AN ASSISTANT EXPERT SYSTEM:  
ASSISTING ASSISTORS IN ASSISTING TAXPAYERS

Thomas J. Beckman
Internal Revenue Service
AI Lab, Research Division
1201 E St., Room 604, NW
Washington, DC 20224
(202) 376-1582

ABSTRACT

The Internal Revenue Service (IRS) employs 5,000 assistors to answer telephone inquiries from taxpayers. This paper describes the design and development of an expert system, the Taxpayer Service Assistant (TSA), that helps assistors provide advice to taxpayers on tax law topics. As measured by GAO [1], these assistors provide correct and complete answers on only 63% of the 18 million calls on tax law that are received yearly. The IRS is exploring text retrieval and expert systems as alternative methodologies for supporting assistors and improving the quality of service. This paper examines the trade-offs made by each approach and shows why expert systems are more likely to be effective.

BACKGROUND

Description of the Existing Task

Taxpayer Service assistors are the IRS personnel who respond to telephone inquiries from taxpayers. Front-line assistors are the first persons with whom taxpayers speak when they call toll-free telephone assistance sites. Initially, front-line assistors identify and refer calls that are not related to Taxpayer Service issues, and then categorize the remaining calls into account, procedural, or technical areas. The TSA addresses only the latter -- technical questions on tax law.

Front-line assistors are employees with limited experience who answer frequently-asked questions on a variety of simple tax topics. Initially, they receive four weeks of classroom training and two weeks of on-the-job training. Even after this intensive training, front-line assistors are not expected to answer questions from memory; rather, they are expected to know where to find answers in IRS publications. Assistors consult IRS publications to locate text or confirm answers on about half of the calls they receive [2]. Surprisingly, although front-line assistors are generalists who may encounter questions on over 130 different tax topics, they are able to answer 84% of taxpayer questions without referral.

When front-line assistors recognize topics in which they have not been trained, they refer callers to more experienced assistors -- technical backups and specialists. Technical backups are assistors who have completed all phases of assistor training and have at least one year of experience; specialists are local domain experts who have completed specialist training and possess considerably more experience. To answer these more difficult inquiries, technical backups and specialists often refer to sources such as IRS Publications, IRS Letter Rulings, and the CCH Standard Federal Tax Reporter.

Task Scope

During fiscal year 1986, 5,000 assistors responded to 38.2 million telephone inquiries, of which 18 million calls (47%) were technical questions on tax law [3]. Questions can fall into one of 139 general tax topic categories. Even simple tax topics require assistors to remember up to 10 key pieces of information; more difficult topics can often exceed 50 parameters. Any inquiry may require research into one of 159 IRS publications or into one of 10 volumes (totaling over 15,000 pages) of the CCH Standard Federal Tax Reporter. According to GAO [1], performance by front-line assistors in answering tax law inquiries prepared by GAO was as follows: 63% were correct and complete, 15% were correct but incomplete, and 22% were incorrect.

DESIGN ISSUES

Two Approaches: Expert Systems Vs. Text Retrieval

Currently, there are two projects under development at IRS that attempt to improve performance on the assistor task. The TSA takes an expert system approach; the Automated Taxpayer Service System (ATSS) focuses on text retrieval. The TSA provides advice and explanations based on
taxpayer responses entered by assistors; the ATSS locates and retrieves relevant pieces of text from a database of IRS publications based on entering appropriate keyword combinations.

Text Retrieval Approach The approach taken by ATSS augments the present way that the assistor task is done. Rather than trying to find relevant sections of text by locating the correct IRS publication(s) and then using tables of contents and indexes, text is accessed directly by entering keywords. When keywords match words in the text, sections of text which are retrieved are displayed. Keywords are often logically combined in order to restrict the amount of text retrieved. Text is retrieved from a database containing IRS publications, the tax law and its IRS interpretation, and other tax reference guides.

One advantage of this approach is that tax law databases and extensive off-the-shelf text retrieval packages are readily available from commercial sources. Also, vendors keep their tax databases current through frequent updating. A further advantage is the relatively low cost-per-user of providing this capability. Finally, this approach requires no extensive software development for its use.

One disadvantage of the text retrieval approach is that assistors must learn how to use keywords in logical combinations in order to select relevant text sections. Revenue Canada's experience, although positive, indicates that at least one week of classroom training and several weeks of usage were needed to reach proficiency [4]. Assistors also must learn how to narrow the scope of retrieval so that only the most relevant pieces of text are selected. An unexpected drawback of text retrieval systems is the limited amount of text that is visible on a screen at any one time, as compared to printed media.

However, the most serious deficiencies of this approach go to root of the assistor task. Namely, text retrieval methods still leave the assistor with the tasks of reading, understanding, interpreting, and restructuring text into forms that can be used directly to answer taxpayer questions. In addition to arriving at accurate and complete answers, assistors must often provide explanations of their answers at a level that is understandable to taxpayers. Here again, the assistor must formulate a cogent explanation from text; in most cases, simply reading relevant text is not adequate because decision parameters and their allowable values are not self-evident.

Expert Systems Approach The second approach, the one embodied in the TSA, uses frames and rules from expert systems technology to restructure IRS publications and tax law text into an active form. The TSA implements algorithms and decision trees so that the domain knowledge can be used directly to perform the assistor task. There are several advantages:

1. Only relevant data and their relationships are accessed.
2. Knowledge is accessible in a readily usable form.
3. Knowledge is better organized.
4. Knowledge is represented more explicitly.
5. Explanations show how decisions and advice were arrived at.
6. At any point in the session, the current state of the system can be explained, along with parameters still needed to reach an answer.

Use of expert systems technology shifts the burden of knowing, remembering, and reasoning about details of technical tax law from the human mind to the computer. The TSA eliminates the need to refer to IRS publications in answering taxpayer questions. The system aids assistors by supplanting their existing tax knowledge, or, if needed, by providing a comprehensive source of knowledge when a topic is unknown to them.

The TSA is expected to reduce referrals of taxpayer questions to specialists, allowing those specialists to concentrate their expertise on answering only the most complex cases. In addition, the author believes that the TSA will reduce "answer shopping" (repeat calls on the same question by taxpayers looking for a more favorable answer) by providing more accurate, complete, consistent, and unambiguous advice. Furthermore, if printouts of consultation sessions were made available to taxpayers upon request, this might reduce the need for taxpayers to consult IRS publications and mailing them to taxpayers. Finally, use of the TSA should result in several training benefits. These measures are expected to improve the overall effectiveness of TPS personnel.

However, there are disadvantages. The main disadvantage is that expert systems are hand-crafted -- parameters, their relationships, and allowable values, must be determined through a lengthy process of knowledge acquisition from printed materials and knowledge elicitation from experts. In addition, screen interfaces must be designed very carefully with the average user in mind. Thus, the time re-
quired for system development is quite significant, especially if the entire breadth of the tax code is to be represented.

A second drawback is that expert systems require considerably greater computing resources than text retrieval systems, resulting in a higher cost-per-user. However, this differential cost will likely moderate over time due to continuing cost efficiencies in both hardware and software. A final unresolved issue is determining the most effective configuration for system deployment.

Cognitive Issues in the Assistor Task

Assistors are in the unusual position of having their knowledge tested on a daily basis. To be successful, they must find solutions to three basic cognitive problems: managing task complexity, making text knowledge usable, and learning heuristic shortcuts and strategies. A related cognitive issue involves training -- namely, what is the most effective way to teach the assistor task? Although the design of the TSA does address these issues, a satisfactory solution has not yet been found for the training issue.

SYSTEM DEVELOPMENT

Objectives for the TSA Prototype

System objectives were derived from analyses of the assistor task, feedback from assistors regarding their needs [2], and constraints imposed by the state of AI technology. The following objectives summarize the basic capabilities desired in the feasibility prototype:

1. Provide expertise that enables front-line assistors to answer technical tax questions directly without accessing research materials or referring inquiries to specialists.
2. Explain the system's reasoning both during and after the consultation session.
3. Justify the system's advice at the end of the session.
4. Send a hardcopy of consultation sessions to taxpayers upon request.
5. Use the system as a training aid to test assistor knowledge of the TSA topic through the use of examples and exploratory learning.

Choice of a Specific Tax Topic

Many factors were considered during the topic selection process, such as potential improvement in the quality of service, topic complexity, ease of exploring system functionality and measurement of system performance, and validation of system accuracy. Potential topics were also evaluated from analyses of report data [5], and from a questionnaire [2]. These efforts turned up two tax topics that were considered most promising -- the dependency test and depreciation.

Report data [5] revealed that even questions on the most frequently-asked topics amount to only 2% of total call volume; and many of the higher volume topics have low error rates of between 3% and 7%. However, the dependency test is one of the few exceptions -- with a volume of 1.2% of the total calls and an error rate of 12%.

Good domain topics require use of heuristics and judgment in arriving at an answer. Surprisingly, in performing the assistor task, not many tax topics actually possess this feature. Because assistors learn only the IRS interpretation of the tax law, heuristics play only a minor role in most tax topics. Depreciation is one of the few exceptions -- a topic that possesses heuristics independent of tax law interpretations.

However, depreciation turned out to require more real-world general knowledge for classifying objects than seemed wise in a prototype. Thus, it was decided that heuristics were not needed to add to the already substantial issues that arose during development of the system. Therefore, for all of the above reasons, the dependency test was chosen as the initial "sliver of knowledge".

Knowledge Acquisition and Elicitation

The process of knowledge elicitation was atypical. The author constructed the initial demonstration prototype by extracting information from IRS publication 501, "Exemptions and Standard Deduction". Once the prototype was completed, domain experts critiqued and evaluated the TSA for accuracy and efficiency. Feedback from these specialists resulted in various changes to the prototype: correcting errors and refining the tax knowledge base itself, restructuring the way in which knowledge is represented, and improving the phrasing of questions. Although as yet unfinished, specialists are working on determining the most efficient sequence of questioning.

Role of the System and Level of Abstraction

An expert system can play one of several roles: agent or decision-maker, advisor, assistant, or tutor [6]. Each of these roles places different demands on the design of the system. The expertise of the average user is crucial in deter-
mining the system's role and level of abstraction. For example, an agent system must embody all relevant domain knowledge, and some common-sense knowledge; or it will not be relied upon. An advisor for experts need only provide answers from minimal inputs using jargon; an explanation is probably not needed or desired. However, novice assistants might prefer a wordy agent, assistant, or tutor system that provides extensive explanations, defined terms, and detailed decisions made by the system during the questioning process.

The TSA was initially envisioned as an assistant to experienced assistants. The design changed substantially when its role was modified to permit its use by novices. The TSA shifts answering of complex, but clear-cut straight-forward topics normally answered by specialists to less-experienced assistants. In these cases, the system serves as an assistant to novices, but also acts as an agent by replacing specialists. In fact, the dependency test has sections on multiple support and divorced/separated support that previously were referred to specialists to answer; these sections are now answered by front-line assistants. This allows specialists to focus their energies on questions that truly require their expertise.

The TSA also plays the role of a crude tutor. It provides remedial tutoring through use of its exploratory mode, or more formal training through use of its examples. In either of these roles, the explanation facility is quite important, as is the ability to define technical terms. In a sense, the TSA arrives at operational meanings of terms by making detailed decisions. Finally, if relationships between data and their taxonomy were displayed graphically, it might help novice assistants to form their own mental models of tax knowledge.

**Reasoning Methods**

In addition to the basic design approaches mentioned above, there are other issues to be considered. A basic design decision involves choosing between forward and backward reasoning. In this task environment, most questions involve validating a single taxpayer hypothesis ("Can I deduct my dog?"), rather than generating and evaluating multiple hypotheses ("What sort of deductions can a traveling salesman take?"). Therefore, a backward-chaining inferencing mechanism would seem preferable, due to the smaller-sized tree traversed.

However, even with the theoretical advantages of backward-chaining, the task domain itself imposes more important constraints on inferencing. It turns out that taxpayer inquiries better resemble unstructured conversations, or taxpayers telling stories [2], rather than structured interchanges in which the assistant controls the order of questions asked and data received. Backward-chaining forces a sequence of questions that may not correspond to the order in which data are received. Only forward-chaining provides this needed flexibility.

The notion of progress, and being able to see more than just the current question being asked, is another issue that impacts on the inferencing method chosen. In preliminary demonstrations, assistants appeared to prefer display of several related pieces of data at one time on a screen, including a summary of the system's current status, rather than a display of only one question at a time. This also tends to make forward-chaining preferable, because most backward-chaining shells follow the MYCIN model in only allowing one question to be answered at a time.

However, from the standpoint of efficiency, assistants want to ask questions in an order that will minimize the time spent per call. This is complicated by the fact that some questions require much longer to answer than others. For example, in the dependency test, taxpayers are more likely not to qualify for the exemption because of the amount of support provided to the dependent, rather than whether the dependent is a U.S. citizen, but it may take much longer to answer the support test questions.

Using prompts in a forward-chaining system yields two advantages: flexibility and optimization. Prompts suggest the best sequence in which to ask questions, based on answers received thus far. This not only improves assistant performance, and provides more prompt service to taxpayers, but also serves as a training aid.

**Knowledge Representation**

The author has learned much about the structure of tax law knowledge through discussions with a domain expert [7]. One outgrowth of these discussions is that tax topics in the TSA are structured into decision taxonomies comprised of "tests" and "options". All tests within a tax topic must be passed in order for the topic to be passed. However, only one option need be passed in order to pass a test. In the dependency test, the decision tree contains both tests (AND nodes -- all parameters must be true) and options (OR
were several reasons for this choice. First, the hierarchical, well-organized structure of the tax law lends itself to a frame lattice hierarchy with inheritance, default values, daemons, and handlers would make this task easier. Finally, by modularizing and structuring the knowledge, code could be shared and reused in other expert system applications.

However, a strictly object-oriented approach had its problems, the most serious being daemons. Without the chronological trace provided by the rule-based system, it quickly became difficult to debug and discern the effects of the daemons, much unstructured Lisp code is written to accomplish what rules can do as efficiently, and with better organization. Also, with daemons, one has very little control over the order of their firing; with a rule-based system, the inference engine controls rule firing through conflict resolution, agendas and other built-in mechanisms. As a result, daemons were used sparingly, mostly for initialization of variables.

Because of the difficulties in implementing the object-oriented approach, ultimately frames and rules were used to represent knowledge in the TSA. Even though there are four levels in the knowledge taxonomy for the dependency test, a flat frame structure proved easier to implement, especially since inheritance was not needed. In the TSA, rules are used for a number of purposes, such as to prompt users, draw conclusions, add text for later explanation, control screen display, and build explanations.

The dependency test requires over 95 slots for parameters and over 160 rules concerning relationships between parameters. In addition, a like number of slots in screen frames are needed to display the parameters, and about 25 rules are required for screen control. Another 20 rules prompt assistants as to which question to ask next. The explanation facility consists of 35 slots, six rules, and roughly 100 lines of custom code.

Uncertainty

There are several aspects to uncertainty in the domain of tax law advice that are rather different than those found in other domains. The assistant's advice is based on the IRS interpretation of the tax law. Although this interpretation is usually consistent, there are some points of conflict, and areas exist in which no formal interpretation exists. In either event, it is important that any advice given be consistent and explainable.

Another problem area resides with data provided by the taxpayer. If data needed to provide a definitive answer are either unknown, uncertain, or inconsistent, the TSA would make this task easier. Finally, by modulating and structuring the knowledge, code could be shared and reused in other expert system applications.

At present, when taxpayers do not have needed pieces of information readily at hand, assistants give general answers and offer to mail out relevant IRS publications to taxpayers. In these situations, taxpayers are forced to determine the correct answer by themselves at some later time when both publication and data are available. A typical example that arises in the dependency test is assistors requests for income and support data -- items that many taxpayers do not immediately have available when they call.

A final issue involves measures of certainty. In the dependency test topic, providing an "unknown" value for most parameters is not only desirable, but essential. Entering "unknown" as a value permits the system to continue collecting data, drawing conclusions, and providing contingent answers that are as accurate and as complete as possible.

However, for topics such as depreciation questions, it is quite possible that multiple answers might exist, presenting a dilemma to the designer: accuracy vs. clarity, comprehension, and confidence.

Training Aide

At present, training is facilitated in several ways. First, the system contains examples to illustrate how
typical calls might be answered. Second, the system can be used in exploratory mode. Assistors can enter values and see consequences in the actions that the TSA takes. When values are changed, assistors can see which data elements are dependent on others. Finally, students can examine two types of explanations that justify the system's answer.

Explanation Facility

The TSA provides two types of explanation, one for taxpayers, and another for assistors (see sample session). Taxpayers receive advice and an explanation in English sentences. Assistors see a screen in an abbreviated, tabular format. In addition to seeing the results from each test, assistors also see results from all options answered by the taxpayer. Each line in the table consists of option or test name, allowable values, and values given by the taxpayer.

The concept of building explanations as rules are fired is an appealing one. The author tried to use the minimal explanation facility that comes with GoldWorks to accomplish this -- it did not work. Even after receiving assistance and code from Gold Hill, their approach of tracing back through the rule tree after an answer had been reached was much too complex. Rules that possessed "or" clauses on several levels required one to trace back through multiple paths, clearly a bad idea. The author next attempted to build explanations using daemons. However, it was impossible to control the sequence of execution of daemons and rule retraction when using Goldwork's logical dependency feature.

Rather than tracing backwards after a result had been reached, the author tried adding text strings to specific slots each time rules containing text fired. This works fine until a parameter's value is changed. Then there is no easy way of identifying and changing the value of the corresponding text. However, by using the logical dependency feature provided with GoldWorks, whenever a parameter value is changed in the antecedent of a previously fired rule, all assertions in the consequent of that rule are retracted, thus deleting the corresponding text. In addition, to ensure that only the correct text string is retracted, the text for each type of explanation that is asserted must be placed in a separate string within separate frame slots. Finally, when an explanation is requested, it is built directly from string slot fragments.

Another advantage of this approach is the ease with which contingent answers can be built. A recent addition to the TSA is the ability to give simplistic contingent explanations when data are either missing and/or unknown. Of course, if any test is failed, then no contingent answer is needed. For all tests that are unanswered or have unknown as an answer, references are made to the relevant IRS publication, and the options and their allowable values are displayed (not yet implemented).

User Interface

How pieces of information and their relationships are displayed can be crucial to user acceptance. Experienced assistors might prefer an abbreviated display of relevant parameters and their legal range of values. On the other hand, novice assistors might prefer a more guided question and answer format with the ability to define terms.

It is important that all actions taken by the system be displayed so assistors can see the effects of entering data. Otherwise, the system's "reasoning process" is opaque, and users will be more tentative in trusting system results. Also, displaying system actions has proved quite helpful in debugging the TSA.

The level of abstraction at which the system is targeted should depend on the expertise of the average user. The TSA assumes an assistant of modest experience. Therefore, two status lines near the top of the screen display the progress made thus far -- which tests are passed, failed, or unknown, and which tests have not yet been answered (see sample session). This provides a summary view of the status of a call and also makes the structure of the domain knowledge more explicit.

Moving towards the direction of system primitives, assistors will be able to request more detailed information from the system. Details such as the meaning of terms or the meaning of questions can be asked for data will be displayed as popup windows. This feature has not yet been implemented.

Actions taken and conclusions drawn by the system that are consequences of assistor actions are displayed as comment lines at the bottom of the screen (see sample session). As much as possible, the system's actions are displayed as comments. Assistors can see the effects and consequences of making changes in data known to the system. Prompts also appear at the bottom of the screen to suggest what question is best to answer next. However, assistors are free to choose their own course of action.
FUTURE DIRECTIONS

future Project Development

The strategies to be used in developing later phases of the project may differ substantially from those used in the prototype. Several experienced practitioners [8] have expressed the belief that development of serious systems will inevitably involve building custom shells and interfaces. Based on limited experience with developing an explanation facility, the author is rapidly approaching the same conclusion.

There are several stages of project work still required to reach a fully operational system. The first stage, the feasibility prototype, is described extensively above. In the second stage, the system's functional capabilities will be developed more fully. The third stage expands the scope of the knowledge base to include many specialized topics. The fourth stage enhances the user interface, and implements the TSA for operational use by several levels of assistants. A final stage might involve constructing a "bombproof" version of the system for direct taxpayer access, either in person at IRS sites, or by computer and modem access at remote sites, such as commercial computer databases.

For the prototype phase, only one "slice" of knowledge is described. In later phases, many specialized technical tax issues will be added. Expanding the TSA's functional capabilities is much more difficult than expanding its breadth of knowledge. Capabilities such as intelligent tutoring, understanding natural language inputs, machine learning to optimize ordering of questions, providing for error-resistant knowledge acquisition and modification by specialists, and generating flexible text are all active research areas, with few successful commercial applications.

The following objectives will be pursued to varying degrees in later phases of the project:

1. Provide extensive tutoring and training of assistants on issues covered by the TSA. Teach novice assistants who are learning the material for the first time. Provide a refresher course for more experienced assistants.
2. Syntactically parse textual inputs with limited vocabularies. Understand limited natural language inputs semantically using conceptual dependency.
3. Learn from experience to optimize the ordering of remaining data needed to be input to the system to reach a decision. The assistant would be prompted as to which data to ask for next. This would be implemented using genetic algorithms built on top of a rule-based system [9].
4. Allow specialists to modify the knowledge base without assistance by designing a front-end knowledge acquisition module that would perform validity and consistency checks on these modifications to ensure an accurate and complete knowledge base [10] [11].
5. Allow assistants to customize the interface to best suit their individual needs and style.
6. Provide interface and explanation facilities that accommodate multiple levels of knowledge of assistants and/or taxpayers.
7. Make the structure of knowledge visible by giving it graphical form as an animated flow chart or algorithm. If users can see hierarchial decision trees in motion, then they can create similar mental models, as well as know what data are still needed for a decision.
8. Facilitate knowledge acquisition among multiple experts [12].

REFERENCES

In order to claim your dependent as an exemption, you must pass all five dependency tests. Based on the information you have given us, you MAY claim the person as your dependent.

EXPLANATION: You have passed all five dependency tests as follows:
- The Relationship Test was met because the person is your CHILD.
- The Citizenship Test was met because the person is a U.S. citizen.
- The Joint Return Test was met because the person was not married.
- The Gross Income Test was met because the child was under 19 years old.
- The Support Test threshold of 50% was met by providing 60% of support.