PROJECT: Research and Development for the Declassification Productivity Initiative

Grant #: DE-FGO1-96IS50317

PERIOD: January - August, 1997

INTRODUCTION: The highlight for the first quarter was the presentation of research progress and findings at the DPI Symposium on March 5, 1997. Since that presentation, additional progress was slowed down due to the decreased budget funding for year two, and consequently, the decrease in time-effort of the principal investigators. This report summarizes the progress in each of the topical areas to date. A research article has been prepared for publication for the Optical Character Recognition project; two progress reports are included for the Logical Analysis project; and two progress reports for the Knowledge Representation project. Research activities for the Tipster Technology project will resume this fall.
DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.
DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, make any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.
1.0 Personal Comments and Background Information

Initially, the general goal of this part of the project was oriented towards the creation of a model of an ideal set of classified documents. Unfortunately, it was discovered that I had not been cleared to receive examples of classified information, with the result that the project experienced a somewhat shaky start because of the unavailability of materials. The original work plan had stated that much of the work would have to be devoted to building models of information types that could be used in a declassification system, so this provided direction for the first year. Of the phases set out in the original work plan, the following tasks became primary:

1) the development of general strategies for knowledge representation (KR) and inferencing;
2) the construction of models of information types that could be used in a model set of classified documents; and
3) the development of particular inferencing and KR techniques that could be used in automated classification/declassification systems.
By the end of July of 1996, I was still not in possession of any actual classification guidelines or classified information, despite what seemed to me to be heroic efforts on the part of Dr. Curtis to make material available to me. However, DOE did provide instruction about how the classification process works, and that proved to be very helpful to the project. It became clear that the activities of classifiers were much less analytically driven than I had anticipated. Applying declassification guidelines was much like rule application in a domain in which open-textured concepts are used. Having had extensive experience in the legal domain, I immediately recognized that both the declassification process and the law were facing similar problems pertaining to the interpretation of rules (lawyers make big bucks off these problems, but I don't know about DOE classifiers). This realization prompted me to consider how the research could be oriented towards the development of technology that could be used to assist original and derivative classifiers in their daily tasks of guideline application.

Careful thought on the matter yielded the idea that perhaps the best plan would be to attempt to develop a system for classifying the types of information with which the classifiers must deal and then determine how best to incorporate that technology into an automated system. This fit well with the original goal of building models of information types for use in a general model of classified documents. I discussed the idea with Dr. Curtis, the DOE supervisor for this component, and after considering all the circumstances, a decision was made to focus research efforts in this area. The conference with Dr. Curtis gave the project the direction it needed to fashion its most promising work plan.

Research on the design of an automated support system for classifiers became the primary focus of the project and produced the results described in the body of this report. The system seemed to call for a collection of automated tools that could be used by classifiers to classify and declassify documents on the fly. The system would have to provide a friendly user interface and would require a strong inference engine to support interaction between the classifier and the system. My investigations into the roles that KR and inferencing would play in the construction of a model of a set of classified documents revealed that features of the model could be represented in a form that would support interactive processes involving the human classifier and the automated tools. It thus seemed feasible to think about building an automated classification system that would integrate a number of technologies that were being developed in DPI.

It was obvious that much work would have to be done on how to relate abstract models of document content to the actual contents of documents. An important finding was that concept mapping should be employed in the system at the same level as text-processing programs and that concept mapping, text processing, and natural language understanding techniques could be incorporated into the system to produce KBs that could be accessed by the automated tools. This paper contains a report on the current status of the design of the system. I welcome your input and suggestions.

2.0 Proposed Automated Classification Support System

Recent advances in object-oriented programming and windows have significantly eased the task of overcoming problems caused by the lack of natural language understanding capabilities. Programmers can now build systems that provide the
user with extensive expressive power while avoiding the use of direct, human-provided natural language input. This has prompted expert systems developers to attend aggressively to the relation between representation and classification. Matters pertaining to the nature and extent of the codependency between the two are of special concern. This is because new programming environments allow researchers to maximize the use classification technologies as means of lessening representation tasks, especially in domains in which natural language text is used extensively. Advanced class-based systems have been developed (e.g. CLOS) in which programmers may define both the classes of objects that will be recognized and the procedures by which instances of those classes may be constructed, modified and extinguished. This is all complemented by programming environments that allow the display of system operations in windows in user-friendly ways [e.g. CAPI, version 3.2.2]. The result is that prompting mechanisms such as menus and list panels can be employed as mediums of direct communication between the system and the user or system developer.

This report describes research on the design of an experts systems technology that would combine and coordinate classification and representation technologies into a powerful expert systems construction tool. The system, called ACSS (Automated Classification Support System), would be used to represent structural and semantic properties of diverse domains while simultaneously building conveniently accessible knowledge bases for them. Current plans call for the system to be implemented in a lispworks environment that would use a CLOS-based system to interact with the human classifier. The system would employ a classification module that would classify domain information and construct sentential representations of key components. A special interpreter would interpret the results and incorporate them into a knowledge base (KB). The system would construct an analytic model of the domain to whatever extent it could make the requisite connections among KB components. This is all to be done with the help of an inferencing support module.

3.0 Overview of ACSS

The basic architecture of the ACSS is diagrammed below.

In the main interface, the user would select the module appropriate for the desired task. The classification module would be used to build knowledge bases for natural-language-text domains, whereas the query/search module would be employed to ask questions and retrieve information from KBs built using the classification module. The automated dictionary would be a by-product of the classification process and would be used for quick look-up and displays of relations between concepts. The classification process being designed for the system is diagrammed below. Arrows
are used to indicate that the inferencing module is to provide interactive support during the entire process of classification and interpretation. The declassification process is to bear a similar design.

One of the special features that will distinguish ACSS from some other systems will be its ability to construct its own query/search module automatically during classificatory and interpretative processes. The system will be able to support participatory navigation at the query/search level based on the structural and semantic processing that was employed to build the KB. This will produce special benefits not found in systems whose query/search mechanisms are constructed as independent enterprises. The querying power of the system, however, will be functionally dependent upon classificatory and interpretive capabilities. The constraining principle is that the retrieval capabilities of the system will be measured by what has been classified and incorporated into the system through interpretation.

ACSS will be particularly distinguished by its behind-the-scenes employment of a robust, sententially-oriented representation language and a specialized interpreter capable of reading the language and of building an optimized KB from input. The system will start with a natural language domain and develop a KB for it over which an optimized search can be conducted. As textual information is classified, the inferencing component will prompt the human classifier for additional input in an attempt to maximize the semantic depth of the classificatory descriptions selected for the text. The prompters, with the help of the inference engine, will generate appropriate sentential descriptions and then pass them to the interpreter. The representation language in which the descriptions are to be expressed will be formally structured in a way that will enable the interpreter to cast descriptive content into a KB that will be especially accommodative of analytical paths that will be mapped out in query/search.

The two key modules of the system will be the classifying module and the inferencing module. These two will work in combination to build the query/search module. The inferencing module will be a spin-off of SMS (Symbolic Manipulation System), an automated reasoning system that reads a special representation language called SL (Symbolic Language) and builds knowledge bases from epistemological elements conveyed by SL [see deBessonet, 1995a; 1991]. SMS and SL were conceived as part of an attempt to develop a unified approach to knowledge representation and reasoning. ACSS will provide SMS-like inferencing support for each of its modules.
A full understanding of ACSS technology would require a familiarity with the technology from which it is derived.

The classification module will be used to classify document content for use in query/search. Success has been reported by others in using concept maps and other conceptual models constructed during the knowledge acquisition process to support intelligent and efficient navigation through various components of the knowledge base. The capabilities of an expert system that employs this approach are in large measure a function of the degree to which domain objects have been classified. The more finely grained and analytic the classification scheme is made, the greater the potential capabilities of the system become. It follows that special efforts must be spent on the design and construction of the classification scheme to achieve best results. The domain must be examined closely to determine the types of information it contains, and models of each information type must be built in advance and incorporated into the classification module.

The scheme of classification used in ACSS will employ a carefully constructed lexicon that will make available not only normal semantic definiens, but also special syntactical structures that govern the use of special terms. The system is to rely heavily on a set of key taxonomies prepared in advance, the most important of which will cover the types of information that will appear in the domain. Referred to as info-types, a set of preliminary classificatory headings or names is to be selected by manually perusing key documents of the domain that is to be classified. In most feasible applications of ACSS, an initial set of info-types could be selected with surprisingly little time and effort, perhaps in a few hours or less for one familiar with the domain. Info-types discovered later during the classification process could be added to the system on the fly. The selected info-types will then be added to the general taxonomic structure and will generally constitute the top level of the scheme. Some of the terms will bear defined relations to one another, whereas others will hold somewhat independent positions. A defined term, for instance, would have a defined relation to its definiens, yet the term itself might stand independently of some other group bound together by common relations. A significant number of the info-types will be transportable to other domains that are predominantly textual. The "definitional info-type," for example, will have obvious relevance for any textual domain in which terms are defined.

ACSS is being designed to accommodate text processing technologies during the classification process. After the text is classified using the classification module, the system will be in a position to build concept maps of the text, to employ text processing programs in building key-word distributions, and to take advantage of whatever natural language understanding results are available. Some of the text processing and natural language understanding technologies that will be used are being developed by other investigators of DPI. However, the inferencing module of ACSS will be able to do some of this on its own. The system will be called upon to produce what is referred to in SMS theory as "atomically normalized forms" of texts and to generate sets of nonstandard relations and inferences for formal representations of the text. The inferencing component will access the atomically normalized forms and nonstandard relations as part of its attempt to detect when a document contains text that is or should be classified under the classification guidelines of DOE.

4.0 Role of Nonstandard Inferencing

In the area of inferencing, ACSS will draw upon SMS theory, which is being
extended to include a general theory of systematic association capable of dealing with nonstandard relations of the sort recognized by human beings in ordinary conversation [see deBessonet, 1995a]. The associative theory is referred to as *penumbral inferencing* (deBessonet, 1991). This kind of inferencing is similar to what is known in AI circles as "plausible reasoning" but is distinguished by its somewhat broader scope (deBessonet 1991). Any given SMS object, whether sentential or nonsentential, may have a set of associated objects that bear "penumbral relations" to the given object. Penumbral relations are nondeductive but are not necessarily understood in terms of probability, or even possibility. SMS thus recognizes certain objects in its knowledge base that are not recognized in standard knowledge bases, say of the first-order sort. Figure 1 schematically illustrates the difference between the two approaches. Uppercase letters are used as variables, and attached numbers distinguish variables having the same range. In Figure 1, Q1 represents a sentential object, and the arrow points to objects associated with Q1. Standard associations, here represented by \{Q1, ..., Qj\}, cover objects related to Q1 based on standard deductive or probabilistic relations. The set \{D1, ..., Dk\} represents nonstandard associations, that is, those that are not related to Q1 by deduction or some defined probability. If, for example, \(Q1 = "\text{John believes Mary is happy,"}\) the assertion "Mary is happy" would be a member of \{D1, ..., Dk\} but not necessarily a member of \{Q1, ..., Qj\}.

\[
\begin{align*}
Q1 & \rightarrow \{Q1, ..., Qj\} \text{ Standard Associations} \\
& \quad \text{SMS Associations} \\
& \quad \{D1, ..., Dk\} \\
\text{(where Q and D range over sentential objects;)} \\
& \quad j \geq 1 \text{ and } k \geq 1 \\
\end{align*}
\]

---

Figure 1. SMS Associations and Standard Associations

Penumbral relations are generated based on named relations and a combination of syntactic and semantic factors. The scope of the semantic base of the system can be increased by generating penumbral associations for the formal assertions that become part of the ACSS knowledge base. As a result, two related regions of the knowledge base become appreciable, one consisting of the "structural given," that is, the formal assertions themselves and their underlying structures (formalisms), and the other consisting of penumbral associations. In Figure 1, the set \{D1, ..., Dk\} represents penumbral associations. Having these associations at hand can enhance the ability of a system to interact with the user. If a query or search is matched in the penumbral region, for instance, the system can give an appropriately qualified response.

For SMS theory to be successfully employed in ACSS it will be necessary to determine the specifications of a knowledge base that would possess sufficient semantic scope to support the classification module, including adequate sets of top-level representations and anchoring, basic-level representations. A system of multiple levels of abstraction is being designed to fix meanings of concepts at the basic level. The example in Figure 2 shows three levels of equivalent meaning.
1. \( <\alpha \ <\text{gave}<\text{at t-1}>\beta \gamma <\text{delighted}<\text{at t-2}>\beta> \) [surface level expression]

2. \( \{ [\alpha \ (\text{caused} \ (\text{bmo} \text{ give})) \ (\beta \ (\text{has \ (at t-1)}) \gamma )) \) caused
\( \) (\beta \ (\text{is \ (at t-2)}) \text{delighted})\) [intermediate representation]

3. \( (\beta \ (\text{has-not \ (at t-0)}) \gamma ) \)
\( \) and
\( \{ [\alpha \ \text{causes} \ (\beta \ (\text{has \ (at t-1)}) \gamma )) \) causes
\( \) \( (\beta \ (\text{has \ (at t-2)}) \text{EMOTIONAL-STATE}) \)
\( \) and
\( (\text{EMOTIONAL-STATE} \ (\text{the})) \text{is \ POSITIVE} \)
\( \) and
\( (\text{EMOTIONAL-STATE} \ (\text{the})) \text{is= \ delighted})\) [lower-level representation]

Figure 2. Levels of Expression

[Note: In Figure 2, \( \alpha \), \( \beta \) and \( \gamma \) are variables (\( \alpha \) and \( \beta \) ranging over persons and \( \gamma \) ranging over things). The arrows indicate that the components of representations 1 and 3 map into representation 2, which functions as an intermediary causal formalism. The term "bmo" flags the term that describes the type of act at the root of the causal relation. The terms "t-0", "t-1", and "t-2" are sequential temporal indicators. The link "has-not" represents the negation of the link "has", and the term "the" attached to the term "EMOTIONAL-STATE" indicates that all the references are back to the first emotional state mentioned. The uppercase terms "EMOTIONAL-STATE" and "POSITIVE" are primitive terms that function to structure lower-level representations.]

The idea is for the system to produce anchoring representations automatically from surface-level representations by employing syntactic functions. A syntactical function \( \Omega \) can be defined to operate on a top-level representation \( \Theta \), for example, to produce an equivalent basic level representation that is optimized for storage and retrieval. Given that \( \Theta \) is expression 1 (surface level representation) of Figure 2, the following would hold:

\[
\Omega (\Theta) = \Delta_{i}<\alpha, \beta, \gamma, t_x, t_y, t_z>
\]

where \( \Delta \) is a name of a basic formalism indexed by \( i \) and where \( <\alpha, \beta, \gamma, t_x, t_y, t_z> \) is an ordered sequence associated with formalism \( \Delta_{i} \) and where members of the sequence have mappings into the levels of representation in Figure 2. In this example \( t_x, t_y, t_z \) map to the temporal indicators in Figure 2. The anchoring representations are to serve as the semantic base of the system.

The design of a semantic pool that can be drawn upon by the system during the course of conducting inferencing and querying operations constitutes an important
step in the creation of a suitable paradigm of a knowledge base that could serve in advanced applications of ACSS technology. The system must have this kind of semantic information available to generate concept maps automatically for classified domains and to establish relations between key terms during text processing operations.

References and Relevant Publications

Books


Articles


The February, 1997, progress report described findings pertaining to the development of an interactive classification tool to be used by original and derivative classifiers to classify and declassify documents. One conclusion was that to be successful in developing such a tool, considerable effort would have to be spent on determining how general models of document content could be related to actual document content. Research during the last period addressed some of the issues presented in this area. The findings were that it might be feasible to combine text processing techniques that employ correlated and weighted word distributions with interpretive processes that access both general models of document content and specific lexical formalisms built for vocabularies chosen for particular domains. The idea would be to construct abstract descriptions of document content from the word distributions by mapping words to semantic operators and lexical formalisms and then relating the formalisms themselves to produce the desired results. The research indicated that the process would be heavily dependent on an ability of the system to classify items at the microscopic level, including the level at which grammatical distinctions are made. The approach would be to proceed in a bottom-up fashion to map content to higher and higher abstraction (formalisms). The overall goal would be to generate the detectable abstract forms for classified information that could be recognized when new documents are processed. Although much more work needs to be done in this area, it should be pointed out that useful results could be produced at varying stages of development. The most promising possibility in this respect would be to develop an interactive tool to be used by original classifiers to classify documents as they are created or first submitted for classification. The next term of research will be spent investigating this possibility in more detail.
Effect of Broken Characters, Touching Characters, and Speckle on OCR Accuracy

Nathan E. Brener, S. Sitharama Iyengar, and Abhinav Rau
Department of Computer Science
Louisiana State University

Michael Cannon
Los Alamos National Laboratory

Thomas A. Nartker
Information Science Research Institute
University of Nevada, Las Vegas

Abstract

Although there are many factors that can affect the performance of OCR engines, it was found that three factors, namely broken characters, touching characters, and speckle, are the most important in determining OCR accuracy. These three factors are commonly found in the DOE Office of Declassification (OD) typewriter-era documents. In order to determine how these three factors effect OCR accuracy, we developed a test suite with 144 pages that contain these factors at various levels and in various combinations. The test suite was used to test the performance of four current OCR engines. The test results showed that all three factors cause substantial OCR errors. The results also showed that of the four OCR products tested, the Xerox OCR engine performs better than the others on all three factors, indicating that the Xerox engine is best suited for processing OD typewriter-era documents.
I. Introduction

Currently the Department of Energy (DOE) Office of Declassification (OD) has approximately 230 million pages of documents that are waiting to be declassified. At present these documents are being reviewed and declassified manually, which is a time-consuming and labor-intensive process. Thus the development of an automated (computerized) review system would lead to a substantial productivity gain in the OD declassification process. The first step in such an automated system is the conversion of the documents to text files using Optical Character Recognition (OCR) engines. In order for the automated system to run successfully, the OCR conversion must be done at a high level of accuracy. Commercial OCR products today are designed primarily for documents produced by laser printers and typically achieve high levels of accuracy on such documents. However, much of the OD document population was produced by typewriters in the 1940's and 1950's and is in sharp contrast to laser printer documents. As a result, modern OCR technologies may not achieve sufficiently high accuracy on the OD typewriter-era documents. In this paper we investigate the factors that cause OCR errors as a first step toward improving OCR accuracy on OD documents.

II. Test Suite of Document Pages

The literature was studied to determine the current state of OCR technology and to compile a list of factors that can cause OCR errors. Although there are many factors that can effect the performance of OCR engines, it was found that three factors, namely broken characters, touching characters, and speckle or noise, are the most important in determining OCR accuracy. These three factors are commonly found in the OD typewriter-era documents.

In order to determine how these three factors effect OCR accuracy, we developed a test suite with 144 pages that contain these factors at various levels and in various combinations. This test suite was selected from approximately 550 pages provided by Scott Crandall and Robert Amsler of DynMeridian. These pages are representative of the lower quality documents that are part of the OD document population waiting to be declassified. These lower quality documents were chosen for the test suite in order to thoroughly study the three factors described above. Using a combination of manual assessment and an automated quality assessment program developed by Michael Cannon [1], each page in the
test suite was marked as having either a low, intermediate, or high level of each of the three factors.

III. Test of OCR Engines

The 144 page test suite described above was used to test the performance of four current OCR products. The OCR testing was done at the Information Science Research Institute (ISRI) at UNLV using the test facilities developed by Tom Nartker’s group [2]. The four OCR engines included in the test are the following:

2. Nankai Reader, Version No. 4.0, Nankai University, Tianjin, China.
4. Xerox OCR engine, Version No. 11.0, Xerox Corp., Peabody, Massachusetts.

The test results are given in the next section.

IV. Test Results and Discussion

The test suite was processed by each of the four OCR engines and a separate performance analysis was obtained for each page in the suite. In addition, an overall performance analysis was done for the entire test suite. Although every page in the suite was processed, the performance analysis presented in this paper will mainly be limited to the pages that have a high level of one or more of the three factors. Pages with intermediate or low levels of all three factors will not be included in most of the analysis given here, although these pages will be included in the overall analysis for the entire suite.

Table 1 gives average character and word accuracies for all pages that have a high level of broken characters, regardless of the levels of the other two factors. Table 2 also represents pages with a high level of broken characters, however, the pages in this table are limited to those in which the levels of the other two factors are low. Similarly, Tables 3 and 5 give average accuracies for all pages with a high level of touching characters and speckle, respectively, and the pages in Table 4 are limited to those with a high level of touching
characters and low levels of the other factors. There is no table given for pages with a high level of speckle and low levels of the other factors because there were not enough pages in this category to form a statistically significant sample.

In Table 6, all of the pages have high levels of both speckle and touching characters, and as a result, the average accuracies in this table are significantly lower than those in the other tables. Table 7 includes all of the pages that have a high level of one or more of the three factors and thus provides a measure of the overall performance of the four OCR engines.

Table 8 gives character and word accuracies for the entire 144 page test suite, including not only the pages with a high level of one or more factors but also those with only intermediate or low levels of all three factors. While the accuracies given in Tables 1-7 were obtained by averaging over all the pages in the table, the accuracies presented in Table 8 were obtained in a different manner. These accuracies were calculated by dividing the total number of correct characters in the entire test suite by the total number of characters in the suite.

In Tables 1-8, the line labeled “None” gives the actual character accuracies achieved by the OCR engines. The following three lines show the accuracies obtained when characters marked by the OCR engines as either reject characters, first level suspect characters, or second level suspect characters are manually corrected. These lines also give the percentage (in parentheses) of the total characters that are so marked. It should be noted that the increase in the accuracy percentage due to the correction of reject characters is often greater than the percentage of characters that are marked as rejects. This is due to the fact that when the user corrects a reject character that is part as a word, he often notices other incorrect characters in the same word and corrects these as well. Although substantial increases in accuracy can be obtained by manually correcting marked characters, the number of such characters is usually large, as shown by the percentages inside parentheses in the tables, and thus this type of correction would not be feasible for an automated system.
Table 1. Average character accuracy (1a) and average word accuracy (1b) for pages with a high level of broken characters. The sample contains 27 pages.

<table>
<thead>
<tr>
<th>Error Correction</th>
<th>Caere Accuracy (Marked)</th>
<th>Nankai Accuracy (Marked)</th>
<th>Recognita Accuracy (Marked)</th>
<th>Xerox Accuracy (Marked)</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>74.96% (-)</td>
<td>79.90% (-)</td>
<td>55.26% (-)</td>
<td>81.29% (-)</td>
</tr>
<tr>
<td>Rejests only</td>
<td>74.96% (0.00%)</td>
<td>87.63% (2.97%)</td>
<td>75.45% (7.07%)</td>
<td>87.80% (2.72%)</td>
</tr>
<tr>
<td>Rejests and 1st Level Markers</td>
<td>83.74% (4.39%)</td>
<td>93.20% (12.13%)</td>
<td>N/A (-)</td>
<td>91.44% (8.08%)</td>
</tr>
<tr>
<td>Rejests, 1st and 2nd Level Markers</td>
<td>86.18% (7.36%)</td>
<td>95.71% (18.93%)</td>
<td>N/A (-)</td>
<td>94.13% (14.62%)</td>
</tr>
</tbody>
</table>

Table 2. Average character accuracy (2a) and average word accuracy (2b) for pages with a high level of broken characters and low levels of touching characters and speckle. The sample contains 8 pages.

<table>
<thead>
<tr>
<th>Error Correction</th>
<th>Caere Accuracy (Marked)</th>
<th>Nankai Accuracy (Marked)</th>
<th>Recognita Accuracy (Marked)</th>
<th>Xerox Accuracy (Marked)</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>76.62% (-)</td>
<td>82.64% (-)</td>
<td>55.51% (-)</td>
<td>85.80% (-)</td>
</tr>
<tr>
<td>Rejests only</td>
<td>76.62% (0.00%)</td>
<td>88.23% (2.25%)</td>
<td>73.73% (6.16%)</td>
<td>90.93% (2.37%)</td>
</tr>
<tr>
<td>Rejests and 1st Level Markers</td>
<td>85.83% (5.20%)</td>
<td>93.30% (9.37%)</td>
<td>N/A (-)</td>
<td>93.81% (7.39%)</td>
</tr>
<tr>
<td>Rejests, 1st and 2nd Level Markers</td>
<td>88.25% (7.85%)</td>
<td>95.54% (15.07%)</td>
<td>N/A (-)</td>
<td>95.83% (12.69%)</td>
</tr>
</tbody>
</table>

Table 1.

Table 2.
Table 3. Average character accuracy (3a) and average word accuracy (3b) for pages with a high level of touching characters. The sample contains 33 pages.

<table>
<thead>
<tr>
<th>Error Correction</th>
<th>Caere Accuracy (Marked)</th>
<th>Nankai Accuracy (Marked)</th>
<th>Recognita Accuracy (Marked)</th>
<th>Xerox Accuracy (Marked)</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>50.27% (-)</td>
<td>58.08% (-)</td>
<td>61.67% (-)</td>
<td>73.42% (-)</td>
</tr>
<tr>
<td>Rejects only</td>
<td>50.27% (0.00%)</td>
<td>73.35% (4.61%)</td>
<td>79.61% (6.34%)</td>
<td>80.00% (1.78%)</td>
</tr>
<tr>
<td>Rejects and 1st Level Markers</td>
<td>70.36% (5.89%)</td>
<td>81.84% (19.75%)</td>
<td>N/A (4.13%)</td>
<td>82.59% (6.07%)</td>
</tr>
<tr>
<td>Rejects, 1st and 2nd Level Markers</td>
<td>75.87% (11.37%)</td>
<td>85.66% (29.89%)</td>
<td>N/A (8.88%)</td>
<td>85.26% (8.88%)</td>
</tr>
</tbody>
</table>

Table 3a

<table>
<thead>
<tr>
<th>Accuracy</th>
<th>Caere %</th>
<th>Nankai %</th>
<th>Recognita %</th>
<th>Xerox %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>30.36%</td>
<td>39.78%</td>
<td>32.99%</td>
<td>59.09%</td>
</tr>
</tbody>
</table>

Table 3b

Table 4. Average character accuracy (4a) and average word accuracy (4b) for pages with a high level of touching characters and low levels of broken characters and speckle. The sample contains 12 pages.

<table>
<thead>
<tr>
<th>Error Correction</th>
<th>Caere Accuracy (Marked)</th>
<th>Nankai Accuracy (Marked)</th>
<th>Recognita Accuracy (Marked)</th>
<th>Xerox Accuracy (Marked)</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>61.30% (-)</td>
<td>69.05% (-)</td>
<td>70.04% (-)</td>
<td>77.56% (-)</td>
</tr>
<tr>
<td>Rejects only</td>
<td>61.30 % (0.00%)</td>
<td>80.64 % (3.72%)</td>
<td>82.87 % (4.64%)</td>
<td>81.38 % (0.86%)</td>
</tr>
<tr>
<td>Rejects and 1st Level Markers</td>
<td>76.21% (4.44%)</td>
<td>88.79 % (18.46%)</td>
<td>N/A % (-)</td>
<td>83.45% (2.25%)</td>
</tr>
<tr>
<td>Rejects, 1st and 2nd Level Markers</td>
<td>80.07 % (7.84%)</td>
<td>91.55 % (28.19%)</td>
<td>N/A % (-)</td>
<td>84.59 % (4.80%)</td>
</tr>
</tbody>
</table>

Table 4a

<table>
<thead>
<tr>
<th>Accuracy</th>
<th>Caere %</th>
<th>Nankai %</th>
<th>Recognita %</th>
<th>Xerox %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>43.46%</td>
<td>50.78%</td>
<td>42.68%</td>
<td>67.65%</td>
</tr>
</tbody>
</table>

Table 4b
### Table 5.

<table>
<thead>
<tr>
<th>Error Correction</th>
<th>Caere</th>
<th>Nankai</th>
<th>Recognita</th>
<th>Xerox</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Accuracy (Marked)</td>
<td>Accuracy (Marked)</td>
<td>Accuracy (Marked)</td>
<td>Accuracy (Marked)</td>
</tr>
<tr>
<td>None</td>
<td>52.21% (-)</td>
<td>42.67% (-)</td>
<td>51.65% (-)</td>
<td>66.13% (-)</td>
</tr>
<tr>
<td>Rejects only</td>
<td>52.21% (0.00%)</td>
<td>66.36% (4.38%)</td>
<td>80.33% (9.21%)</td>
<td>75.62% (2.67%)</td>
</tr>
<tr>
<td>Rejects and 1st Level Markers</td>
<td>73.19% (5.49%)</td>
<td>76.03% (16.46%)</td>
<td>N/A (-)</td>
<td>79.72% (6.55%)</td>
</tr>
<tr>
<td>Rejects, 1st and 2nd Level Markers</td>
<td>77.79% (9.71%)</td>
<td>80.80% (24.42%)</td>
<td>N/A (-)</td>
<td>83.15% (12.33%)</td>
</tr>
</tbody>
</table>

Table 5. Average character accuracy (5a) and average word accuracy (5b) for pages with a high level of speckle. The sample contains 24 pages.

### Table 6.

<table>
<thead>
<tr>
<th>Error Correction</th>
<th>Caere</th>
<th>Nankai</th>
<th>Recognita</th>
<th>Xerox</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Accuracy (Marked)</td>
<td>Accuracy (Marked)</td>
<td>Accuracy (Marked)</td>
<td>Accuracy (Marked)</td>
</tr>
<tr>
<td>None</td>
<td>30.33 % (-)</td>
<td>20.51% (-)</td>
<td>40.71 % (-)</td>
<td>50.56% (-)</td>
</tr>
<tr>
<td>Rejects only</td>
<td>30.33 % (0.00%)</td>
<td>41.89 % (3.90%)</td>
<td>77.01 % (11.67%)</td>
<td>60.94 % (2.58%)</td>
</tr>
<tr>
<td>Rejects and 1st Level Markers</td>
<td>58.33 % (7.31%)</td>
<td>52.64 % (14.21%)</td>
<td>N/A (-)</td>
<td>65.21 % (6.12%)</td>
</tr>
<tr>
<td>Rejects, 1st and 2nd Level Markers</td>
<td>65.63 % (12.83%)</td>
<td>57.81 % (19.52%)</td>
<td>N/A (-)</td>
<td>69.86 % (12.49%)</td>
</tr>
</tbody>
</table>

Table 6. Average character accuracy (6a) and average word accuracy (6b) for pages with high levels of speckle and touching characters and a low level of broken characters. The sample contains 6 pages.
Table 7. Average character accuracy (7a) and average word accuracy (7b) for pages with a high level of one or more of the factors broken characters, touching characters, and speckle. The sample contains 73 pages.

<table>
<thead>
<tr>
<th>Error Correction</th>
<th>Caere</th>
<th>Nankai</th>
<th>Recognita</th>
<th>Xerox</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>61.34%</td>
<td>64.26%</td>
<td>59.08%</td>
<td>75.41%</td>
</tr>
<tr>
<td>Rejets only</td>
<td>61.34%</td>
<td>78.52%</td>
<td>79.27%</td>
<td>82.20%</td>
</tr>
<tr>
<td>Rejets and 1st Level Markers</td>
<td>77.14%</td>
<td>86.11%</td>
<td>N/A</td>
<td>85.49%</td>
</tr>
<tr>
<td>Rejets, 1st and 2nd Level Markers</td>
<td>81.03%</td>
<td>89.52%</td>
<td>N/A</td>
<td>88.19%</td>
</tr>
</tbody>
</table>

(7a)

<table>
<thead>
<tr>
<th></th>
<th>Caere</th>
<th>Nankai</th>
<th>Recognita</th>
<th>Xerox</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>46.92%</td>
<td>52.63%</td>
<td>38.97%</td>
<td>63.29%</td>
</tr>
</tbody>
</table>

(7b)

Table 8. Character accuracy (8a) and word accuracy (8b) for the entire 144 page test suite.

<table>
<thead>
<tr>
<th>Error Correction</th>
<th>Caere</th>
<th>Nankai</th>
<th>Recognita</th>
<th>Xerox</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>69.83%</td>
<td>70.34%</td>
<td>67.19%</td>
<td>81.46%</td>
</tr>
<tr>
<td>Rejets only</td>
<td>69.83%</td>
<td>83.40%</td>
<td>83.15%</td>
<td>86.14%</td>
</tr>
<tr>
<td>Rejets and 1st Level Markers</td>
<td>81.48%</td>
<td>89.24%</td>
<td>N/A</td>
<td>88.34%</td>
</tr>
<tr>
<td>Rejets, 1st and 2nd Level Markers</td>
<td>84.74%</td>
<td>92.01%</td>
<td>N/A</td>
<td>90.34%</td>
</tr>
</tbody>
</table>

(8a)

<table>
<thead>
<tr>
<th></th>
<th>Caere</th>
<th>Nankai</th>
<th>Recognita</th>
<th>Xerox</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>55.95%</td>
<td>60.82%</td>
<td>48.32%</td>
<td>71.57%</td>
</tr>
</tbody>
</table>

(8b)
V. Conclusions

Conclusions Based on Character Accuracy

The character accuracy conclusions are derived from the line in the tables labeled "None" in which no OCR errors are manually corrected. For the three factors considered (broken characters, touching characters, and speckle) and the four OCR engines tested (Caere, Nankai, Recognita, and Xerox), we can draw the following general conclusions based on character accuracy:

1. All three factors cause substantial OCR errors.

2. Most of the OCR engines (Xerox, Nankai, Caere) made fewer errors on broken characters than on the other two factors.

3. Most of the OCR engines (Xerox, Nankai, Recognita) made more errors on speckle than on the other two factors.

4. Thus of the three factors considered, broken characters cause the fewest OCR errors and speckle causes the most OCR errors.

5. Of the four OCR engines tested, Xerox performs better than the others on all three factors.

6. Nankai has the second best performance on broken characters, Recognita has the second best performance on touching characters, and Caere has the second best performance on speckle.

7. The difference between Xerox and the OCR engine with the second best performance is small for broken characters but intermediate to large for touching characters and speckle.

8. Overall (averaging over all three factors), the difference between Xerox and the OCR engine with the second best performance is substantial.

9. Of the four OCR engines tested, Xerox is best suited for processing OD typewriter-era documents.
Conclusions Based on Word Accuracy

Conclusions 1, 2, 5, 8, 9 above for character accuracy are the same for word accuracy. The other conclusions are different for word accuracy, as follows:

3. Most of the OCR engines (Nankai, Caere, Recognita) made more errors on touching characters than on the other two factors.

4. Thus of the three factors considered, broken characters cause the fewest OCR errors and touching characters cause the most OCR errors. An exception to this general conclusion is Xerox, for which speckle causes the most errors.

6. Nankai has the second best performance on broken characters and touching characters, and Caere has the second best performance on speckle.

7. The difference between Xerox and the OCR engine with the second best performance is small to intermediate for broken characters but intermediate to large for touching characters and speckle.

Acknowledgment

This work was supported by DOE under Contract No. DE-FG01-961550317. The authors would like to thank Tom Curtis of DOE for a number of useful discussions.

References


Classification/Declassification of Text Documents via a Logical Analysis Approach, Progress Report No. 3

By Evangelos Triantaphylou, Ph.D.

Department of Industrial and Manufacturing Systems Engineering
3128 CEBA Building, Louisiana State University
Baton Rouge, LA 70803-6409
e-mail: ietrian@lsuvvm.sncc.lsu.edu
(February 24, 1997)

This research project is confined to the development of a computerized system to automatically classify documents emitted by the Department of Energy (DOE). The primary objective of this research study is to extract the features from a set of documents that will determine the documents' classification. In this study, the documents' features will take the form of lexical compounds (i.e., single words, two consecutive words, three consecutive words, etc.)

At the current stage of the project, we are approaching the solution of our first research question: How to use a set of classification guides (given or extracted) in classifying a text document? This research question corresponds to the first problem, of the two problems, defined in our previous two reports. Currently, we are able to extract up to seven lexical compounds from a collection of documents and our next steps are: (1) test and analyze the quality of the output of our computer system with "real" documents (e.g., news with preassigned subjects extracted from the Internet), and (2) generate the rules (i.e., combination of lexical compounds) that will ratify the documents' already assigned subjects.

WORK COMPLETED

After the submission of the second progress report, we have accomplished the following tasks:

1. We generated the specifications for the computer programs to be developed and the specifications for eleven data files required. We completed this task using our findings in the reviewed literature reported in our previous progress report. This work was completed on 10/30/96.

2. We coded eighty-five Turbo Pascal (ver. 1.5 for Windows) computer programs and created eleven data files. The composition of these files is as follows: Sixty programs are used directly for the extraction of the lexical terms whereas the remaining twenty programs were needed to test each of the systems modules. It is important to mention here that all these eighty-five programs totaled two hundred procedures or, equivalently, twelve thousand lines of code (including in-program documentation). This was completed on 2/21/97.
3. All these programs were fully verified according to our initial specifications using seven documents with controlled vocabulary. As a result of these verification, we found that the available lists of stop words does not fully apply to the DOE documents. One example, among others, is the word "not" which was included as a stop word. According to published literature, a stop word should be eliminated for further analysis because it does not convey meaning. This work was also completed on 2/21/97.

FUTURE WORK

Our immediate plan to wrap-up our first problem is as follows:

1. Complete the Phase 0 or Text Preprocessing. In this first activity we will retrieve from the Internet a set of two hundred news-documents (i.e., national news, international news, government, etc.). Then using 80% of these documents, we will generate the rules (i.e., combination of lexical compounds) that will attempt to predict the subject of the remaining 20% of the documents. Next, we will review and modify our computer logic accordingly. (The term subject can be seen as a category, which is the term used in our study and in our previous progress reports). We plan to complete this activity by 3/15/97.

2. The second activity will be similar to the one just described, but this we will use available DOE documents. We plan to complete this activity by 3/15/97.

3. Our third activity will be to Process the documents of each of the categories. In this activity we will use the lexical terms extracted from each of the categories to generate the rules (i.e., combination of lexical compounds) that will attempt to predict the classification/declassification status of the DOE documents. Our expectation of the generated rules is to corroborate the DOE's preassigned tags (i.e., classified, declassified, etc.) with a minimum error. We plan to complete this activity by 04/30/97.

CONCLUSION

Our results in this progress report included very little experimentation; rather, our main concern was the completion of the required programs for the experimentation. Now, with most of the coding done, we expect to have a demonstration by the middle of May 1997.
This report summarizes the progress of the research project "Classification/Declassification of Text Documents via a Logical Analysis Approach" for the period ending August 26, 1997. This research project is confined to the development of a computerized system to classify automatically new documents given a set of classification guides. The primary objective of this study is to scan the text of a document and extract the lexical compounds (i.e., single words, word-pairs, phrases, etc.) that would determine the classification.

The tasks regarding the first problem (i.e., how to use source document, for which a classification status is known, and extract the underlying classification guides) were completed on early June 1997. Some interesting results are described below. Currently, we are working on the second problem (i.e., How to use a set of classification guides in classifying a given document) of the two problems presented in our previous progress reports. Our next step will consist in defining a set of rules to relate the extracted data from a set of documents to the guidelines. A promising approach consists in translating the guidelines into Boolean expressions. The parameters of these Boolean expressions could be condition such as: "a specific data given", "types and amounts of nuclear materials used", "location of a military exercise", etc. Our system, then, should be able to read through a document and decide which of these conditions are satisfied or not.

WORK COMPETED

After the emission of the third progress report we have accomplished the following activities:

1) About twenty thousand lines of code were written in the programming language Turbo Pascal for Windows ver. 1.5. The core of these programs consists in deriving a dictionary of unique indexing terms, derivation of some classification rules by using a Logical Analysis approach, derivation of document categories centroids by using the Vector Space Model, and the computation of the accuracies and CPU times.

2) Two thousand four hundred tests have been conducted on eighty news documents randomly extracted from the CNN website. The CNN news subjects selected were "Science and Technology (ST)", "World (W)", "Sports (S)", and "Health (H)." In these tests we investigated the magnitude of various parameters such as the accuracy of the methodologies, CPU times, number of indexing terms generated, and size of the various data structures used.
The proposed methodology (i.e., the One Clause at a Time, OCAT, algorithm) was benchmarked against the results of the Vector Space Model, which is a popular categorization methodology.

3) The experiments were conducted as follows. First, six category pairs were formed with the four CNN subjects selected, as follows: (a) ST and W, (b) ST and S, (c) ST and H, (d) W and S, (e) W and H, and (f) S and H.

Then, a set of all the cooccurring words was extracted from the documents in each category pair (the size of the cooccurring words was up to 1, 2, 3, 4, and 5 consecutive words, one size at a time was considered). These cooccurring words were considered as the indexing terms for all the documents in each category pair.

Finally, this preliminary experimentation consisted in applying the Round-Robin test on all six category pairs. That is, the number of conducted experiments were: 6 category pairs * 80 documents * 5 indexing terms sizes = 2,400 experiments.

4) Results. After the Round-Robin tests the following main results were obtained.

(a) Accuracy per indexing term size: The OCAT algorithm always outperformed the corresponding average accuracy of the VSM by at least 52 percent. Some examples of the rules obtained by with OCAT algorithm for single words are

i. For the category pair Science and Technology Vs. World:
   IF (the words "CNN" OR "Internet" OR "orbit" OR "using" OR "2000" occur in a document) THEN categorize the document as Science and Technology ELSE categorize it as World.

ii. For the category pair Science and Technology Vs. Sports:
    IF ((the word "July" does not occur) OR the words "expected" OR "Clinton" occur in a document) THEN categorize the document as Science and Technology ELSE categorize it as Sports.

iii. For the category pair Science and Technology Vs. Health:
     IF the words ("mission" OR "Internet" OR "cnn" OR "super") occur in a document THEN categorize the document as Science and Technology ELSE categorize it as Health.

iv. For the category pair World and Sports:
    IF (the words "president" OR "minister" OR "miles" OR "removed" OR "force" occur in a document) THEN categorize the document as World ELSE categorize it as Sports.
v. For the category pair World and Health:
IF (The words "minister" OR "power" OR "police" OR "parties" OR "worker" occur in a document) THEN categorize the document a World ELSE categorize it as Health.

vi. For the category pair Sports and Health:
IF (the words "play" OR "left" OR "player" OR "fivyear" occur in a document) THEN categorize the document as Sport ELSE categorize it as HEALTH.

(b) CPU Times. On these results two issues were obtained: (1) On the average, the OCAT algorithm was four times faster than the VSM and (2) the CPU time of the VSM appeared to be highly sensitive to the number of generated indexing terms in comparison to the CPU time of the OCAT algorithm.

FUTURE WORK

Currently, our efforts are put defining a set of rules to relate the extracted data from a set of documents to some classifying guidelines. A promising approach consists in translating the guidelines into Boolean expressions. The parameters of these of these Boolean expressions could be condition such as: "a specific data given", "types and amounts of nuclear materials used", "location of a military exercise", etc. Our system, then, should be able to read through a document and decide which of these conditions are satisfied or not. The estimated time to complete this activity is December 1997.