Georgia Tech Research Institute

# Advanced Decision Support for Archival Processing of Presidential Electronic Records: Annual Technical Status Report (July 2007- June 2008)

William Underwood
Sheila Isbell
Sandra Laib
Matthew Underwood

Technical Report ITTL/CSITD 08-06

August 2008

Computer Science and Information Technology Division
Information Technology and Telecommunications Laboratory
Georgia Tech Research Institute
Georgia Institute of Technology

The Army Research Laboratory (ARL) and the National Archives and Records Administration (NARA) sponsor this research under Army Research Office Cooperative Agreement W911NF-06-2-0050. The findings in this paper should not be construed as an official ARL or NARA position unless so indicated by other authorized documentation.

## ABSTRACT

The performance of the previously developed Information Extraction tool has been improved by the inclusion of additional wordlists and JAPE rules. Additional semantic categories such as facility names, legislative bills and statutes, names of governments, and relative temporal expressions are now annotated.

Speech acts are acts of speech or writing in which one does something just by saying something, e.g., "I appoint you…", "I hereby proclaim…". One hundred eighteen Presidential records were analyzed with regard to the expression of speech acts with performative verbs and speech acts about the author's past or future speech acts or other's speech acts. More than 60 kinds of speech acts were discovered in the corpus. The analysis confirms that performative verbs are used to express the actions carried out by records. A method has been formulated for identifying the speech acts occurring in e-records. It will be implemented, tested using records from the analyzed corpus and then experimentally evaluated

A corpus of 50 presidential records was analyzed to determine the topic(s) of a record. An approach to identifying the topics of an e-record has been formulated. The approach is an extension of the methods for recognizing document types and speech acts.

A method for automatic document type recognition has been implemented and successfully tested. The method is based on the method for automatically annotating semantic categories such as person's names, dates, and postal addresses. Currently, it extends this method by (1) identifying about 80 types of intellectual elements of documents, (2) parsing these elements using context-free grammars defining the documentary form of document types, (3) interpreting the meaning of the form of the document to identify some or all of the following: the chronological date, author(s), addressee(s), and topic.

Progress in implementing the Access Restriction Checker includes the interface of GATE and PROLOG to the Java Expert System Shell (JESS). GATE includes the resources for document type recognition and extraction of facts about a record such as author, addressee, chronological date and sometimes topic. PROLOG includes the facts and rules representing background knowledge. Additional rules have been constructed for identifying possible access restrictions. Still needed are the capabilities for speech act and topic recognition. Methods have been formulated for providing these capabilities and plans are to implement them this coming year.

Due to the rapid changes in computer technology, archivists must be concerned not only with the obsolescence of e-record file formats, but with the obsolescence of the operating systems, database management systems and integrated development environments of their Archival Repository and Archival Processing System. These issues are investigated and discussed using the Presidential Electronic Records Pilot System (PERPOS) as a case in point.

# TABLE OF CONTENTS

# 1. Introduction

## *1.1 Background*

Knowledge of the action conveyed by a record and of what a record is about — its topic(s) — is knowledge that is essential to determining whether a record is a personal misfiled record or a Presidential record that might have restrictions on its disclosure. Knowledge of the topics of a record is also required for archival description of individual records, folders of records or series of folders containing records. Automatically derived knowledge of the topics of e-records also has potential to significantly improve the precision and recall of records from massive record collections that are retrieved in response to Freedom of Information Act (FOIA) requests and in legal discovery.

Document types have a role in archival description and review. Archival descriptions include the names of the types of documents that occur in a record series, for example, correspondence, memoranda or agenda. Also, knowing a documents type helps in understanding the action communicated by a document. Knowing document type can help in discriminating personal records from Presidential records. It can also help in determining PRA restrictions. The ability to recognize document type also contributes to determining records relevant to a FOIA request, for example, one might be able to request memoranda to the President from the Chief of Staff between certain dates.

Presidential e-records must be reviewed for Presidential Records Act restrictions and FOIA exemptions before they can be disclosed to the public. Decision support is needed in this intellectually demanding task.

One of the challenges that archivists face in preserving electronic records is that the operating system and programming language technologies that support an application for processing e-records are changing so rapidly. To maintain the viability of an archival processing system, it must be possible to cost effectively migrate archival processing software to new programming languages or development environments, new operating systems and database systems.

## *1.2 Purpose*

The research tasks for the second year of research were:

1. Develop Methods to Recognize the Actions and Topics of E-Records. Use in Metadata Extraction and Archival Description.

2. Extend Document Type Learning to Additional Document Types and Larger Sample Sizes. Develop a Method to Generate Item, File Unit and Record Series Descriptions.

3. Analyze Presidential Records to Determine Criteria for Determining PRA Restrictions and FOIA Exceptions.

4. Investigate Issues in Maintaining the Archival Research Prototype in the Face of Technological Change. Implement Security Measures Needed to Protect the Prototype Archival System.

The purpose of this report is to summarize research progress during the second year of this research project. In the next section, progress and results for each of the research tasks is described. The final section summarizes research progress.

# 2. Research Tasks

## *2.1 Recognition of Actions and Topics of E-Records for Metadata Extraction and Automatic Description*

This research task is to develop and evaluate methods to automatically recognize and annotate communication acts and topics of electronic records.

### 2.1.1 Improvements of Annotation of Semantic Categories in Records

Recognition of the actions and topics of e-records is dependent on the capability to recognize the proper names of persons and organizations who perform the actions or who may be the topic of a record. In prior research, a capability to recognize the names of persons, organizations, locations, dates and monetary amounts was demonstrated [Isbell et al 2006]. However, the performance of this method was not high enough to support the development of reliable methods for topic and action recognition. The performance of that method of semantic annotation was substantially enhanced by creation of additional and enlarged wordlists and additional and refined pattern annotation rules.

Tools for extracting files from containers, for converting the files to text formats, and for annotating their contents are installed on the PERPOS system in the Virtual Laboratory at Archives II. They were used in an experiment to evaluate the performance of the semantic annotation of actual, born digital, personal computer records from the Bush Administration.

A corpus of 50 records was selected from a record series accessioned into PERPOS. They are records from the Office of Legislative Affairs. The Office of Legislative Affairs provides advice and support regarding the President's legislative agenda and legislation is general, and liaison between the White House staff and members of Congress.

The experiment evaluated the performance of the semantic annotator with regard to the named entities addressed in the previous two experiments, namely, annotation of person,

location, and organization names, dates, money and percents. The results are shown in Figure 1.

| Annotation Type | Correct | Partially Correct | Missing | Spurious | Precision | Recall | F-Measure |
|---|---|---|---|---|---|---|---|
| Person | 515 | 11 | 42 | 57 | 0.8928 | 0.9164 | 0.9044 |
| Location | 270 | 15 | 54 | 24 | 0.8981 | 0.8186 | 0.8565 |
| Organization | 509 | 31 | 31 | 50 | 0.889 | 0.9186 | 0.9035 |
| Date | 456 | 1 | 1 | 1 | 0.9967 | 0.9967 | 0.9967 |
| Money | 28 | 1 | 0 | 8 | 0.7703 | 0.9828 | 0.8636 |
| Percent | 6 | 0 | 0 | 0 | 1.0 | 1.0 | 1.0 |

Overall average precision: 0.9178 Overall average recall: 0.9282 Overall average F-measure: 0.9108

**Figure 1. The Performance of Semantic Annotation on Corpus 3**

The eight examples that were spuriously annotated as money were all instances of "mark up" of legislation. The term "mark" was annotated as Money (German Mark). This error is easily repaired by a rule that differentiates "mark up" of legislation from the monetary term "Mark".

Most of the "Missing" annotations for locations were state abbreviations (OK, MA, ME, IN, VA) that appeared in parentheses after a legislator's name, indicating the state they represented. These missing location abbreviations were in a list of ambiguous state abbreviations. However, they can easily be disambiguated since they appear after a person's name and are in parentheses.

The table in Figure 2 shows:

1. The performance on Corpus 1 of the "vanilla" Semantic Annotator provided with the GATE distribution [Underwood 2004]
2. The performance on Corpus 2 of the Semantic Annotator after improvements made based on an analysis of Experiment 1 [Isbell et al 2006]
3. The performance on Corpus 3 of the Semantic Annotator after improvements to the wordlists and JAPE rules [Underwood and Isbell 2008b].

| | Experiment 1: Default ANNIE Corpus 1 | Experiment 2: GaTech IE Vers 1 Corpus 2 | Experiment 3: GaTech IE Vers 2 Corpus 3 |
|---|---|---|---|
| Person | 0.6768 | 0.7269 | 0.9044 |
| Location | 0.7926 | 0.7922 | 0.8565 |
| Organization | 0.6342 | 0.7182 | 0.9035 |
| Date | 0.8934 | 0.9471 | 0.9967 |
| Money | 1.0000 | 0.9286 | 0.8636 |
| Percent | 0.8182 | 0.9273 | 1.0000 |
| Overall Average F-measure | 0.7490 | 0.8316 | 0.9108 |

**Figure 2. Improvements in F-measure in Three Experiments**

In the three experiments, the performance has increased in all cases with the exception of the annotation of money terms in Experiment 3. As previously explained, in experiment 3, the decrease in performance in annotation of money terms is due to the incorrect annotation of “mark” in the “ mark up” of legislation.

The current performance is very good. Without this level of performance, methods for speech act, topic and document type recognition, which are dependent on the semantic annotation method, cannot achieve a high-level of performance. However modifications will be made to correct the annotations in Experiment 3 that were missed, partially correct and spurious.

### 2.1.2 Actions and Archives

According to the UBC Project, to be a record of an activity, a record must be associated with some action [Duranti et al 1997]. The record must:

(1) Carry out an action, e.g., an advertisement for a job, an offer of employment, and acceptance of employment (a dispositive record), or
(2) Provide evidence of an action already carried out, e.g., report of a job interview (a probative record), or
(3) Provide information on which to base action, e.g., an applicant's curriculum vita (a supporting record).

Each of these types of record carries out a communication act. The dispositive records, by definition, carry out an act — advertise, offer, accept. The probative records, e.g., a report of a job interview, carry out the action of reporting an action already carried out. A supporting record, e.g., an applicant's curriculum vita, is a *statement* of facts, and a statement of fact carries out the action of stating facts.

### 2.1.3 Speech Acts

Philosophers of Language [e.g., Austin 1962] and Linguists [e.g., Sadoc 1972], have studied communication acts in their spoken rather than written forms, and have developed a theory of Speech Acts. *Performative verbs* are verbs whose action is accomplished merely by saying them or writing them, e.g., "I promise, "I recommend," "I advise," I nominate," or "I appoint." English is believed to have around 250-300 performative verbs. Vanderveken [1990] defines the semantics of 271 performative verbs. Wierzbicka [1987] defines the semantics of about 230 speech act verbs, not all performative.

A speech act can be represented by the *name of the speech act*, a *speaker* (author), who is the utterer (writer) of a message and a hearer (*addressee*) who is any of the immediate intended recipients of the speaker's (writer's) communication, the *propositional content* that consists of a subject and a predicate which expresses something about the subject, and its *illocutionary force* (purpose). According to Searle’s taxonomy of elementary

illocutionary acts, there are only five *illocutionary points* that speakers can attempt to achieve in expressing a propositional content with an illocutionary force. These are the *assertive, commissive, directive, declarative* and *expressive* illocutionary points [Searle 1969, 1979].

The speech act research literature was reviewed with specific focus on identification of speech acts in spoken dialog and text. Holdcroft and Smith [1992] discuss the computational problem of identifying speech acts. Rose et al [1995] discuss patterns for identifying thirteen speech acts occurring in dialogues.

### 2.1.4 Analysis of Speech Acts Expressed in Presidential E-Records

A corpus of 118 copies of Presidential records was analyzed to determine the occurrence of (1) speech acts expressed with performative verbs, (2) speech acts not expressed with performative verbs, and (3) speech acts that are the subject or object of speech acts or are in past or future tense [Underwood and Isbell 2008a]. All 118 records express speech acts. 83 of the records use one or more of 46 performative verbs. Those performative verbs are shown in the list below. In the report of the analysis, definitions and examples of the use of the verb in the corpus is provided.

| | | | |
|---|---|---|---|
| accept | commit | nominate | revoke |
| acknowledge | confirm | offer | salute |
| advise | congratulate | order(1) | suggest(2) |
| agree | declare | pledge | tell |
| amend | delegate | proclaim | tender |
| announce | determine | prohibit | terminate |
| appoint | direct | propose | thank |
| appreciate | encourage | recommend | urge |
| ask(1) | endorse | report | veto |
| ask(2) | forecast | request | welcome |
| authorize | inform | resign | |
| certify | invite | retire | |

The 35 records that did not use performative verbs includes speech acts such as asking questions, requesting actions, making recommendations, directing someone to do something, and informing someone of some facts. Section headings are one of the structural features that can be used to recognize these speech acts.

Many of the records in the corpus contain sentences in which speech acts are the subject or object of the sentence or performative verbs in the past or future tense. These speech acts include most of the performative verbs discovered in their performative use and shown in the table above. Additional performative verbs that were discovered include the following.

| | | | |
|---|---|---|---|
| apologize | convene | predict | stipulate |
| claim | define | rule | warn |
| contend | estimate | state | |

This analysis of Presidential records confirms the role of speech acts, and in particular performative verbs, in expressing the action carried out by records. Even when the performative verbs are not used performatively, but are in the past or future tense or in nominalized forms as the subject or object of sentences, they often represent the subject matter or topic of the record.

### 2.1.5 Method for Recognizing the Action Carried Out by a Record

An approach has been formulated for automatically recognizing the action carried out by a record, that is, to recognize the speech (or communication) act performed by the author (writer) of the record [Underwood and Isbell 2008b]. The approach is based on an extension of the methods for annotating the named entities in records [Underwood and Isbell 2008a] and for recognizing the document type of a record (see section 2.2).

The approach to annotating the speech acts of the writer of a document adds additional GATE processor resources [Cunningham et al 2007] (shown underlined in the list below) and resources that we construct (shown in italics).

Tokenizer
Wordlist Lookup
Sentence splitter
Hepple POS Tagger + Lexicon
JAPE transducer + Rules for Annotating Named Entities
JAPE Transducer + Rules for Annotating Intellectual Elements
SUPPLE Parser + Document Type Grammars
Morphological Analyzer
SUPPLE Parser +English Grammar
Orthomatcher
*Pronominal Coreferencer*
*Speech act Transducer*
*Extract Record's Communicative Action*

The Hepple Part of Speech (POS) Tagger uses a lexicon of approximately 17,800 words. The lexicon associates parts-of-speech with each of these words, and the Hepple POS Tagger selects a part-of-speech for the particular words in a sentence based on transition probabilities between parts of speech. There are many performative verbs that are not in the lexicon provided with the vanilla version of GATE. These verbs are being added to the lexicon.

The part of speech, e.g., verb, is a syntactic feature of words. That a verb is a performative verb is a semantic/pragmatic feature of verbs. The Performative verb

Tagger will add the semantic feature PERFVB to those verbs recognized by the Hepple Tagger that are also performative verbs.

The morphological analyzer provided with GATE is used by the SUPPLE parser, which is needed in recognizing and annotating syntactic patterns containing performative verbs indicating speech acts of the author of a record.

#### 2.1.5.1 Pronominal Coreference Resolution

Reference to an entity already mentioned in text, most often with a pronoun or a different name, is called *anaphora*. The reference that points back to some entity is called the *anaphor* while the entity it refers to is called the *antecedent.*

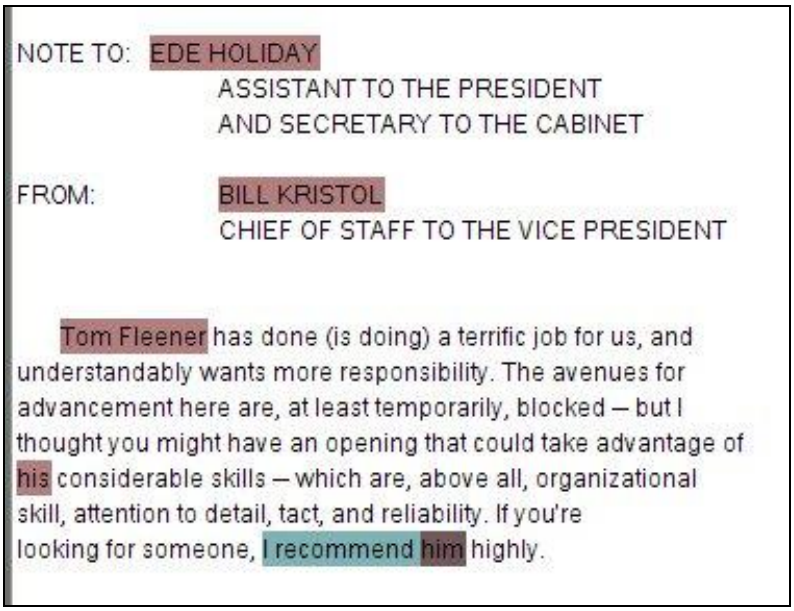
The process of finding the proper antecedent for each anaphor in text is called *anaphora resolution*. In the case all anaphors that refer to the same entity are to be found, the process is called coreference resolution. There are different kinds of coreference, e.g., pronominal, proper names, apposition, part-whole, but not all of them are equally important for the speech act recognition task.

Pronominal coreference is the most common type of coreference. It includes finding the proper antecedent for the following types of pronouns:

personal: I, you, me, him, her, we
possessive: my, your, our
reflexive: myself, yourself

Pronominal coreference resolution is very important to speech act recognition because without finding the proper antecedent one cannot know whether it is the speaker (writer) of a document who is performing the speech (communication) act or whether the speaker (writer) is commenting on the speech act of some other person(s).

The importance of correct correlation of personal pronouns with their antecedents is illustrated in the following example. In this example, person's names are annotated in brown and speech acts in light blue.


NOTE TO: EDE HOLIDAY
ASSISTANT TO THE PRESIDENT
AND SECRETARY TO THE CABINET

FROM: BILL KRISTOL
CHIEF OF STAFF TO THE VICE PRESIDENT

Tom Fleener has done (is doing) a terrific job for us, and understandably wants more responsibility. The avenues for advancement here are, at least temporarily, blocked -- but I thought you might have an opening that could take advantage of his considerable skills -- which are, above all, organizational skill, attention to detail, tact, and reliability. If you're looking for someone, I recommend him highly.

In this note, the recommendation isn't directly connected to *Tom Fleener* as the text says, "I recommend him". *Him* is correctly linked back to *Tom Fleener* and to *his* by use of the pronominal coreferencer.

The GATE version 4 distribution includes a pronominal coreferencer [Dimitrov 2002]. That approach to resolving pronominal coreference depends on the antecedent being a named entity. It does not rely extensively on linguistic and domain knowledge. Our initial evaluation of this coreferencer indicates that it does not perform well in coreferencing the pronouns *I, we, my* and *our* to the author's (writer's) names in memoranda and correspondence. We have reviewed the literature in pronominal coreferencing and are formulating an appropriate method for this task.

It does not perform well at all in coreferencing the pronouns *I, me, my* and *myself* to the author's names in memoranda and correspondence. This latter failure is due to the fact that it handles these pronouns only when the name is associated with quoted speech, for example, *President Bush said, "I'm trying to set high standards for government service."* This limitation is due in large part to the fact that the Sheffield pronominal coreferencer has been applied primarily to press wires and transcripts of broadcast news.

A method is being developed that uses knowledge of the author's and recipient's names as determined by a Document Type Recognizer (described in task 2) in coreferencing the pronouns *I, me, my, myself, we, you, your, our* and *ours*. It will be tested on a corpus of 50 documents and then an experiment will be conducted to evaluate its performance on a corpus of Bush administration e-records that was not used in development.

2.1.5.2 Speech Act Transducer

A writer may perform a communication act by explicitly using a performative verb or by using declarative, interrogative or imperative sentences. They may also make an assertion

without using a declarative sentence, ask a question without using an interrogative sentence, and give an order without using an imperative sentence. These kinds of speech acts are called indirect speech acts [Stefanowitsch 2003]. There is no reliable method to determine the intended illocutionary force of sentences that do not include the performative use of performative verbs.

Examples of the use of some 60 performative verbs have been analyzed to determine the syntactic patterns in which those verbs express a speech act. These patterns will be represented in JAPE rules that annotate speech acts in the text. The performative verb expressing a speech act may appear in nominalized form. For example, "My *request* is for an analysis of the War Powers Act." Thus, it must be possible to recognize the nominalized form of speech act verbs.

The speech act transducer will take the results of the SUPPLE parser (with a grammar for English) and the enhanced pronominal coreferencer. This consists of a list representation of the syntactic form of the sentences in the text of a document and a quasi-logical representation of the semantics of those sentences. Performative verbs will have a semantic feature indicating that they are performative. For the pronouns *I, you, we, your, my* and *our*, the pronouns will be referenced to author(s) names and addressee's names.

The function of the semantic transducer is to produce a semantic representation of the speech acts carried out by the document. That semantic representation will include elements such as the following.

qlf = [speech-act(e1), name(e1, VERB), author(e1, PERSON_NAME),
recipient(e2, PERSON_NAME2), proposition(e1, PROPOSITION),
(illocutionary-point(e1,POINT)]

To accomplish this, the speech act transducer must first determine whether the verb is ambiguous, and if so disambiguate it. For instance, the verb *agree* has three related but different meanings as a performative verb. First, a person can be in agreement with something somebody else said. For example, "I agree with Senator Cook that the resources of the nation belong to the nation, not to the multinational oil companies or to any individual." Second, a person can agree to do something or agree to a condition. For example, "I agree to attend the meeting." Third, persons with different ideas as to how to do something can by mutual concession or discussion agree on the same solution. For example, "We agree on whom to elect chairman." The sentence pattern "I (or we) agree that *clause*" can be used to recognize the first meaning. The sentence pattern "I (or we) agree *infinitive*" can be used to recognize the second meaning. The sentence pattern "I (we) agree *prepositional phrase*" can be used to recognize the third meaning. The form against which to match the patterns is provided by SUPPLE as a parse tree of a sentence.

There are other meanings for the verb *agree*, for example, something can be agreeable or suitable. For instance, "White wine doesn't agree with me." This meaning can be disambiguated from the performative meanings of agree by the fact that the grammatical

subject is not a person. Furthermore, the speech act in this case is not a performative sentence, but an assertion.

One hundred eighteen presidential records have been analyzed with regard to the speech acts expressed in the records. Forty-six different performative verbs were identified in the corpus. This corpus will be used in initial testing and demonstration of speech act recognition.

### 2.1.6 The Topics of Records

NARA's Lifecycle Data Requirements Guide specifies guidelines for the metadata of items, file units and record series in the Archival Research Catalog (ARC) [NARA 2007]. Those guidelines require that some of the metadata values come from authority lists [NARA 2000]. The later document includes a Topical Subject Thesaurus. This thesaurus is supposed to be used by Archivists for indicating the topics of records.

A record may have more than one topic. For example, a transcript of a Presidential News conference typically has many topics. Similarly, a Memorandum of Conversation (MEMCON) usually involves multiple topics.

In written records, the topic of a paragraph is often introduced in the first sentence of the paragraph. The topic of a document may be introduced in the first paragraph, but the other paragraphs within the document (or section of a document) may have different but related topics.

Linguists have tried to explicate the concept of the *topic of discourse*. Spoken discourse or written records that comment on, remark on, say something about, inform someone on something, tell someone about something, or advise someone of something have a *topic* or *theme.* The topic of discourse can be a person, thing, action, situation, or event. The information about the topic (theme) is called the *comment* (or *rheme*).

The syntactic subject of English sentences is more often a pronoun than a noun phrase. The syntactic direct object of English sentences is more often a noun phrase than a pronoun. Noun phrases are often used to introduce new topics, while pronouns are used to refer to previously introduced topics.

English uses several means to signal a new topic, for example

- Stating it explicitly as the syntactic subject.
- Stating it explicitly as the direct object of the verb.
- Using passive voice to transform a direct object into a subject.
- Emphasizing the topic using clefting, putting a constituent in focus by using a main clause and a subordinate clause, .e.g., “It’s John Edwards that I admire.”

- Through constructions such as *there*-insertion, e.g., “There arose a problem in the research design.”
- Through constructions like "As for...", "Speaking of...", etc.
- Through topicalization, that is, moving the topic to the beginning of the sentence.

*Topic segmentation* is the problem of identifying the segments of text that are about a topic. *Topic identification* is the problem of identifying the subject of the segment.

2.1.6.1 A Method for Recognizing the Topic(s) of a Record

An approach to identifying the topics of an e-record has been formulated [Underwood and Isbell 2008b]. The approach is an extension of the methods for recognizing document types and speech acts.

Document type recognition is included in the method for topic recognition because document structure as indicated by the document type can often be used to determine the topic(s) of a record. For instance, the topic of a memo is the topic on the subject line. The topic of a report is usually its title. Section headings indicate the topic of a section. Captions indicate the topic of a figure.

The capability to recognize speech acts is required because the topic of a record is sometimes the object of a performative verb, or more generally the proposition of the speech act. For instance, the topic of a Presidential Proclamation, while indicated in the title is also the proposition of the speech act “I hereby proclaim…” Sometimes speech acts by the author in the past or speech acts by other persons are the topic of a record.

We are constructing JAPE rules to annotate the topics of sentences based on their syntax, speech acts, pronominal reference, and named entities. Other JAPE rules determine the topics of paragraphs and the topics of sections.

These rules are being developed from an analysis of the topics in a corpus of 50 Presidential records. The method of topic identification will also be tested on this corpus. Then experiments will be conducted using records selected from the Bush Presidential e-record collection.

2.1.6.2 Related Research in Automatic Topic Recognition

Bigi et al [2001] compare several statistical methods for topic identification on two kinds of textual data—newspaper articles and e-mails. Five methods are tested on these two corpora: topic unigrams, cache model, TFIDF classifier, topic perplexity, and weighted model. One of the methods achieves a topic identification of 80% on a general newspaper corpus but does not exceed 30% on the e-mail corpus. Another method gives the best result on e-mails, but does not exhibit the same behavior on a newspaper corpus. They conclude that statistical topic identification methods depend not only on a corpus, but also on its type.

Lin [1995] presents a method for automatically identifying the central ideas in a text based on a knowledge-based concept-counting model. To represent and generalize concepts, he uses the hierarchical concept taxonomy in WordNet. By setting appropriate cutoff values for such parameters as concept generality and child-to-parent frequency ratio, the amount and level of generality of concepts extracted from the text are controlled. Tiun et al [2001] investigate a similar method. Stein and Eissen [2004] provide a formal framework for a similar method.

## *2.2 Recognition of Document Types*

During the prior year of research, it was found that there was substantial work needed to automatically identify the intellectual and physical elements of documentary form before the grammatical induction technique could be applied. More specifically, it is necessary to be able to distinguish paragraphs of records from other intellectual elements such as section headings, horizontal lines, lists, tables, columns, and page numbers. It is also necessary to convert records in a variety of legacy and current proprietary file formats into a common representation such as html or enriched text so as to have access to physical elements of form such as boldface, underline and italics. To focus on developing the capability to recognize intellectual and physical elements of documentary form, it was decided to focus on document type recognition, rather than induction of grammars for document types.

### 2.2.1 Method of Document Type Recognition and Metadata Extraction

Our approach to document type recognition is represented by processing and language resources entirely within the GATE architecture [Underwood and Laib 2008].

Tokenizer
Wordlist Lookup + Wordlists
Sentence splitter
Hepple POS Tagger + Lexicon
Named Entity Transducer + Rules for Named Entities
*Intellectual Element Transducer + Intellectual Element Rules*
SUPPLE Parser + *Document Type Grammars*
*Extract Record Metadata*

The intellectual element transducer uses JAPE rules to recognize and annotate the intellectual elements of a variety of document types. The intellectual elements are currently recognized in five phases. First, the key terms or phrases that compose intellectual elements are annotated. Fig. 3 shows a JAPE rule for recognizing the intellectual element called *to*, the addressee caption of a memorandum.

```
//MEMORANDUM FOR -> TO
//TO: -> TO
Rule: TO
(
 (
  (Token.string == "MEMORANDUM")
  (Token.string == "FOR")
 )  |
 (
  (Token.string.string == "TO")
  (Token.string == ":")
 )
)
: to
-->
      :to.PossElement = (eletype = "to")
```

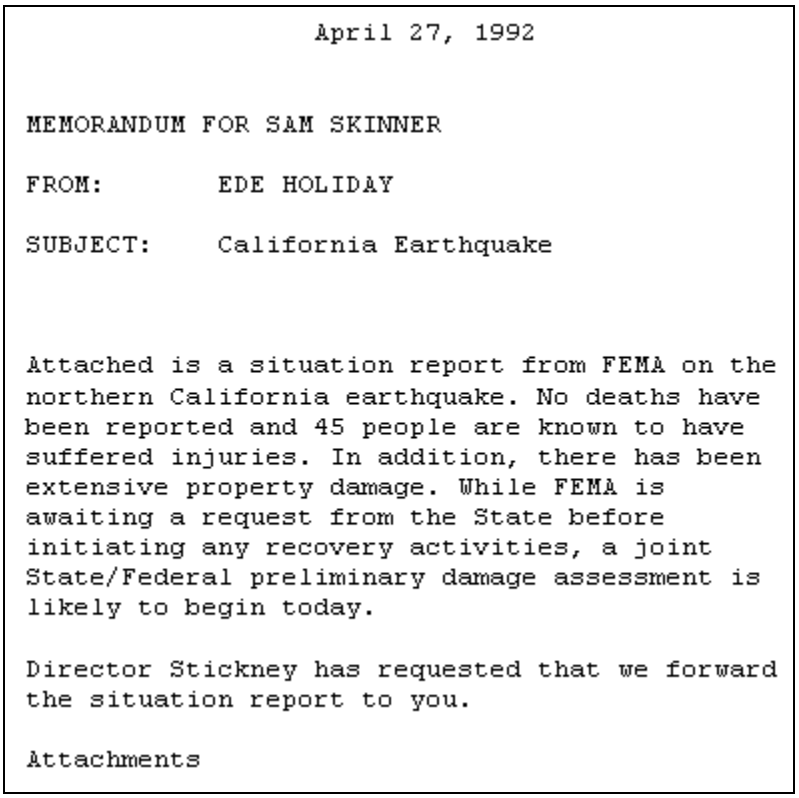
**Figure 3. JAPE Rule for Intellectual Element "to"**

In the second phase, additional intellectual elements are annotated based on named entities recognized during information extraction and on annotations created during the earlier phases. The Intellectual Element Transducer currently annotates about 80 intellectual elements that may occur in seventeen documentary forms of Bush Presidential records.

The SUPPLE parser is a bottom up chart parser that uses a context-free grammar for English to parse a sequence of word tokens (with their parts of speech) and named entities (usually proper nouns) in a sentence. To recognize the documentary form of an e-record, we provide SUPPLE with context-free grammars for documentary forms and pass to SUPPLE a sequence of intellectual elements of the entire document.

Figure 4 shows an example of a White House memorandum.[1] Memoranda such as this were printed on White House stationery. The electronic copies of memoranda typically did not contain a letterhead, but in some cases a letterhead was typed at the top of the memorandum.

[1] Bush Presidential Library, Bush Presidential Records, WHORM Subject File, Disasters-Natural, ID#324869.



April 27, 1992

MEMORANDUM FOR SAM SKINNER

FROM: EDE HOLIDAY

SUBJECT: California Earthquake

Attached is a situation report from FEMA on the northern California earthquake. No deaths have been reported and 45 people are known to have suffered injuries. In addition, there has been extensive property damage. While FEMA is awaiting a request from the State before initiating any recovery activities, a joint State/Federal preliminary damage assessment is likely to begin today.

Director Stickney has requested that we forward the situation report to you.

Attachments

**Figure 4. An example of a White House Memorandum**

The context-free grammar shown in Figure 5 defines the documentary form of the memo shown in Fig. 4 as well as the forms of other memoranda in the Bush e-record collection.

MEMO → HEAD BODY
MEMO → HEAD BODY OPTIONAL
HEAD → DATE ADDRLINE SNDRLINE SUBJLINE
HEAD → DATE ADDRLINE SNDRLINE THRULINE SUBJLINE
ADDRLINE → TO ENTITIES
SNDRLINE → FROM ENTITIES
THRULINE → THRU ENTITY
SUBJLINE → SUBJ TOPIC
ENTITIES → ENTITY ENTITIES
ENTITIES → ENTITY
ENTITY→ PERSON
ENTITY → PERSON JOBTITLE
ENTITY → ORGANIZATION
BODY → PARAS
BODY → SECTS
BODY → PARAS SECTS
PARAS → PARA PARAS
PARAS → PARA
SECTS → SECT SECTS

SECTS → SECT
SECT → SECTHDG PARAS
OPTIONAL → ATTACH
OPTIONAL → COPIES
OPTIONAL → ATTACH COPIES
ATTACH→ ATTACHMENT
ATTACH → ATTACHMENT TITLES
TITLES → TITLE TITLES
TITLE → TITLE
COPIES → CC ENTITIES
COPIES → CC ADDRESSES
ADDRESSES → ENTITY ADDRESS ADDRESSES
ADDRESSES → ENTITY ADDRESS
COPIES → JOBTITLES
JOBTITLES → JOBTITLE JOBTITLES
JOBTITLES → JOBTITLES

**Figure 5. A Context-free Grammar for the Documentary Form of Memoranda**

This and other context-free grammars for document types are translated into the notation used by the SUPPLE parser and semantic notations are added to the rules to enable the interpretation of the text appearing in the intellectual elements. Figure 6 shows an example of rules of the augmented grammar for memoranda.

```
%% MEMO → HEAD BODY
rule(memo(s_form:F,sem:D^[[document,D],[document_form,D,memo],
[author,D,SNDRList],[addressee,D,AddrList],[topic,D,TOPIC],
[date,D,DATE]]),
 [head(s_form:F,sem:[DATE,AddrList,AuthorList,TOPIC]),
 body(s_form:F)]).

%% HEAD → DATE ADDRLINE SNDRLINE SUBJLINE
rule(head(s_form:F,sem:[Date,ADDRList,SNDRList,TOPIC]),
 [chrondate(s_form:F,date:DATE),
 addrline(s_form:F,sem:AddrList),
 sndrline(s_form:F,sem:SNDRList),
 subjline(s_form:F,topic:TOPIC)]).
```

**Figure 6. A Fragment of the SUPPLE Grammar for the Documentary Form of Memoranda**

Figure 7 shows the parse tree of the structure of the document shown in Fig. 3 as recognized by the SUPPLE parser.

```
{best_parse=
(memo (head (chrondate (sem_cat "April 27, 1992"))
            (addrline (for (sem_cat "MEMORANDUM FOR"))
                  (entities (entity (person (sem_cat "SAM SKINNER")))))
            (sndrline (from (sem_cat "FROM:"))
                  (entities (entity (person (sem_cat "EDE HOLIDAY")))))
            (subjline (subj (sem_cat "SUBJECT:"))
```

```
            (topic (sem_cat "California Earthquake"))))
      (body (paras (para
            (sem_cat "Attached is a situation report from FEMA on the
            northern California earthquake. No deaths have been
            reported and 45 people are known to have suffered injuries.
            In addition, there has been extensive property damage.
            While FEMA is awaiting a request from the State
            before initiating any recovery activities, a joint
            State/Federal preliminary damage assessment is likely to
            begin today."))
            paras (para
            (sem_cat "Director Stickney has requested that we forward
            the situation report to you."))))))
      (optional (attachment (sem_cat "Attachments")))))}
```

**Figure 7.The Documentary Form (Parse tree) of the Sample Memorandum**

Figure 8 shows the semantics of the memorandum in a quasi-logical form (qlf). This also represents the metadata of the memoranda that are needed for describing the e-record.

```
{qlf=[document(e1), document_form(e1, memo), author(e1, 'EDE HOLIDAY'),
      addressee(e1, 'SAM SKINNER'), topic(e1, 'California Earthquake'),
      date(e1, 'April 27, 1992')]}
```

**Figure 8. The Semantics of the Document Shown in Fig. 3**

This logical notation is read as "e1 is a document," "The documentary form of e1 is memo," the author of e1 is EDE HOLIDAY," etc.

2.2.1.1 Archival Description

From the metadata shown in Fig. 8, the following item description can be automatically generated.

> A memorandum dated April 27, 1992 from Ede Holiday to Sam Skinner regarding California Earthquake.

Suppose that there was a directory in *Edith E. Holiday's Files* titled *Petrolia, Calif. (Cape Mendocino) Earthquake.* The following folder (directory) description can be automatically generated from the descriptions of items in the folder.

> This file unit contains materials relating to the 1992 Petrolia, California Earthquake. It includes memoranda, situation reports and correspondence.

The document type recognizer is run inside the GATE graphical user interface (GUI). To conduct experiments with the Bush Presidential e-records, this recognizer must be interfaced to the PERPOS prototype. GATE provides an Application Programmers Interface (API) that allows GATE to be run inside a Java program. A GATE persistent

pipeline application was created that can be loaded inside a JAVA program. A JavaBean (Doc Parser) was created to perform the parsing of documents. The DocParser JavaBean was then packaged and registered using the Java packager command. This created the DocParser.dll file that was registered as the DocParser Bean Control. Next, the PERPOS Archival Processing Tool (APT) was modified to use the DocParser Bean Control for recognizing documentary forms and extracting metadata for filling in the fields of record withdrawal forms and for generating item descriptions.

## *2.3 PRA Restrictions and FOIA Exceptions*

Figure 9 illustrates the components of the access restriction checker. When assistance in checking for access restrictions is requested, the automatic speech act and topic recognition methods in GATE are applied to a plain text or html copy of the record. Metadata such as document type, author(s), addressee(s), and dates can be previously determined during automatic description and associated with the record in the manifest. These facts about an e-record are stored into the working memory of the Java Expert System Shell (JESS). The JESS Inference Engine checks the antecedent criteria of rules for identifying possible access restrictions against the facts in the working memory. If the criteria include background knowledge not included in the record or the working memory, it is sought in a long-term memory of background knowledge. If all the criteria of a rule are satisfied, the consequent part of the rule asserts facts, which may include conclusions about possible access restrictions, into the working memory. The conclusions are then displayed to the review archivist.

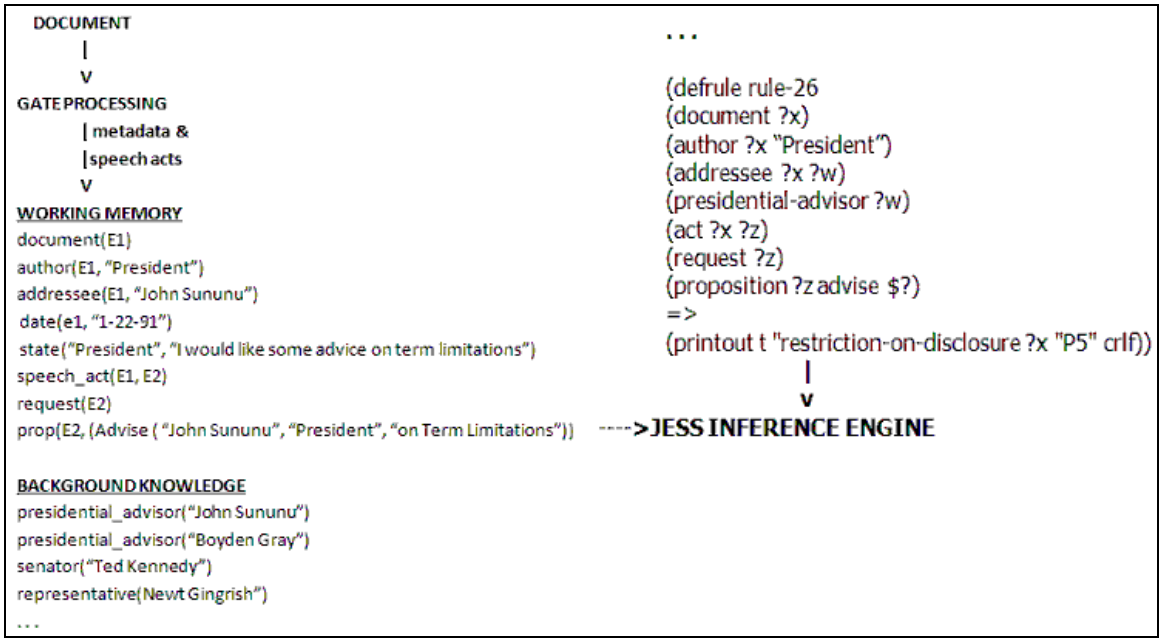



**Figure 9. Components of the Access Restriction Checker**

During the PERPOS II Project, a representative sample of 150 Presidential e-records was analyzed to determine features of those records that were important in determining whether they were personal record misfiles or subject to PRA restrictions P-2 Appointments to Federal Office or P-5 Confidential Advice. During that project an access restriction checker was designed and prototyped. This prototype allows information about a record to be asserted into a working memory so that the system can then reason with these facts. It applies a set of rules to determine if a document or passages therein might have an access restriction, and displays the results to an archivist [Harris et al 2005]. This research task is to develop a fully functional prototype access restriction checker and evaluate its performance.

The corpus of 150 Presidential e-records is too small to adequately develop criteria for determining access restrictions. The Bush Presidential Library has released documents formerly withheld under Presidential Records Act restrictions P-2 (appointment to federal office) and P-5 (confidential advice between the President and his advisors or between those advisors). We analyzed some of these documents to better identify criteria for determining P-2 and P-5 restrictions.

The background knowledge that is needed is too extensive to be stored in working memory. It includes names and titles of the president's advisors, names and titles of white house staff, names, titles and dates of nominations of presidential nominees to Federal office and names of Senators and Congressmen of the 101st and 102nd Congress. Due in part to the fact that the GATE environment includes Prolog as a resource, it was decided to represent this background knowledge as Prolog facts and rules that can be accessed when needed as criteria of JESS rules.

The information about a record that is needed to determine whether an access restriction might apply includes document type, communication act and topic. The results of research tasks 1 and 2 provide those capabilities. JESS and the rules for access restrictions have been interfaced to GATE and the document type and speech act recognition resources.

During the winter quarter of 2007, archivists at the Bush Presidential Library pilot tested PERPOS in support of FOIA Processing. They made recommendations for modification of features and inclusion of new features in PERPOS. This provides NARA with a better understanding of the functions needed to support archival review of e-records. This test included the review of Presidential e-records for PRA restrictions and FOIA exemptions. The reviewed test FOIA collections are a corpus that could be used for evaluating the Access Restriction Checker.

## *2.4 Maintaining the Archival Research Prototype in the Face of Technological Change*

One of the challenges that archivists face in preserving electronic records is that the operating system and programming language technologies that support an application for

processing e-records are changing so rapidly. For instance, Microsoft no longer supports the operating system in which PERPOS operates, Windows 2000. The integrated development environment (IDE) used to develop PERPOS is Visual Basic 6 in Visual Studio. Microsoft no longer supports this development environment. To maintain the viability of the prototype, the Visual Basic code must be converted to a current development environment, and be portable to multiple operating environments, e.g., Microsoft Windows and Linux.

The current implementation of PERPOS, with some additional features supporting FOIA processing, has been migrated from Windows 2000 to Windows 2003. This version of PERPOS is being Beta tested before distribution to other sites.

A security policy for the PERPOS prototype [Molavi 2004] has been implemented. Security patches for the operating system and database and the signature files for the Virus scanner is provided using a firewall that only allows access to a secure server providing these services. Security policies for user accounts, auditing, etc. have been implemented using the Microsoft Management Console (MMC) 3.0. The security policy and its implementation will be documented in the *PERPOS Administrator's Guide*.

In PERPOS, a file type identifier is used in filtering, viewing files, converting files to other formats, repairing corrupt (damaged) files, recognizing archives and self-extracting archives, and in recognizing password-encrypted files. The PERPOS file type identifier is being replaced by the Linux file command (written in C) and a greatly enhanced magic file. Most of the criteria for recognizing file types have been converted from Visual Basic to the magic file used by the Linux file command. There are an additional 50 file types that require modification of the file command's C-code in order to be recognized. The new file type identifier recognizes 478 File types. Examples for 353 of these 478 file types have been collected and used to test the new file type identifier. It correctly recognizes these 353 file types.

### 2.4.1 Investigation of Migration of PERPOS from VB6 to VB.NET

The tools available for migration of VB6 programs to VB.NET programs include the Code Advisor for VB6 and the Upgrade Wizard for VB. The Code Advisor for VB6 should be run before the upgrade Wizard. The code advisor identifies 28 types of code that will not work in VB.NET and that will not be upgraded by the Upgrade Wizard [Laib and Underwood 2008]

To investigate the detailed issues in migration of VB6 code to VB.NET, the ManifestLibrary.dll of PERPOS was chosen as an example. There are 5 classes and 4 module files in the dll. Running the Code Advisor on this dll produced a report listing 88 issues relating to 5 types of code that the Upgrade Wizard will not upgrade. The table below shows those 5 categories of code that are not upgradeable, a description of each category, and the number of instances of that category of code.

| Name | Description | Issue Count |
|---|---|---|
| Late Binding of Variant or Object | Variables, parameters, and return values typed as Variant or Object can cause problems when upgrading. | 20 |
| Variant-Returning String Function | Variant-returning string functions are not supported in Visual Basic .NET. Use the String-returning version of the function, which has a '$' suffix. | 14 |
| Non Zero Lowerbound Arrays Not Supported | Visual Basic .NET does not support the use of arrays that have a lower-bound index other than zero. | 10 |
| As Any Not Supported | API Declare statements that include parameters typed using 'As Any' will not be upgraded. | 4 |
| '#If' blocks are not reliably upgraded | When a #If condition evaluates to False, the #If...#End If block is not upgraded. The Upgrade Wizard does not reliably evaluate whether #If conditions are True or False. | 40 |

The "Late Binding of Variant or Object" issues need to be deferred until after the Upgrade Wizard has run. This category has two subcategories. One subcategory is code that can be replaced with so-called "overloaded methods." These are methods with the same name but different arguments or return values. The other category of code has to do with the way nulls are handled in database return values.

The "Variant-Returning String Function" code is easily fixed by replacing the variant returning function by a corresponding function with a $ suffix that returns a string. The Code Advisor places 'FIXIT:' comments in the code. It also places a button on the tool bar that allows the user to go to the next FIXIT comment.

The "Non Zero Lowerbound Arrays Not Supported" category of code is difficult to fix. Code that calculates indexes as well as the "Dim" and Redim" statements that define arrays must be modified.

The "As Any Not Supported" code category is caused by the "CopyMemory" and the "ZeroMemory" Windows APIs. This code is replaced after the code has gone through the Upgrade Wizard.

The "'#If' blocks are not reliably upgraded" category is easy to fix but time consuming. These blocks are debugging blocks that are in most class files.

After the needed changes are made, the Upgrade Wizard can be run to replace the VB6 code with VB.NET code. It creates an Upgrade Report that lists "Warnings" and "Compile Errors." "Warnings" include code that has been generated that will not behave exactly as the code it replaces. For instance, "Get" and "Put" methods are replaced with FileGet and FilePut, which do not behave exactly as the methods they replace.

*Marshalling* (also known as serialization) is the process of transforming the memory representation of an object to a data format suitable for storage or transmission. Marshalling was done behind the scenes in VB6. In VB.NET explicit marshalling instructions are needed to tell the system how to pass structures. Since fixed-length strings are not allowed in VB.NET, the VB upgrade Wizard replaces fixed-length strings with a character array with the length specified in the marshalling instructions.

Once the code from the Upgrade Wizard has been modified to remove the "Compile Errors" and the "Warnings", it needs to be compiled again. This may result in different errors and warnings. Once a clean compile is achieved it is time to create a test application that can make COM calls to the dll to test that behavior has not changed.

After a clean compile was achieved, it was found that methods that convert numeric variables to octal variables, or vice versa, have changed behaviors, even though no VB Upgrade Warning was given of this change in behavior. Where the octal numbers were converted to integer, they caused numeric overflow errors. Where a negative number was converted to octal, unexpected values resulted. The change of behavior was due to the fact that there are unsigned numbers in VB.NET, but not in VB6. The Upgrade Wizard changed all "int" variables to "Short" and all "long" variables to "Integer". In the cases of octal conversion, where "int" was used in VB6, it should have been replaced with "UShort". Where "long" was used in VB6, it should have been replaced with "UInteger".

"Compile Errors" include deferred changes to the API calls. These errors are fixed using code found on the "Equivalent of CopyMemory in .NET" website.[2]

*Code refactoring* is any change to source code which improves its readability or simplifies its structure without changing its results. An example of the need for refactoring is when there is duplicate code in different locations. Refactoring would consist of replacing the code with a single method.

The Upgrade Wizard uses the "Microsoft.VisualBasic.Compatibility" libraries to mimic some VB6 behavior. To utilize the full capacity of the NET Framework, some, if not all, of this code should be refactored by replacing it with Net Framework equivalents. Similarly, the FileGet and the FilePut methods generated by the Upgrade Wizard from VB6 Gets and Puts should be replaced with Stream and Buffer objects.

Before drawing conclusions on the ease and advisability of migration to VB.NET or JAVA, we need to investigate the tools that are available for the conversion of VB6 to JAVA.[3]

[2] url: http://ww.codeproject.com/vb/net/CopyMemory_in_Net.asp
[3] VB Converter, JAVA Edition, Diamond Edge Products.

# 3. Summary of Results

The performance of the previously developed Information Extraction tool has been improved by the inclusion of additional wordlists and JAPE rules. Additional semantic categories such as facility names, legislative bills and statutes, names of governments, and relative temporal expressions are now annotated.

Speech acts are acts of speech or writing in which one does something just by saying something, e.g., "I appoint you…", "I hereby proclaim…". One hundred eighteen Presidential records were analyzed with regard to the expression of speech acts with performative verbs and speech acts about the author's past or future speech acts or other's speech acts. More than 60 kinds of speech acts were discovered in the corpus. The analysis confirms that performative verbs are used to express the actions carried out by records. A method has been formulated for identifying the speech acts occurring in e-records. It will be implemented, tested using records from the analyzed corpus and then experimentally evaluated

A corpus of 50 presidential records was analyzed to determine the topic(s) of a record. An approach to identifying the topics of an e-record has been formulated. The approach is an extension of the methods for recognizing document types and speech acts.

A method for automatic document type recognition has been implemented and successfully tested. The method is based on the method for automatically annotating semantic categories such as person's names, dates, and postal addresses. Currently, it extends this method by (1) identifying about 80 types of intellectual elements of documents, (1) parsing these elements using context-free grammars defining the documentary form of document types, (3) interpreting the meaning of the form of the document to identify some or all of the following: the chronological date, author(s), addressee(s), and topic.

Progress in implementing the Access Restriction Checker includes the interface of GATE and PROLOG to the Java Expert System Shell (JESS). GATE includes the resources for document type recognition and extraction of facts about a record such as author, addressee, chronological date and sometimes topic. PROLOG includes the facts and rules representing background knowledge. Additional rules have been constructed for identifying possible access restrictions. Still needed are the capabilities for speech act and topic recognition. Methods have been formulated for providing these capabilities and plans are to implement them this coming year.

Due to the rapid changes in computer technology, archivists must be concerned not only with the obsolescence of e-record file formats, but with the obsolescence of the operating systems, database management systems and integrated development environments of their Archival Repository and Archival Processing System. These issues are investigated and discussed using the Presidential Electronic Records Pilot System (PERPOS) as a case in point.

## Presentations

B. Clement and W. Underwood. Evolution of a Prototype Archival System for Preserving and Reviewing Electronic Records. Archives 2008: R/Evolution & Identities. Society Of American Archivists. San Francisco, August 26-30, 2008

W. E. Underwood. Metadata Extraction, Archival Description, and FOIA Search in PERPOS. OCLC Western Digital Forum in San Diego, August 9-10, 2007.

W. E. Underwood. NLP Technologies Applied to Archival Tasks, NARA Administration Building, Allegany Ballistics Laboratory, Rocket Center, West Virginia, December 14, 2007

W. E. Underwood. NLP Technologies Applied to Archival Tasks, NARA, Archives II, College Park, April 2008

W. Underwood. Automatic Metadata Extraction for Archival Description and Access. Society of American Archivists: Research Forum, San Francisco, August 26, 2008

# References


[Austin 1962] J. L. Austin. *How to do things with words: The William James Lectures delivered at Harvard University in 1955*. Ed. J. O. Urmson. Oxford: Clarendon, 1962.

[Bigi et al 2001] B. Bigi, A. Brun, J-P Haton, K. Smali and I. Zitouni. A Comparative Study of Topic Identification on Newspaper and E-mail. *Proceedings Eighth International Symposium on String Processing and Information Retrieval 2001* (SPIRE 2001). 13-15 Nov. 2001 pp. 238 - 241.

[Cunningham et al 2007] H. Cunningham, D. Maynard, K. Bontcheva, V. Tablan, C. Ursu, M. Dimitrov, M. Dowman, N. Aswani, I. Roberts, Y. Li, and A. Shafirin. Developing Language Processing Components with GATE Version 4, The University of Sheffield, December 1, 2007.

[Dimitrov 2002] M. Dimitrov. *A Light-weight Approach to Coreference Resolution for Named Entities in Text*. MSc Thesis, University of Sofia, Bulgaria, 2002.

[Duranti et al 1997] L Duranti, T. Eastwood and H. McNeil. The Preservation of the Integrity of Electronic Records. www.interpares.org/UBCProject/intro.htm

[Holdcroft and Smith 1992] D. Holdcroft and P. Smith. Speech Acts and Computation. Chapter 6, *Machinations: Computational Studies of Logic, Language and Cognition* (R. Spenser-Smith and S. B. Torrance eds.) Intellect Books, 1992.

[Isbell 2006 et al] S. Isbell, M. Underwood and W. Underwood. The PERPOS Information Extractor Applied to Presidential E-Records, PERPOS TR ITTL/CSITD 05-10, November 2006.

[Lin 1995] C-Y Lin. Knowledge-based Automatic Topic Identification. Meeting of the Association for Computational Linguistics, 1995

[Molavi 2004] D. Molavi. PERPOS Network Use and Security Policy, PERPOS Technical Report 04-6, Information Technology and Telecommunications Laboratory, Georgia Tech Research Institute, June 2004.

[NARA 2007] National Archives and Records Administration. Lifecycle Data Requirements Guide June 18, 2007 www.archives.gov/research/arc/**lifecycle-data-requirements**.doc

[NARA 2000] National Archives and Records Administration. *Life-Cycle Systems Data Standards Authorities.* Washington, DC: NARA Office of Records Administration, 2000, plus updates.

[Rosé et al 1995] C. P. Rosé, B. Di Eugenio, L. Levin, and C. Van Ess-Dykema. Discourse processing of dialogues with multiple threads. *Proc. 33rd Annual Meeting of the Association for Computational Linguistics*, pages 31-38, 1995.

[Sadock 1972] J. Sadock. *Toward a Linguistic Theory of Speech Acts*. Academic Press, 1974.

[Searle 1969] J. R. Searle. *Speech Acts*. Cambridge University Press. 1969.

[Searle 1979] J. R. Searle. *Expression and Meaning*. Cambridge University Press. 1979

[Stefanowitsch 2003] A. Stefanowitsch. A construction-based approach to indirect speech acts. In Klaus-Uwe Panther and Linda Thornburg (Eds), *Metonymy and pragmatic inferencing*. Amsterdam and Philadelphia: Benjamins, 2003, pp. 105-126.

[Stein and Meyer 2004] B. Stein and S. Meyer. Topic Identification: Framework and Application. *Proceedings of I-KNOW '04, 4th International Conference on Knowledge Management. Journal of Universal Computer Science*, pp. 353-360.

[Tiun 2001] S. Tiun, R. Abdullah and E. Tang. Automatic Topic Identification Using Ontology Hierarchy. *Proc. of the 2nd International Conference on Intelligent Text Processing and Computational Linguistics* (CICLing-2001), Mexico City, Mexico (2001).

[Underwood and Isbell 2008a] W. Underwood and S. Isbell. Recognizing Speech Acts in Presidential E-records, Working paper ITTL/CSITD 08-03, Georgia Tech Research Institute, June 2008.

[Underwood and Isbell 2008b] W. Underwood and S. Isbell Semantic Annotation of Presidential E-records, Technical Report ITTL/CSITD 08-01, Georgia Tech Research Institute, May 2008.

[Underwood and Laib 2008] W. Underwood and S. Laib. Recognition of Documentary Forms. TR 08-02, Information Technology and Telecommunications Laboratory, Georgia Tech Research Institute, Atlanta, Georgia, May 2008.

[Vanderveken 1990] D. Vanderveken. *Meaning and Speech Acts. Vol 1, Principles of Language Use*. Cambridge University Press, 1990.

[Wierzbicka 1987] A. Wierzbicka. *English Speech Act Verbs: A semantic dictionary* Academic Press 1987.