Access Restriction Checker

Brian Harris
Elizabeth Whitaker
Robert Simpson

PERPOS TR ITTL/CSITD 05-07
18 August 2005

Georgia Tech Research Institute
Information Technology and Telecommunications Laboratory
Atlanta, Georgia

The Army Research Laboratory (ARL) and the National Archives and Records Administration (NARA) sponsored this research under Army Research Office Cooperative Agreement DAAD19-03-2-0018. The findings in this paper should not be construed as an official ARL or NARA position unless so indicated by other authorized documentation.

Abstract

Part of the PERPOS project has been to analyze the kinds of knowledge that archivists use to review Presidential Records for Presidential Record Act (PRA) restrictions and Freedom of Information Act (FOIA) exceptions, and to develop an automated tool that could use this knowledge to support archivist's decisions in reviewing Presidential Records. We have begun prototyping such a tool, which we call the Access Restriction Checker. The results of our initial exploration show great promise for such a tool and believe it would be a great labor saver as a component in the future archivist's tool kit. Such a tool is not a replacement for the judgment of archivists, whose responsibility it is to review the records; rather the tool is a decision support tool. This Technical report provides an overview of our initial work on the Access Restriction Checker. Additional work is required to broaden the knowledge coverage to other types of access restrictions.

Table of Contents

INTRODUCTION **1**

BACKGROUND 1
PURPOSE 2
SCOPE 2

RELEVANT TECHNOLOGIES **3**

RULE-BASED REASONING 3
CASE-BASED REASONING 3
MACHINE LEARNING 5

OVERVIEW OF THE CURRENT PROTOTYPE **5**

ARCHITECTURE 7
THE PROCESS 9

GRAPHICAL USER INTERFACE FOR ARCHIVAL REVIEW **10**

DEVELOPMENT OF DECISION RULES FOR ACCESS RESTRICTIONS **15**

SUMMARY AND FUTURE RESEARCH **19**

REFERENCES **20**

Introduction

Background

The Presidential Electronic Records PilOt System (PERPOS) project has developed tools to assist archivists in processing electronic records created by office applications on personal computers. The tools, called the Archival Repository Tool (ART) and the Archival Processing Tool (APT), support archivists in accessioning, arranging, preserving, reviewing, and describing record series. During the review activity, a reviewer can view records in a file system and review them for access restrictions. The records (files) can be opened for public access, closed to public access, redacted and opened for public access, marked as a Personal Record Misfile, or transferred to a software library, because they are misfiled software that was used to create the records, rather than being a record [Underwood et al 2005].

Review of Presidential electronic records for access restrictions is an intellectually demanding task that requires page-by-page review of Presidential Library accessions. Due to the increasing volume of electronic records from Presidential administrations, the need to review these records and the cost of the limited human resources that can be applied to the review process, the review process is an archival processing bottleneck.

Another objective of PERPOS project has been to analyze the kinds of knowledge that archivists use to review Presidential Records for Presidential Record Act (PRA) restrictions and Freedom of Information Act (FOIA) exceptions, and to develop an automated tool that could be used with this knowledge to support archivist's decisions in reviewing Presidential Records. Such a tool is not a replacement for the judgment of archivists, whose responsibility it is to review the records; rather the tool is a decision support tool.

There are many potential benefits of such a tool.

1) It might identify an access restriction not identified by the reviewer, thus reducing the risk of opening a record or passage of a record whose access should have been restricted.
2) It might be used as a tutor during training of review archivists.
3) Novice reviewers could use the tool to check their work.
4) The tool might provide additional evidence in case a reviewer's judgment was uncertain, or point out uncertainties, where the reviewer thought the decision was certain.
5) It might give a rapid review to records responsive to a FOIA request to estimate the workload in terms of the number of restrictions and types of restrictions likely to apply.
6) It might estimate which unprocessed electronic record series are likely to have many restrictions, and which are likely to have few or no restrictions. The

systematic review of those with no or few restrictions could result in more records being opened to the public at an earlier date.

7) Experienced reviewers are eventually promoted or retire and NARA and Presidential Libraries lose their expertise. The tool might accumulate review knowledge so that the knowledge resource is not lost.
8) The tool will support PRA and FOIA review decisions for Presidential records, so it would also support review of Federal Records for FOIA exemptions.
9) Since most states have Open Record Acts, state records need to be reviewed for access restrictions before release to the public. The technology might be transferred to support archivists performing review of state government records.
10) Although the records being considered in this study are unclassified records, the technology might transfer to declassification review.
11) Paper records can be scanned to produce digital images of the records. These images can be converted to machine readable records using OCR technology. Thus, the access restriction checker could be applied to review of machine readable, OCRed, paper records, as well as records originally created digital.

We call this tool an Access Restriction Checker (or Classifier) because of its similarity to spell checkers or style checkers.

Purpose

The research objective is to demonstrate the feasibility of automatically classifying records:

- as Personal Record Misfiles, or
- Presidential Records, or passages of Presidential Records, whose access should be restricted because they are exempt from release due to a paragraph of the Freedom of Information Act, or restricted from release due to a paragraph of the Presidential Records Act, or
- can be opened because they are not subject to a FOIA exemption or PRA Restriction.

The purpose of this report is to describe progress in achieving this research objective through the development of a prototype access restriction checker.

Scope

The next section of this paper discusses advanced information technologies that can be applied to this problem. The third section provides an overview of the architecture and process. The fourth section describes the graphical user interface. In the fifth section, the use of the access restriction checker in the development of decision rules is described. Finally, the research results are summarized and future research is discussed.

Relevant Technologies

There are several technologies which are being brought together to help in semi-automated archival review, i.e., using automation to aid an archivist in the interactive review of documents. This hybrid approach will provide more robustness than a tool which will make use of only one technology.

Rule-Based Reasoning

Rule-based reasoning is an artificial intelligence technique that makes use of the knowledge of human experts gathered through interviews, literature reviews of the expert process, in this case an archival process, and observation of the experts, i.e., the archivists at work. This knowledge is encoded as production rules that can be used by the system to reason about a set of facts. A production rule is a condition-action pair. Whenever the system recognizes the pattern or condition in working memory, the rule is executed and asserts new facts to working memory, based on the action performed. These new facts represent the reasoning of the system.

This is the technique that was used to implement the first large expert systems of the 1980s. We are not implementing a large expert system, since these are hard to manage and maintain, but are, rather implementing small specialized modules in the approach of the newer intelligent agents. A collection of these modules will be brought to the problem of identifying the access restrictions in a collection of documents. This method of development will allow the prototype to evolve by adding new modules implemented as rule-based agents or as one of the other techniques described in this section. The modules, as the prototype evolves can be distributed, i.e., multiple modules can execute simultaneously on a cluster of processors, bringing more computing power to the problem of aiding the archivist in reviewing the massive amounts of data that must be reviewed. This approach, in combination with other efficiency techniques, will enable the scaling that will be necessary for this problem.

In the Access Restriction Checker, rule-based reasoning is useful for recognizing patterns in text that are well understood and are able to be articulated by the archivist. Pieces of knowledge which are not so easily expressed by the expert can be discovered using other techniques such as case-based reasoning or machine learning.

Case-Based Reasoning

Case-Based Reasoning (CBR)[Kolodner 1993] solves new problems by applying stored experiences. Past experiences are stored as cases in a case library that may be implemented as a database. A case-based software system solves new problems by retrieving similar cases from its case memory and selecting one or more that most resemble the new problem. The system adapts the retrieved solution into a new solution and evaluates it for the current problem. After repairing any faults in the new solution,

the system stores it along with feedback from its execution as a new case for possible reuse. There have been a number of studies that have shown that humans use their memories of previous experiences as a means of solving new problems.

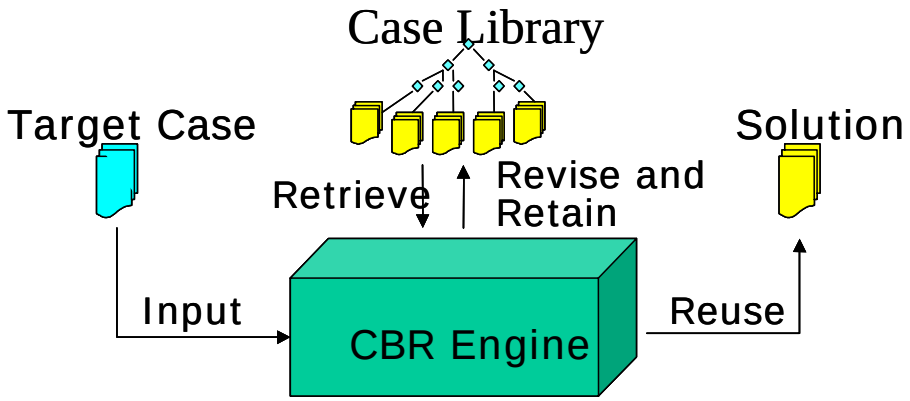


Figure 2 - Case-based Reasoning Process

The processes involved in CBR can be represented by a schematic cycle (see Figure 2). Aamodt and Plaza [1994] among others have described CBR typically as a cyclical process:

1. RETRIEVE the most similar case(s) from the case library
2. REUSE the case(s) to attempt to solve the problem
3. REVISE or ADAPT the proposed solution if necessary
4. RETAIN the new solution as a part of a new case that gets added to the case library

A new problem is matched against cases in the case base and one or more similar cases are *retrieved*. A solution suggested by the matching cases is then *reused* and tested for success. Unless the retrieved case is a very close match the solution will probably have to be *revised* (adapted) producing a new case that can be *retained*.

Case Representation for CBR: A case is a highly contextualized piece of knowledge representing an experience. It contains the past lesson that is the content of the case and the context in which the lesson can be used in the future. Typically a case comprises:

5. *Problem/situation description:* The state of the world when the episode recorded in the case occurred, and, if appropriate, the problem that needed to be solved
6. *Solution:* The stated or derived solution to the problem
7. *Outcome:* The result of carrying out the solution in the given situation

The case problem representation includes a set of features that describe the important characteristics of the problem and are used to index cases in the case library. Cases that include problems and their solutions can be used to derive solutions to new problems. Cases can be represented in a variety of forms using the full range of artificial intelligence representation formalisms including frames, objects, predicates, semantic nets and rules. The frame/object representation that will be used in this project is typical of approaches used in the majority of recent CBR software implementations.

For the Access Restriction Checker we envision features that describe the most important characteristics of cases of restrictions will be the same features as those involved in the

rule-based reasoning approach, e.g., document type, topic, job titles of author and addressee, political issues, and economic issues. These characteristics will be stored along with the document and the results of an expert archivist's review. A new document being reviewed will be matched by the system with documents which have similar characteristics. The results of the archival reviews of those similar documents can be used as a starting point for the review of the new document.

Machine Learning

Machine learning techniques can be used to improve the reasoning of a system by adding to and fine tuning the decision-making capabilities of the software. For the Access Restriction Checker "induction algorithms" will be used to "induce" or learn new rules for identifying PRA or FOIA restrictions from large collections of examples of documents marked up with their important characteristics and their related archival annotations. These collections of documents from which the rules are learned are called "training sets." The induction algorithms use clustering and statistical pattern recognition techniques to identify the features of a document which are associated with a particular type of PRA or FOIA restriction. The learned rules can be applied to new documents to help an archivist identify the PRA or FOIA restrictions.

At first glance, CBR may seem similar to the rule-induction algorithms of machine learning. Like a rule-induction algorithm, CBR starts with a set of cases or training examples; it forms generalizations of these examples, albeit implicit ones, by identifying commonalities between a retrieved case and the target problem. The key difference, however, between the implicit generalization in CBR and the generalization in rule induction lies in when the generalization is made. A rule-induction algorithm draws its generalizations from a set of training examples before the target problem is even known; that is, it performs eager generalization. The difficulty for the rule-induction algorithm is in anticipating the different directions in which it should attempt to generalize its training examples. This is in contrast to CBR, which delays (implicit) generalization of its cases until testing time -- a strategy of lazy generalization. CBR therefore tends to be a good approach for rich, complex domains in which there are myriad ways to generalize a case.

We believe that rules learned by induction, in combination with rules elicited from experts and cases applied through case-based reasoning can be applied in a hybrid system to produce a robust tool for interactive archival review.

Overview of the Current Prototype

Our focus in this early stage of development of the Access Restriction Checker has been on rule-based reasoning. We have built a framework for experimentation with rules that represent the well-understood knowledge of the restrictions that can be articulated by the archivists. It is hoped that the application of rule-based reasoning can cover a significant subset of the access restrictions so that obvious productivity increases can be seen for the

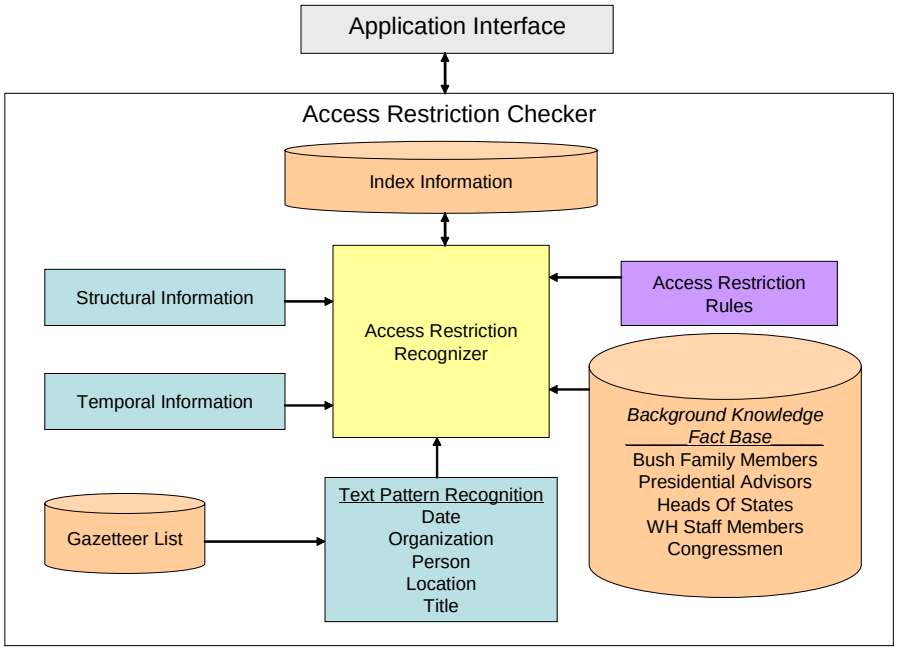
archivists. Later phases of the prototype development will tackle those areas where rule-based reasoning is less productive such as exception handling. In these areas we will apply case-based reasoning or another technique, if appropriate.

The prototype extracts named entities from a presidential record, identifies record or document type, determines the primary communication action of the document, and assents this information into a working memory so that the system can then reason with them, applies a set of rules to determine if the document or passages thereof might have an access restriction, and displays the results to an archivist.

As it currently exists, our framework allows for the creation of alternate experimental rule sets for identification of PRA and FOIA restrictions. The rule-based approach will be combined with case-based reasoning and machine learning to provide a more robust, hybrid approach to aiding the archivists.

There are a number of tools that can be used to extract information from documents. Many of these tools are optimized for specific tasks such as recognizing symbols within text as entities or processing knowledge. We have integrated two such tools to create our platform for access restriction checking. The first tool is Gate/Annie [Cunningham 2003]. Gate/Annie is an information extraction tool that locates in the text the names of persons, places, or things. After these entities are located we pass that information into Jess (Java Expert System Shell). Jess is a rule-based production system that can be embedded into a java program to allow that program to reason on facts to create new knowledge. This reasoning is formulated through rules that are specified by the programmer that are activated when the conditions for a rule are satisfied [Friedman-Hill 2004]. The prototype takes the pieces of information extracted from Gate/Annie and asserts them as facts into Jess. Jess then takes these pieces of information and reasons about them, asserting new information into the fact base.

Architecture



The above figure shows the major components of the current prototype. Below is a brief description of each component:

Text Pattern Recognition using Gate/Annie

Gate/Annie is used to extract information from the incoming documents. It annotates the named entities such as persons and locations and sends the annotated information back the main program.

Access Restriction Rules using the Jess Rules Engine

In this module, rules are executed to decide if a particular document has a restriction. These rules are written in the Jess language and execute in the Jess interpreter. This module takes in as input all of the annotations from Gate/Annie and the Segmentation Engine reasons over this information and returns restriction annotations to the main program if there are restrictions found.

Background Knowledge and Fact Base

The fact base consists of all of the background knowledge needed in identifying restrictions. This includes cabinet member names, white house staff mames, etc. The fact base currently resides in the Jess Working Memory. As we begin to scale the prototype to deal with large quantities of data and more complete knowledge about the state of the world referred to in the documents, we will use a database to represent the long term

information. Intermediate results will still be stored in the Jess Working Memory. Eventually the Background Knowledge or Fact Base will add considerable power to the reasoning capability of the system, include the ability to reason temporally and spatially about the context provided by world and political events, heads of state, members of the administration, members of the US government, and presidential friends and family members.

Main Application

The main application is the main control over all of the modules. The main application passes data between modules and determines what to do with the output.

Application User Interface

The user interface (or UI) is what the archivist sees and uses to interact with the system. It wraps all of the control features of the main application into a set of button clicks.

Gate/Annie

Gate is a platform for doing information extraction. Developers can customize execution phases called processing resources to extract the information within the text that the developers want. Currently the set of processing resources that we are using constitute Annie (or sometimes referred to as Default Annie). Annie contains six processing resources that execute as a pipeline of sequentially executing processes; the output of one flows as the input into the other. The table below is a list of the processing resources in Annie in execution order (top to bottom) and a brief description on what each resource does.

Name	**Description**
Annie English Tokeniser	It separates all of the words within the document.
Annie Gazetteer	The gazetteer consists of lists of different entities such as first names, last names, locations, job titles, etc… During this phase all of the words that were found in the tokeniser are cross referenced within the gazetteer lists. If the words are found in the gazetteer list they are given certain attributes for later processing.
Annie Sentence Splitter	The sentence splitter takes the document and tries to ascertain where sentences begin and end.
Annie Part of Speech Tagger	This processing resource gives parts of speech labels to each word within the sentences passed to it by the sentence splitter.
Annie Named Entity Transducer	This resource is a set of rules for how to extract information from the text. For example, if a text string, *John*, is found within the first name gazetteer list, a **Lookup** annotation will be created with the attribute of **firstperson**. Rules in the transducer are

	defined that annotate this string of text up as a person. There are also rules that will try to see if *John* is followed by another word with a capital letter. If *John* is followed by a word with a word starting with a capital letter, let's say *Doe*, then the entire section *John Doe* will be resolved to a person.
Annie OrthoMatcher	This resource tries to locate similarities within the annotations. For example, if *John Doe* is found in the text and marked up as a person in a previous resource and *John* elsewhere in the text is written then it tries to relate the former occurrences of *John* with the previous *John Doe*.

We have currently only made slight modifications to two resources within Annie, the Annie Gazetteer and the Annie Named Entity Transducer. We have added names, and position titles to the default gazetteer set and have added a few rules for extracting this information in the named entity transducer.

Java Expert System Shell

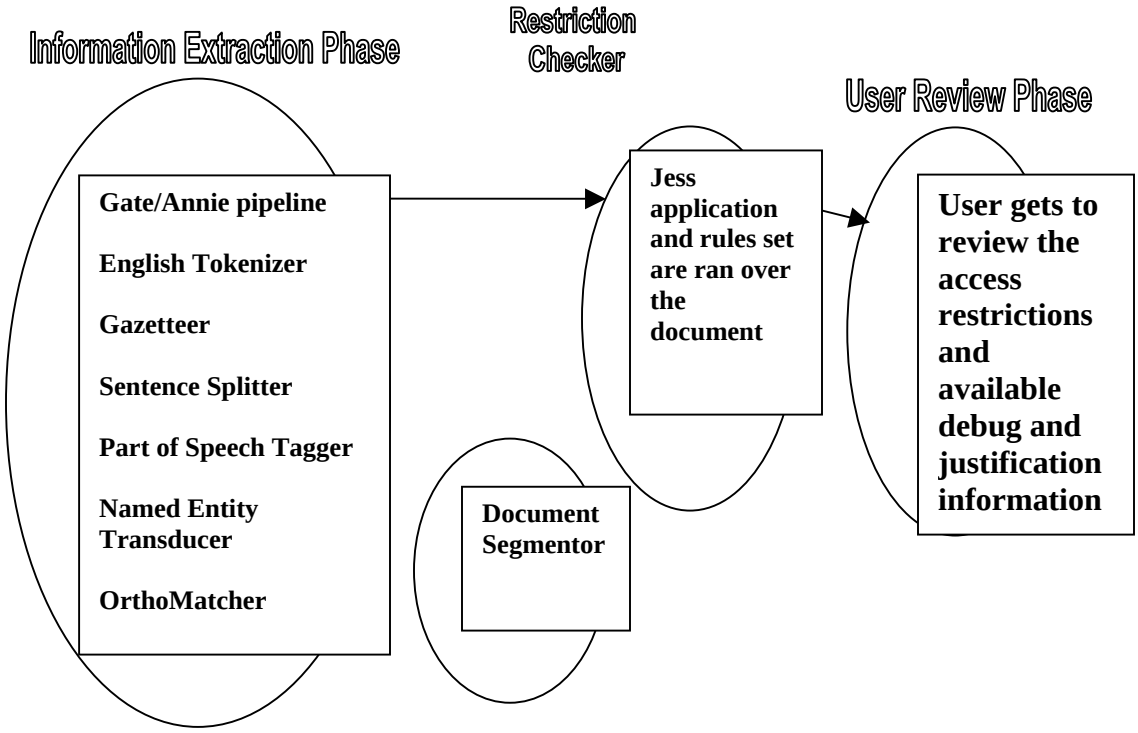
Jess is an embedded production system API accessible from java. A Production system model is a way of expressing a computer program as a set of "If X Then Y" rules or production rules. These rules are not run in sequential order; when conditions X are met, instructions Y are executed. The knowledge use to determine restrictions within a document is encoded as production rules. The Jess inference engine provides facilities for reasoning over facts, and executing of rules.

The Process

The system process consists of five phases: **Information Extraction**, **Document Type Classification**, **Communication Act Identification**, **Restriction Checker** and the **User Review Phase**. In the **Information Extraction phase** the document is passed through the Gate/Annie pipeline. The output after running the Gate/Annie pipeline is a set of annotations (person names, location names, organization names, etc…) giving the starting and ending positions of the names within the text [Underwood 2004]. These annotations along with the document's structure information such as text segmentation are passed to a **Document Type Classifier** that determines the documentary form (record type) of the document, e.g., memoranda, letter, agenda, press release [Harris and Underwood 2005]. The annotated, segmented document is passed with its document type to the **Communication Act Identifier** that uses the document type and annotations to determine the communication act, the participants in the act, its purpose and its propositional content. These are represented in a Jess template. The annotated document, the document type, and the communication act that the document represents in input into the **Restriction Checker.**

In the next phase, all of the annotations found in the **Information Extraction Phase** are asserted as facts into the Jess rule engine. We run the Jess application to reason over the collected information to see if there are any restrictions within the document. If there are restrictions the prototype annotates them in the text and passes them back to the GUI application starting the **User Review Phase**. While these annotations have starting and ending positions within the text, they also contain information such as what Jess rules were fired, and explanation as to why the identified part of a text restricted.

The **User Review Phase** allows the user to view all of the information that the **Restriction Checker** outputs. Currently, the user can just view the information, not update it or correct it.



Graphical User Interface for Archival Review

To illustrate the intended use of the tool being developed we present the following scenarios of use for discussion and envisioning of the evolving tool.

The prototype Access Restriction Checker is designed for two modes of use:

- Use by researchers to edit and experiment with rules for helping archivists identify documents with specific kinds of access restrictions

This mode of use is meant to aid the researcher in writing, editing and experimenting with rules that will successfully identify particular kinds of restrictions in documents in the experimental corpus.

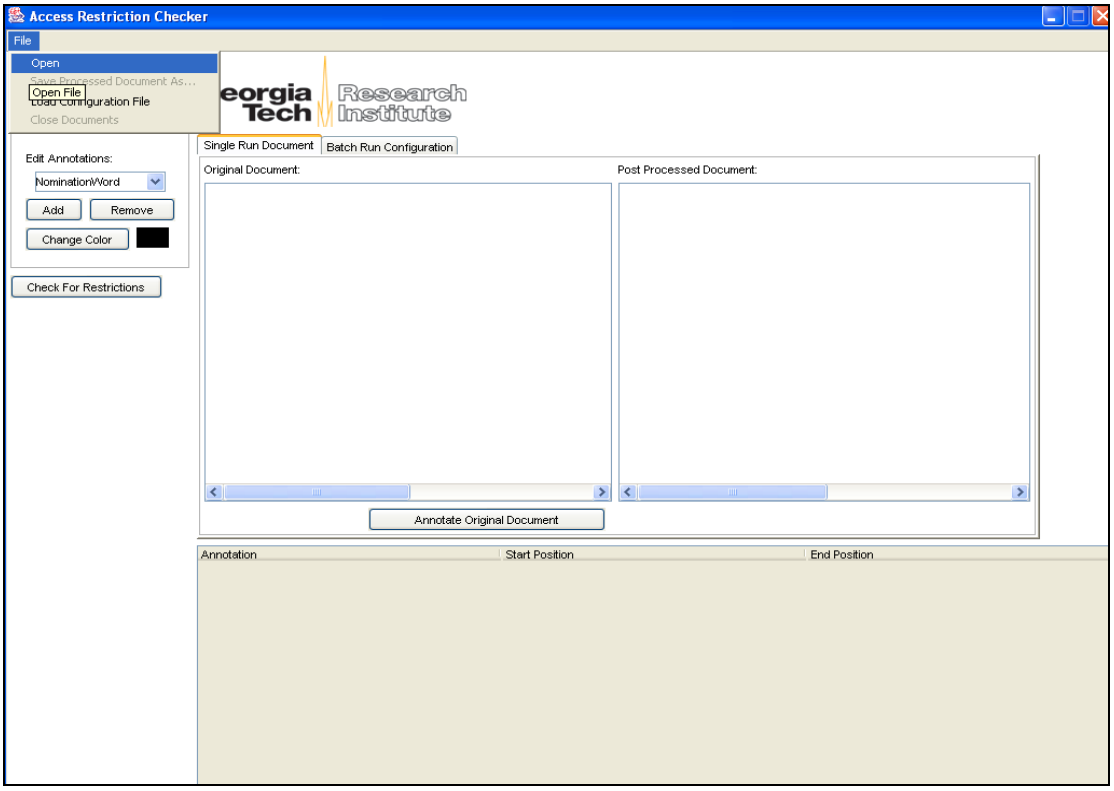
- Use by archivist in identifying documents that should be closed to public access or that should be redacted because of FOIA exemptions or PRA restrictions.

This mode is designed to work in collaboration with the archivist. The intent is for the system to attempt to identify quickly (by an automated process) documents which may contain FOIA or PRA restrictions. The archivist can then quickly scan the identified portions and their associated potential restriction types. If the archivist agrees with the identified restriction, he or she authorizes the appropriate action. Otherwise the archivist attaches an annotation to the document or passage indicating that they disagree with the access restriction and the archivist takes the appropriate action.

Single Document Scenario for Archivist Review

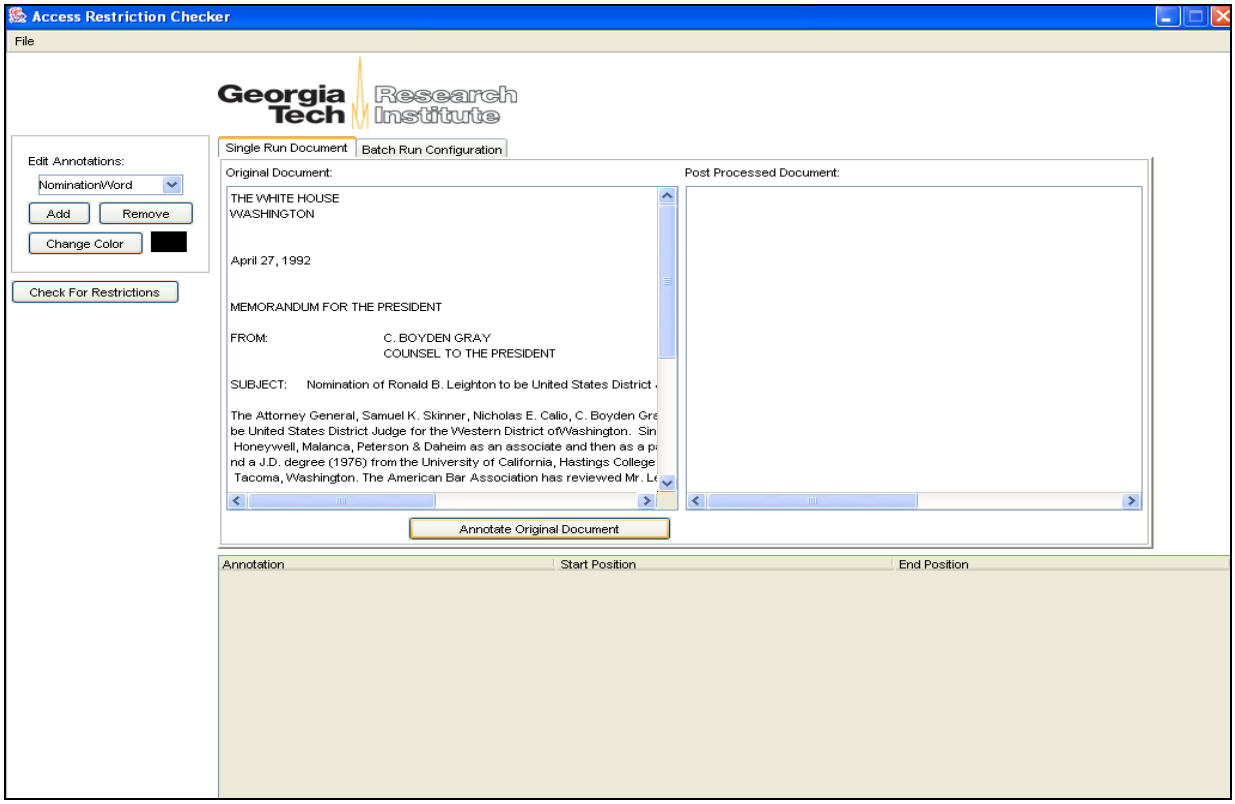
Step One

Application loads into Single Document mode. You must first load a document to annotate. You can do this by selecting File→Open. This will bring up a file dialog that will allow you to select a file.

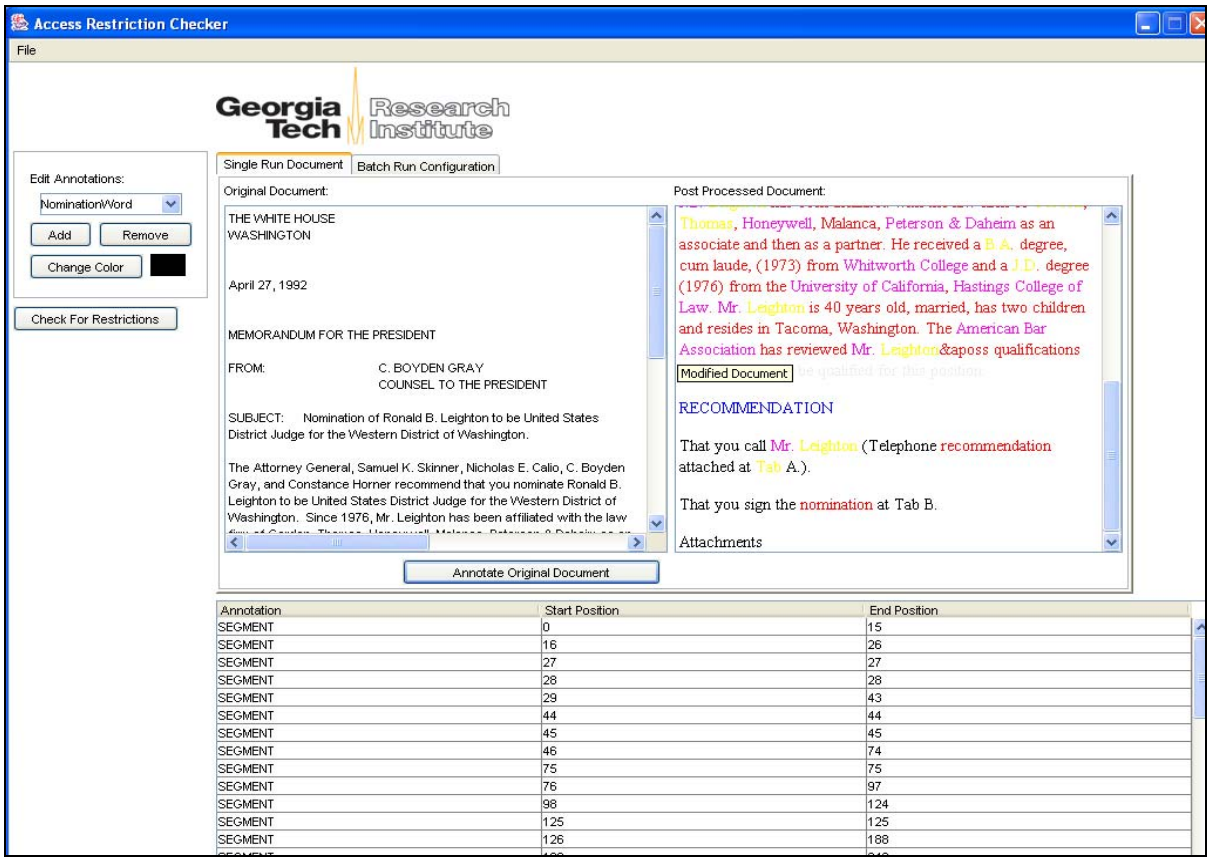


Step Two

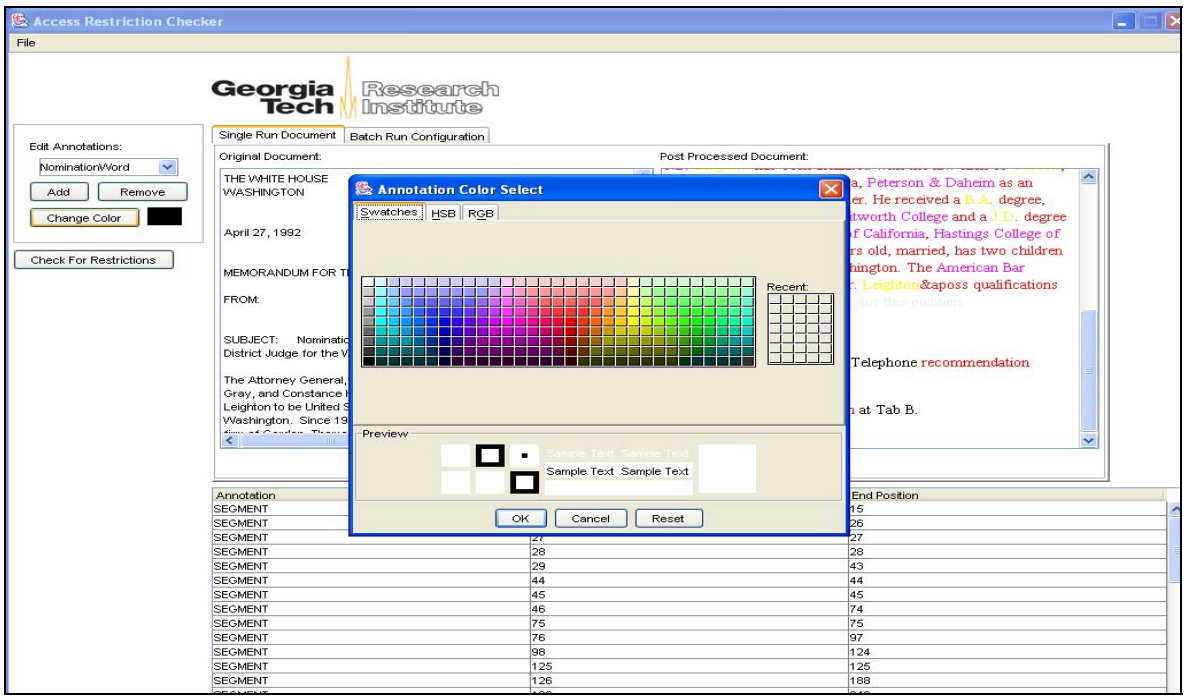
After selecting your file it should load in the Original Document into a text window as illustrated in the figure below. If the file format is not html, xml, or plain text document, it is converted to a plain text document.



To identify and annotate the document with named entities, click on the *Annotate Original Document* button. The Post Processed Document area will load the annotated version of the document as shown in the next figure.

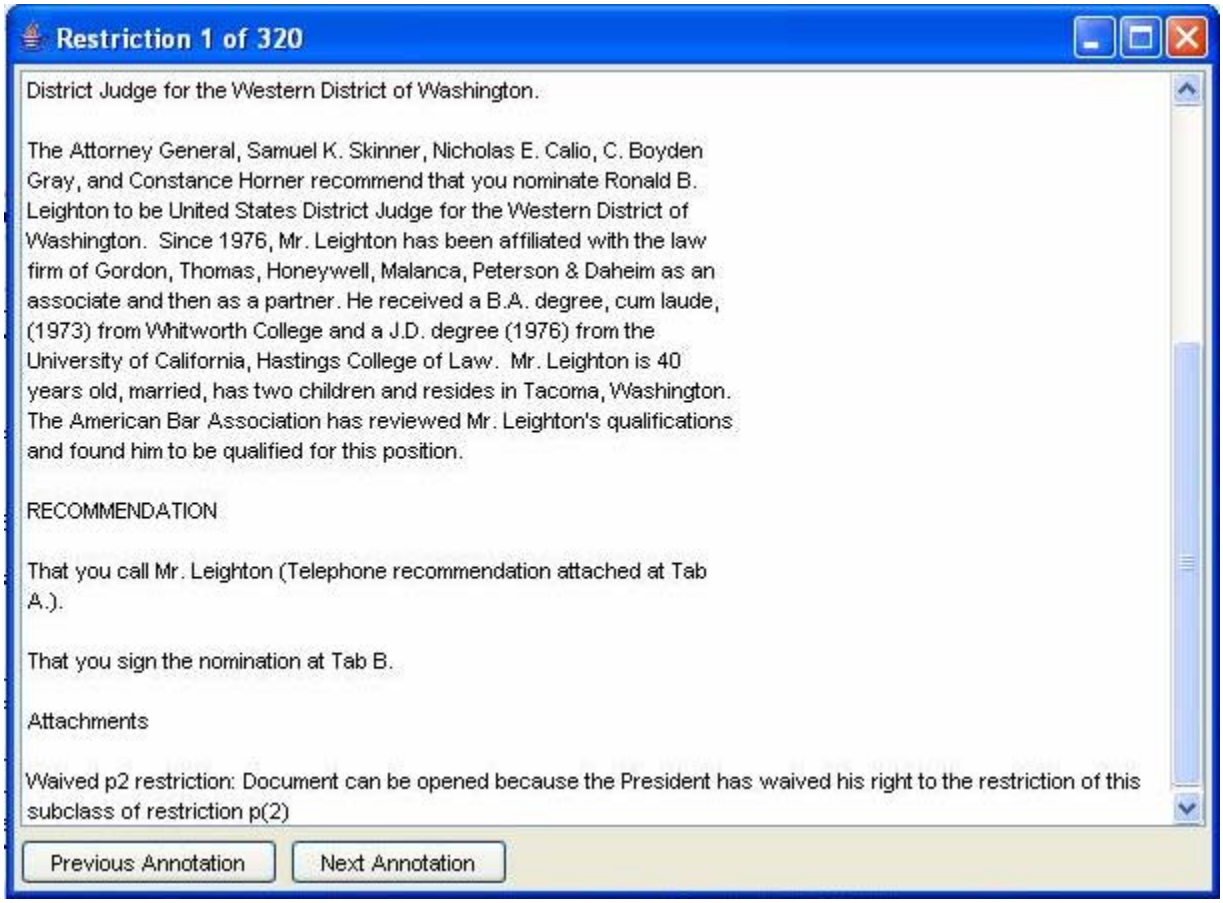


Different entity types are identified by different color text. The colors are defined in the properties file that is loaded but they can be changed in the edit annotations box on the left of the application. The figure below shows a color change in progress.



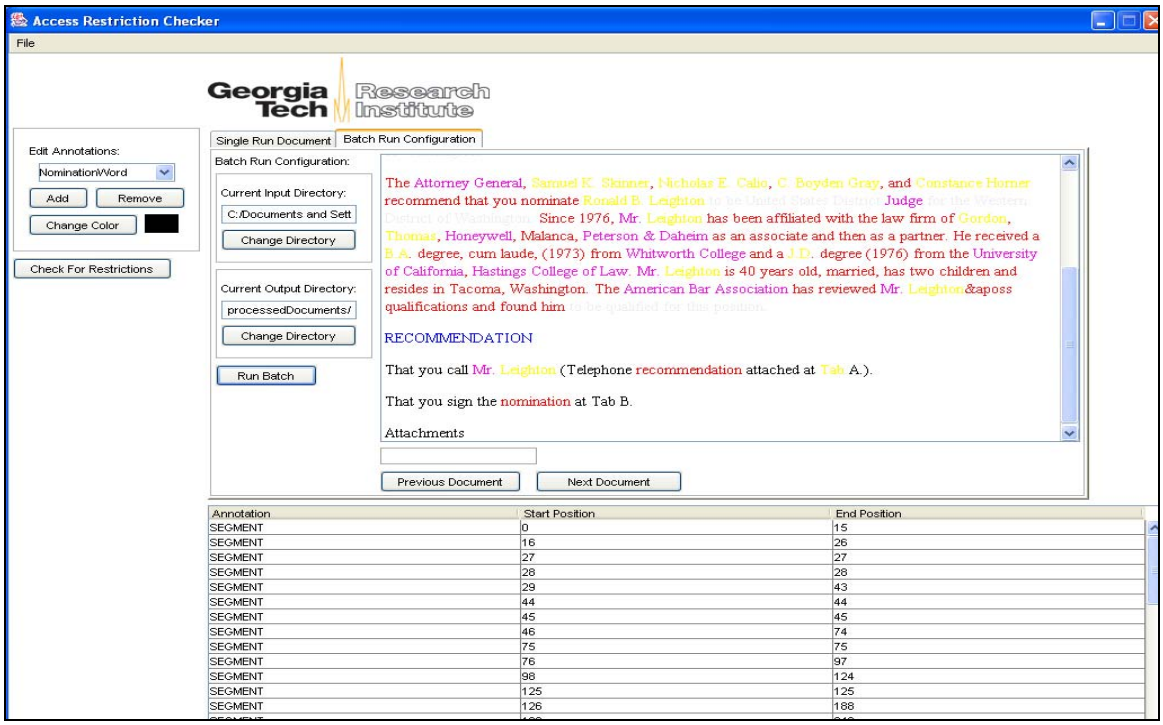
Step Three

The next step is to check for restrictions. The document must be loaded and annotated before this can be accomplished. To check for restrictions click on the *Check For Restrictions* button. The tool will identify any restrictions that satisfy it restriction rules. If there are restrictions identified, another window will pop up that will allow you to cycle through the recommended restrictions on the document. An example of the restriction window is shown in the figure below. The possible restriction is indicated at the bottom of the window. In this case the restriction has actually been waived by the President.



Batch Run Mode for Archivist Review

The only difference between Batch Run Mode and Single document mode is that the program takes an input directory and annotates all of the documents in that directory upon selecting the *Run Batch* button. The annotated documents are placed in the specified output directory and the first file in the directory is loaded onto the screen.



Users can cycle through the documents in the corpus by selecting the *Previous Document* or *Next Document* buttons. Restrictions can be checked in a similar manner to that of the Single Document mode by selecting the *Check For Restrictions* button.

Development of Decision Rules for Access Restrictions

We are using the prototype Access Restriction Checker to develop and test decision rules for distinguishing Personal Record Misfiles(PRMs) from Presidential Records and to recognize Freedom of Information Act (FOIA) exemptions and Presidential Record Act (PRA) restrictions [Underwood and Harris 2005]. About 150 sample Presidential Records previously classified as PRMs, and open or restricted presidential records are being analyzed to develop the decision rules [Underwood & Hayslet-Keck 2004].

In addition to decision rules, other kinds of knowledge are needed in interpret records, and to reason about the semantic relationships of concepts. There is the knowledge needed to interpret the record to determine what action it represents, who the participants in the action are, the purpose of the communication act, and the propositional content of the act. The communication (speech) acts include resignation, appointment, nomination, advice, recommendation, requesting, briefing, reporting and many other human actions that are carried out in presidential records [Underwood and Harris 2005]

Second, there is the knowledge needed to determine what is entailed by the propositional content of the communication act represented by the record. Third, there is the knowledge needed to determine semantic relations such as synonymy (the semantic relation that holds between two words that can express the same meaning, and meronymy

(the part to whole relation). Fourth, there is the knowledge and reasoning capability to a determine one concept (noun, noun phrase, verb, verb phrase) subsumes another concept. Conversely, that a concept is subsumed (is a specialization) of another [Underwood and Harris 2005].

In the following paragraphs, examples of decision rules for distinguishing Personal Record Misfiles from Presidential records, and for recognizing some subclasses of PRA restrictions P2 and P5 are given.

Example of Decision Rules for Recognizing a PRA Restriction P5, Confidential Advice

If the author of the record is the President and the addressee is a presidential advisor, or the author of record is a presidential advisor and the addressee is the President, then the record is a communication between the President and an advisor.

If the author of the record is a presidential advisor and the addressee is a presidential advisor, then the record is a communication between presidential advisors.

If record is a communication between the President and a presidential advisor, or the record is a communication between presidential advisors, and the purpose of the communication is a request (for action, information) or an order, and the content involves Domestic Economic Policy issues, then access is restricted under PRA a(5).

Domestic Economic Policy addresses economic growth and tax revenues. Fiscal and Monetary policy is a part of Domestic Economic policy and addresses the budget, especially taxation and borrowing. This knowledge is not represented as decision rules, but as rules such as the following:

```
domestic_economic_policy_issue(X), if equal(X, "economic growth") or
                                       subsumes("economic growth", X) or
                                       equal(X, "tax revenues"), or
                                       subsumes("tax revenues", X), or
                                       fiscal_and_monetary_policy_issue(X).

fiscal_and_monetary_policy_issue(X), if equal(X, "federal budget"), or
                                        subsumes("federal budget", X), or
                                        equal(X, "taxation"), or
                                        subsumes("taxation", X), or
                                        equal(X, "federal borrowing"), or
                                        subsumes("federal borrowing", X).
```

The decision rules are represented in Jess as follows.

```
(defrule p5r1
        (communication_act
```

```
                (author ?person&:(= ?person ?presidentID))
                (addressee ?person_id &:(presidential_advisor ?to_person_id))
        )
        =>
        (assert communication_between_president_and_advisor)
        (printout t " communication between president and advisor")
)

(defrule p5r1a
        (communication_act
                (author ?person&:(=?person ?presidental_advisor)
                (addressee ?person &:(=?person ?presidentID)
        )
=>
        (assert communication_between_president_and_advisor)
        (printout t "communication between president and advisor")
)

(defrule p5r2
        (communication_act
                (author ?person&:(= ?person ?presidential_advisor))
                (addressee ?person_id &:(=?presidential_advisor ?to_person_id))
        )
=>
        (assert communication_between_presidential_advisors)
        (printout t "communication between presidential advisors")
)

(defrule p5r3
        "Confidential advice on domestic economic policy issues"
        (communication_between_president_and_advisors)
        (communication_act
                (purpose "directive")
                (content domestic_economic_policy_issue)
        )
=>
        (assert (review_class (type P5) (rule p5r3) (waived (is_waived P5r3)))
        (printout t " review_class type p5r3")
)
(defrule p5r3a
        "Confidential advice on domestic economic policy issue"
        (communication_between_presidential_advisors)
        (communication_act
                (purpose "directive")
                (content domestic_economic_policy_issue)
        )
```

```
=>
        (assert (review_class (type P5) (rule p5r3a) (is_waived P5r3a)))
        (printout t " review_class type p5R1")
)
```

Example of Decision Rules for Recognizing Personal Record Misfiles

If the record is addressed to the President or the First Lady, and is from a person who is a member of the Republican National Committee (RNC), or the record is addressed to a person who is a member of the RNC and is from the President or First Lady, then the record is a communication between the President or First Lady and the RNC.

If the record is a communication between the President or First Lady and the RNC, and is about political issues, then the document is a PRM because it is personal/political.

These rules are expressed in Jess as follows.

```
(defrule prmr1
        (communication_act

        ((author ?person&:(=?person ?rnc_staff_member))
        (addressee ?person&: ((=?person ?presidentID)
        |(=?person ?firstLadyId)))
         |
        ((author ?person&: ((=?person ?presidentID) | (=?person ?firstLadyId)))
        (addressee ?person&: (=?person ?rnc_staff_member)))
        )
        =>
        (assert communication_between_president_or_first_lady_and_rnc)
        (printout t "Communication between President or First Lady and RNC")
)

(defrule prmr2
        (communication_between_president_or_first_lady_and_rnc)
        (communication_act
                (content political_issue)
        =>
        (assert review_class (type PRM) (rule prmr1))
        (printout t "review_class type PRM PRMR1")
)
```

Example of Decision Rules for Recognizing PRA Restriction P2, Appointments to Federal Office

If record is a decision memo, and addressed to President by one of his advisors, and communication act of the memo is a recommendation that the President take an action,

and that action is to sign a nomination of person to a federal office, and that person was factually nominated to that office, then the records review class is P2 (3a).

If record is restricted under P2 (3a), then records may be opened because President waived his restriction rights to this subclass of records.

```
(defrule p2r1
        "
        (communication_act
                (act recommend)
                (addressee ?person&:(= ?person ?presidentID))
                (purpose "directive")
                (content president_sign_nomination)
        )
=>
        (assert (review_class (type P2) (rule p2r1) (waived (is_waived p2r1)))
        (printout t " review_class type p2r1")
)
```

Summary and Future Research

In this paper we have describes our initial development of the Access Restriction Checker. Our technical emphasis has been on the application of rule-based reasoning. We have built a framework for experimentation with rules that represent the well-understood knowledge of the restrictions that can be articulated by the archivists. Even though we know that the application of rule-based reasoning can cover a significant subset of the access restrictions, we know that there are both technical and practical limitations to this approach. Later phases of the prototype development will need to tackle those areas where rule-based reasoning is less productive such as exception handling. In these areas we will apply case-based reasoning or another technique, if appropriate.

The prototype illustrates the overall process: the extraction of named entities from a presidential record, identifies types, determines the communication act of the document and asserts this information into a working memory so that the system can then reason with the information about the document.

After testing the "Access Restriction Checker" on the "Test Corpus", experiments will be conducted at the Bush Presidential Library with actual Presidential Records and Personal Record Misfiles from the Bush Administrations personal computer files. The performance of the "Access Restriction Checker" will be assessed for each of the categories that it is engineered to check.

References

[Aamodt and Plaza 1994] A. Aamodt & E. Plaza. Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. *AI Communications*, 7(i), 1994, pp 39-59.

[Cunningham 2003] H. Cunningham, D. Maynard, K. Bontcheva, V. Tablan, C. Ursu and M. Dimitrov. Developing Language Processing Components with GATE: A User Guide. Department of Computer Science, University of Sheffield. February 2003. PDF

[Friedman-Hill 2004] E. J. Friedman-Hill. Jess, The Rule Engine for the Java Platform, Version 6.1p7. SAND98-8206 (revised), Sandia National Laboratory, Livermore, CA, 7 May 2004. http://herzberg.ca.sandia.gov/jess/docs/61/

[Harris and Underwood 2004] B. Harris and W. Underwood. Factual Knowledge Needed for Information Extraction and FOIA Review, PERPOS Working Paper 04-7, December.

[Harris and Underwood 2005].B. Harris and W. Underwood, Learning and Classifying Document Types. PERPOS Working paper 05-8, June 2005.

[Kolodner 1993] J. L. Kolodner,. Case-Based Reasoning. Morgan Kaufmann, 1993

[Underwood 2004] M. G. Underwood. Recognizing Named Entities in Presidential Electronic Records**,** PERPOS Technical Report ITTL/CISTD 04-4, June, 2004 (Revised Nov 2004).

[Underwood and Harris 2005] W. E. Underwood and B. Harris The Knowledge and Reasoning Required to Recognize Presidential Record Act Restrictions and Personal Record Misfiles. PERPOS Working Paper 05-3, ITTL/CSITD, Georgia Tech Research Institute, 2005.

[Underwood and Hayslett-Keck 2004] William Underwood, Marlit Hayslett-Keck. A Corpus of Presidential, Federal and Personal Records for use in Information Extraction, Description and FOIA/PRA Review Experiments, PERPOS Technical Report 04-5. CSITD/ITTL/GTRI, June 2004.

[Underwood et al 2005] W. E. Underwood, M. Hayslett-Keck and S. Laib. The PERPOS Tools: User's Guide (Version 3.0) PERPOS Technical Report ITTL/CSITD 05-02, Revised March, 2005.