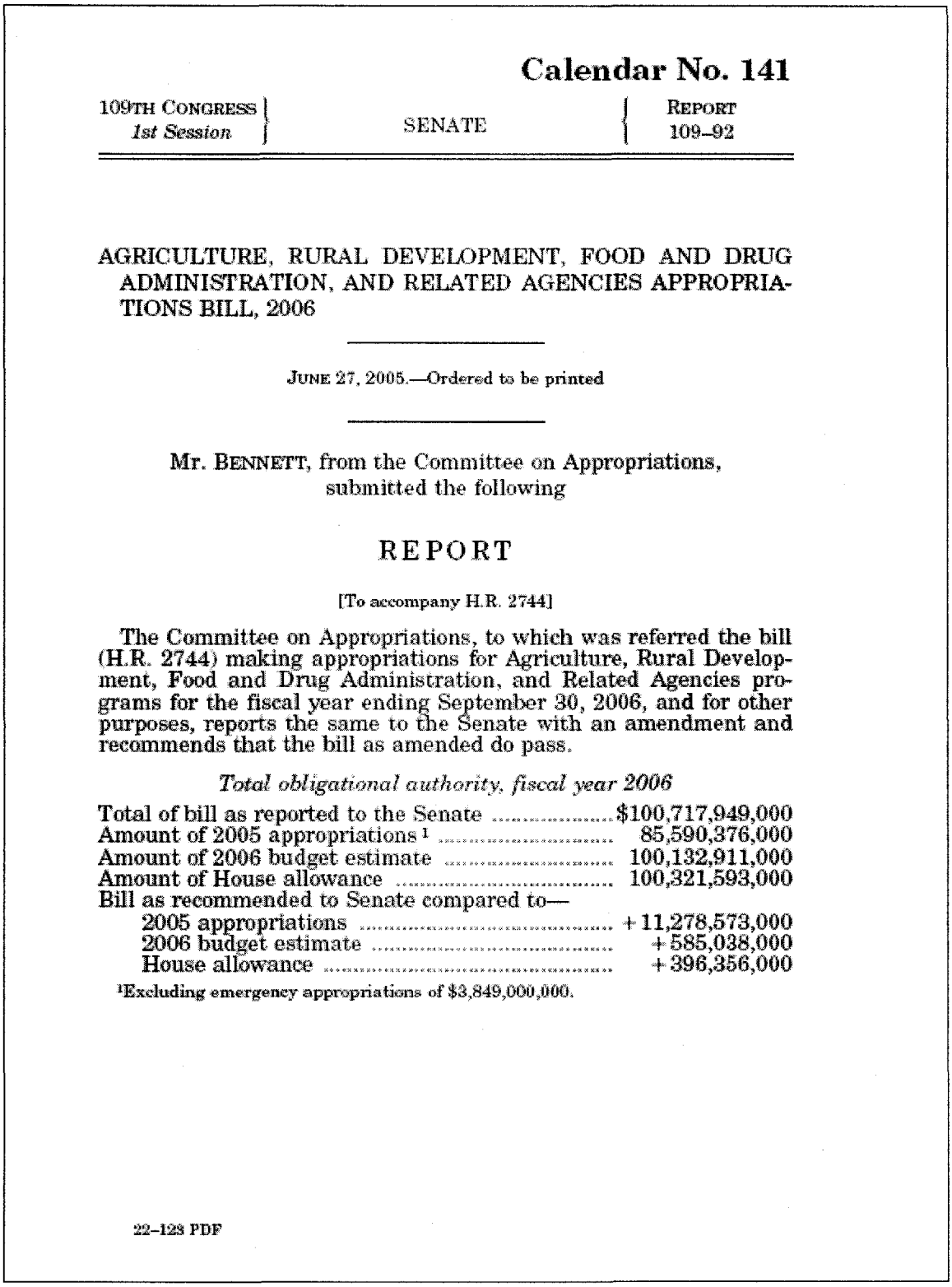
Printed for the use of the
Committee on Homeland Security and Governmental Affairs

U.S. GOVERNMENT PRINTING OFFICE

20-169 PDF WASHINGTON : 2005

For sale by the Superintendent of Documents, U.S. Government Printing Office
Internet: bookstore.gpo.gov Phone: toll free (866) 512-1800; DC area (202) 512-1800
Fax: (202) 512-2250 Mail: Stop SSOP, Washington, DC 20402-0001

Fig. 42. Metadata page sample of group "GPONonForm"



Calendar No. 141

109TH CONGRESS, 1st Session | SENATE | REPORT 109–92

AGRICULTURE, RURAL DEVELOPMENT, FOOD AND DRUG ADMINISTRATION, AND RELATED AGENCIES APPROPRIATIONS BILL, 2006

JUNE 27, 2005.—Ordered to be printed

Mr. BENNETT, from the Committee on Appropriations, submitted the following

REPORT

[To accompany H.R. 2744]

The Committee on Appropriations, to which was referred the bill (H.R. 2744) making appropriations for Agriculture, Rural Development, Food and Drug Administration, and Related Agencies programs for the fiscal year ending September 30, 2006, and for other purposes, reports the same to the Senate with an amendment and recommends that the bill as amended do pass.

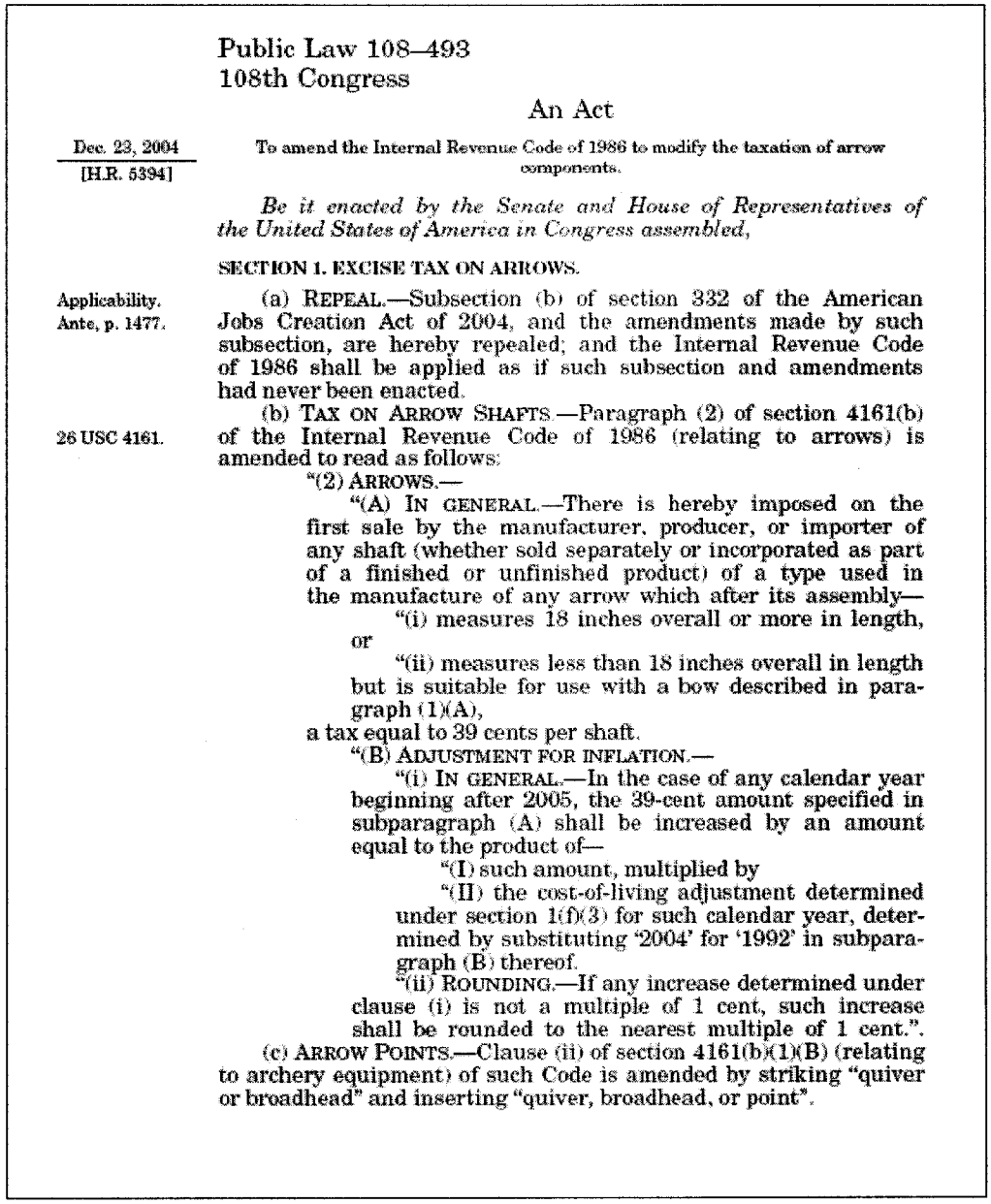
Total obligational authority, fiscal year 2006

Total of bill as reported to the Senate	$100,717,949,000
Amount of 2005 appropriations [1]	85,590,376,000
Amount of 2006 budget estimate	100,132,911,000
Amount of House allowance	100,321,593,000
Bill as recommended to Senate compared to—	
2005 appropriations	+11,278,573,000
2006 budget estimate	+585,038,000
House allowance	+396,356,000

[1]Excluding emergency appropriations of $3,849,000,000.

22–128 PDF

Fig. 43. Metadata page sample of group "Congress Report"



Public Law 108–493
108th Congress

An Act

Dec. 23, 2004
[H.R. 5394]

To amend the Internal Revenue Code of 1986 to modify the taxation of arrow components.

Be it enacted by the Senate and House of Representatives of the United States of America in Congress assembled,

SECTION 1. EXCISE TAX ON ARROWS.

Applicability.
Ante, p. 1477.

(a) REPEAL.—Subsection (b) of section 332 of the American Jobs Creation Act of 2004, and the amendments made by such subsection, are hereby repealed; and the Internal Revenue Code of 1986 shall be applied as if such subsection and amendments had never been enacted.

26 USC 4161.

(b) TAX ON ARROW SHAFTS.—Paragraph (2) of section 4161(b) of the Internal Revenue Code of 1986 (relating to arrows) is amended to read as follows:

"(2) ARROWS.—

"(A) IN GENERAL.—There is hereby imposed on the first sale by the manufacturer, producer, or importer of any shaft (whether sold separately or incorporated as part of a finished or unfinished product) of a type used in the manufacture of any arrow which after its assembly—

"(i) measures 18 inches overall or more in length, or

"(ii) measures less than 18 inches overall in length but is suitable for use with a bow described in paragraph (1)(A),

a tax equal to 39 cents per shaft.

"(B) ADJUSTMENT FOR INFLATION.—

"(i) IN GENERAL.—In the case of any calendar year beginning after 2005, the 39-cent amount specified in subparagraph (A) shall be increased by an amount equal to the product of—

"(I) such amount, multiplied by

"(II) the cost-of-living adjustment determined under section 1(f)(3) for such calendar year, determined by substituting '2004' for '1992' in subparagraph (B) thereof.

"(ii) ROUNDING.—If any increase determined under clause (i) is not a multiple of 1 cent, such increase shall be rounded to the nearest multiple of 1 cent.".

(c) ARROW POINTS.—Clause (ii) of section 4161(b)(1)(B) (relating to archery equipment) of such Code is amended by striking "quiver or broadhead" and inserting "quiver, broadhead, or point".

Fig. 44. Metadata page sample of group "Public Law"

We created a template for each group. The templates are available in Appendix D. Without changing our metadata extraction code, we applied our template-based approach to these documents. The metadata extraction results of group "GPOForm" are shown in Table 19. Even though the metadata pages in this group are different from those in our

DTIC collection, we succeeded to get high accuracy for most metadata fields without changing the metadata extraction code. We got a low recall/precision for the field "performing_organization". In some documents, the value of this field has more than one column, however in our current implementation, we order the lines based on their coordinates. As the result, the extracted data were out of order.

TABLE 19
METADATA EXTRACTION RESULTS OF GROUP "GPOFORM"

Field	#doc	#c	#p	#in	Recall		Precision		F-Measure	
					compl	partial	compl	partial	compl	partial
report_num	14	14	0	0	100%	100%	100%	100%	100%	100%
government_accession_num	0	0	0	0	N/A	N/A	N/A	N/A	N/A	N/A
recipient_catalog_num	0	0	0	0	N/A	N/A	N/A	N/A	N/A	N/A
Title	14	14	0	0	100%	100%	100%	100%	100%	100%
Reportdate	14	14	0	0	100%	100%	100%	100%	100%	100%
Performing_organization_code	4	4	0	0	100%	100%	100%	100%	100%	100%
Creator	14	14	0	0	100%	100%	100%	100%	100%	100%
Performing_number	0	0	0	0	N/A	N/A	N/A	N/A	N/A	N/A
Performing_organization	14	9	5	0	64.29%	100%	64.29%	100%	64.29%	100%
Work_unit_num	1	1	0	0	100%	100%	100%	100%	100%	100%
contract_grant_num	6	6	0	0	100%	100%	100%	100%	100%	100%
Sponsor	14	14	0	0	100%	100%	100%	100%	100%	100%
report_type_coverage	14	14	0	0	100%	100%	100%	100%	100%	100%
sponsor_code	14	14	0	0	100%	100%	100%	100%	100%	100%
Notes	10	10	0	0	100%	100%	100%	100%	100%	100%
Abstract	14	14	0	0	100%	100%	100%	100%	100%	100%
Keyword	14	13	1	0	92.86%	100%	92.86%	100%	92.86%	100%
dist_statement	14	14	0	0	100%	100%	100%	100%	100%	100%
sec_classification_report	14	14	0	0	100%	100%	100%	100%	100%	100%
sec_classification_page	14	14	0	0	100%	100%	100%	100%	100%	100%
Num_page	14	14	0	0	100%	100%	100%	100%	100%	100%
Price	0	0	0	0	N/A	N/A	N/A	N/A	N/A	N/A

Table 20 shows the metadata extraction results of group "GPONonForm". The completely correct results of these metadata fields, except field "title" and field "serialno", are desirable. For the field "title", 12 out of 19 partially correct data contains just one single character error. If we ignore this error, the recall/precision under the "compl" column will be 85.96%. All extracted data of the field "serialno" contain one single character error. If we replace the character '?' with the character '-' in the extracted data, the recall/precision under the "compl" column will be 100%.

TABLE 20
METADATA EXTRACTION RESULTS OF GROUP "GPONONFORM"

Field	#doc	#c	#p	#in	Precision		Recall		F-measure	
					compl	partial	compl	partial	compl	partial
Title	57	38	19	0	66.67%	100%	66.67%	100%	66.67%	100%
Type	57	56	1	0	98.25%	100%	98.25%	100%	98.25%	100%
Session	57	55	2	0	96.49%	100%	96.49%	100%	96.49%	100%
Date	55	52	0	0	100%	100%	94.55%	94.55%	97.20%	97.20%
Serialno	30	0	30	0	0%	100%	0%	100%	N/A	100%
Use	55	51	0	0	100%	100%	92.73%	92.73%	96.23%	96.23%

Table 21 shows the metadata extraction results of the group "Congress Report". The results of most metadata fields are desirable. However, we got low precision for the field "date" and the field "creator". This is because that the smallest unit of our current metadata extraction code is a line, but the metadata "date" or "creator" on a metadata page in this group is just a part of line. Extending our engine to make it be able to work with smaller units such as a word or a phrase can improve the results. We failed to extract the field "session" correctly due to two reasons. First, most extracted data of the field "session" have OCR errors. Second, our current implementation has limitation to order

the text in multiple columns. All the data extracted for the field "session" are partially correct.

TABLE 21
METADATA EXTRACTION RESULTS OF GROUP "CONGRESS REPORT"

Field	#doc	#c	#p	#in	Precision		Recall		F-measure	
					compl	partial	compl	partial	compl	partial
candno	6	6	0	0	100%	100%	100%	100%	100%	100%
session	16	0	16	0	0.00%	100%	0.00%	100%	N/A	100%
Title	16	14	2	0	87.50%	100%	87.50%	100%	87.50%	100%
Date	16	0	13	1	0.00%	92.86%	0.00%	81.25%	N/A	86.67%
Creator	14	0	13	1	0.00%	92.86%	0.00%	92.86%	N/A	92.86%
Type	16	13	3	0	93.94%	100%	91.18%	100%	92.54%	100%
accompany	16	15	1	0	85.00%	100%	100%	100%	91.89%	100%

Table 22 shows the metadata extraction results of the group "Public Law". We got high accuracy results for the field "congress_number" and the field "type". In our current implementation, we cannot locate a label in a text string if the label does not occur at the beginning of the text string. This is the main reason that we got a low recall/precision for the metadata "bill_number". In some documents the field "bill_number" occurs in the middle of a line. Extending our current engine to add new features for locating a label or a special pattern (e.g. a regular expression) will improve the results. We failed to get desirable results for the field "date" because in our current implementation, the small unit is line and the field "date" is just a part of the line. Instead of extracting the extract information, we extracted the whole line.

TABLE 22
METADATA EXTRACTION RESULTS OF GROUP "PUBLIC LAW"

Field	#doc	#c	#p	#in	Recall		Precision		F-Measure	
					compl	partial	compl	partial	compl	partial
Date	16	0	16	0	0.00%	100%	0.00%	100%	N/A	100%
Bill_number	16	6	3	0	66.67%	56.25%	37.50%	100%	48.00%	72.00%
Congress_num	16	16	0	0	100%	100%	100%	100%	100%	100%
Type	16	16	0	0	100%	100%	100%	100%	100%	100%

Our experiments with the sample collection of GPO documents demonstrated the adaptability of our template-based approach. In our experiments, we succeeded to get desirable results for some fields without changing our template processing code. For most other fields we got partially correct results. However, our current implementation has a few limitations that affect the adapting of it to another collection. The following are a list of these limitations:

1) The smallest unit is a line in our current implementation. Therefore, we have problems with extracting a metadata correctly if it is only a part of a line. For example, in our experiments, we got very low precisions for the field "creator" and the field "date" of the group;
2) Incomplete feature set; our currently implemented features are not complete. Sometimes, we have problems with extracting some metadata fields. For example, for the field "bill_number" of the group "Public Law", we need to add a new feature for searching a specific label or even a specific pattern in a text string;
3) Our current implementation ordered the lines by their coordinates. It has limitation to process the multiple-column text.

The first limitation can be addressed by extending our engine to work on the hierarchy structure of a document, including on smaller units such as a phrase or a word. The second limitation can be addressed by developing a relatively complete feature set or making the feature set extensible (i.e. a new feature can be defined based on the existing feature set). For the third limitation, a more sophisticated algorithm to order the text is required. A possible refinement is to detect the columns in the text.

6.3.5. Complexity

Our template-based approach addressed the complexity issue by classifying documents into fine-grained groups to simplify the task of creating templates. In this section, we will introduce our experiment with aims to show whether our template-based approach simplifies the tasks of creating templates.

6.3.5.1. Complexity Measures

We used software complexity measure - Halstead Complexity Measures [75] to evaluate the complexity of our templates. Halstead Complexity Measures are based on the numbers of operators and operands used in source code. There are four complexity measures: Measure of Program Length N, Measure Volume V, Measure Difficulty D, and Measure Effort E. These four measures can be defined as following equations:

$$N=N_1+N_2$$

$$V = N \times Log_2(n_1 + n_2)$$

$$D = \frac{n_1}{2} \times \frac{N_2}{n_2}$$

$$E = D \times V$$

Where n_1 stands for the unique number of operators, n_2 for the unique number of operands, N_1 for the total number of operators, and N_2 for the total number of operands.

To use Halstead Complexity Measures to evaluate our templates, we first convert any feature into a XML element. The feature "begin" is converted to *<loc type= "begin">*. The feature "end" is converted to *<loc type= "end">*. For all other features the feature names are used as the element names. Any parameter is either converted to an attribute or text content. We call a template after this conversion a "normalized template". For example, the feature "size(1300,1700)" will be converted to "<size min="1300" max="1700" />". Fig. 45 shows a sample template after we converted the features to their corresponding XML elements. Then we treat each element as an operator and the attributes and the text content of the element as its operands. For a template sample shown in Fig. 45, for example, we can think of "stringmatch" as an operator and the operands are its text content and its attributes such as whether it is case sensitive, which string to match, whether it is an extract match or a partial match.

```
<structdef>
  <meta name="title" min="1" max="1">
     <begin inclusive="current"> <size min="1300" max="1700" /></begin>
     <end inclusive="before">
          <stringmatch case="yes" loc="beginwith">HEARING</stringmatch>
     </end>
  </meta>
  <meta name="reporttype" min="1" max="1">
      <begin inclusive="current">
         <stringmatch case="yes" loc="beginwith">HEARING</stringmatch>
      </begin>
      <end inclusive="before"> <size min="800" max="1000"/></end>
  </meta>
  <meta name="date" min="0" max="1">
    <begin inclusive="current"><dateformat/></begin>
    <end inclusive="current"><onesection/></end>
  </meta>
</structdef>
```

Fig. 45. A Template Sample

The Halstead Complexity Measures of the sample shown in Fig. 45 are computed as follows:

$$N_1 = 16;\ n_1 = 8;\ N_2 = 25;\ n_2 = 15;$$

$$N=N_1+N_2=16+25=41$$

$$V = N \times Log_2(n_1 + n_2) = 41 \times Log_2(8+15) \approx 185.47$$

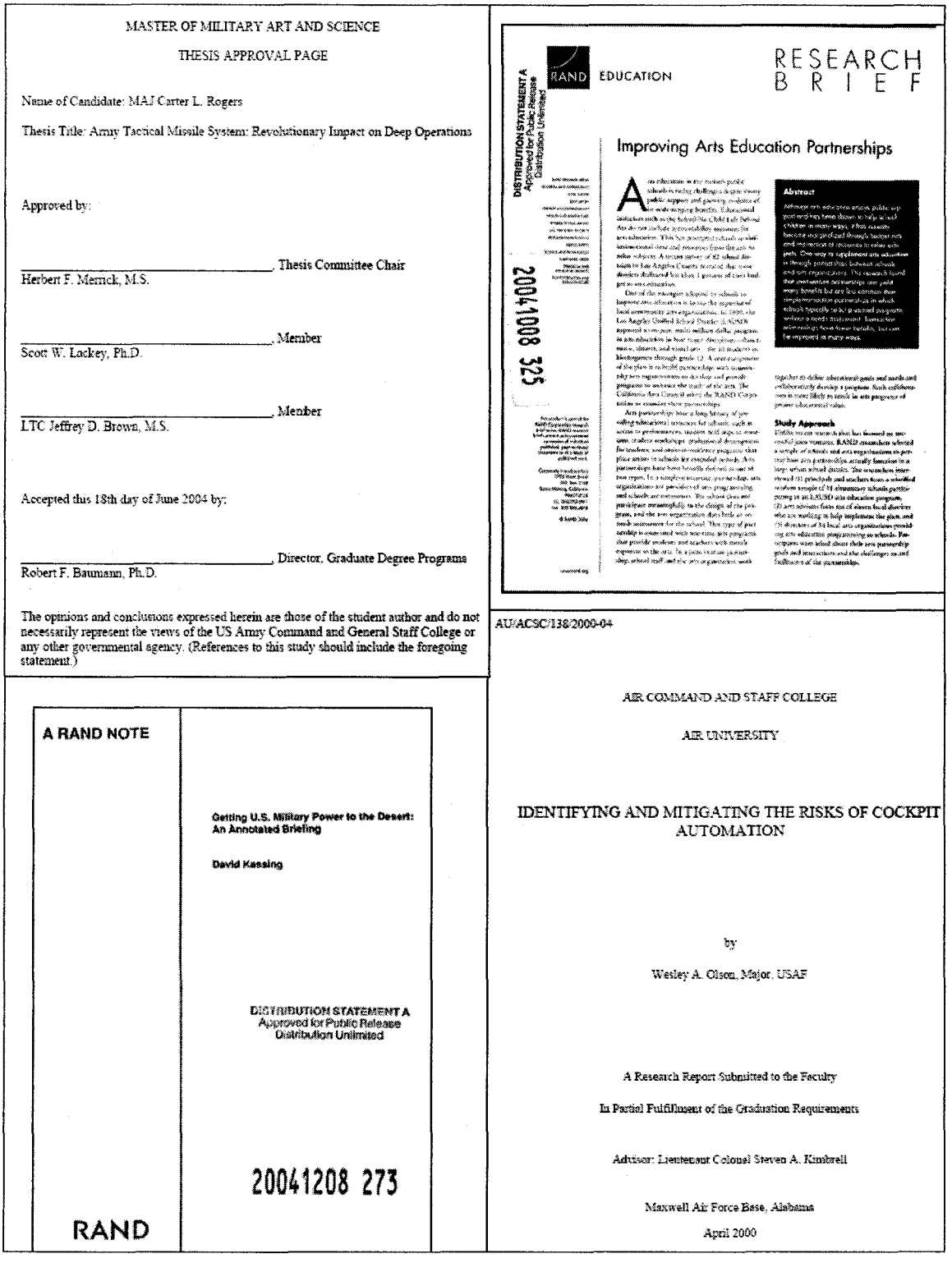
$$D = \frac{n_1}{2} \times \frac{N_2}{n_2} = \frac{8}{2} \times \frac{25}{15} \approx 6.67$$

$$E = D \times V \approx 6.67 \times 185.47 \approx 1236.44$$

6.3.5.2. Experiment

The basic idea of our experiment is to compare the complexity of creating templates with classification with the complexity of creating a generic template without a template for a collection.

First, we selected a subset of documents from our DTIC test bed. This subset consists of four groups. The sample metadata pages of these four documents are shown in Fig. 46. Then we created one template for each group, and one generic template for all these four groups (i.e. create a template without classification). Finally, we measured and compared the complexity of the templates with classification and the complexity of the generic template without classification.



MASTER OF MILITARY ART AND SCIENCE

THESIS APPROVAL PAGE

Name of Candidate: MAJ Carter L. Rogers

Thesis Title: Army Tactical Missile System: Revolutionary Impact on Deep Operations

Approved by:

_____, Thesis Committee Chair
Herbert F. Merrick, M.S.

_____, Member
Scott W. Lackey, Ph.D.

_____, Member
LTC Jeffrey D. Brown, M.S.

Accepted this 18th day of June 2004 by:

_____, Director, Graduate Degree Programs
Robert F. Baumann, Ph.D.

The opinions and conclusions expressed herein are those of the student author and do not necessarily represent the views of the US Army Command and General Staff College or any other governmental agency. (References to this study should include the foregoing statement.)

DISTRIBUTION STATEMENT A
Approved for Public Release
Distribution Unlimited

20041008 325

RAND EDUCATION

RESEARCH BRIEF

Improving Arts Education Partnerships

Abstract

AU/ACSC/138/2000-04

AIR COMMAND AND STAFF COLLEGE

AIR UNIVERSITY

IDENTIFYING AND MITIGATING THE RISKS OF COCKPIT AUTOMATION

by

Wesley A. Olson, Major, USAF

A Research Report Submitted to the Faculty

In Partial Fulfillment of the Graduation Requirements

Advisor: Lieutenant Colonel Steven A. Kimbrell

Maxwell Air Force Base, Alabama

April 2000

A RAND NOTE

Getting U.S. Military Power to the Desert: An Annotated Briefing

David Kassing

DISTRIBUTION STATEMENT A
Approved for Public Release
Distribution Unlimited

20041208 273

RAND

Fig. 46. Metadata page samples

The normalized templates for these four groups are shown in Fig. 47, Fig. 48, Fig. 49, and Fig. 50.

```
<?xml version="1.0" ?>
<structdef>
  <meta name= "degree" min="1" max="1">
     <begin inclusive="current"><loc type="begin"/></begin>
     <end inclusive="current"><onesection/></end>
  </meta>
  <meta name= "creator" min="1" max="1">
     <begin inclusive="current">
      <stringmatch loc= "beginwith" case= "yes">Name of Candidate</stringmatch>
    </begin>
     <end inclusive="current"><onesection/></end>
  </meta>
  <meta name= "title" min="1" max="1">
      <begin inclusive="current">
         <stringmatch case="yes" loc="beginwith">Thesis Title</stringmatch>
      </begin>
      <end inclusive="before"><rvspace min="1" /></end>
  </meta>
</structdef>
```

Fig. 47. Template 1

```
<?xml version="1.0" ?>
<structdef>
  <meta name= "title" min="1" max="1">
      <begin inclusive="current">
          <largeststr start= "0" end= "0.5" minwc= "4"/>
      </begin>
      <end inclusive="before"><sizechange/></end>
  </meta>
</structdef>
```

Fig. 48. Template 2

```
<?xml version="1.0" ?>
<structdef>
  <meta name="title" min="1" max="1">
      <begin inclusive="current">
          <largeststr start= "0" end= "0.5" minwc= "4"/>
      </begin>
     <end inclusive="before"><rvspace min="1"/></end>
  </meta>
  <meta name="creator" min="1" max="1">
      <begin inclusive="after"><rmeta ref="title"/></begin>
      <end inclusive="before"><rvspace min="1" /></end>
  </meta>
</structdef>
```

Fig. 49. Template 3

```
<?xml version="1.0" ?>
<structdef>
  <meta name="identifier" min="1" max="1">
     <begin inclusive="current"><loc type="begin"/></begin>
     <end inclusive="before"><onesection/></end>
  </meta>
  <meta name="contributor" min="1" max="1">
     <begin inclusive="after"><rmeta ref="identifier"/></begin>
     <end inclusive="before"><rvspace min="2" /></end>
  </meta>

  <meta name="title" min="1" max="1">
     <begin inclusive="after"><rmeta ref="contributor"></begin>
     <end inclusive="before"><rvspace min="2" /></end>
  </meta>
  <meta name="creator" min="1" max="1">
      <begin inclusive="after">
         <stringmatch loc="equal" case="no">by</stringmatch>
      </begin>
      <end inclusive="before"><rvspace min="2" /></end>
  </meta>
  <meta name="date" min="1" max="1">
      <begin inclusive="current"><dateformat/></begin>
      <end inclusive="before"><onesection/></end>
  </meta>
</structdef>
```

Fig. 50. Template 4

To create a generic template for documents from all these four groups, we simply extended our template language so that we can use a logic combination of multiple rules to locate a metadata field. Three new elements "or", "and", and "not" were added for specifying the logic relations between rules. A generic template written in this extended language is shown in Fig. 51. It is for documents from all the four groups.

```
<?xml version="1.0" ?>
<structdef>
  <meta name="identifier" min="0" max="1">
     <begin inclusive="current">
               <stringmatch case="yes" loc="beginwith">AU</stringmatch>
     </begin>
     <end inclusive="before"><onesection/></end>
  </meta>
  <meta name= "title" min="1" max="1">
      <or>
        <begin inclusive="current">
               <stringmatch case="yes" loc="beginwith">Thesis Title</stringmatch>
        </begin>
        <begin inclusive="current">
               <largeststr start= "0" end= "0.5" minwc= "4"/>
        </begin>
      </or>
      <end inclusive="before"><rvspace min="1" /></end>
  </meta>
  <meta name= "creator" min="0" max="1">
    <or>
       <begin inclusive="current">
               <stringmatch loc= "beginwith" case= "yes">Name of Candidate</stringmatch>
       </begin>
       <begin inclusive="after">
               <stringmatch loc= "onesection" case= "no">by</stringmatch>
       </begin>
       <and>
           <begin inclusive="current"><nameformat/></begin>
           <begin inclusive="after"><rmeta ref="title"/></begin>
       </and>
     </or>
     <end inclusive="before"><rvspace min= "1" /></end>
  </meta>

  <meta name="contributor" min="1" max="1">
     <begin inclusive="after"><rmeta ref="identifier"/></begin>
     <end inclusive="before"><rvspace min="2" /></end>
  </meta>

  <meta name="degree" min="1" max="1">
     <begin inclusive="current">
               <stringmatch case="yes" loc="beginwith">MASTER OF|DOCTOR OF</stringmatch>
     </begin>
     <end inclusive="before"><onesection/></end>
  </meta>

  <meta name="date" min="0" max="1">
      <begin inclusive="current"><dateformat/></begin>
      <end inclusive="before"><onesection/></end>
  </meta>
</structdef>
```

Fig. 51. Generic Template

Table 23 shows the complexity of the templates with classification and the complexity of creating a generic template.

TABLE 23
COMPLEXITY COMPARISON

	N1	N2	n1	n2	N	V	D	E
Template1	15	23	8	14	38	169.46	6.57	1113.58
Template2	6	8	6	8	14	53.30	3.00	159.91
Template3	10	10	7	12	20	84.96	2.92	247.80
Template4	25	34	10	12	59	263.11	14.17	3727.34
Sum 1-4								5248.63
Generic Template	42	57	13	29	99	533.84	12.78	6820.26

From Table 23, the total Halstead effort of creating four separate templates for four groups is a slightly smaller than the effort of creating one generic template. Our results indicate that for a small number of groups the difference between the effort of creating a generic template and the total effort of creating separate templates can be little. However, the effort to create a template for an individual group is smaller than the effort to create a generic template. Our results also indicate that the effort to create a template varies from one group to another group. The effort to create a template for some group (e.g. template 2 or template 3 in our experiment) can be significantly less than the effort to create a generic template.

The complexity of creating templates is just one aspect. In aspect of the complexity of maintenance, our template-based approach has some advantages over the approach of using one generic template. First, a template in our template-based approach is simpler and easier to understand than a generic template. In this way, our template-based approach not only reduces the possibility of having errors in a template, but also simplifies the task of fixing the bugs. Furthermore, in our template-based approach a template for one group is independent of templates for other groups. Therefore, changing one template will not affect the results of other groups. Moreover, the creation of a new

template does not require understanding the existing templates. However, in the approach of using one generic template, whenever you want to make a change, you have to understanding the template. In addition, your change for handling new kinds of documents may affect the results of documents in existing types.

CHAPTER VII

CONCLUSIONS AND FUTURE WORK

7.1. Conclusions

Using metadata not only helps resource discovery, but can also make a collection interoperable with the help of OAI-PMH. The high cost of the manual creation of metadata for a large collection implies a great demand on tools for automatically extracting metadata from a collection. However, existing automatic metadata extraction approaches have limitations on working with a large heterogeneous collection. This dissertation has proposed a template-based approach to automate the task of extracting metadata from a large legacy collection. This dissertation has addressed the following questions: How do we achieve a high accuracy for a heterogeneous collection? How do we apply our template-based approach to a very large collection? How does the template-based approach handle new documents that added to a collection over time? How do we apply our approach to a new document collection? How complex are the document features that are used in our template-based approach?

The template-based approach first classifies documents into groups, and then creates a template for each group. In this way, a heterogeneous collection is converted to a set of homogeneous sub-collections. Templates are written in a designed language, which can be understood by the metadata extraction code. As such, the template-based approach should be able to work with different collections. Ideally, by creating new templates, the template-based approach should work with new kinds of documents that

are added to a collection over time or be adapted to a different collection without changing the metadata extraction code.

As we have described in Chapter 1, our objectives are:

- To develop a flexible and adaptable approach for extracting metadata from physical collections, with the focus on the DTIC collections;
- To develop an efficient approach of classifying documents into document groups;
- To integrate the techniques and tools developed for DTIC test bed into an interoperable digital library framework;

This research has met these objectives. First, a template-based approach has been developed for extracting metadata from physical collections. Our template-based approach has the flexibility to use different templates for different document groups. Our template-based approach can also be adapted to a different collection by creating a new set of templates even though there are some limitations in our current implementation. Secondly, we have developed an approach of classifying documents into groups based on documents' metadata pages. We first divide metadata pages into structured metadata pages and unstructured metadata pages, and then classify metadata pages into fine-grained groups. Lastly, we have integrated the techniques and tools developed for DTIC test bed into an interoperable digital library framework OAI-PMH.

There are a number of projects that extract metadata from legacy collections. Most do not target a large heterogeneous collection. Few have addressed scaling issue, adaptability issue, and evolution issue. The function of locating the metadata pages among documents is not seen in other projects. Our template-based approach is unique since it finds metadata pages from documents, classifies documents into group based on

its metadata pages, decouples the templates from metadata extraction code, and loads templates at running time.

The template-based approach has developed for Defense Technical Information Center to process its legacy collection. We expect that our template-based approach will help other organizations with extracting metadata from their collections as well. We also expect that with the ability of automatically extracting metadata from documents, our template-based approach of metadata extraction will be beneficial to the users of publishing tools such as Kepler (http://kepler.cs.odu.edu), whose users have to create metadata manually at this time. This dissertation has also demonstrated a feasible way to automate the task of building an OAI compliant digital library from a large legacy collection. An automated tool like this will simplify the task of creating a data provider, and therefore may attract more organizations to join OAI-PMH framework.

7.2. Future Work

We have demonstrated that our template-based approach is a feasible way to achieve high accuracy for heterogeneous collections. In this section, we will briefly discuss some potential areas for future work.

One possible enhancement is to integrate metadata from different kinds of pages. A document may have more than one page containing metadata. For example, a document may have a cover page, a title page and a form page. The cover page might have a title, an author, and a publication date. The title page might have a title, an author, and an abstract. The form page might have a title, a report number, and sponsoring organization. Extracting metadata from all the three pages will get more information than extracting metadata from only one page. Integrating information from multiple pages

may increase the quality of metadata because redundant occurrences of a metadata field also give a chance to correct the errors in OCR or metadata extraction.

Another possible development is to extend our metadata extraction code to work with a hierarchy document structure instead of working on the line level only. The feature set and rule language could be also improved.

Other possible enhancements include: the use of machine-learning techniques to evaluate the quality of extracted metadata, the integration of machine-learning approaches and rule-based approaches for metadata extraction, the use of knowledge bases for metadata extraction, OCR error correction, and the use of machine-learning techniques for document classification.

REFERENCES

[1] Adobe Acrobat Capture 3.0 White Paper, "The Four Flavors of Adobe PDF for Paper-based Documents," <http://www.adobe.com/products/acrcapture/pdfs/aacflavors.pdf> (18 February 2006) .

[2] M. Aiello, C. Monz, L. Todoran, and M. Worring. Document understanding for a broad class of documents. IJDAR, 2002.

[3] P. E. Black. "Levenshtein distance," 10 November 2005. <http://www.nist.gov/dads/HTML/Levenshtein.html> (17 March 2006).

[4] O. Altamura, F. Esposito, and D. Malerba. Transforming Paper Documents into XML Format with WISDOM++. International Journal of Document Analysis and Recognition, 3(2):175--198, 2000.

[5] A. Anjewierden. AIDAS: Incremental Logical Structure Discovery in PDF Documents. In 6th International Conference on Document Analysis and Recognition (ICDAR), pages 374-378, Seattle, September 2001.

[6] D. E. Appelt and D. J. Isarel. Introduction to Information Extraction Technology, A Tutorial Prepared for IJCAI-99.

[7] "Archon: A Digital Library that federates Physics with varying degrees of metadata richness," <http://archon.cs.odu.edu/> (10 September 2005).

[8] W. Arms. Digital Libraries. MIT Press, Cambridge, MA, 1999.

[9] D. Bergmark. Automatic Extraction of Reference Linking Information from Online Documents. CSTR 2000-1821, November 2000.

[10] D. P. Bertsekas, A. Nedic, and A. E. Ozdaglar. "Convexity, Duality, and Lagrange Multipliers," <http://citeseer.ist.psu.edu/499868.html> (10 February 2006).

[11] M. E. Blake and F. L. Knudson. Metadata and Reference Linking. Library Collections, Acquisitions, & Technical Services 26 (2002) 219-230.

[12] V. Borkar, K. Deshmukh, and S. Sarawagi. Automatic segmentation of text into structured records. In SIGMOD, 2001.

[13] "British English Word Lists for Spell Checkers," 2003, <http://www.curlewcommunications.co.uk/wordlist.html> (6 May 2004).

[14] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery, 2(2): 955-974, 1998.

[15] P. Caplan. Reference Linking for Journal Articles: Promise, Progress and Perils. Portal: Libraries and the Academy, vol. 1, no. 3, pp. 352-356.

[16] P. Caplan and W. Y. Arms. "Reference linking for journal articles," *D-Lib Magazine* [online journal], July 1999, <http://www.dlib.org/dlib/july99/caplan/07caplan.html> (8 August 2004).

[17] F. Cesarini, M. Lastri, S. Marinai, and G. Soda. Encoding of modified X-Y trees for document classification. In Proc. Sixth ICDAR, pages 1131–1136, 2001.

[18] F. Cesarini, M. Gori, S. Marinai, and G. Soda. *Structured document segmentation and representation by* the *modified X-Y tree.* In Proc. Fifth ICDAR, pages 563--566, 1999.

[19] C.-C. Chang and C.-J. Lin, "LIBSVM : A Library for Support Vector Machines," 2001. <http://www.csie.ntu.edu.tw/~cjlin/libsvm> (10 November 2003).

[20] "CLIPS: A Tool for Building Expert Systems," <http://www.ghg.net/clips/CLIPS.html> (12 February 2006).

[21] A. Crystal and P. Land. Metadata and Search: Global Corporate Circle DCMI 2003 Workshop, Seattle, Washington, USA, 2003.

[22] T. G. Dietterich. Machine Learning. Nature Encyclopedia of Cognitive Science, London: Macmillan, 2003.

[23] "D-Lib Magazine," <http://dlib.org> (21 April 2005).

[24] M. Doane. Metadata, Search and Meaningful ROI, Global Corporate Circle, DCMI Workshop, Seattle, Washington, USA, 2003.

[25] "DocBook.org," <http://www.docbook.org/> (15 October 2004).

[26] "DTIC Public STINET (Scientific & Technical Information Network)," <http://stinet.dtic.mil/str/index.html> (6 July 2005).

[27] "Dublin Core Metadata Initiative: Making It Easier to Find Information," <http://dublincore.org/> (15 December 2005).

[28] R. Dugad and U. B. Desai. "A Tutorial on Hidden Markov Models," May 1996. <http://uirvli.ai.uiuc.edu/dugad/hmm_tut.html> (25 January 2006).

[29] J. Greenberg, K. Spurgin, and A. Crystal. Final Report for the AMeGA (Automatic Metadata Generation Applications) Project. Retrived on April 2005 from http://www.loc.gov/catdir/bibcontrol/lc_amega_final_report.pdf.

[30] H. Han, C. L. Giles, E. Manavoglu, H. Zha, Z. Zhang, and E. A. Fox. Automatic Document Metadata Extraction Using Support Vector Machine. 2003 Joint Conference on Digital Libraries (JCDL'03), Houston, Texas USA, May 2003.

[31] X. Hao, J. Wang, M. Bieber, P. Ng. A Tool for Classifying Office Documents. ICTAI 1993: 427-434.

[32] S. Hitchcock, D. Bergmark, T. Brody, C. Gutteridge, L. Carr, W. Hall, C. Lagoze, and S. Harnad. "Open citation linking: The way forward," *D-Lib Magazine* [online journal], October 2002, <http://www.dlib.org/dlib/october02/hitchcock/10hitchcock.html> (12 September 2005).

[33] S. Hitchcock, L. Carr, Z. Jiao, D. Bergmark, W. Hall, C. Lagoze, and S. Harnad. Developing services for open eprint archives: globalisation, integration and the impact of links. 5th ACM Conference on Digital Libraries, San Antonio, Texas, USA, June 2000.

[34] C.-W. Hsu and C.-J. Lin. A comparison of methods for multi-class support vector machines , *IEEE Transactions on Neural Networks*, Vol. 13, Pages 415-425, 2002.

[35] C.-W. Hsu, C.-C. Chang, and C.-J. Lin. "A Practical Guide to Support Vector Classification," <http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf> (4 December 2003).

[36] J. Hu, R. Kashi, and G. Wilfong. Document Image Layout Comparison and Classification. In Proc. of the Intl. Conf. on Document Analysis and Recognition (ICDAR), 1999.

[37] T. Hu and R. Ingold. A mixed approach toward efficient logical structure recognition from document images. Electronic Publishing -- Origination, Dissemination and Design, 6(4): 457-468, 1994.

[38] T. Joachims. Making large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning, B. Schölkopf and C. Burges and A. Smola (ed.), MIT-Press, 1999.

[39] T. Joachims, Learning to Classify Text Using Support Vector Machines. Dissertation, Kluwer, 2002.

[40] "Karush-Kuhn-Tucker conditions," *Wikipedia, the free encyclopedia*, 2006 <http://en.wikipedia.org/wiki/Karush-Kuhn-Tucker_conditions>

[41] J. Kim, D. X. Le, and G. R. Thoma. Automated labeling algorithms for biomedical document images. Proc. 7th World Multiconference on Systemics, Cybernetics and Informatics, Vol. V, pages 352-57, Orlando FL, July 2003.

[42] S. Klink and T. Kieninger. Rule-based Document Structure Understanding with a Fuzzy Combination of Layout and Textual Features. IJDAR 4(1): 18-26 (2001).

[43] S. Klink, A. Dengel, and T. Kieninger. Document structure analysis based on layout and textual features. In Proc. of Fourth IAPR International Workshop on Document Analysis Systems, DAS2000, pages 99--111, Rio de Janeiro, Brazil, 2000.

[44] R. Kohavi and F. Provost. Glossary of Terms. Editorial for the Special Issue on Applications of Machine Learning and the Knowledge Discovery Process, Vol. 30, No. 2/3, February/March 1998.

[45] J.T. Kwok. Automated text categorization using support vector machine. In Proceedings of the International Conference on Neural Information Processing, Kitakyushu, Japan, Oct. 1998, pp. 347-351.

[46] C. Lagoze, H. Van de Sompel, M. Nelson, and S. Warner. Implementation Guidelines for the Open Archives Initiative Protocol for Metadata Harvesting. http://www.openarchives.org/OAI/2.0/guidelines.htm

[47] C. Lagoze, H. Van de Sompel, M. Nelson, and S. Warner. "The Open Archives Initiative Protocol for Metadata Harvesting," *Open Archives Initiative,* 12 October 2004, <http://www.openarchives.org/OAI/openarchivesprotocol.html> (8 February 2006).

[48] "Latex – A document preparation system," <http://www.latex-project.org/> (5 January 2006).

[49] D. X. Le and G. R. Thoma, Page Layout Classification Technique for Biomedical Documents. In Proc. World Multiconference on Systems, Cybernetics and Informatics (SCI), X: 348-52, July 2000.

[50] M. Lesk. Practical Digital Libraries: books, bytes, and bucks, Morgan Kaufmann Publishers, California, 1997.

[51] X. Li, Cheng Z, Sheng F, Fan X, and Ng P. A Document Classification and Extraction System with Learning Ability. Proceedings of the Fifth World Conference on Integrated Design and Process Technology, Dallas, Texas, June 2000.

[52] J. Liang. Document Structure Analysis and Performance Evaluation. Ph.D dissertation, University of Washington, 1999.

[53] X. Liu. Federating Heterogeneous Digital Libraries by Metadata Harvesting, PhD dissertation, Old Dominion University, 2002.

[54] K. Maly, M. Zubair, M. Nelson, X. Liu, H. Anan, J. Gao, J. Tang, and Y. Zhao. Archon—a digital library that federates physics collections. In DC-2002: Metadata for e-Communities: Supporting Diversity and Convergence, October 2002.

[55] S. Mao, A. Rosenfeld, and T. Kanungo. Document Structure Analysis Algorithms: A literature Survey. In Proc. SPIE Electronic Imaging, 5010:197-207, 2003.

[56] D. Carlise, P. Ion, R. Miner, and N. Poppelier, "Mathematical Markup Language (MathML) Version 2.0 (Second Edition)," *World Wide Web Consortium,* 21 October 2003, <http://www.w3.org/TR/MathML2/> (16 January 2006).

[57] "Microsoft Office 2003 XML Reference Schemas," <http://rep.oio.dk/Microsoft.com/officeschemas/welcome.htm> (1 February 2006).

[58] G. Nagy and Seth S. Hierarchical representation of optically scanned documents. Proc. Of ICPR, pp. 347-349, 1984

[59] D. Niyogi and S. N. Srihari. The Use of Document Structure Analysis to Retrieve Information from Documents in Digital Libraries. Proceedings of EI '97, SPIE/IS&T Symposium on Electronic Imaging: Science & Technology, San Jose, CA, February, 1997

[60] D. Niyogi and S. N. Srihari. Knowledge-based derivation of document logical structure. In Proceedings of ICDAR '95 (Third International Conference on Document Analysis and Recognition), Montreal, Canada, August 1995.

[61] D. Niyogi and S. Srihari. Using domain knowledge to derive the logical structure of documents. SPIE, pp. 114--125, 1996.

[62] "Prolog," *Wikipedia, the free encyclopedia*, 2006, <http://en.wikipedia.org/wiki/Prolog>

[63] L. R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proceedings of the IEEE, vol. 77, no. 2, pp. 257-286, 1989.

[64] B. Rosenfeld, R. Feldman, and Y. Aumann. Structural extraction from visual layout of documents. In Proceedings of the eleventh international conference on Information and knowledge management 2002, McLean, Virginia, USA, November, 2002.

[65] K. Seymore, A. McCallum, and R. Rosenfeld. Learning hidden Markov model structure for information extraction. In AAAI Workshop on Machine Learning for Information Extraction, 1999.

[66] B. Stehno and G. Retti. Modeling the logical structure of books and journals using augmented transition network grammars. In: Journal of Documentation, Vol. 59 No. 1 p. 69-83, 2003.

[67] K. M. Summers. Automatic Discovery Of Logical Document Structure. Ph.D. Dissertation, Cornell University 1998.

[68] L. Todoran, M. Aiello, C. Monz and M. Worring. Logical Structure Detection for Heterogeneous Document Classes. In 7th Document Recognition and Retrieval (SPIE), San Jose, pp. 99-110, 2001.

[69] "U. S. Government Printing Office", <http://www.gpo.gov> (8 October 2005).

[70] H. Van de Sompel and O. Beit-Arie. "Open linking in the scholarly information environment using the OpenURL framework," *D-Lib Magazine* [online journal], March 2001, <http://www.dlib.org/dlib/march01/vandesompel/03vandesompel.html> (25 January 2005).

[71] V. N. Vapnik. The nature of Statistical Learning Theory. Springer, Berlin, 1995.

[72] N. Warakagoda. "Definition of Hidden Markov Model," 10 May 1996, <http://jedlik.phy.bme.hu/~gerjanos/HMM/node4.html> (23 February 2006).

[73] XML by example, December 1999, QUE. ISBN: 0-7897-2242-9

[74] "XMLCities: content for a new era", <http://www.xmlcities.com/> (11 July 2005).

[75] H. Zuse, Software Complexity: measures and methods. New York, 1991. ISBN 0-89925-640-6.

APPENDIX A

TEMPLATE SCHEMA FOR STRUCTURED METADATA PAGE

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="template" type="OneTemplate" />
<xs:complexType name="OneTemplate">
     <xs:sequence>
        <xs:element name="form" minOccurs="0" maxOccurs="unbounded"
type="OneForm" />
     </xs:sequence>
</xs:complexType>

<xs:complexType name="OneForm">
     <xs:sequence>
        <xs:element name="match" minOccurs="0" maxOccurs="unbounded"
type="StrMatch"/>
        <xs:element name="fixed" minOccurs="0" maxOccurs="unbounded"
type="Fixed" />
        <xs:element name="extracted" minOccurs="0"
maxOccurs="unbounded" type="Extracted" />
        <xs:element name="exclude" minOccurs="0" maxOccurs="unbounded"
type="xs:string" />
    </xs:sequence>
    <xs:attribute name="max" type="xs:int" />
</xs:complexType>

<xs:complexType name="StrMatch">
     <xs:sequence>
        <xs:element name="line" minOccurs="0" maxOccurs="unbounded"
type="xs:string" />
    </xs:sequence>
    <xs:attribute name="max" type="xs:int" />
</xs:complexType>

<xs:complexType name="Fixed">
     <xs:sequence>
            <xs:element name="field" minOccurs="0"
maxOccurs="unbounded" type="Field"/>
     </xs:sequence>
</xs:complexType>

<xs:complexType name="Field">
     <xs:sequence>
        <xs:element name="line" minOccurs="0" maxOccurs="unbounded"
type="xs:string"/>
    </xs:sequence>
    <xs:attribute name="num" type="xs:string" />
    <xs:attribute name="optional" type="xs:string" />
</xs:complexType>
```

APPENDIX A (continued)

```
<xs:complexType name="Extracted">
     <xs:sequence>
        <xs:element name="metadata" minOccurs="0" maxOccurs="unbounded"
type="Metadata"/>
    </xs:sequence>
</xs:complexType>

<xs:complexType name="Metadata">
     <xs:sequence>
        <xs:element name="rule" minOccurs="0" maxOccurs="unbounded"
type="FRelation"/>
        <xs:element name="exclude" minOccurs="0" maxOccurs="unbounded"
type="xs:string" />
    </xs:sequence>
    <xs:attribute name="name" type="xs:string" />
    <xs:attribute name="default" type="xs:string" />
</xs:complexType>

<xs:complexType name="FRelation">
    <xs:attribute name="relation" type="xs:string" />
    <xs:attribute name="field" type="xs:string" />
</xs:complexType>
</xs:schema>
```

APPENDIX B

COVCLASS SCHEMA

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="covclasses" type="CoverClasses" />
<xs:complexType name="CoverClasses">
     <xs:sequence>
        <xs:element name="covclass" minOccurs="0" maxOccurs="unbounded"
type="CovClass" />
     </xs:sequence>
</xs:complexType>
<xs:complexType name="CovClass">
     <xs:sequence>
        <xs:element name="layoutstruct" minOccurs="0"
maxOccurs="unbounded" type="LayoutStruct"/>
        <xs:element name="block" minOccurs="0" maxOccurs="unbounded"
type="Block" />
        <xs:element name="blockrelation" minOccurs="0"
maxOccurs="unbounded" type="BlockRelation" />
    </xs:sequence>
    <xs:attribute name="name" type="xs:string" />
</xs:complexType>
<xs:complexType name="LayoutStruct">
    <xs:attribute name="compare" type="xs:string" />
    <xs:attribute name="type" type="xs:string" />
    <xs:attribute name="min" type="xs:decimal" />
</xs:complexType>
<xs:complexType name="Block">
     <xs:sequence>
            <xs:element name="stringmatch" minOccurs="0"
maxOccurs="unbounded" type="StringMatch"/>
     </xs:sequence>
    <xs:attribute name="name" type="xs:string" />
    <xs:attribute name="align" type="xs:string" />
    <xs:attribute name="xsize" type="xs:string" />
    <xs:attribute name="loc" type="xs:string" />
    <xs:attribute name="allupcase" type="xs:boolean" />
    <xs:attribute name="firstupcase" type="xs:boolean" />
</xs:complexType>
<xs:complexType name="BlockRelation">
    <xs:attribute name="begin" type="xs:string" />
    <xs:attribute name="end" type="xs:string" />
    <xs:attribute name="relation" type="xs:string" />
    <xs:attribute name="adjacent" type="xs:boolean" />
</xs:complexType>
<xs:complexType name="StringMatch">
  <xs:simpleContent>
      <xs:extension base="xs:string">
            <xs:attribute name="case" type="xs:boolean"/>
            <xs:attribute name="loc" type="xs:string"/>
```

APPENDIX B (continued)

```
            <xs:attribute name="distance" type="xs:int"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>
</xs:schema>
```

APPENDIX C

SAMPLES, TEMPLATES, AND METADATA EXTRACTION RESULTS

This appendix includes the samples, templates, and metadata results of individual groups from the experiments in section 6.2.1.1.

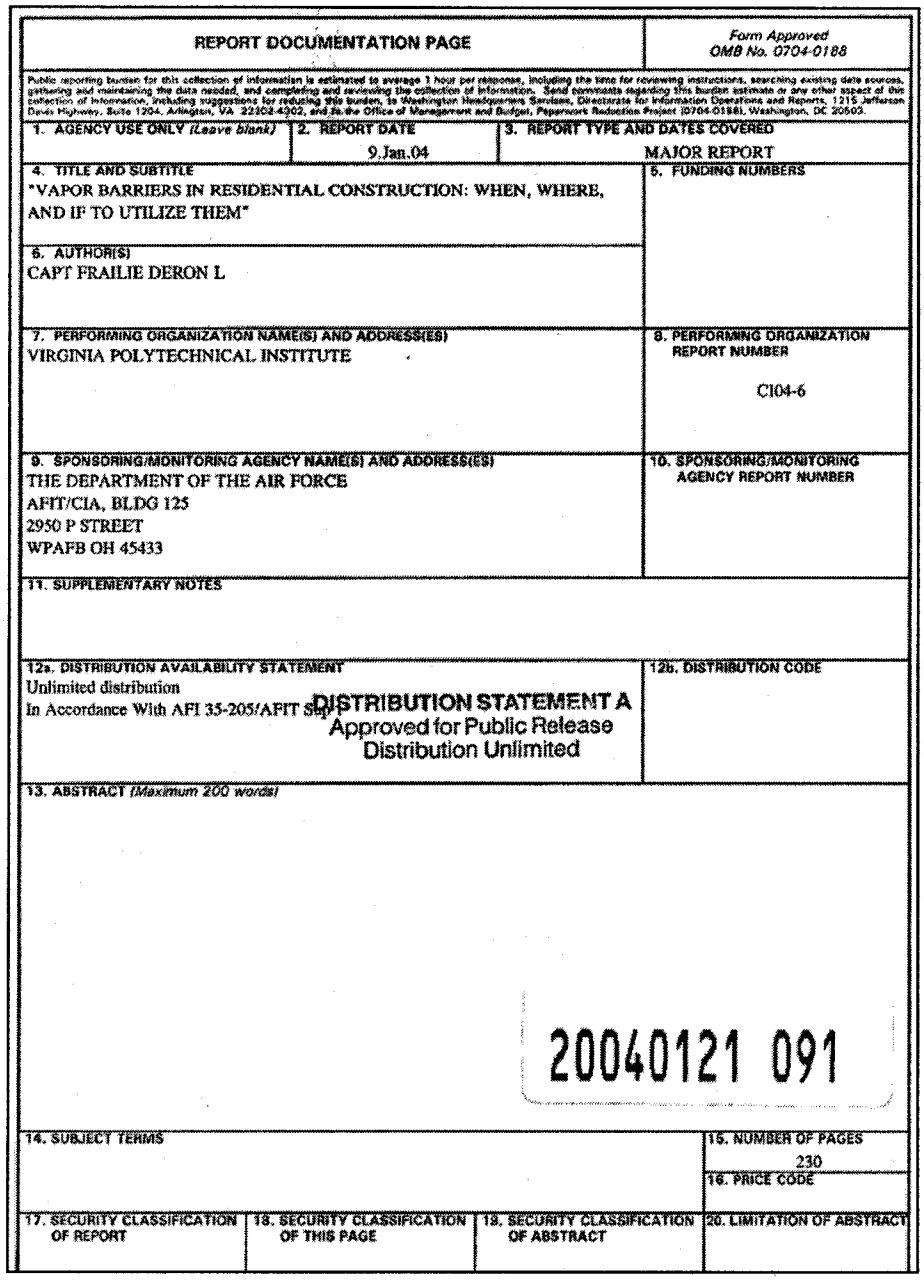
C.1. Data Set

Group	Doc #	List of IDs[2]
sf298_1	3	415238, 419415, 416827
sf298_2	3	415915, 416353, 417145
generic	14	410612, 416050, 410979, 411614, 415826, 415847, 412708, 416321, 416786, 417006, 418118, 418517, 419272, 420017
thesis	3	416562, 416557, 410621
letter	1	411910
issuedby	1	418055
usawc	2	414953, 415399
afrl	5	419305, 417912, 417477, 412971, 412244
arl	5	420016, 417242, 414778, 413912,411840
edgewood	4	417162, 416864, 416809, 415715
nps	15	420437, 420436, 420315, 418556, 418310, 418307, 417634, 417506, 417443, 417333, 417087, 415282, 415013, 415009, 414879
usnce	5	418489, 417681, 417310, 415165, 414926
afit	6	415472, 413433, 413228, 412963, 412907, 412678
text	33	412114, 413622, 414677, 415249, 415510, 415609, 416149, 416657, 416666, 416713, 416719, 416722, 416749, 417014, 417022, 417068, 417125, 417782, 417880, 418018, 418064, 418083, 418657, 418677, 418720, 418864, 418907, 418938, 419141, 419215, 419362, 420073, 420158

[2] An ID is a part of the *"AD Number"* that is unique in the public STINET collection. You can search its corresponding document in the website http://stinet.dtic.mil/ by using this ID with a prefix "ADA", e.g. "ADA420158".

APPENDIX C (continued)

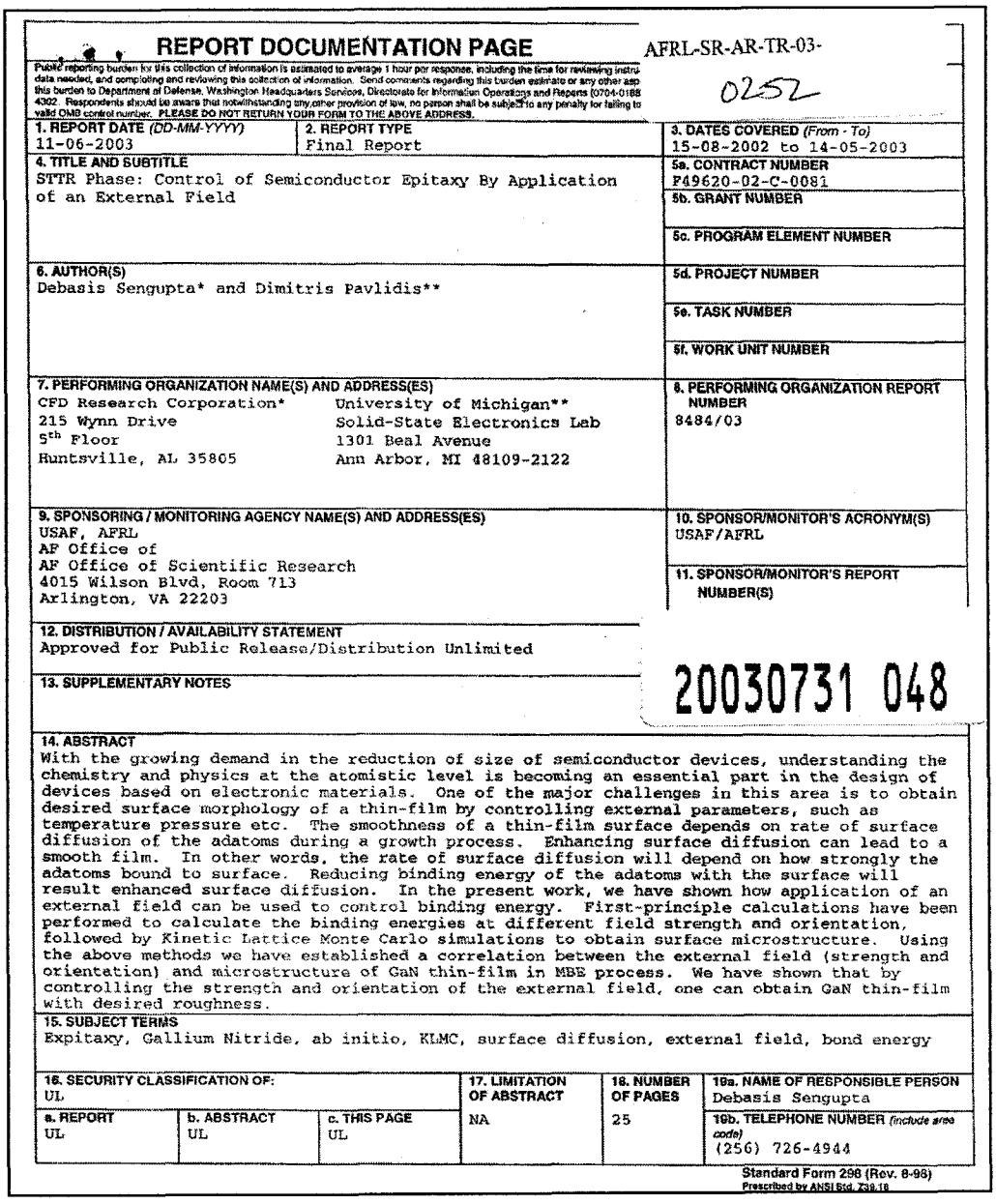
C.2. Metadata Page Samples



REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.			
1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE 9.Jan.04	3. REPORT TYPE AND DATES COVERED MAJOR REPORT	
4. TITLE AND SUBTITLE "VAPOR BARRIERS IN RESIDENTIAL CONSTRUCTION: WHEN, WHERE, AND IF TO UTILIZE THEM"			5. FUNDING NUMBERS
6. AUTHOR(S) CAPT FRAILIE DERON L			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) VIRGINIA POLYTECHNICAL INSTITUTE			8. PERFORMING ORGANIZATION REPORT NUMBER CI04-6
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) THE DEPARTMENT OF THE AIR FORCE AFIT/CIA, BLDG 125 2950 P STREET WPAFB OH 45433			10. SPONSORING/MONITORING AGENCY REPORT NUMBER
11. SUPPLEMENTARY NOTES			
12a. DISTRIBUTION AVAILABILITY STATEMENT Unlimited distribution In Accordance With AFI 35-205/AFIT Sup 1 DISTRIBUTION STATEMENT A Approved for Public Release Distribution Unlimited			12b. DISTRIBUTION CODE
13. ABSTRACT *(Maximum 200 words)* 20040121 091			
14. SUBJECT TERMS			15. NUMBER OF PAGES 230 16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT

Fig. 52. Metadata page sample of the group "sf298_1"

APPENDIX C (continued)



REPORT DOCUMENTATION PAGE

AFRL-SR-AR-TR-03-

0252

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (From - To)
11-06-2003	Final Report	15-08-2002 to 14-05-2003

4. TITLE AND SUBTITLE
STTR Phase: Control of Semiconductor Epitaxy By Application of an External Field

5a. CONTRACT NUMBER
F49620-02-C-0081

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Debasis Sengupta* and Dimitris Pavlidis**

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

CFD Research Corporation*
215 Wynn Drive
5th Floor
Huntsville, AL 35805

University of Michigan**
Solid-State Electronics Lab
1301 Beal Avenue
Ann Arbor, MI 48109-2122

8. PERFORMING ORGANIZATION REPORT NUMBER
8484/03

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
USAF, AFRL
AF Office of
AF Office of Scientific Research
4015 Wilson Blvd, Room 713
Arlington, VA 22203

10. SPONSOR/MONITOR'S ACRONYM(S)
USAF/AFRL

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for Public Release/Distribution Unlimited

13. SUPPLEMENTARY NOTES

20030731 048

14. ABSTRACT
With the growing demand in the reduction of size of semiconductor devices, understanding the chemistry and physics at the atomistic level is becoming an essential part in the design of devices based on electronic materials. One of the major challenges in this area is to obtain desired surface morphology of a thin-film by controlling external parameters, such as temperature pressure etc. The smoothness of a thin-film surface depends on rate of surface diffusion of the adatoms during a growth process. Enhancing surface diffusion can lead to a smooth film. In other words, the rate of surface diffusion will depend on how strongly the adatoms bound to surface. Reducing binding energy of the adatoms with the surface will result enhanced surface diffusion. In the present work, we have shown how application of an external field can be used to control binding energy. First-principle calculations have been performed to calculate the binding energies at different field strength and orientation, followed by Kinetic Lattice Monte Carlo simulations to obtain surface microstructure. Using the above methods we have established a correlation between the external field (strength and orientation) and microstructure of GaN thin-film in MBE process. We have shown that by controlling the strength and orientation of the external field, one can obtain GaN thin-film with desired roughness.

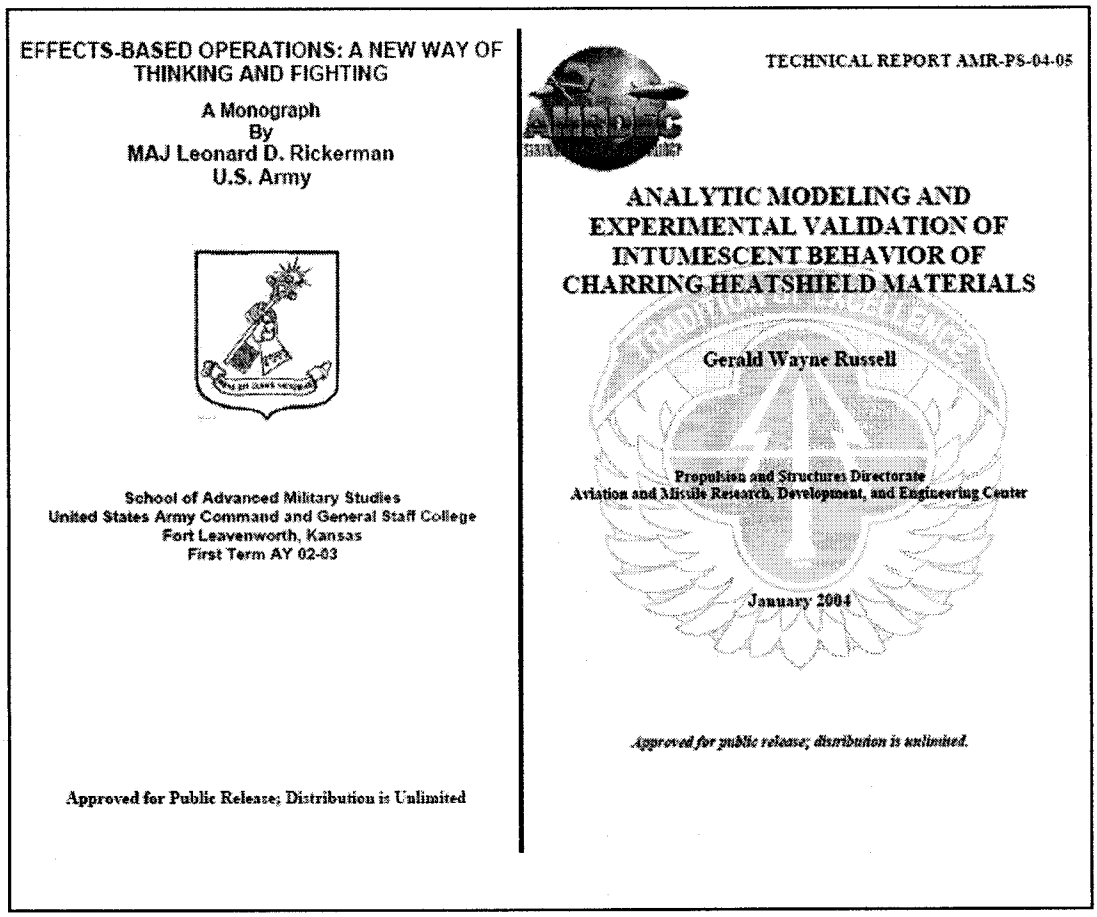
15. SUBJECT TERMS
Expitaxy, Gallium Nitride, ab initio, KLMC, surface diffusion, external field, bond energy

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Debasis Sengupta
UL	UL	UL	NA	25	19b. TELEPHONE NUMBER (include area code) (256) 726-4944

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

Fig. 53. Metadata page sample of group "sf298_2"

APPENDIX C (continued)



EFFECTS-BASED OPERATIONS: A NEW WAY OF THINKING AND FIGHTING

A Monograph
By
MAJ Leonard D. Rickerman
U.S. Army

School of Advanced Military Studies
United States Army Command and General Staff College
Fort Leavenworth, Kansas
First Term AY 02-03

Approved for Public Release; Distribution is Unlimited

TECHNICAL REPORT AMR-PS-04-05

ANALYTIC MODELING AND EXPERIMENTAL VALIDATION OF INTUMESCENT BEHAVIOR OF CHARRING HEATSHIELD MATERIALS

Gerald Wayne Russell

Propulsion and Structures Directorate
Aviation and Missile Research, Development, and Engineering Center

January 2004

Approved for public release; distribution is unlimited.

Fig. 54. Metadata page sample of group "generic"

APPENDIX C (continued)

THE OBJECTIVES OF UNITED STATES MILITARY INTERVENTION IN NORTHERN IRAQ BETWEEN OPERATION DESERT STORM AND OPERATION IRAQI FREEDOM

A thesis presented to the Faculty of the U.S. Army Command and General Staff College in partial fulfillment of the requirements for the degree

MASTER OF MILITARY ART AND SCIENCE
General Studies

by

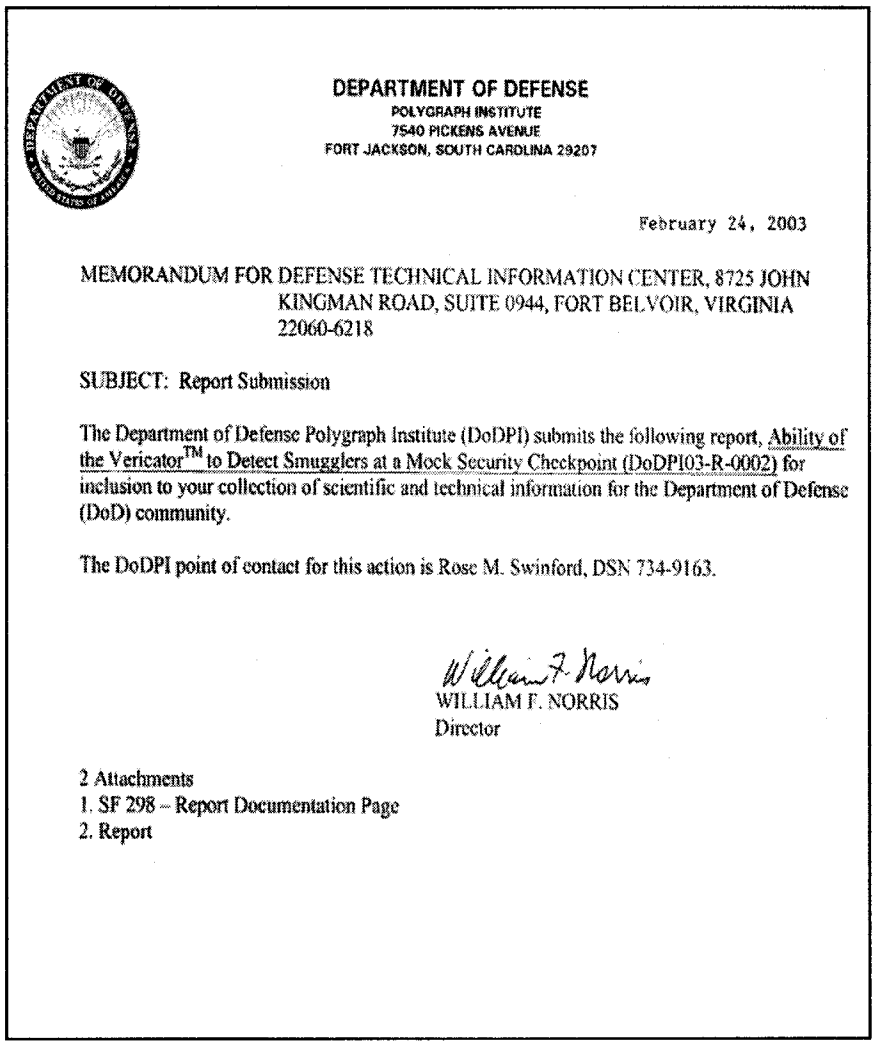
MICHAEL A. SCHIESL, MAJ, USA
B.S., Missouri Western State College, Saint Joseph, Missouri, 1991

Fort Leavenworth, Kansas
2003

Approved for public release; distribution is unlimited.

Fig. 55. Metadata page sample of the group "thesis"

APPENDIX C (continued)



DEPARTMENT OF DEFENSE
POLYGRAPH INSTITUTE
7540 PICKENS AVENUE
FORT JACKSON, SOUTH CAROLINA 29207

February 24, 2003

MEMORANDUM FOR DEFENSE TECHNICAL INFORMATION CENTER, 8725 JOHN KINGMAN ROAD, SUITE 0944, FORT BELVOIR, VIRGINIA 22060-6218

SUBJECT: Report Submission

The Department of Defense Polygraph Institute (DoDPI) submits the following report, Ability of the Vericator™ to Detect Smugglers at a Mock Security Checkpoint (DoDPI03-R-0002) for inclusion to your collection of scientific and technical information for the Department of Defense (DoD) community.

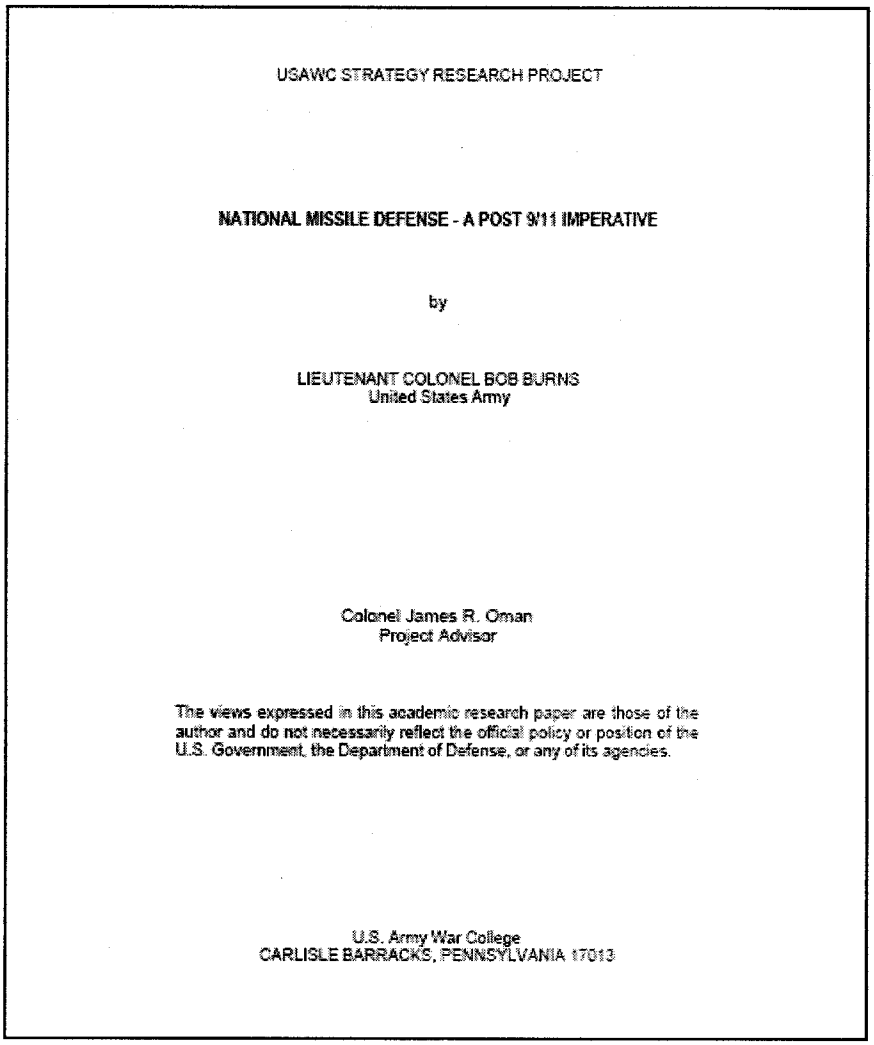
The DoDPI point of contact for this action is Rose M. Swinford, DSN 734-9163.

William F. Norris
WILLIAM F. NORRIS
Director

2 Attachments
1. SF 298 – Report Documentation Page
2. Report

Fig. 56. Metadata page sample of the group "letter"

APPENDIX C (continued)



USAWC STRATEGY RESEARCH PROJECT

NATIONAL MISSILE DEFENSE - A POST 9/11 IMPERATIVE

by

LIEUTENANT COLONEL BOB BURNS
United States Army

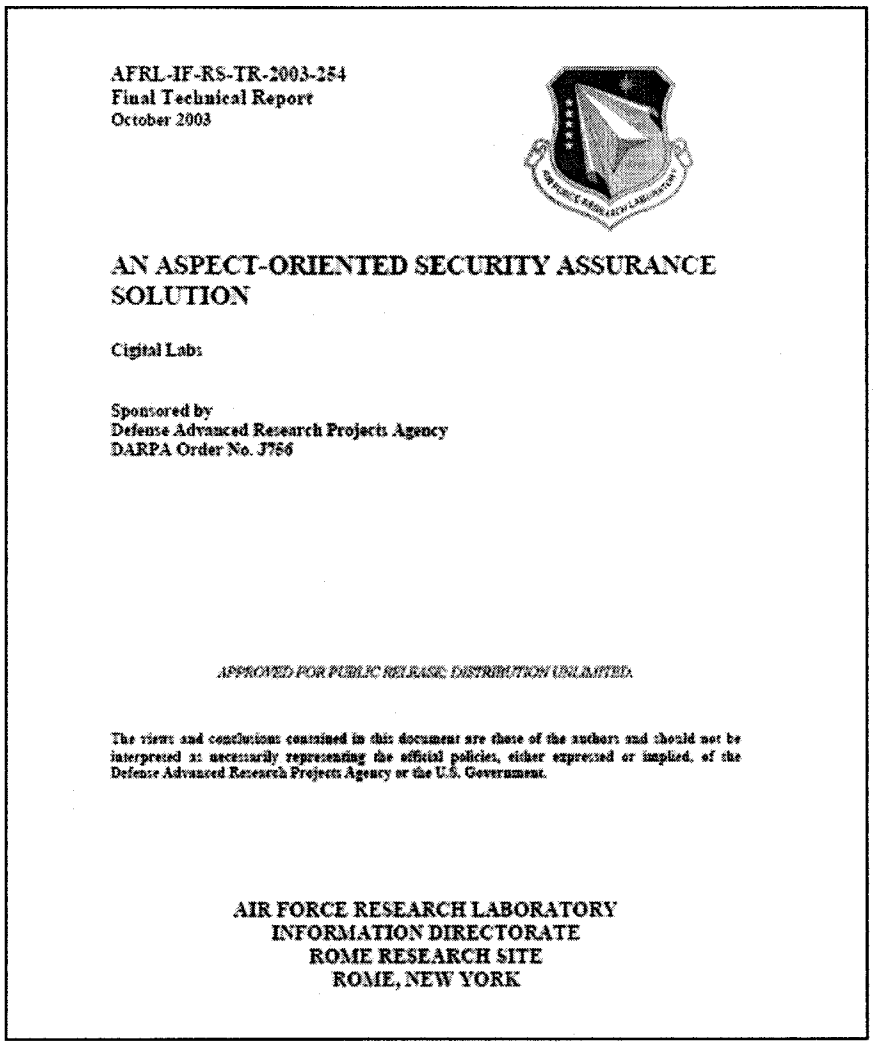
Colonel James R. Oman
Project Advisor

The views expressed in this academic research paper are those of the author and do not necessarily reflect the official policy or position of the U.S. Government, the Department of Defense, or any of its agencies.

U.S. Army War College
CARLISLE BARRACKS, PENNSYLVANIA 17013

Fig. 57. Metadata page sample of the group "usawc"

APPENDIX C (continued)



AFRL-IF-RS-TR-2003-254
Final Technical Report
October 2003

AN ASPECT-ORIENTED SECURITY ASSURANCE SOLUTION

Cigital Labs

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. J756

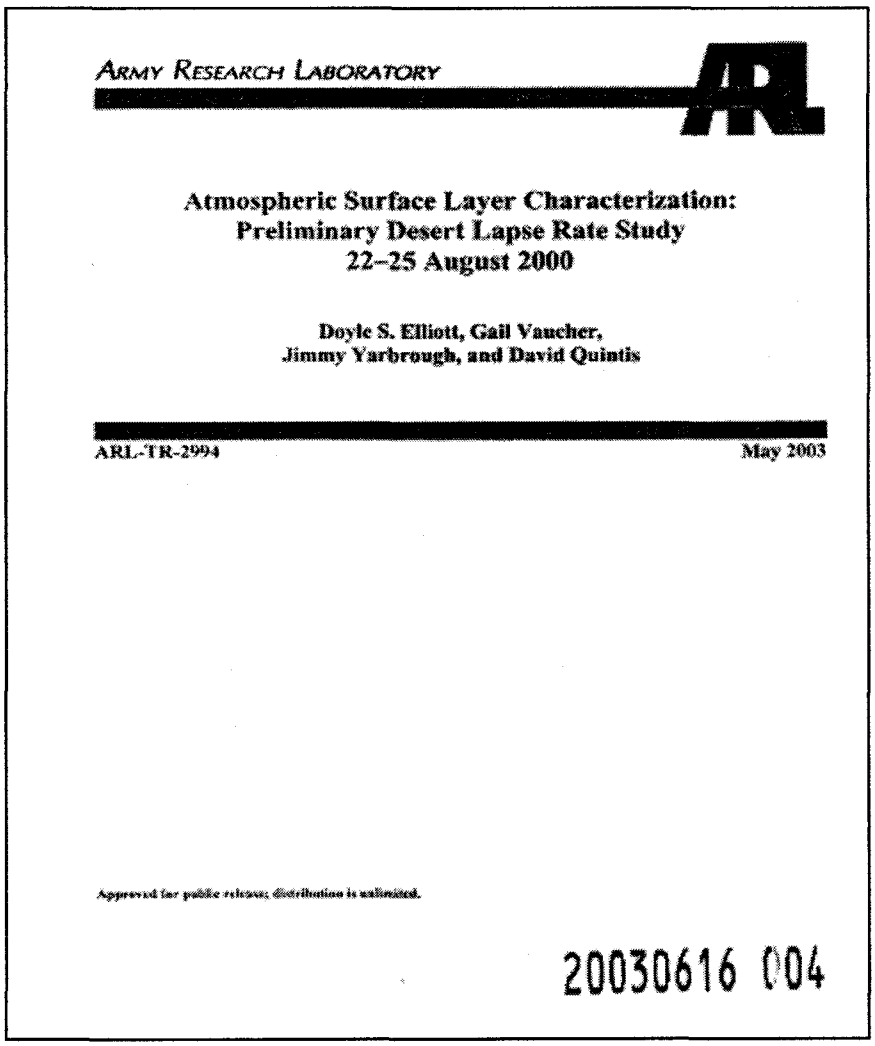
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

Fig. 58. Metadata page sample of the group "afrl"

APPENDIX C (continued)



ARMY RESEARCH LABORATORY

**Atmospheric Surface Layer Characterization:
Preliminary Desert Lapse Rate Study
22–25 August 2000**

**Doyle S. Elliott, Gail Vaucher,
Jimmy Yarbrough, and David Quintis**

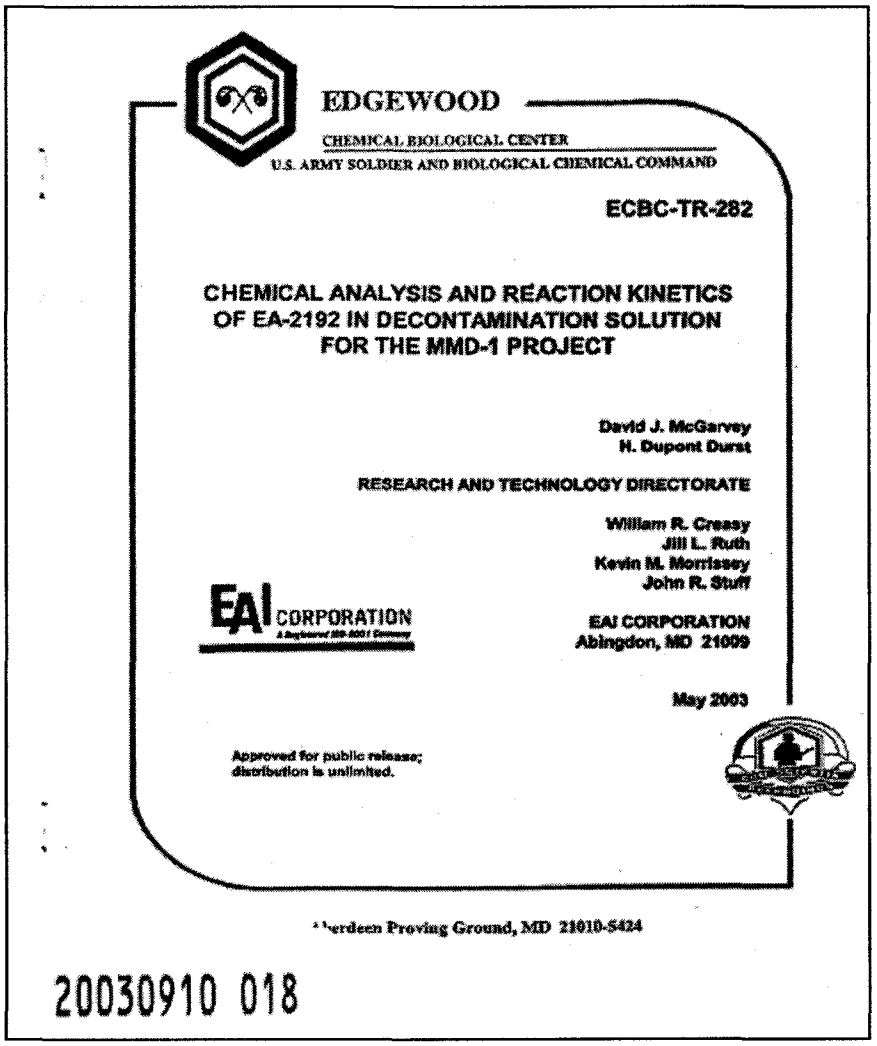
ARL-TR-2994 May 2003

Approved for public release; distribution is unlimited.

20030616 004

Fig. 59. Metadata page sample of the group "arl"

APPENDIX C (continued)



EDGEWOOD

CHEMICAL BIOLOGICAL CENTER

U.S. ARMY SOLDIER AND BIOLOGICAL CHEMICAL COMMAND

ECBC-TR-282

CHEMICAL ANALYSIS AND REACTION KINETICS OF EA-2192 IN DECONTAMINATION SOLUTION FOR THE MMD-1 PROJECT

David J. McGarvey
H. Dupont Durst

RESEARCH AND TECHNOLOGY DIRECTORATE

William R. Creasy
Jill L. Ruth
Kevin M. Morrissey
John R. Stuff

EAI CORPORATION

EAI CORPORATION
Abingdon, MD 21009

May 2003

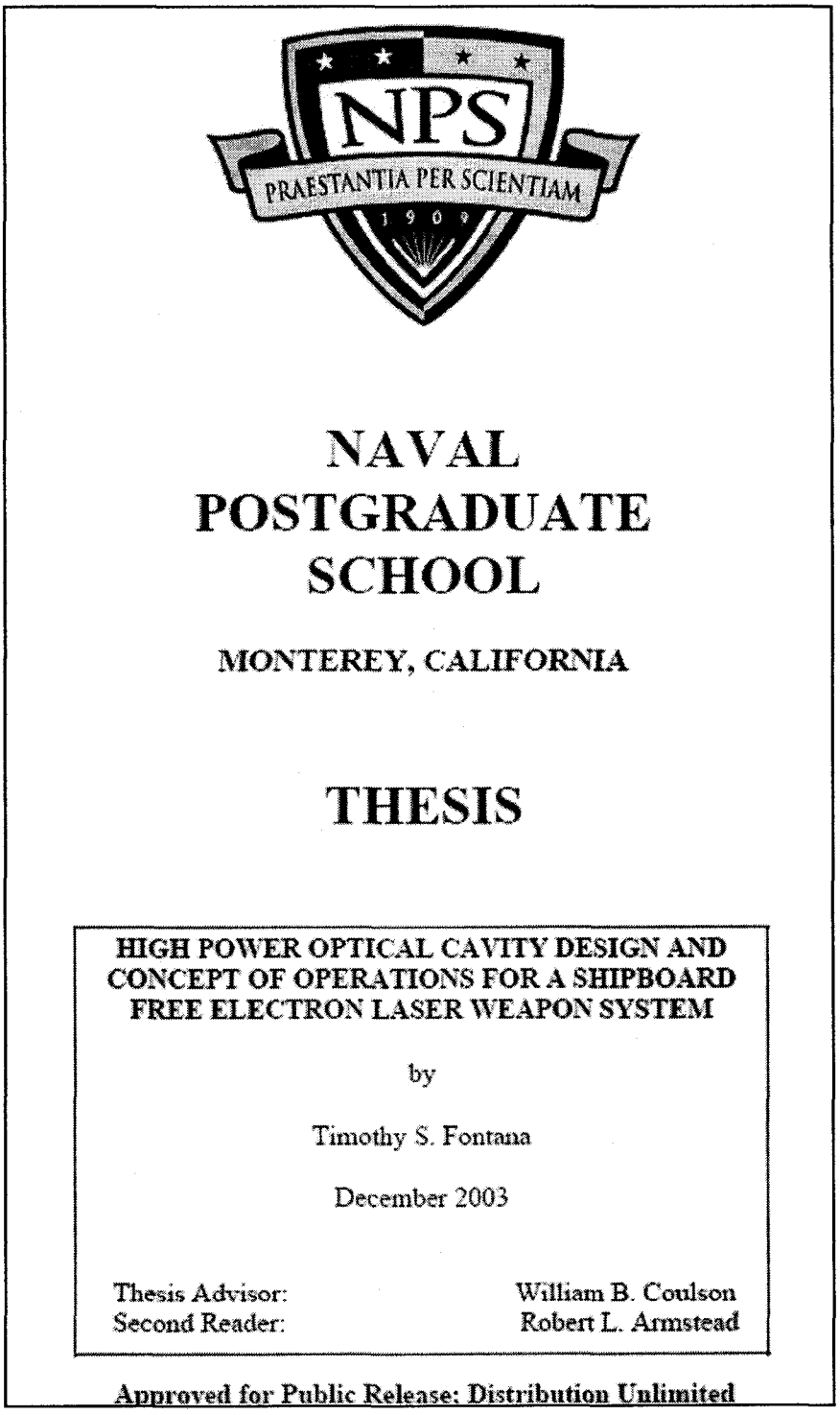
Approved for public release; distribution is unlimited.

erdeen Proving Ground, MD 21010-5424

20030910 018

Fig. 60. Metadata page sample of the group "edgewood"

APPENDIX C (continued)



NAVAL
POSTGRADUATE
SCHOOL

MONTEREY, CALIFORNIA

THESIS

HIGH POWER OPTICAL CAVITY DESIGN AND CONCEPT OF OPERATIONS FOR A SHIPBOARD FREE ELECTRON LASER WEAPON SYSTEM

by

Timothy S. Fontana

December 2003

Thesis Advisor: William B. Coulson
Second Reader: Robert L. Armstead

Approved for Public Release: Distribution Unlimited

Fig. 61. Metadata page sample of the group "nps"

APPENDIX C (continued)

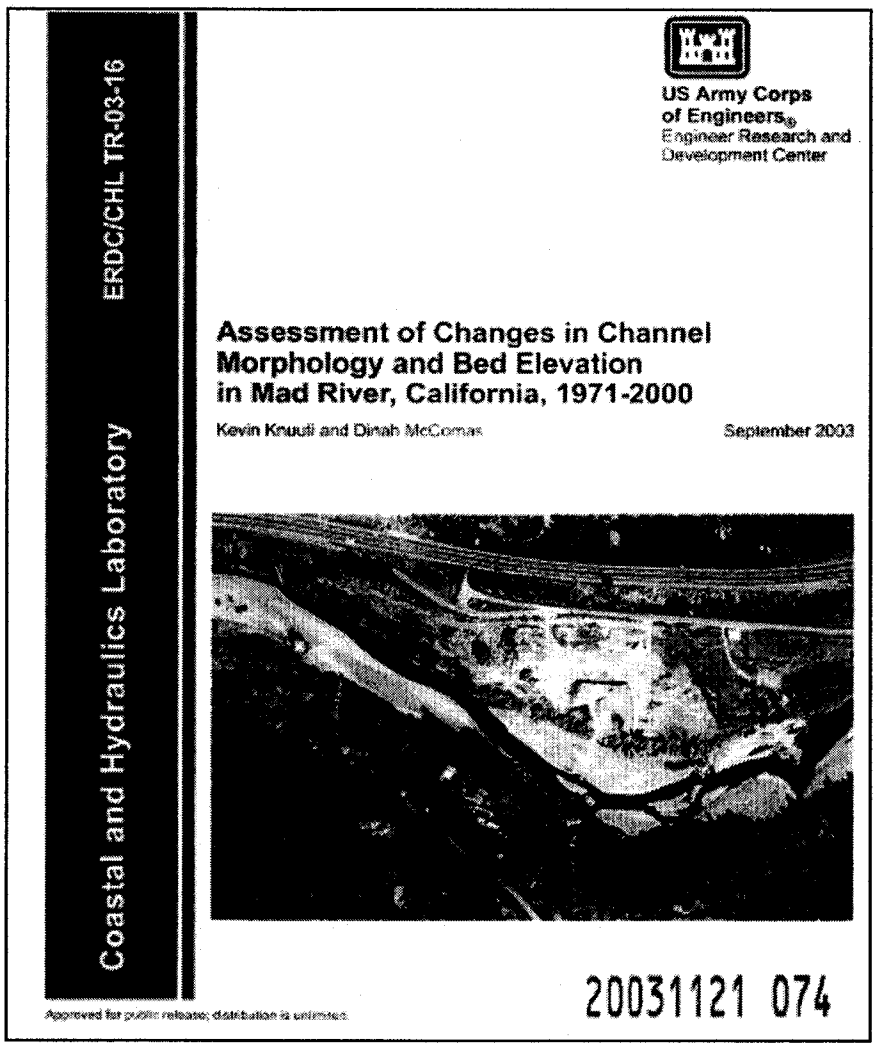
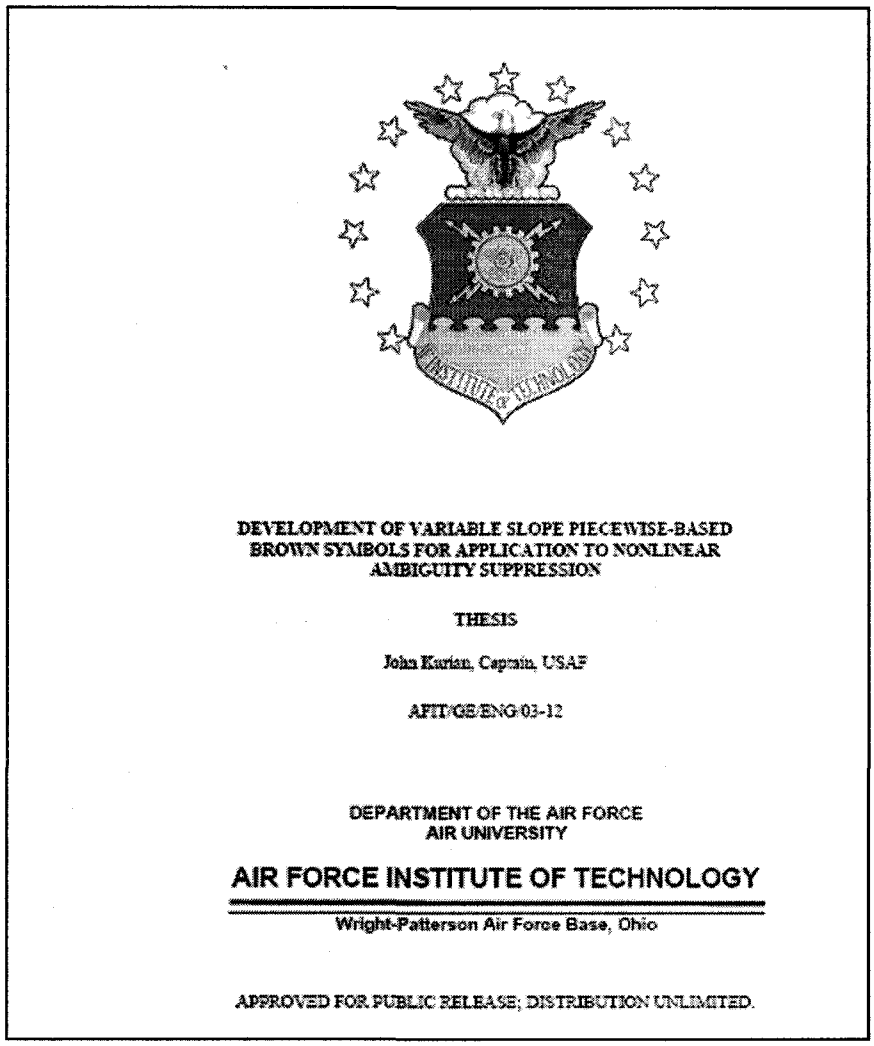


Fig. 62. Metadata page sample of the group "usnce"

APPENDIX C (continued)



DEVELOPMENT OF VARIABLE SLOPE PIECEWISE-BASED BROWN SYMBOLS FOR APPLICATION TO NONLINEAR AMBIGUITY SUPPRESSION

THESIS

John Kurian, Captain, USAF

AFIT/GE/ENG/03-12

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

Fig. 63. Metadata page sample of the group "afit"

APPENDIX C (continued)

AD_____

Award Number: DAMD17-99-1-9501

TITLE: Chronic Stress and Neuronal Pathology: Neurochemical, Molecular and Genetic Factors

PRINCIPAL INVESTIGATOR: George F. Koob, Ph.D.
Pietro P. Sanna, M.D.
Amanda Roberts, Ph.D.

CONTRACTING ORGANIZATION: The Scripps Research Institute
La Jolla, California 92037

REPORT DATE: January 2003

TYPE OF REPORT: Final

PREPARED FOR: U.S. Army Medical Research and Materiel Command
Fort Detrick, Maryland 21702-5012

DISTRIBUTION STATEMENT: Approved for Public Release;
Distribution Unlimited

The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision unless so designated by other documentation.

20030328 269

Fig. 64. Metadata page sample of the group "text"

APPENDIX C (continued)

C.3. Templates

Template for the group "sf298_1"

```
<template>
<form max="-1">
<match max="5">
      <line>REPORT DOCUMENTATION PAGE</line>
</match>
<fixed>
      <field num="1"><line>1. AGENCY USE ONLY (Leave
blank)</line></field>
      <field num="1"><line>1. AGENCY USE ONLY</line></field>
      <field num="2"><line>2. REPORT DATE</line></field>
      <field num="3"><line>3. REPORT TYPE AND DATES
COVERED</line></field>
      <field num="4"><line>4. TITLE AND SUBTITLE</line></field>
      <field num="5"><line>5. FUNDING NUMBERS</line></field>
      <field num="6"><line>6. AUTHOR(S)</line></field>
      <field num="7"><line>7. PERFORMING ORGANIZATION NAME(S) AND
ADDRESS(ES)</line></field>
      <field num="8"><line>8. PERFORMING ORGANIZATION REPORT
NUMBER</line></field>
      <field num="9"><line>9. SPONSORING / MONITORING AGENCY NAME(S)
AND ADDRESS(ES)</line></field>
      <field num="10"><line>10. SPONSORING / MONITORING AGENCY REPORT
NUMBER</line></field>
      <field num="11"><line>11. SUPPLEMENTARY NOTES</line></field>
      <field num="12a"><line>12a. DISTRIBUTION / AVALILABILITY
STATEMENT</line></field>
      <field num="12b"><line>12b. DISTRIBUTION CODE</line></field>
      <field num="13"><line>13. ABSTRACT (Maximum 200
Words)</line></field>
      <field num="13"><line>ABSTRACT (Maximum 200 Words)</line></field>
      <field num="13"><line>13. ABSTRACT</line></field>
      <field num="14"><line>14. SUBJECT TERMS</line></field>
      <field num="15"><line>15. NUMBER OF PAGES</line></field>
      <field num="16"><line>16. PRICE CODE</line></field>
      <field num="17"><line>17. SECURITY CLASSIFICATION OF
REPORT</line></field>
      <field num="18"><line>18. SECURITY CLASSIFICATION OF THIS
PAGE</line></field>
      <field num="19"><line>19. SECURITY CLASSIFICATION OF
ABSTRACT</line></field>
      <field num="20"><line>20. LIMITATION OF ABSTRACT</line></field>
</fixed>
<extracted>
      <metadata name="date">
            <rule relation="belowof" field="2"/>
            <rule relation="rightof" field="1"/>
            <rule relation="leftof" field="3"/>
```

APPENDIX C (continued)

```
      <rule relation="aboveof" field="4|5"/>
</metadata>
<metadata name="typecoverage">
      <rule relation="belowof" field="3"/>
      <rule relation="rightof" field="2"/>
      <rule relation="aboveof" field="4|5"/>
</metadata>
<metadata name="title">
      <rule relation="belowof" field="4"/>
      <rule relation="leftof" field="5"/>
      <rule relation="aboveof" field="6"/>
</metadata>
<metadata name="funding_number">
      <rule relation="belowof" field="5"/>
      <rule relation="aboveof" field="7|8"/>
      <rule relation="rightof" field="4|6|7"/>
</metadata>
<metadata name="creator">
      <rule relation="belowof" field="6"/>
      <rule relation="aboveof" field="7|8"/>
      <rule relation="leftof" field="5|8"/>
</metadata>

<metadata name="performing_org">
      <rule relation="belowof" field="7"/>
      <rule relation="aboveof" field="9|10"/>
      <rule relation="leftof" field="8|10"/>
</metadata>
<metadata name="performing_number">
      <rule relation="belowof" field="8"/>
      <rule relation="aboveof" field="9|10"/>
      <rule relation="rightof" field="7|9|6"/>
</metadata>
<metadata name="sponsor">
      <rule relation="belowof" field="9"/>
      <rule relation="aboveof" field="11"/>
      <rule relation="leftof" field="10|8|5"/>
</metadata>
<metadata name="sponsor_num">
      <rule relation="belowof" field="10"/>
      <rule relation="aboveof" field="11"/>
      <rule relation="rightof" field="9|7"/>
</metadata>
<metadata name="notes">
      <rule relation="belowof" field="11"/>
      <rule relation="aboveof" field="12a|12b"/>
</metadata>
<metadata name="dist_statement">
      <rule relation="belowof" field="12a"/>
      <rule relation="aboveof" field="13"/>
      <rule relation="leftof" field="12b"/>
</metadata>
```

APPENDIX C (continued)

```
        <metadata name="dist_code">
              <rule relation="belowof" field="12b"/>
              <rule relation="aboveof" field="13"/>
              <rule relation="rightof" field="12a"/>
        </metadata>
        <metadata name="abstract">
              <rule relation="belowof" field="13"/>
              <rule relation="aboveof" field="14|15"/>
        </metadata>
        <metadata name="subject">
              <rule relation="belowof" field="14"/>
              <rule relation="aboveof" field="17|18|19|20"/>
              <rule relation="leftof" field="15|16"/>
        </metadata>
        <metadata name="no_of_page">
              <rule relation="belowof" field="15"/>
              <rule relation="aboveof" field="16"/>
              <rule relation="rightof" field="14|19"/>
        </metadata>
        <metadata name="price_code">
              <rule relation="belowof" field="16"/>
              <rule relation="aboveof" field="20"/>
              <rule relation="rightof" field="14|19"/>
        </metadata>
        <metadata name="sec_report">
              <rule relation="belowof" field="17"/>
              <rule relation="leftof" field="18"/>
        </metadata>
        <metadata name="sec_page">
              <rule relation="belowof" field="18"/>
              <rule relation="leftof" field="19"/>
              <rule relation="rightof" field="17"/>
        </metadata>
        <metadata name="sec_abstract">
              <rule relation="belowof" field="19"/>
              <rule relation="leftof" field="20"/>
              <rule relation="rightof" field="18"/>
        </metadata>
        <metadata name="lim_abstract">
              <rule relation="belowof" field="20"/>
              <rule relation="rightof" field="19"/>
        </metadata>
  </extracted>
  <exclude>^NSN\s([\d-])*</exclude>
  <exclude>\QStandard Form 298\E.*</exclude>
  <exclude>\QPrescribed by ANSI\E.*</exclude>
  <exclude>\Q298-102\E.*</exclude>
  </form>
  </template>
```

APPENDIX C (continued)

Template for the group "sf298_2"

```
<template>
<form max="-1">
<match max="5">
      <line>Report Documentation Page</line>
</match>
<fixed>
      <field num="1"><line>1. REPORT DATE (DD-MM-YYYY)</line></field>
      <field num="1"><line>1. REPORT DATE</line></field>
      <field num="2"><line>2. REPORT TYPE</line></field>
      <field num="3"><line>3. DATES COVERED (FROM - TO)</line></field>
      <field num="3"><line>3. DATES COVERED</line></field>
      <field num="4"><line>4. TITLE AND SUBTITLE</line></field>
      <field num="5a"><line>5a. CONTRACT NUMBER</line></field>
      <field num="5b"><line>5b. GRANT NUMBERS</line></field>
      <field num="5c"><line>5c. PROGRAM ELEMENT NUMBER</line></field>
      <field num="5d"><line>5d. PROJECT NUMBER</line></field>
      <field num="5e"><line>5e. TASK NUMBER</line></field>
      <field num="5f"><line>5f. WORK UNIT NUMBER</line></field>
      <field num="6"><line>6. AUTHOR(S)</line></field>
      <field num="7"><line>7. PERFORMING ORGANIZATION NAME(S) AND
ADDRESS(ES)</line></field>
      <field num="8"><line>8. PERFORMING ORGANIZATION REPORT
NUMBER</line></field>
      <field num="9"><line>9. SPONSORING/MONITORING AGENCY NAME(S) AND
ADDRESS(ES)</line></field>
      <field num="10"><line>10. SPONSOR/MONITOR'S
ACRONYM(S)</line></field>
      <field num="11"><line>11. SPONSOR/MONITOR'S REPORT
NUMBER(S)</line></field>
      <field num="12"><line>12. DISTRIBUTION/AVALILABILITY
STATEMENT</line></field>
      <field num="13"><line>13. SUPPLEMENTARY NOTES</line></field>
      <field num="14"><line>14. ABSTRACT"</line></field>
      <field num="15"><line>15. SUBJECT TERMS</line></field>
      <field num="16"><line>16. SECURITY CLASSIFICATION
OF:</line></field>
      <field num="16->a"><line>a. REPORT</line></field>
      <field num="16->b"><line>b. ABSTRACT</line></field>
      <field num="16->c"><line>c. THIS PAGE</line></field>
      <field num="17"><line>17. LIMITATION OF ABSTRACT</line></field>
      <field num="18"><line>18. NUMBER OF PAGES</line></field>
      <field num="19a"><line>19a. NAME OF RESPONSIBLE
PERSON</line></field>
      <field num="19b"><line>19b. TELEPHONE NUMBER (include area
code)</line></field>
</fixed>
<extracted>
      <metadata name="date">
            <rule relation="belowof" field="1"/>
```

APPENDIX C (continued)

```
      <rule relation="leftof" field="2"/>
      <rule relation="aboveof" field="4|5a"/>
</metadata>
<metadata name="reporttype">
      <rule relation="belowof" field="2"/>
      <rule relation="rightof" field="1"/>
      <rule relation="leftof" field="3"/>
      <rule relation="aboveof" field="4|5a"/>
</metadata>
<metadata name="datecoverage">
      <rule relation="belowof" field="3"/>
      <rule relation="rightof" field="2"/>
      <rule relation="aboveof" field="4|5a"/>
</metadata>
<metadata name="title">
      <rule relation="belowof" field="4"/>
      <rule relation="leftof" field="5a|5b|5c|3|5d|5e"/>
      <rule relation="aboveof" field="6|5d"/>
</metadata>
<metadata name="contract_number">
      <rule relation="belowof" field="5a"/>
      <rule relation="aboveof" field="5b"/>
      <rule relation="rightof" field="4|6|7"/>
</metadata>
<metadata name="grant_number">
      <rule relation="belowof" field="5b"/>
      <rule relation="aboveof" field="5c"/>
      <rule relation="rightof" field="4|6|7"/>
</metadata>
<metadata name="program_number">
      <rule relation="belowof" field="5c"/>
      <rule relation="aboveof" field="5d|6"/>
      <rule relation="rightof" field="4|6|7"/>
</metadata>
<metadata name="creator">
      <rule relation="belowof" field="6"/>
      <rule relation="aboveof" field="7|8"/>
      <rule relation="leftof" field="5d|5e|5f|8|5c"/>
</metadata>
<metadata name="project_number">
      <rule relation="belowof" field="5d"/>
      <rule relation="aboveof" field="5e"/>
      <rule relation="rightof" field="4|6|7"/>
</metadata>
<metadata name="task_number">
      <rule relation="belowof" field="5e"/>
      <rule relation="aboveof" field="5f"/>
      <rule relation="rightof" field="4|6|7"/>
</metadata>
<metadata name="work_unit_number">
      <rule relation="belowof" field="5f"/>
      <rule relation="aboveof" field="8|7"/>
```

APPENDIX C (continued)

```
      <rule relation="rightof" field="4|6|7"/>
</metadata>

<metadata name="performing_org">
      <rule relation="belowof" field="7"/>
      <rule relation="aboveof" field="9|10"/>
      <rule relation="leftof" field="8|10|5f"/>
</metadata>
<metadata name="report_number">
      <rule relation="belowof" field="8"/>
      <rule relation="aboveof" field="9|10"/>
      <rule relation="rightof" field="7|9|6"/>
</metadata>
<metadata name="sponsor">
      <rule relation="belowof" field="9"/>
      <rule relation="aboveof" field="12"/>
      <rule relation="leftof" field="10|11|8|5f"/>
</metadata>
<metadata name="sponsor_acronym">
      <rule relation="belowof" field="10"/>
      <rule relation="aboveof" field="11"/>
      <rule relation="rightof" field="9|7|4"/>
</metadata>
<metadata name="sponsor_report_number">
      <rule relation="belowof" field="11"/>
      <rule relation="aboveof" field="12"/>
      <rule relation="rightof" field="9|7|4"/>
</metadata>
<metadata name="dist_statement">
      <rule relation="belowof" field="12"/>
      <rule relation="aboveof" field="13"/>
</metadata>
<metadata name="notes">
      <rule relation="belowof" field="13"/>
      <rule relation="aboveof" field="14"/>
</metadata>
<metadata name="abstract">
      <rule relation="belowof" field="14"/>
      <rule relation="aboveof" field="15"/>
</metadata>
<metadata name="subject">
      <rule relation="belowof" field="15"/>
      <rule relation="aboveof" field="16|17|18|19a"/>
</metadata>
<metadata name="no_of_page">
      <rule relation="belowof" field="18"/>
      <rule relation="rightof" field="17"/>
      <rule relation="leftof" field="19a"/>
</metadata>
<metadata name="responsible_person">
      <rule relation="belowof" field="19a"/>
      <rule relation="rightof" field="18"/>
```

APPENDIX C (continued)

```
            <rule relation="aboveof" field="19b"/>
      </metadata>
      <metadata name="responsible_phone">
            <rule relation="belowof" field="19b"/>
            <rule relation="rightof" field="18"/>
      </metadata>
      <metadata name="sec_report">
            <rule relation="belowof" field="16->a"/>
            <rule relation="leftof" field="16->b"/>
      </metadata>
      <metadata name="sec_page">
            <rule relation="belowof" field="16->c"/>
            <rule relation="leftof" field="17"/>
            <rule relation="rightof" field="16->b"/>
      </metadata>
      <metadata name="sec_abstract">
            <rule relation="belowof" field="16->b"/>
            <rule relation="leftof" field="16->c"/>
            <rule relation="rightof" field="16->a"/>
      </metadata>
      <metadata name="lim_abstract">
            <rule relation="belowof" field="17"/>
            <rule relation="rightof" field="16->c"/>
            <rule relation="leftof" field="18"/>
      </metadata>
</extracted>
<exclude>\QStandard Form 298\E.*</exclude>
<exclude>\QPrescribed by ANSI\E.*</exclude>
</form>
</template>
```

Template for the group "generic"

```
<?xml version="1.0" ?>
<structdef>
      <meta name="title" min="1" max="1">
            <begin inclusive="current">largeststrsize(0,0.5)</begin>
            <end inclusive="before">layoutchange</end>
      </meta>
      <meta name="creator" min="0" max="1">
            <begin inclusive="current">nameformat</begin>
            <end inclusive="before">!nameformat</end>
      </ meta>

      < meta name="date" min="0" max="1">
            <begin inclusive="current"  scope="global">
                  dateformat
            </begin>
            <end>onesection</end>
```

APPENDIX C (continued)

```
      </meta>
      < meta name="rights" min="0" max="1">
            <begin inclusive="current" scope="global">
                  <stringmatch case="no" loc="beginwith">
                        Approved for
                  </stringmatch>
            </begin>
            <end inclusive="before">featurechange</end>
      </meta>
</structdef>
```

Template for the group "thesis"

```
<?xml version="1.0" ?>
<structdef>

      <meta name="title" min="1" max="1">
            <begin inclusive="current">largeststrsize(0,0.3)</begin>
            <end inclusive="before">featurechange</end>
      </meta>

      <meta name="creator" min="0" max="1">
            <begin inclusive="after" scope="global">
                  <stringmatch case="no" loc="onesection">
                        By
                  </stringmatch>
            </begin>
            <end>onesection</end>
      </meta>
      <meta name="thesis">
            <begin inclusive="current" scope="global">
                  <stringmatch case="no" loc="beginwith">
                        A thesis
                  </stringmatch>
            </begin>
            <end inclusive="before">
                  <stringmatch case="no" loc="beginwith">
                        Master
                  </stringmatch>
            </end>
      </meta>

      <meta name="degree">
            <begin inclusive="current" scope="global">
                  <stringmatch case="no" loc="beginwith">
                        Master
                  </stringmatch>
            </begin>
            <end inclusive="current">onesection</end>

      </meta>
```

APPENDIX C (continued)

```
      <meta name="program">
            <begin inclusive="after">degree</begin>
            <end inclusive="current">onesection</end>

      </meta>

      <meta name="date" min="0" max="1">
            <begin inclusive="current"
scope="global">dateformat</begin>
            <end inclusive="current">onesection</end>
      </meta>

      <meta name="rights" min="0" max="1">
            <begin inclusive="current" scope="global">
                  <stringmatch case="no" loc="beginwith">
                        Approved for
                  </stringmatch>
            </begin>
            <end inclusive="before">featurechange</end>
      </meta>

</structdef>
```

Template for the group "letter"

```
<?xml version="1.0" ?>
<structdef>

      <meta name="contributor" min="1" max="1">
            <begin inclusive="current">begin</begin>
            <end inclusive="before">dateformat</end>
      </meta>

      <meta name="date" min="0" max="1">
            <begin inclusive="current">dateformat</begin>
            <end inclusive="current">onesection</end>
      </meta>

      <meta name="title" min="0" max="1">
            <begin inclusive="after">date
            </begin>
            <end inclusive="before">
                  <stringmatch case="no" loc="beginwith">
                        SUBJECT
                  </stringmatch>
            </end>
      </meta>
```

APPENDIX C (continued)

```
        <meta name="subject" min="0" max="1">
              <begin inclusive="current">
                    <stringmatch case="no" loc="beginwith">
                          SUBJECT
                    </stringmatch>
              </begin>
              <end inclusive="current">onesection
              </end>
        </meta>

        <meta name="content" min="0" max="1">
              <begin inclusive="after">subject
              </begin>
              <end inclusive="before">nameformat</end>
        </meta>

        <meta name="creator">
              <begin inclusive="current">nameformat
              </begin>
              <end inclusive="current">onesection
              </end>
        </meta>

</structdef>
```

Template for the group "issuedby"

```
<?xml version="1.0" ?>
<structdef>

        <meta name="title" min="1" max="1">
              <begin inclusive="current">begin</begin>
              <end inclusive="before">dateformat</end>
        </meta>

        <meta name="date" min="0" max="1">
              <begin inclusive="current">dateformat</begin>
              <end inclusive="current">onesection</end>
        </meta>

        <meta name="sponsor" min="0" max="1">
              <begin inclusive="after">
                    <stringmatch case="no" loc="beginwith">
                          Sponsored by
                    </stringmatch>
              </begin>
              <end inclusive="before">
```

APPENDIX C (continued)

```
            <stringmatch case="no" loc="beginwith">
                  Issued by
            </stringmatch>
      </end>
</meta>

<meta name="issuedby" min="0" max="1">
      <begin inclusive="current">
            <stringmatch case="no" loc="beginwith">
                  Issued by
            </stringmatch>
      </begin>
      <end inclusive="before">
            <stringmatch case="no" loc="beginwith">
                  Contract No.
            </stringmatch>
      </end>
</meta>

<meta name="contract_no" min="0" max="1">
      <begin inclusive="current">
            <stringmatch case="no" loc="beginwith">
                  Contract No.
            </stringmatch>
      </begin>
      <end inclusive="current">onesection</end>
</meta>

<meta name="creator">
      <begin inclusive="current">nameformat</begin>
      <end inclusive="before">!nameformat</end>
</meta>
<meta name="effectivedate">
      <begin inclusive="after">
            <stringmatch case="no" loc="beginwith">
                  Effective Date
            </stringmatch>
      </begin>
      <end inclusive="before">
            <stringmatch case="no" loc="beginwith">
                  Contract Expiration Date
            </stringmatch>
      </end>
</meta>
<meta name="expiredate">
      <begin inclusive="after">
            <stringmatch case="no" loc="beginwith">
                  Contract Expiration Date
            </stringmatch>
      </begin>
      <end inclusive="before">
            <stringmatch case="no" loc="beginwith">
```

APPENDIX C (continued)

```
                        Reporting Period
                    </stringmatch>
              </end>
        </meta>
        <meta name="Coverage">
              <begin inclusive="after">
                    <stringmatch case="no" loc="beginwith">
                        Reporting Period
                    </stringmatch>
              </begin>
              <end inclusive="before">
                    <stringmatch case="no" loc="beginwith">
                        DISCLAIMER|The view and conclusions
                    </stringmatch>
              </end>
        </meta>
        <meta name="rights" min="0" max="1">
              <begin inclusive="current">
                    <stringmatch case="no" loc="beginwith">
                        Approved for
                    </stringmatch>
              </begin>
              <end inclusive="before">featurechange</end>
        </meta>

</structdef>
```

Template for the group "usawc"

```
<?xml version="1.0" ?>
<structdef>

        <meta name="type" min="1" max="1">
              <begin inclusive="current">begin</begin>
              <end inclusive="current">onesection</end>
        </meta>

        <meta name="title" min="1" max="1">
              <begin inclusive="after">type</begin>
              <end inclusive="before">
                    <stringmatch case="no" loc="onesection">
                        by
                    </stringmatch>
              </end>
        </meta>

        <meta name="creator" min="1" max="1">
              <begin inclusive="after">
                    <stringmatch case="no" loc="onesection">
                        by
                    </stringmatch>
              </begin>
```

APPENDIX C (continued)

```
            <end inclusive="current">onesection
            </end>
      </meta>

      <meta name="note" min="0" max="1">
            <begin inclusive="current">
                  <stringmatch case="no" loc="beginwith">
                        The views expressed
                  </stringmatch>
            </begin>
            <end inclusive="before">featurechange
            </end>
      </meta>

      <meta name="Publisher" min="0" max="1">
            <begin inclusive="after">note
            </begin>
            <end inclusive="before">end</end>
      </meta>


</structdef>
```

Template for the group "afrl"

```
<?xml version="1.0" ?>
<structdef>
      <meta name="identifier" min="1" max="1">
            <begin inclusive="current">begin</begin>
            <end inclusive="current">onesection</end>
      </meta>
      <meta name="type" min="1" max="1">
            <begin inclusive="after">identifier</begin>
            <end inclusive="current">onesection</end>
      </meta>
      <meta name="date" min="1" max="1">
            <begin inclusive="current">beginwithmonth</begin>
            <end inclusive="current">onesection</end>
      </meta>
      <meta name="title" min="1" max="1">
            <begin inclusive="after">date</begin>
            <end inclusive="before">sizechange</end>
      </meta>
      <meta name="creator" min="1">
            <begin inclusive="after">title</begin>
            <end inclusive="current">onesection</end>
      </meta>
      <meta name="contributor" min="0" max="1">
            <begin inclusive="after">
                  <stringmatch case="no" loc="beginwith">
                        Sponsored by
```

APPENDIX C (continued)

```
                    </stringmatch>
              </begin>
              <end inclusive="before">
                    <stringmatch case="no" loc="beginwith">
                          APPROVED FOR
                    </stringmatch>
              </end>
        </meta>
        <meta name="publisher" min="0" max="1">
              <begin inclusive="current">size=30</begin>
              <end inclusive="current">end</end>
        </meta>
</structdef>
```

Template for the group "edgewood"

```
<?xml version="1.0" ?>
<structdef>
        <meta name="identifier" min="1" max="1">
              <begin inclusive="current">size(26,36)</begin>
              <end inclusive="current">onesection</end>
        </meta>
        <meta name="title" min="1" max="1">
              <begin inclusive="after">identifier</begin>
              <end inclusive="before">sizechange</end>
        </meta>
        <meta name="creator" min="1">
              <begin inclusive="after">title</begin>
              <end inclusive="before">sizechange</end>
        </meta>
        <meta name="date" min="0" max="1">
              <begin inclusive="current">beginwithmonth</begin>
              <end inclusive="current">onesection</end>
        </meta>
</structdef>
```

Template for the group "nps"

```
<?xml version="1.0" ?>
<structdef>
        <meta name="title" min="1" max="1">
              <begin inclusive="after">
                    <stringmatch case="yes" loc="beginwith">
                          THESIS
                    </stringmatch>
              </begin>
              <end inclusive="before">
                    <stringmatch case="yes" loc="beginwith">
```

APPENDIX C (continued)

```
                        by
                    </stringmatch>
                </end>
        </meta>
        <meta name="creator" min="1">
                <begin inclusive="after">
                        <stringmatch case="yes" loc="beginwith">
                                by
                        </stringmatch>
                </begin>
                <end inclusive="before">beginwithmonth</end>
        </meta>
        <meta name="date" min="0" max="1">
                <begin inclusive="after">creator</begin>
                <end inclusive="current">onesection</end>
        </meta>
</structdef>
```

Template for the group "usnce"

```
<?xml version="1.0" ?>
<structdef>
        <meta name="title" min="1" max="1">
                <begin inclusive="current">size(34,43)</begin>
                <end inclusive="before">sizechange</end>
        </meta>
        <meta name="creator" min="1">
                <begin inclusive="after">title</begin>
                <end inclusive="before">beginwithmonth</end>
        </meta>
        <meta name="date" min="0" max="1">
                <begin inclusive="after">creator</begin>
                <end inclusive="current">onesection</end>
        </meta>
</structdef>
```

Template for the group "afit"

```
<?xml version="1.0" ?>
<structdef>
        <meta name="title" min="1" max="1">
                <begin inclusive="current">begin</begin>
                <end inclusive="before">
                        <stringmatch case="yes" loc="beginwith">
                                THESIS
                        </stringmatch>
                </end>
        </meta>
```

APPENDIX C (continued)

```
      <meta name="creator" min="1">
            <begin inclusive="after">
                  <stringmatch case="yes" loc="beginwith">
                        THESIS
                  </stringmatch>
            </begin>
            <end inclusive="before">
                  <stringmatch case="no" loc="beginwith">
                        AFIT/

                  </stringmatch>
            </end>
      </meta>
      <meta name="identifier" min="0" max="1">
            <begin inclusive="after">creator</begin>
            <end inclusive="current">onesection</end>
      </meta>
      <meta name="Publisher" min="0" max="1">
            <begin inclusive="after">identifier</begin>
            <end inclusive="before">
                  <stringmatch case="no" loc="beginwith">
                        APPROVED FOR
                  </stringmatch>
            </end>
      </meta>
      <meta name="Rights" min="0" max="1">
            <begin inclusive="after">contributor</begin>
            <end inclusive="current">end</end>
      </meta>
</structdef>
```

Template for the Group "text"

```
<?xml version="1.0" ?>
<structdef>
      <meta name="title" min="1" max="1">
            <begin inclusive="current">
                  <stringmatch case="yes" loc="beginwith">
                        TITLE
                  </stringmatch>
            </begin>
            <end inclusive="before">
                  <stringmatch case="yes" loc="beginwith">
                        PRINCIPAL INVESTIGATOR
                  </stringmatch>
            </end>
      </meta>
      <meta name="creator" min="1">
            <begin inclusive="after">title</begin>
```

APPENDIX C (continued)

```
            <end inclusive="before">
                  <stringmatch case="yes" loc="beginwith">
                        CONTRACTING ORGANIZATION
                  </stringmatch>
            </end>
      </meta>
      <meta name="contributor" min="0" max="1">
            <begin inclusive="after">creator</begin>
            <end inclusive="before">
                  <stringmatch case="yes" loc="beginwith">
                        REPORT DATE
                  </stringmatch>
            </end>
      </meta>
      <meta name="date" min="0" max="1">
            <begin inclusive="after">contributor</begin>
            <end inclusive="before">
                  <stringmatch case="yes" loc="beginwith">
                        TYPE OF REPORT
                  </stringmatch>
            </end>
      </meta>
      <meta name="type" min="0" max="1">
            <begin inclusive="after">date</begin>
            <end inclusive="current">onesection</end>
      </meta>
</structdef>
```

APPENDIX C (continued)

C.4. Metadata Extraction Results

TABLE 24
METADATA EXTRACTION RESULTS OF THE GROUP "SF298_1"

Field	#doc	#C	#p	#in	Recall		Precision		F-measure	
					compl	partial	compl	partial	compl	partial
agency	0	0	0	0	N/A	N/A	N/A	N/A	N/A	N/A
date	3	2	1	0	66.67%	100%	66.67%	100%	66.67%	100%
typecoverage	3	2	1	0	66.67%	100%	66.67%	100%	66.67%	100%
Title	3	3	0	0	100%	100%	100%	100%	100%	100%
funding number	2	2	0	0	100%	100%	100%	100%	100%	100%
creator	3	3	0	0	100%	100%	100%	100%	100%	100%
perform_org	6	6	0	0	100%	100%	100%	100%	100%	100%
Report_no	3	3	0	0	100%	100%	100%	100%	100%	100%
sponsor	3	3	0	0	100%	100%	100%	100%	100%	100%
sponsor_no	2	0	0	2	0%	0%	0%	0%	N/A	N/A
Notes	2	2	0	0	100%	100%	100%	100%	100%	100%
distribution	3	2	1	0	66.67%	100%	66.67%	100%	66.67%	100%
dis_code	0	0	0	0	N/A	N/A	N/A	N/A	N/A	N/A
abstract	2	2	0	1	100%	100%	66.67%	66.67%	80.00%	80.00%
subject	2	1	0	0	50.00%	50.00%	100%	100%	66.67%	66.67%
no_page	3	3	0	0	100%	100%	100%	100%	100%	100%
price_code	0	0	0	0	N/A	N/A	N/A	N/A	N/A	N/A
cls_report	2	1	1	0	50.00%	100%	50.00%	100%	50.00%	100%
page_cls	2	2	0	0	100%	100%	100%	100%	100%	100%
abs_cls	2	2	0	0	100%	100%	100%	100%	100%	100%
lim_abstract	2	2	0	0	100%	100%	100%	100%	100%	100%

APPENDIX C (continued)

TABLE 25
METADATA EXTRACTION RESULT OF THE GROUP "SF298_2"

Field	#doc	#C	#p	#in	recall		precision		F-measure	
					compl	partial	compl	partial	compl	partial
date	3	2	0	0	66.67%	66.67%	100%	100%	80.00%	80.00%
report type	2	2	0	0	100%	100%	100%	100%	100%	100%
dates covered	2	2	0	0	100%	100%	100%	100%	100%	100%
title	3	3	0	0	100%	100%	100%	100%	100%	100%
contract num	2	1	0	0	50.00%	50.00%	100%	100%	66.67%	66.67%
grant_no	1	1	0	0	100%	100%	100%	100%	100%	100%
program_no	0	0	0	0	N/A	N/A	N/A	N/A	N/A	N/A
creator	3	3	0	0	100%	100%	100%	100%	100%	100%
project_no	0	0	0	0	N/A	N/A	N/A	N/A	N/A	N/A
task_no	0	0	0	0	N/A	N/A	N/A	N/A	N/A	N/A
work_no	0	0	0	0	N/A	N/A	N/A	N/A	N/A	N/A
perform_org	3	2	0	0	66.67%	66.67%	100%	100%	80.00%	80.00%
report_no	1	1	0	1	100%	100%	50.00%	50.00%	66.67%	66.67%
sponsor	3	2	0	0	66.67%	66.67%	100%	100%	80.00%	80.00%
sponsor_acr	3	2	0	0	66.67%	66.67%	100%	100%	80.00%	80.00%
sponsor_no	1	1	0	0	100%	100%	100%	100%	100%	100%
distribution	3	3	0	0	100%	100%	100%	100%	100%	100%
notes	1	1	0	0	100%	100%	100%	100%	100%	100%
abstract	2	2	0	1	100%	100%	66.67%	66.67%	80.00%	80.00%
subject	3	3	0	0	100%	100%	100%	100%	100%	100%
no_page	2	2	0	0	100%	100%	100%	100%	100%	100%
Responsible_person	2	2	0	0	100%	100%	100%	100%	100%	100%
cls_report	2	2	0	0	100%	100%	100%	100%	100%	100%
page_cls	2	2	0	0	100%	100%	100%	100%	100%	100%
abs_cls	2	2	0	0	100%	100%	100%	100%	100%	100%
lim_abstract	2	2	0	0	100%	100%	100%	100%	100%	100%
Responsible_phone	2	2	0	1	100%	100%	66.67%	66.67%	80.00%	80.00%

APPENDIX C (continued)

TABLE 26
METADATA EXTRACTION RESULTS OF THE GROUP "GENERIC"

field	#doc	#C	#p	#in	Recall		precision		F-measure	
					compl	partial	compl	partial	compl	Partial
title	14	12	2	0	85.71%	100%	85.71%	100%	85.71%	100%
creator	14	6	7	0	42.86%	92.86%	46.15%	100%	44.44%	96.30%
date	10	8	0	0	80.00%	100%	100%	100%	88.89%	100%
Rights	14	12	0	0	85.71%	85.71%	100%	100%	92.31%	92.31%
type	11	Did not try								
identifier	4	Did not try								
Publisher	8	Did not try								

TABLE 27
METADATA EXTRACTION RESULTS OF THE GROUP "THESIS"

field	#doc	#C	#p	#in	recall		precision		F-measure	
					compl	partial	compl	partial	compl	Partial
title	3	3	0	0	100%	100%	100%	100%	100%	100%
creator	3	3	0	0	100%	100%	100%	100%	100%	100%
date	3	1	0	0	33.33%	33.33%	100%	100%	50.00%	50.00%
Rights	2	2	0	0	100%	100%	100%	100%	100%	100%
thesis	3	3	0	0	100%	100%	100%	100%	100%	100%
degree	3	3	0	0	100%	100%	100%	100%	100%	100%
program	3	3	0	0	100%	100%	100%	100%	100%	100%

TABLE 28
METADATA EXTRACTION RESULTS OF THE GROUP "LETTER"

field	#doc	#C	#p	#in	recall		precision		F-measure	
					compl	partial	compl	partial	compl	partial
Publisher	1	1	0	0	100%	100%	100%	100%	100%	100%
date	1	1	0	0	100%	100%	100%	100%	100%	100%
title	1	1	0	0	100%	100%	100%	100%	100%	100%
subject	1	1	0	0	100%	100%	100%	100%	100%	100%
content	1	1	0	0	100%	100%	100%	100%	100%	100%
creator	1	1	0	0	100%	100%	100%	100%	100%	100%

APPENDIX C (continued)

TABLE 29
METADATA EXTRACTION RESULTS OF THE GROUP "ISSUEDBY"

field	#doc	#C	#p	#in	recall		precision		F-measure	
					compl	partial	compl	partial	compl	partial
title	1	0	1	0	0.00%	100%	0.00%	100%	N/A	100%
date	1	1	0	0	100%	100%	100%	100%	100%	100%
sponsor	1	1	0	0	100%	100%	100%	100%	100%	100%
issuedby	1	1	0	0	100%	100%	100%	100%	100%	100%
contact_no	1	1	0	0	100%	100%	100%	100%	100%	100%
creator	1	1	0	0	100%	100%	100%	100%	100%	100%
effectivedate	1	1	0	0	100%	100%	100%	100%	100%	100%
expiredate	1	1	0	0	100%	100%	100%	100%	100%	100%
Coverage	1	1	0	0	100%	100%	100%	100%	100%	100%
rights	1	1	0	0	100%	100%	100%	100%	100%	100%

TABLE 30
METADATA EXTRACTION RESULTS OF THE GROUP "USAWC"

field	#doc	#C	#p	#in	recall		precision		F-measure	
					compl	partial	compl	partial	compl	partial
Type	2	2	0	0	100%	100%	100%	100%	100%	100%
Title	2	2	0	0	100%	100%	100%	100%	100%	100%
creator	2	2	0	0	100%	100%	100%	100%	100%	100%
Note	2	2	0	0	100%	100%	100%	100%	100%	100%
Publisher	2	0	2	0	0.00%	100%	0.00%	100%	N/A	100%
Contributor	2	Did not try								

TABLE 31
METADATA EXTRACTION RESULTS OF THE GROUP "AFRL"

field	#doc	#C	#p	#in	recall		precision		F-measure	
					compl	partial	compl	partial	compl	partial
Identifier	5	5	0	0	100%	100%	100%	100%	100%	100%
Type	5	5	0	0	100%	100%	100%	100%	100%	100%
Date	5	5	0	0	100%	100%	100%	100%	100%	100%
Title	5	5	0	0	100%	100%	100%	100%	100%	100%
Creator	5	5	0	0	100%	100%	100%	100%	100%	100%
Contributor	5	5	0	0	100%	100%	100%	100%	100%	100%
Publisher	5	5	0	0	100%	100%	100%	100%	100%	100%
Rights	5	0	0	0	Did not try					

APPENDIX C (continued)

TABLE 32
METADATA EXTRACTION RESULTS OF THE GROUP "ARL"

field	#doc	#C	#p	#in	recall		precision		F-measure	
					compl	partial	compl	partial	compl	partial
Identifier	5	5	0	0	100%	100%	100%	100%	100%	100%
Date	5	5	0	0	100%	100%	100%	100%	100%	100%
Title	5	4	1	0	80.00%	100%	80.00%	100%	80.00%	100%
Creator	5	4	1	0	80.00%	100%	80.00%	100%	80.00%	100%
Rights	5	0	0	0	Did not try					

TABLE 33
METADATA EXTRACTION RESULTS OF THE GROUP "EDGEWOOD"

field	#doc	#C	#p	#in	recall		precision		F-measure	
					compl	partial	compl	partial	compl	partial
Identifier	4	4	0	0	100%	100%	100%	100%	100%	100%
Date	4	3	0	0	75.00%	75.00%	100%	100%	85.71%	85.71%
Title	4	3	1	0	75.00%	100%	75.00%	100%	75.00%	100%
Creator	4	0	4	0	0%	100%	0%	100%	N/A	100%
Rights	4	0	0	0	Did not try					

TABLE 34
METADATA EXTRACTION RESULTS OF THE GROUP "NPS"

field	#doc	#C	#p	#in	recall		precision		F-measure	
					compl	partial	compl	partial	compl	partial
Creator	15	14	0	0	93.33%	93.33%	100%	100%	96.55%	96.55%
Date	15	14	1	0	93.33%	100%	93.33%	100%	93.33%	100%
Title	15	13	0	0	86.67%	86.67%	100%	100%	92.86%	92.86%
contributor	15	0	0	0	Did not try					
Rights	15	0	0	0	Did not try					

APPENDIX C (continued)

TABLE 35
METADATA EXTRACTION RESULTS OF THE GROUP "USNCE"

field	#doc	#C	#p	#in	recall		precision		F-measure	
					compl	partial	compl	partial	compl	partial
Creator	5	4	1	0	80%	100%	80%	100%	80%	100%
Date	5	5	0	0	100%	100%	100%	100%	100%	100%
Title	5	5	0	0	100%	100%	100%	100%	100%	100%
Contributor	5	0	0	0	Did not try					
Rights	5	0	0	0	Did not try					

TABLE 36
METADATA EXTRACTION RESULTS OF THE GROUP "AFIT"

Field	#doc	#C	#p	#in	Recall		Precision		F-measure	
					compl	partial	compl	partial	compl	partial
Title	6	6	0	0	100%	100%	100%	100%	100%	100%
Creator	6	6	0	0	100%	100%	100%	100%	100%	100%
Publisher	6	6	0	0	100%	100%	100%	100%	100%	100%
Identifier	6	6	0	0	100%	100%	100%	100%	100%	100%
Rights	6	6	0	0	100%	100%	100%	100%	100%	100%
Type	6	Did not try								

TABLE 37
METADATA EXTRACTION RESULTS OF THE GROUP "TEXT"

Field	#doc	#C	#p	#in	Recall		Precision		F-measure	
					compl	partial	Compl	partial	compl	partial
Title	33	30	3	0	90.91%	100%	90.91%	100%	90.91%	100%
Creator	33	32	1	0	96.97%	100%	96.97%	100%	96.97%	100%
Contributor	33	32	1	0	96.97%	100%	96.97%	100%	96.97%	100%
Date	33	26	7	0	78.79%	100%	78.79%	100%	78.79%	100%
Type	33	32	1	0	96.97%	100%	96.97%	100%	96.97%	100%
Rights	33	0	0	0	Did not try					

APPENDIX D

DATA SET AND TEMPLATES USED IN EXPERIMENTS IN THE SECTION 6.2.4.

D.1. Data Set

Group	doc #	List of IDs
GPOForm	14	LPS64485 , LPS64487 , LPS64488 , LPS64490 , LPS64494 , LPS64495 , LPS64496 , LPS64497 , LPS64498 , LPS64499 , LPS64500 , LPS64542 , LPS64547 , LPS64548
GPONonForm	57	LPS60590 , LPS60600 , LPS60632 , LPS60634 , LPS60640 , LPS60646 , LPS60654 , LPS60659 , LPS60668 , LPS60672 , LPS60679 , LPS60685 , LPS60692 , LPS60700 , LPS60701 , LPS60708 , LPS60715 , LPS60718 , LPS60719 , LPS60821 , LPS60926 , LPS60939 , LPS60940 , LPS60945 , LPS60951 , LPS60970 , LPS61006 , LPS61022 , LPS61126 , LPS61147 , LPS61275 , LPS61350 , LPS61368 , LPS61372 , LPS61382 , LPS61412 , LPS61785 , LPS61838 , LPS62107 , LPS62120 , LPS62297 , LPS62341 , LPS62344 , LPS62362 , LPS62378 , LPS62380 , LPS62382 , LPS62384 , LPS62419 , LPS62426 , LPS62763 , LPS62862 , LPS62888 , LPS63173 , LPS63485 , LPS63488 , LPS63610
Congress Report	16	LPS61663 , LPS61830 , LPS62091 , LPS62154 , LPS62171 , LPS62225 , LPS62236 , LPS62466 , LPS62497 , LPS62578 , LPS62613 , LPS62705 , LPS62710 , LPS62816 , LPS61612 , LPS62237
Public Law	16	LPS60020 , LPS60022 , LPS60024 , LPS61432 , LPS61457 , LPS61459 , LPS61461 , LPS62472 , LPS62622 , LPS62628 , LPS62660 , LPS62739 , LPS63165 , LPS63332 , LPS62656 , LPS62658

D.2. Templates

Template of the Group "GPOForm"

```
<template>

<form max="-1">
<match max="5">
      <line>Technical Report Documentation Page</line>
</match>
<fixed>
      <field num="1"><line>1. Report No.</line></field>
      <field num="2"><line>2. Government Accession No.</line></field>
      <field num="3"><line>3. Recipient's Catalog No.</line></field>
      <field num="4"><line>4. Title and Subtitle</line></field>
```

APPENDIX D (continued)

```
      <field num="5"><line>5. Report Date</line></field>
      <field num="6"><line>6. Performing Organization
Code</line></field>
      <field num="7"><line>7. Author(s)</line></field>
      <field num="8"><line>8. Performing Organization Report
No.</line></field>
      <field num="9"><line>9. Performing Organization Name and
Address</line></field>
      <field num="10"><line>10. Work Unit No. (TRAIS)</line></field>
      <field num="11"><line>11. Contract or Grant No.</line></field>
      <field num="12"><line>12. Sponsoring Agency Name and
Address</line></field>
      <field num="13"><line>13. Type of Report and Period
Covered</line></field>
      <field num="14"><line>14. Sponsoring Agency Code</line></field>
      <field num="15"><line>15. Supplementary Notes</line></field>
      <field num="16"><line>16. Abstract</line></field>
      <field num="17"><line>17. Key Words</line></field>
      <field num="18"><line>18. Distribution Statement</line></field>
      <field num="19"><line>19. Security Classif. (of this
report)</line></field>
      <field num="20"><line>20. Security Classif. (of this
page)</line></field>
      <field num="21"><line>21. No. of pages</line></field>
      <field num="22"><line>22. Price</line></field>
</fixed>
<extracted>
      <metadata name="report_num">
            <rule relation="belowof" field="1"/>
            <rule relation="leftof" field="2"/>
            <rule relation="aboveof" field="4"/>
      </metadata>
      <metadata name="government_accession_num">
            <rule relation="belowof" field="2"/>
            <rule relation="rightof" field="1"/>
            <rule relation="leftof" field="3"/>
            <rule relation="aboveof" field="4"/>
      </metadata>
      <metadata name="recipient_catalog_num">
            <rule relation="belowof" field="3"/>
            <rule relation="rightof" field="2"/>
            <rule relation="aboveof" field="4|5"/>
      </metadata>
      <metadata name="title">
            <rule relation="belowof" field="4"/>
            <rule relation="leftof" field="5|6"/>
            <rule relation="aboveof" field="7"/>
      </metadata>
      <metadata name="reportdate">
            <rule relation="belowof" field="5"/>
            <rule relation="aboveof" field="6"/>
            <rule relation="rightof" field="4"/>
```

APPENDIX D (continued)

```
</metadata>
<metadata name="performing_organization_code">
      <rule relation="belowof" field="6"/>
      <rule relation="aboveof" field="8"/>
      <rule relation="rightof" field="4"/>
</metadata>
<metadata name="authors">
      <rule relation="belowof" field="7"/>
      <rule relation="aboveof" field="9|10"/>
      <rule relation="leftof" field="8|10"/>
</metadata>
<metadata name="performing_number">
      <rule relation="belowof" field="8"/>
      <rule relation="aboveof" field="9"/>
      <rule relation="leftof" field="8"/>
</metadata>
<metadata name="performing_organization">
      <rule relation="belowof" field="9"/>
      <rule relation="aboveof" field="12"/>
      <rule relation="leftof" field="10|11"/>
</metadata>
<metadata name="work_unit_num">
      <rule relation="belowof" field="10"/>
      <rule relation="aboveof" field="11"/>
      <rule relation="rightof" field="9"/>
</metadata>
<metadata name="contract_grant_num">
      <rule relation="belowof" field="11"/>
      <rule relation="aboveof" field="13"/>
      <rule relation="rightof" field="9"/>
</metadata>
<metadata name="sponsor">
      <rule relation="belowof" field="12"/>
      <rule relation="aboveof" field="15"/>
      <rule relation="leftof" field="13|14"/>
</metadata>
<metadata name="report_type_coverage">
      <rule relation="belowof" field="13"/>
      <rule relation="aboveof" field="14"/>
      <rule relation="rightof" field="12"/>
</metadata>
<metadata name="sponsor_code">
      <rule relation="belowof" field="14"/>
      <rule relation="aboveof" field="15"/>
      <rule relation="rightof" field="12"/>
</metadata>
<metadata name="notes">
      <rule relation="belowof" field="15"/>
      <rule relation="aboveof" field="16"/>
</metadata>
<metadata name="abstract">
      <rule relation="belowof" field="16"/>
```

APPENDIX D (continued)

```
            <rule relation="aboveof" field="17|18"/>
      </metadata>
      <metadata name="keywords">
            <rule relation="belowof" field="17"/>
            <rule relation="aboveof" field="19|20"/>
            <rule relation="leftof" field="18"/>
      </metadata>
      <metadata name="dist_statement">
            <rule relation="belowof" field="18"/>
            <rule relation="rightof" field="17"/>
            <rule relation="aboveof" field="20|21|22"/>
      </metadata>
      <metadata name="sec_classification_report">
            <rule relation="belowof" field="19"/>
            <rule relation="leftof" field="20"/>
      </metadata>
      <metadata name="sec_classification_page">
            <rule relation="belowof" field="20"/>
            <rule relation="rightof" field="19"/>
            <rule relation="leftof" field="21"/>
      </metadata>
      <metadata name="num_pages">
            <rule relation="belowof" field="21"/>
            <rule relation="rightof" field="20"/>
            <rule relation="leftof" field="22"/>
      </metadata>
      <metadata name="price">
            <rule relation="belowof" field="22"/>
            <rule relation="rightof" field="21"/>
      </metadata>
</extracted>
<exclude>\QForm DOT F1700\E.*</exclude>
<exclude>\QReproduction of completed page authorized\E.*</exclude>
</form>
</template>
```

Template of the Group "GPONonform"

```
<?xml version="1.0" ?>
<structdef pagenumber="1">

      <meta name="title" min="1" max="1">
            <begin inclusive="current">size(1199,1701)</begin>
            <end inclusive="before"><stringmatch case="yes"
loc="beginwith">HEARING|ROUNDTABLE|JOINT HEARING|FIELD
HEARING</stringmatch></end>
      </meta>

      <meta name="type" min="1" max="1">
```

APPENDIX D (continued)

```
         <begin inclusive="current"><stringmatch case="yes"
loc="beginwith">HEARING|ROUNDTABLE|JOINT HEARING|FIELD
HEARING</stringmatch></begin>
         <end inclusive="before">size(800,1000)</end>
      </meta>

      <meta name="session" min="1" max="1">
            <begin inclusive="after">type</begin>
            <end inclusive="before">beginwithmonth</end>
      </meta>

      <meta name="date" min="0" max="1">
            <begin inclusive="current">dateformat</begin>
            <end inclusive="current">onesection</end>
      </meta>
  <meta name="serialno">
         <begin inclusive="current"><stringmatch case="yes"
loc="beginwith">Serial No</stringmatch></begin>
            <end inclusive="current">onesection</end>
</meta>
   <use>
         <begin inclusive="current"><stringmatch case="yes"
loc="beginwith">Printed for the use</stringmatch></begin>
        <end inclusive="before"><stringmatch case="yes"
loc="beginwith">Serial No|Available via the World Wide Web|U.S.
GOVERNMENT</stringmatch></end>
</meta>
</structdef>
```

Template of the Group "Congress Report"

```
<?xml version="1.0" ?>
<structdef pagenumber="1">
      <meta name="candno">
         <begin inclusive="current"><stringmatch case="yes"
loc="beginwith">Calendar No</stringmatch></begin>
        <end inclusive="current">onesection</end>
</meta>
      <meta name="session">
            <begin inclusive="current">size(500,801)</begin>
            <end inclusive="before">largeststrsize(0,0.5)</end>
      </meta>
      <meta name="title" min="1" max="1">
            <begin inclusive="current">largeststrsize(0,0.5)</begin>
            <end inclusive="before">layoutchange</end>
      </meta>

      <meta name="date" min="0" max="1">
            <begin inclusive="after">title</begin>
            <end inclusive="before">sizechange(50)</end>
      </meta>
```

APPENDIX D (continued)

```
      <meta name="creator" min="0" max="1">
            <begin inclusive="after">date</begin>
            <end inclusive="before">featurechange</end>
      </meta>
      <meta name="type">
         <begin inclusive="current"><stringmatch case="yes"
loc="beginwith">R E P O R T|ADVERSE REPORT</stringmatch></begin>
         <end inclusive="before"><stringmatch case="yes"
loc="beginwith">[to accompany|[To accompany</stringmatch></end>
      </meta>
      <meta name="accompany">
         <begin inclusive="current"><stringmatch case="yes"
loc="beginwith">[to accompany|[To accompany</stringmatch></begin>
        <end inclusive="current">onesection</end>
   </meta>
   <meta name="cost">
         <begin inclusive="current"><stringmatch case="yes"
loc="beginwith">[Including cost estimate</stringmatch></begin>
        <end inclusive="current">onesection</end>
   </cost>
   <meta name="notes">
        <begin inclusive="current">size(990, 1110)</begin>
        <end inclusive="before">sizechange(100)</end>
   </meta>
</structdef>
```

Template of the Group "Public Law"

```
<?xml version="1.0" ?>
<structdef pagenumber="1">
      <meta name="date">
         <begin inclusive="current"><stringmatch case="no"
loc="beginwith">PUBLIC LAW|118 STAT|119 STAT</stringmatch></begin>
        <end inclusive="current">onesection</end>
      </meta>

      <meta name="bill_number">
        <begin inclusive="current"><stringmatch case="no"
loc="beginwith">[H.R.|[S.</stringmatch></begin>
        <end>onesection</end>
      </meta>

      <meta name="congress_num">
         <begin inclusive="current">largeststrsize(0,0.3)</begin>
        <end inclusive="before">layoutchange</end>
      </meta>

      <meta name="type">
         <begin inclusive="after">congress_num</begin>
        <end inclusive="before">layoutchange</end>
      </meta>
   </structdef>
```

VITA

JIANFENG TANG

Computer Science Department

Old Dominion University

Norfolk, VA 23539

EDUCATION

B.E. Computer Science, July 1995, Beijing University of Aeronautics and Astronautics, China

M.S. Computer Science, July 1998, Institute of Computing Technology, China

Ph.D. Computer Science, December 2006, Old Dominion University