The Intelligent Banking System

Kenan Sahin and Keith Sawyer

Consultants for Management Decisions, Inc.
One Broadway
Cambridge, MA 02142

Abstract

This paper describes the Intelligent Banking System (IBS), a family of applications developed for Citibank, New York, by Consultants for Management Decisions (CMD) to increase the productivity and effectiveness of English text message processing. These messages were previously processed manually. Data entry operators would read and analyze the message and then type information at a standard ASCII terminal interface. IBS applies a combination of natural language processing and rule-based expert system techniques to analyze the message and to generate a formatted equivalent. IBS also provides a sophisticated intelligent user interface which aids users by applying the system's domain knowledge to the interactive session.

Introduction

International Banking relies heavily on the electronic transfer of messages for basic transactions. Until the mid-seventies, messages were transmitted over the telex carrier networks, as natural language text. In the mid- to late-seventies, the major international banks developed several industry-wide structured formats to represent the most common banking messages, such as funds transfers. This then allowed the banks to develop computer software which could automatically process the structured transaction, precluding the need for manual intervention. (This parallels the more recent moves to EDI in other industries.)

Despite the widespread success of this strategy in reducing processing costs and increasing bank productivity, a significant minority of the international message traffic remained natural language text. This traffic still required costly and error-prone manual processing. Proficient operators needed a significant understanding of international banking transactions, creating high training costs and limiting staffing flexibility.

Because of the need to process English text input, and the need to incorporate a significant amount of domain expertise, traditional programming techniques were inadequate. Artificial Intelligence technology was identified as the appropriate solution. AI offers two groups of techniques which are used by IBS: *Natural Language Processing* techniques, and *Rule-based Expert System* techniques.

The goal of the Intelligent Banking System (IBS) is to use a combination of these techniques to allow the computer to scan and "understand" a natural language text message. Automating the task in this manner would reduce banking costs, increase operator productivity, and reduce the chance of manual error. The task seemed appropriate for this technology, since the application satisfied many of the accepted criteria (Davis, 1982, and Prerau, 1985):

- The domain is characterized by the use of expert knowledge, judgement and experience
- Conventional programming solutions are inadequate
- There are recognized experts that solve the problem today
- The completed system is expected to have a significant payoff for the corporation
- The task requires the use of heuristics, or "rules of thumb"
- The task is neither too easy nor too difficult
- The system can be phased into use gracefully

IBS System Description

IBS was developed on a custom basis by Consultants for Management Decisions (CMD) for Citibank, New York. IBS was originally developed for the Funds Transfer class of banking messages. The methods and approaches applied to this domain proved readily extensible to other types of banking messages, and IBS has since been extended to several other message domains. Thus, IBS is actually a full family of applications, including the following modules:

- Funds Transfer message processing
- Letter of Credit Issuance message processing
- Letter of Credit Reimbursement message processing
- Funds Transfer problem inquiry message processing
- Message classification (involving an analysis of all telex traffic to determine which domain-specific module is appropriate)
- Testkey parameter identification

All of these modules are fully integrated in various production environments.

Several of the modules are implemented on-site at Citibank. These systems are fully embedded in the existing bank processing systems, and are processing live telex traffic on a daily basis. Several of the modules are also installed in two telex carriers' data processing facilities: Western Union/ITT, and TRT. These telex carriers are offering IBS message enhancement as a service to their banking customers. IBS is fully integrated with the production processing environment at these sites as well.

The funds transfer application, because it was the first to be developed, is perhaps the most mature system. This system is accompanied by the Intelligent Banking Workstation (IBW), which allows an operator to review an IBS-processed message using a window-and-mouse based interface. IBW allows intelligent entry, which provides the user with full access to the knowledge capabilities of the system. IBW also provides the user with explanations of the various actions taken by the system during the parsing and resolution phases. IBW increases operator productivity further by allowing the operator to make use of partial information identified by the system.

The other applications are not currently accompanied by their own user interfaces, but instead are integrated so closely with the existing production systems that the standard terminal interfaces of the bank processing systems can be used.

The Funds Transfer module has been in production since 1985. This effort followed the original prototype, which was completed by CMD in mid-1984. The total calendar time for the effort to transform the prototype to full production was approximately eight months. This initial production system was implemented on a dedicated LISP machine and was networked using a custom-developed protocol. This implementation proved to be inappropriate for full production, and the system was ported to the VAX environment in 1986.

The later applications began during 1986 and 1987, and were each designed from the beginning for production implementation on the VAX platform (only a brief prototype phase was included).

Technical Description

The knowledge domain for IBS is the reading and translating of English text messages into a structured format. The messages are sent to Citibank electronically via both internal, proprietary networks, and external telex networks. Depending on the message type, the message can be from 80 words long to several pages long.

Message Characteristics

The subset of English used in these messages is highly terse and abbreviated. The people typing in the message are under time pressures, so abbreviations and typographical mistakes are common. Many of the messages are entered by people for whom English is a second language. Often information which is necessary for the recipient, but not required of the sender, will be omitted to reduce the sender's message entry time. For example, the name of a bank is often specified without the corresponding account number. In many messages, information is supplied which is not needed by the recipient; this information must be ignored.

The domain is such that a direct mapping from individual phrases to structured values is not possible. The structured values depend on the context of the entire message. Some structured values depend on several different phrases in combination. Some values may depend on a particular combination of yet other structured values. The possible combinations of situations resulting in a given value are thus very large. Application domains in which combinatorial effects become significant usually do not submit to a cost effective, traditional programming solution. These

complexities are an indication that artificial intelligence techniques may be appropriate.

In addition to these domain requirements, the production environment required that each message be processed in under sixty seconds.

System Design

IBS makes use of a hybrid approach, borrowing ideas from several significant concepts in computational linguistics and in rule-based expert systems. The abbreviated version of English found in these messages led to the use of a *flexible parser* approach (Hayes & Mouradian 1981). The system combines elements of *case frame grammars* (Fillmore 1968) and *semantic grammars* (Hendrix 1977) to arrive at the final linguistic formalism.

This formalism was designed using a variation of the *augmented transition network* (Woods 1970) to build *semantic units*. Each semantic unit is responsible for the identification of one key piece of information from the telex. As information is identified, it is stored within the semantic unit.

The characteristics of our formalism satisfied the domain requirements:

1. The formalism was capable of identifying single phrases and incomplete sentence fragments

2. The formalism was able to identify useful information and ignore irrelevant information

3. The formalism provided for the identification of abbreviations and misspellings

4. By taking maximum advantage of the domain constraints, the formalism provides for highly efficient processing of the English text

In addition to this linguistic formalism, we employed a rule-based expert system to incorporate domain knowledge. The expert system receives input from the semantic unit values. This expert system is used to make decisions based on overall message content, to infer values using combinations of phrases, and to implement constraints among different structured values. This expert system was also custom-developed to achieve production-level speed performance. In the current version of IBS, the rules have been rewritten directly in LISP code, resulting in a tenfold performance increase.

The true originality of IBS lies in its unique blending of several different research concepts to result in a system which satisfies a specific business goal. Despite the use of these fairly advanced concepts, IBS can still process an average telex in 30 seconds on a machine as small as an IBM PC.

The Intelligent User Interface

IBS was designed to process a message fully, then to pass the message and the corresponding structured information to a user edit interface. IBS identifies an average of over 80% of the structured information. An operator must complete the remaining structured information, usually one or two values.

Many bank's data entry stations cannot support display of both the message and the structured equivalent. These interfaces were designed to be used with a printed copy of the message, and provided for structured value entry only. Designing IBS to print out the telex for these operators would have reduced the cost-effectiveness of the process considerably. Instead, we implemented an *intelligent assistant*, the Intelligent Banking Workstation (IBW) as a companion to IBS. IBW was conceptualized as a low-level assistant to a human operator which would provide much of the processing expertise, freeing the operator to perform higher level conceptual activities (Rich & Waters 1981).

IBW employs mouse cursor control, multiple windows, and pop-up menus and windows to improve operator productivity. Two primary windows are displayed: one containing the original message, and the other containing the structured values identified automatically. The mouse can be used to mark a region of text in the message window, and move that text into one of the structured values.

Incomplete or ambiguous values identified by IBS are made available to the user through pop-up windows. One such window is for English-text *notifications* of problems encountered during processing. A second window contains *suggested values* which are each mouse-selectable. For example, if several branches are found in the IBS bank database for the name "CREDIT SUISSE," the notification window would say "Several branches found for CREDIT SUISSE," and the suggested values menu would display each of the branches, with the corresponding city and account numbers. This mechanism allows the user to benefit even when IBS

cannot uniquely identify a value.

The linguistic and domain knowledge used in the automatic processing is also available to the user. For example, when a region of text is moved to a structured value using the mouse, the user can request *intelligent processing* for that text. The parsers and domain rules are then invoked to process the text. A correctly processed value will be entered by IBW. In addition, other values which may have been affected by this change will be *flagged* with a notification for the user.

This intelligent user interface is a significant value-added component of IBS. The power of the "intelligent assistant" concept, employing mouse cursor control, multiple windows, and pop-up menus, significantly increases the productivity of the users. Providing a broad interface between the users and the intelligence in the system results in maximum value for the knowledge engineering effort.

Success Criteria and Payoff

The application was determined to be successful if over 80% of the information in an average telex was identified automatically, and if each telex could be processed in a short-enough period of time to be cost-effective (approximately 30 seconds for most message types). Each of the many IBS modules has met or exceeded this criteria. At this success level, implementation of the module is considered cost-effective.

The payoff for each of the modules varies depending on the specifics of the installation site. Since each module is installed in varying configurations (telex carrier site vs. money-center bank) the cost savings do vary. Generally, the savings can be characterized by reduced head count, increased customer satisfaction, and lower cost resulting from data entry error.

Acknowledgments

The authors would like to emphasize that IBS is the result of a collaborative effort involving many people too numerous to mention. The authors would like to recognize the key efforts of John Hodgkinson at CMD, and the critical efforts of all involved at Citibank.

References

Davis, R. "Expert Systems: Where Are We? and Where Do We Go From Here?," MIT AI Memo No. 665, 1982.

Fillmore, C. "The Case for Case." In E. Bach and R. Harms (eds.) *Universals in Linguistic Theory*. New York: Holt, Rinehart, and Winston, 1968.

Hayes, J.H. and G.V. Mouradian. "Flexible Parsing." *American Journal of Computational Linguistics*, 7 (1981), 4, pp. 232-242.

Hendrix, G.G. "The LIFER Manual: A Guide to Building Practical Natural Language Interfaces," Technical Note 138, SRI International, 1977.

Prerau, D. S. "Selection of an Appropriate Domain for an Expert System." *AI Magazine*, 6 (1985), 2, pp. 26-30.

Rich, C., and Waters, R.C. "Abstraction, Inspection, and Debugging in Programming." MIT AI Memo No. 634, 1981.

Woods, W.A. "Transition Network Grammars for Natural Language Analysis." *Communications of the ACM*, 13 (1970), pp. 591-606.