Certain types of data, for example, texts and free-form interviews, tend to lend themselves to being analyzed in a qualitative fashion because they are hard to interpret and hence code. The ability to analyze and code such data often requires one to have an in-depth knowledge of the sociocultural environment related to this protocol, to be, in effect, a social expert. Social expertise involves both knowing more than the novice about the sociocultural environment, and knowing the level of consensus among members of the sociocultural environment to each piece of information. Knowing the level of consensus allows the social expert to make explicit the information that is implicit in the verbal protocol. It is often prohibitively expensive, and sometimes not even possible, to provide novice coders with the extensive training necessary to interpret and code such verbal data as would an expert on that sociocultural environment. By integrating theories of the nature of knowledge, social expertise, and coding it is possible to develop a cognitive foundation on which an expert system for coding verbal protocols can be built. In this article a two-stage computer-assisted procedure for coding verbal data is presented. This procedure minimizes the amount of training needed by novices by utilizing the expert's knowledge of social knowledge to improve the novice's coding of the verbal data. In stage 1 the novice uses CODEF, a computer-assisted procedure for coding verbal data on a particular topic as a knowledge base. In stage 2 the knowledge base is processed by an expert system, SKI. This expert system has as its knowledge base social knowledge and as its inference engine a cognitive model of the way the social expert uses this knowledge. SKI uses the expert's knowledge of social knowledge to explicate implicit information in the protocol and thereby diagnose and correct errors of explication in the novices coding of the protocol. By employing SKI it is possible to increase the level of reliability and replicability in the coded data without having the expert code each verbal protocol and without spending extensive time training the novice. This research is indicative of the way expert systems and the findings in artificial intelligence can be used in the social sciences.

Formalizing the Social Expert's Knowledge

KATHLEEN CARLEY
Carnegie-Mellon University

EDITOR’S NOTE: This article was accepted under the editorship of George W. Bohrnstedt.

being analyzed in a qualitative fashion because they are hard to interpret and hence code. Some individuals are more expert than others at coding verbal protocols either because they know more about the process of coding and what needs to be coded or because they know the necessary social background information to make explicit information that is implicit in the protocol. Due to time and resource constraints the expert can code only a limited number of protocols. This suggests that expert system technology might be an appropriate methodology for automating the coding of verbal protocols. Cognitively based theories of the nature of knowledge, the nature of social expertise, and the nature of coding make it possible to develop such an expert system. In this article a computer-aided system for coding verbal protocols that utilizes the social background information known by the social expert,¹ that is, an expert system,² is presented. Such a system makes it possible to do in-depth studies utilizing a large number of cases. Further, such a system, because it emulates the social expert’s coding behavior, releases the social expert from the tedious job of coding all the protocols and obviates the necessity of extensively training novice coders. And finally, automation increases reliability as it ensures that the implicit rules used by the social expert to code the data are explicated, recorded, and then routinely applied.

This process of using an expert system to improve protocol coding is illustrated using data drawn from a large study of the dynamic interrelationship between the social structure and cognitive perceptions of a group of undergraduates at MIT involved in the selection of a new tutor, that is, graduate resident, of their living group, Third East. The original study focused on whether or not the students’ positions in the social structure affected their definition of the concept tutor as their knowledge about tutoring evolved during the tutor selection process (Carley, 1984, 1986a). CODEF and SKI³ were used to code those portions of the interviews with the students that dealt with their definition of the concept tutor. The expert’s vocabulary contains the set of generalized concepts necessary to express the ideas of the current and past residents of this living group relating to the concept tutor.
The expert’s knowledge base contains the social knowledge of this living group relative to the concept tutor.

This article is divided into 5 parts. In the first section cognitively based theories about the nature of knowledge, social expertise, and coding are integrated to form a unified theoretical framework on which an expert system for coding verbal protocols can be constructed. The specific expert system used for doing the analysis, SKI, is then described. Following is an illustration of this approach to the coding of verbal protocols. In the fourth section it is shown that by using SKI the reliability and replicability of the coded data is increased. And in the fifth section, existing methods for textual analysis are contrasted to the proposed method.

KNOWLEDGE, SOCIAL EXPERTISE, AND CODING

In order to form a cognitively based system for coding verbal protocols it is necessary to have a theory about the nature of knowledge, its acquisition, dissemination, and relation to the language of the protocol. It is also necessary to have a theory about the way in which the expert uses knowledge to code a protocol.

THE NATURE OF KNOWLEDGE

It is assumed that there is a distinction between individual knowledge and social knowledge, individual language and social language (Carley, 1987a, 1988). Information is viewed as separable into discrete units. Individual knowledge is the set of information known by the individual. As the individual gathers more information knowledge accumulates. Social knowledge is the set of information shared by the members of the society. All knowledge is articulable (Carley, 1986b). The individual’s language is viewed as an articulation of his or her knowledge (Carley, 1987a, 1988). Knowledge is viewed as a relational phenomena; that is, the pieces of information interrelate. Meaning is a function of this
interrelationship. In order to increase the precision of these ideas, and to specify knowledge acquisition and dissemination schemes, a knowledge representation scheme is needed.

Social knowledge, the information in a verbal protocol, the social expert's knowledge, and so on are all articulable. Consequently, they can all be represented using the same scheme. Following is a description of a scheme for representing such information regardless of whether it is communicated directly (spoken), exists in a verbal protocol (written), or exists in the individual's memory. Relative to protocol analysis, the scheme is aimed at allowing the researcher to codify that information in the protocol that collectively serves as the individual's description, that is, mental framework or definition of a particular event, idea, or object. Essentially, all articulable information is represented as a network of concepts and relationships. This representation scheme is general and can be used in many ways depending on one's interpretation of the relationships. For example, in this article, this representation scheme is used for coding data on concept formation and the relationships are interpreted loosely as "implies." The proposed approach could also be used for coding data on the way an individual thinks about a decision (Carley, 1984). In this case, it subsumes the approach forwarded by Axelrod (1976) for representing decision structures.

Fact—A Piece of Information

The term fact is used for representation of a piece of information as two concepts and the relationship between them (Minsky, 1975; Carley, 1986a, 1986b). Some examples of facts are Jay loves Ann, Gnerds aren't friendly, and Someone who fits in with the hall won't insist on quietness. A knowledge base is a collection of facts. An individual's knowledge base is the collection of facts that are known by the individual. Within these knowledge bases, some of the facts will interrelate; for example, Jay loves Ann, Ann loves Greg, and Jay and Greg hate each other. By sharing concepts, facts form networks, some of which are semantic networks (Simmons, 1973; Schank and Colby, 1973), augmented transition
networks (Bobrow and Fraser, 1969; Winston, 1977), conceptual networks (Sowa, 1984), or inheritance networks. It may be that all of the facts in a particular knowledge base are connected; however, this is not necessary. In other words, a knowledge base may contain networks and yet not be one (Carley, 1986a).

*Concepts—the nodes.* A concept can be a single word, for example, *Jay or friendly,* a clause such as *insist on quietness* or a concept category such as emotions (Brooks, 1978; Brown, 1958, 1979). The level of detail chosen depends on the focus of the research. A concept is an ideational kernel: A concept is a single idea totally bereft of meaning except as it is connected to other concepts (Carley, 1986b). All concepts in isolation are thought to be meaningless (Carley, 1986a, 1986b). Concepts are nothing more than symbols whose meaning is dependent on their use; that is, their relationship to other symbols (Gollob, 1968; Heise, 1969, 1970; Minsky, 1975; Carley, 1986a, 1986b, 1988). The concepts known by the individual form his or her vocabulary. There is presumed to be a countable, and generally finite, number of concepts at any one time in any one sociocultural environment. Two concepts are said to be equivalent if their symbolic representation is the same; for example, *dog* is equivalent to *dog* and not to *dog-01, collie,* or *cat.* Since there are two concepts in a fact one is referred to as the anterior concept and the other as the posterior concept vis-à-vis that fact.

Relative to a specific topic, concepts can be classified or typed. One way of typing concepts is by their relative hierarchical position in the articulable structure or frame that the expert uses to interpret specific information on this topic. For example, in analyzing the students definition of the concept *tutor* it was determined that four types of concepts were used. These concept types are concepts that denote jobs that the tutor can perform (such as providing a *social* influence), concepts that denote the characteristic features or requirements of these jobs (such as *fits in with hall*), concepts that denote measurable features of the tutor candidate whose existence determines whether or not the candidate meets the job requirements (such as *friendly* and *egocentric*).
Relationships—the ties. The relationship that ties together the concepts can be either a single word or a clause; for example, loves, does, or is less likely than. In network representation, the relationship is the tie or connection between the nodes. The ties can have directionality, strength, sign, and meaning (Carley, 1984).

Directionality: The relationship between two concepts can be unidirectional; for example, Jay loves Ann, or bidirectional, for example, two equals one plus one.

Strength: The relationship between two concepts can vary in strength; for example, Chocolate is very good and Chocolate is good. Strength can be thought of as the level of emphasis; for example, the number of times that the fact is mentioned explicitly or implicitly in a protocol. Note, this is a measure of presence and not of certainty or belief as is used in many diagnostic expert systems, such as MYCIN (Buchanan and Shortliffe, 1984).

Sign: The relationship between two concepts can vary in sign; for example, Chocolate is very good and Chocolate is not very good.

Meaning: The relationship between two concepts can vary in meaning; for example, the tutor is a hacker and the tutor should be a hacker. In theory, meaning is simply the verbal phrase—is a or should be. As operationalized, meaning is defined by the pairing of concept types.

Comment. The representation scheme as presented here is very general. A consequence of this representation scheme is that there can be more than one relationship between a pair of concepts; for example, Jay loves Ann and Ann likes Jay. In most applications, the researcher will want to limit the number of types of ties allowed by limiting the number of concepts, by limiting meanings, by limiting strengths, and by not using directionality.

Sets of Facts

As previously noted, facts may be connected thus forming networks. Three types of networks, differing in the level of
complexity, are identified—definitions, frames, and maps. Definitions, frames, and maps are network representations of information on a particular topic, whereas a knowledge base is a list representation of this information. Note, there may be multiple definitions, frames, or maps in a single knowledge base.

A definition is a focused network of facts such that the focus is the concept being defined and the other concepts in the network serve to define the focal concept by their relationship to it and each other. A definition contains only those facts that relative to some context give meaning to the focal concept. There may be more than one definition for a single concept. If the networks of concern are conceptual graphs then definitions are canonical graphs (Sowa, 1984: chap. 3.4). The definition of a concept is dependent on what concepts that concept is related to. This is the principle of relative definition (Carley, 1986a, 1986b, 1987a). The set of definitions in the individual's knowledge base is his or her lexicon.

A frame, as defined by Minsky, is a data structure for representing situations or objects as a hierarchical network (not necessarily a tree) of nodes and relations such that the top levels are always true and the lower levels can be filled with specific instances (Minsky, 1975). A frame contains all of the facts that are related in any way to a particular topic as a network organized in a hierarchical fashion; for example, from the general to the specific, or from the socially valid to the individual. A frame can be thought of as the collection of all the definitions of the focal concept, and all of the definitions of related concepts that are needed to understand the definitions of the focal concept, organized into a network. Frames also contain rules for generating and combining related subgraphs. Such rules are stored as relationships. Definitions can be drawn from different contexts, therefore, the frame contains cross-context links.

A map is a slice of a frame at a particular level of complexity or relative to some set of contexts (Carley, 1984, 1986a, 1986b). For example, a map might contain all facts about selecting a tutor that are socially valid. A map might contain several definitions of one item or an elaborated definition of a concept.5
A frame is the most complete of these three networks. A frame may contain more than one map, and a map may contain more than one definition. To the extent that the frame is a conceptual graph map and definitions are projections of frames.

Relating the Representation
Scheme to Verbal Protocols

An individual's cognitive structure can be represented as a network of frames. A verbal protocol can be thought of as a sample of the information that is cognitively available to the individual who spoke or wrote that protocol. The protocol is thus a slice of one or more frames at a particular level of complexity; that is, the protocol is a knowledge base containing one or more maps. The verbal protocol is a sample of the information in the individual's mind and the structure of that information. To the extent that the interaction that produced the protocol encouraged retrieval of known information the protocol contains a sample of memory. To the extent that the interaction that produced the protocol encouraged creativity and knowledge acquisition the protocol contains a sample of constructed knowledge. In both cases, the protocol is a sample, that is, a window on the mind.

The very richness of verbal protocols makes them difficult to code. In a protocol of an individual playing chess we might find samples of several frames; for example, some of the logic behind a sequence of moves, a partial description of a particular board arrangement, and a discussion of the relative power of knights and bishops. Each conversational topic can be thought of as a sample of a different frame, a different map. Thus coding difficulty stems, in part, from multiple maps.

All of the information in a single verbal protocol or a set of protocols can be coded as a set of facts, a knowledge base. In this knowledge base, one would be able to discern the different conversational topics as different networks. Given the chess example, one should be able to discern three networks. Admittedly, these networks would overlap, and, at some level, even be connected; for example, all might contain the concepts knight and bishop.
Limited time and resources, as in the Third East case study, may cause the social expert to code only those sections of the protocol that relate directly to a single topic. In which case, the knowledge base will contain a single map.

Knowledge Acquisition
and Dissemination

Individuals acquire knowledge only if that knowledge relates to previously acquired knowledge (Carley, 1986a, 1986b, 1988). This is the principle of immediate comprehension. This can be restated in terms of the representation scheme. Individuals acquire a new fact only if at least one of the concepts in that fact is currently in the individual's vocabulary. New concepts are acquired as facts, which contain new concepts, are acquired. Individuals can not acquire isolated concepts. Individuals only disseminate information that they know. Time and effort constraints lead individuals to disseminate information that they expect to be new to their communication partner. Thus social knowledge is likely to be implicit in the communication, whereas individual attitudes and beliefs are likely to be explicit. Additional details on a model of the acquisition and dissemination of knowledge can be found in Carley (1986b, 1988).

Social Knowledge

Social knowledge is shared knowledge. That is, the social knowledge base contains those facts that “everyone knows.” Social knowledge is thus knowable and constraining (Polanyi, 1962; Whorf, 1956; Cicourel, 1974; Bar-Hillel, 1960). Note, even simple models of individual cognition leads to the result that social knowledge will not be explicit in the exchanges between individuals whether the speaker expects the listener to be a member of the society (Carley, 1988). Clearly some facts can be more social than others depending on the level of sharing. It is assumed that information does not have
an absolute truth value; rather, its validity is dependent on the
degree to which it is shared by the members of the society (Carley,
1986b). The more individuals in the society who know a piece of
information, the higher the social validity of that piece of infor-
mation. Tacit consensus is thus sufficient to produce validity at
the social level. Social knowledge is simply that knowledge that
has a certain level of validity. Social validity is defined in terms of
two types of tacit consensus; consensus to existence and consen-
sus to sign. Consensus to existence refers to tacit agreement on
whether or not such a piece of information exists; that is, consen-
sus to there being a relationship. Consensus to sign refers to tacit
agreement on whether the concepts in that piece of information
are positively or negatively related; that is, consensus to the
relationship being of a certain type. Different levels of social
validity can be identified by the level of agreement and the type of
consensus.

THE NATURE OF SOCIAL EXPERTISE

Two types of expertise appear necessary for coding verbal
protocols. The first type of expertise, knowing more, makes it
possible for the coder to parse the protocol. The second type of
expertise, knowing consensual information, makes it possible for
the coder to translate the protocol.

Parsing a verbal protocol includes both being able to encode
English and to extract from the protocol those sections that are
relevant to the topic under analysis. Despite developments in
sentence parsing, there is currently no system that can completely
encode English (Reddy et al., 1973; Newell, 1975; Erman, 1980;
Haas and Hendrix, 1980; Fain et al., 1981; Hayes-Roth et al.,
1983; Marcus, 1980). Extracting the relevant section of the verbal
protocol is difficult for a variety of reasons: the information in the
protocol does not necessarily follow in a logical fashion, the
information in the protocol often needs to be interpreted relative
to the research design, and information other than that which is of
interest to the researcher is also often present in the protocol. In
fact, extracting the relevant sections of the protocol may be even
more difficult than sentence parsing as it requires knowledge of English, the social environment, and the expert's focus of concern. In order to combat these difficulties the coder is often required to have an in-depth knowledge of the research topic; that is, the coder must effectively be an expert on that topic. In this case, expertise is demonstrated by knowing more pieces of information. The quantity of information known by the expert aids in coding explicit information. This point, that to be an expert is to know more, is corroborated by the experience of pioneering expert system researchers who found that expertise is a function of having a large quantity of task-specific knowledge (McDermott, 1981; Buchanan and Shortliffe, 1984).

When the protocol contains information relative to the socio-cultural environment, such as is the case with general newspaper clippings, children's stories, and so on, further complications arise due to the reliance on implicit social knowledge to provide meaning. That is, text processing requires the coder to explicate implicit information (Collins et al., 1977; Schank and Abelson, 1977; Schank and Riesbeck, 1981; Bruce and Newman, 1978; Merleau-Ponty, 1964; Bar-Hillel, 1960). As noted by Merleau-Ponty (1964: 29) "The totality of meaning is never fully rendered: there is an immense mass of implications, even in the most explicit of languages." According to a variety of researchers the existence of shared behavioral scripts makes it possible for the protocol creator to leave out information (e.g., Schank and Abelson, 1977; Schank and Riesbeck, 1981; Goffman, 1974). Information can be left out precisely because it is expected that everyone knows that information. In contrast, Garfinkel (1968, Garfinkel et al., 1981) argues that the detailed and physical nature of the tasks being performed makes explicit articulation in the short period of time in which the communication occurs unfeasible and perhaps impossible. Consequently, it is through reenactment that social knowledge is created. Regardless of whether it is the scripted nature of life or the inarticulable nature of the task that leads to the lack of explication during communication, there appears to be a consensus that social or shared knowledge makes explication possible. And, as Cicourel (1974: 86) notes, without such a
shared knowledge base “everyday interaction would be impossible for nothing could pass as ‘known’ or ‘obvious,’ and all dialogue could become an infinite regress of doubts.”

Social background information\(^{10}\) enables the reader/listener to understand the protocol because it reduces the uncertainty as to which of the possible implications is correct (Rolloff and Berger, 1982). Social knowledge, as shared meaning, both admits and constrains interpretation (Polanyi, 1962; Whorf, 1956; Cicourel, 1974; Bar-Hillel, 1960). The individual who does not have this social knowledge will not be able to extract from the protocol the semantic content implicit in the protocol at the time it was produced. This suggests that the coder must also be an expert on the sociocultural environment, that is, a social expert. In this case, expertise is demonstrated by knowing for each piece of information whether it is social or not, that is, knowing the level of consensus to or social validity of a piece of information. Consensual or social knowledge allows the expert coder to go beyond the explicit information to the implicit information. Social knowledge serves the expert as a lexicon for translating concepts that appear in protocols.

Translating a verbal protocol requires translation of the words and meaning of the speaker/writer into the generalized words and meaning of the listener/reader. Such a translation process requires the translator, in this case the coder, to use social knowledge to make explicit information that is implicit in the verbal protocol. In 1960, Bar-Hillel argued that because translation required the use of social knowledge a computer could not do such translation: “What such a suggestion amounts to, if taken seriously, is the requirement that a translation machine should not only be supplied with a dictionary but also a universal encyclopedia. This is surely utterly chimerical and hardly deserves any further discussion.” This article is grounded on the idea that with respect to a specific sociocultural environment and with respect to a specific coding task, machine translation is possible provided that the expert on that sociocultural environment creates for the machine both a vocabulary list and a set of definitions predicated on social knowledge (that is, provides the relevant portion of the
dictionary and encyclopedia). In other words, we may not be able to build a general translation machine; but, we can build a special purpose one.

While the ideal would be a full expert system that does both parsing and translation, such an ideal is currently unfeasible. In this article a more modest proposition is forwarded: a two-stage process where, first, the novice uses a computer-assisted procedure CODEF to code the verbal protocol and, second, the expert system SKI is used to diagnose and correct errors in this coding. Basically, the novice coder using CODEF parses the verbal protocol and does an initial surface translation. CODEF utilizes the expert's vocabulary for describing the sociocultural environment (social vocabulary) to aid the user in coding the verbal protocol as a knowledge base. Then the computer via SKI does the final translation of the verbal protocol by explicating the implicit social knowledge. SKI utilize the expert's knowledge of the topic. The expert's knowledge is composed of the expert's vocabulary and the expert's characterization of social knowledge (social knowledge base). SKI acts as a post-processor, using social knowledge to diagnose and correct errors in the coded protocol produced while using CODEF. This two-stage process eliminates the need to have the computer parse the protocol. Stage 1 relies on the human coder's ability to extract the relevant sections and code the information that is explicit in the protocol. Stage 2 utilizes the expert's knowledge of the topic to diagnose and correct errors of explication; errors caused by the human coder leaving out social information that is implicit in the protocol.

Social knowledge enables the expert to make explicit the implicit information in a communication. Consider the following illustration. If a student uses the phrase *Aaron is a hacker*, unless one knows what a *hacker* is, there is too much uncertainty to interpret the phrase. If you know that in the micro-world in question a *hacker is someone who does things other than studying*, then the phrase has the interpretation *Aaron is a hacker and therefore does things other than studying*. Further social background information about the micro-world, in this case about MIT and 3E in particular, admits a more rich and detailed inter-
pretation of the phrase (see Figure 1). Social knowledge is necessary to make explicit the implicit information, thus allowing the construction of a rich detailed interpretation by the listener.

Social knowledge reduces uncertainty because it makes it possible to extract the expected interpretation of phrases, as in Figure 1. Note, the expected interpretation is not the only interpretation of a phrase, nor is it necessarily the correct interpretation from the speaker's point of view. The expected interpretation is expected in the sense that it contains the information that the relevant majority attaches to this phrase. Hence the expected interpretation is correct in the sense that it is congruent with the interpretation shared by the relevant majority of the members of the social unit.

In the process of acquiring expertise on a sociocultural environment the individual acquires a set of facts. And for each fact the individual also acquires information on who knows it. By collating this information the expert can determine the level of social validity for each fact. Consequently, the social knowledge base is a subset of the expert's knowledge base. The social knowledge base contains the social vocabulary and a lexicon. The expert uses social knowledge to interpret protocols. For example, if the expert locates a concept in the protocol that is in the social vocabulary then the expert can interpret, that is, determine the meaning of the concept using the lexicon. In this way, social knowledge is used to make implicit information explicit. The social expert uses different levels of social validity to define meta-rules for doing this explication. For example, the social expert might have the meta-rule that if a fact is completely valid, that is, it is known by everyone, then if one of the concepts occurs in the protocol the expert will assume that this fact is implied. In the case of analyzing Third Easter's talk about tutors three different levels of validity are identified, each with their own associated meta-rule—definitives, logical connectives, and simple connectives. Further, if the expert bases these rules on a cognitive model of the way in which the individual acquires knowledge then the validity of the results of this explication should be increased. For example, the expert can employ the principle of immediate
Figure 1: Definition of a Hacker

NOTE: This map shows the socially shared definition of the concept hacker on Third East. The solid lines indicate a positive relationship between the concepts, whereas a dashed line indicates a negative relationship. Note: all relationships have the same strength and are bidirectional. Each concept type is in a different column.
comprehension to limit explication to just those facts that are potentially comprehensible by the individual who produced the protocol. As another example, the social expert can employ conceptual hierarchies (Sowa, 1984; Bruner, 1973) to extend or simplify the explication process.

Social experts use their formalization of social knowledge as a set of rules for interpreting and analyzing data obtained on the sociocultural environment. Through systematic application of these rules the expert can extract an interpretation of information communicated by a member of the social unit that, while consistent with social knowledge, is not typically extracted by other members of the social unit who may not have all of the rules, and who may be less systematic in the application of the rules that they do have. The social expert, unlike the typical member of the social unit, has formalized this knowledge and knows the level of social validity to attach to all information communicated by members of this society. Hence the expert’s ability to make explicit implicit information should be greater than the typical member of the society, and perhaps even less prone to error.

There are several types of error of explication to which the social expert should be less prone. First, error of inclusion—coding a piece of information as present that is not explicitly present in the protocol and that is not socially valid. Second, error due to oversight—failure to code a piece of information as present that is not explicitly present in the protocol and that is socially valid. Third, error in directionality—coding the directionality of a piece of information that is not explicit as the opposite of the directionality of the corresponding socially valid piece of information. Fourth, error in sign—coding the sign of a piece of information that is not explicit as the opposite of the sign of the corresponding socially valid piece of information. These forms of error are illustrated in Figure 2.

**SKI—SOCIAL KNOWLEDGE INTERPRETER**

Social knowledge as determined by the social expert is articulable. Hence it can be stored and utilized by a machine. An expert
Figure 2: Errors of Explication

NOTE: There are four possible errors of explication: error of inclusion, error of exclusion, error of directionality, and error of sign. Arrows indicate directionality. If there is no arrow then no directionality can be assumed. A dotted line indicates a negative sign. A solid line indicates a positive sign. Begin by assuming that only the information a-b was stated in the protocol (upper left-hand corner). Based on this piece of information a coder may produce the explication shown in the middle. In terms of social knowledge (upper right-hand corner) this explication is incorrect. The inclusion of the information b-x is an error of inclusion as that information is not in the social knowledge base. The failure to include the information b-g is an error of exclusion. The inclusion of the information b-d while correctly included shows an error of sign; the included information is positive, whereas it is negative in social knowledge. The piece of information b-c has been correctly included. The inclusion of the information c-f is an error of directionality. Moreover, since the directionality of b-f is really the opposite of that shown, this information should not have been included. At the bottom, the correct explication of a-b relative to social knowledge is shown.
system can routinely apply this knowledge, even as the human social expert does, and with less chance of explication error. SKI is such an expert system. SKI does not code verbal protocols directly. Rather, SKI applies the knowledge of the social expert to improve the coding of a verbal protocol produced by a novice. SKI takes as input a knowledge base that is the coded version of a verbal protocol, and using the social knowledge base produces as output a modified version of the input knowledge base (see Figure 3). In order to determine the validity of the results it is necessary to understand the way in which SKI uses social knowledge to explicate implicit information.

Currently, expert systems are typically thought of as being composed of two parts—an inference engine and a knowledge base (Hayes-Roth et al., 1983; Davis and Lenat, 1980; Waterman, 1986). The inference engine has built-in rules for applying the rules in the knowledge base and making inferences on the basis of such knowledge. The inference engine can be viewed as a model of cognitive processing. Thus SKI can be viewed as a model of the way in which a social expert uses social knowledge to interpret the explicit information in protocols. For SKI, the inference engine, ADDSOC, is separate from the knowledge base; consequently, ADDSOC can be used with social knowledge bases other than that used for Third East tutor selection.

SKI has several features that increase its utility as a research tool. To begin with, it can be used in conjunction with other programs that also handle knowledge bases; for example, CODEF. Other properties of SKI include closure, completeness, simplicity, and a separate inference engine.

**Closure:** Output and input are both knowledge bases. The expert's knowledge base, that is, the social knowledge base, the original coded protocol, and the modified coded protocol produced by SKI all have the same format.

**Completeness:** SKI completes the modification in one passage. That is, even though the output coded protocol can be given as input to SKI doing so will not alter the output knowledge base. When a modified knowledge base is submitted to SKI no changes will be made in the knowledge base.

**Simplicity:** The mechanism for modifying the knowledge base is simple. The inference engine is a forward chaining fact includer,
Figure 3: Interface Between Programs

NOTE: CODEF utilizes information in the social expert's vocabulary to aid the coder in producing a coded representation of the verbal protocol—a coded protocol. The vocabulary can be constructed as protocols are coded. The entire protocol need not be coded at once so CODEF has the ability both to read and write the protocol being coded. SKI is used to diagnose errors in the coded protocol by utilizing the social knowledge base constructed by the expert and the corresponding vocabulary. The result is a new, modified version of the coded protocol. Note: the social knowledge base like the novice-coded knowledge base can be coded using CODEF. Since the social knowledge base is not used by CODEF to produce the coded protocol the line between the social knowledge base and CODEF is dotted.
where the decision to add a fact is adjusted by the level of social validity attached to that fact.

Separate inference engine: In SKI the inference engine, ADDSOC, and the knowledge base are completely separate. ADDSOC is a general purpose program not directly tied to the domain knowledge; hence ADDSOC can be used with other knowledge bases. Thus ADDSOC can be used for research on a variety of questions.

THE KNOWLEDGE BASE

The knowledge base for SKI is the social expert's knowledge base. This knowledge base contains that information that the social expert considers to be socially valid. It is assumed to be composed of three data files, each containing a different type of fact. The three types of facts differ in their level of social validity—definitives, logical connectives, and simple connectives. Definitives have both a high level of consensus to existence and a high level of consensus to design. Local connectives have a moderate level of consensus to existence and a high level of consensus to sign. Simple connectives have a high level of consensus to existence and a moderate level of consensus to sign. Coding details are provided in the next section.

SKI assumes that in a data file all relationships are of the same directionality and uniform in strength. Within a data file relationships are allowed to differ in sign. SKI does not utilize information about differences in the types of concepts or the meaning of relationships.

ADDSOC—THE INference ENGINE

The inference engine can be thought of as the logical structure for determining when to apply the rules in the knowledge base. Inference engines vary in generality and complexity from simple task-specific engines—such as the one presented in this article or EMYCIN (Scott et al., 1977)—to very general complex systems used for multiple tasks, such as OPS5 (Forgy, 1981) or SOAR (Laird, 1987; Laird et al., 1986). There appears to be a trade-off
between the generality of the inference engine and the speed with which the expert system is developed (Hayes-Roth et al., 1983: chap. 10). Typically, the more task-specific the inference engine, the more limited the knowledge representation scheme and the available meta-level inference procedures, and the more rigid the control structure; but the quicker it is to encode the expert’s knowledge and develop the expert system. Whereas, the more general the inference engine, the more flexible the knowledge representation scheme, the more generic and potentially powerful the inference procedures, the more flexible the control structure; but the longer it takes to encode the expert’s knowledge and develop the expert system. Given this trade-off, it was decided to build a simple and fairly task-specific inference engine. There are four inference rules and an implicit inference net across types of facts based on social validity.

ADDSOC is an inference engine for emulating the social expert’s use of social knowledge to make explicit the implicit knowledge in verbal protocol. ADDSOC emulates the social expert’s behavior by acting as a lexical analyzer. The social expert’s knowledge base contains social knowledge as a set of definitions. ADDSOC, as lexical analyzer, extracts the concepts in the input coded verbal protocol, retrieves the definition of these concepts from the social knowledge base, and then uses the retrieved social definition to augment the individual definition present in the coded protocol. In this way, social knowledge is used to reduce the uncertainty of interpretation.

Definition augmentation is done in a generative fashion. The social definition is not stored as a definition that is a network in the social knowledge base; rather, this definition is implicit within the set of facts that form the social knowledge base. ADDSOC builds these definitions as needed, while explicating, that is, diagnosing and correcting errors in the input coded verbal protocol. ADDSOC utilizes fact type and relationship strength in this generation process. ADDSOC reads in the coded protocol and forms the implicit network. Then, for each node in this network, that is, each concept, ADDSOC, using a forward chaining fact, including mechanism, augments the coded definition of that con-
cept by adding the social definition of that concept as contained in the social knowledge base (see Figure 4). Forward chaining is a procedure used in expert systems that produces decisions (adds data) in a recursive fashion by building consequences from antecedents. The user enters the antecedents (in this case the coded protocol that is knowledge base) and using a set of inference rules, returns the consequence (in this case more facts).

The assumption that the social expert uses social knowledge to explicate implicit information controls the overall procedure. The inference rules used in ADDSOC follow from the previous assumptions about the social expert's cognitive behavior during protocol coding. Each of these rules allows ADDSOC to add a fact to or modify the sign of a fact in the coded protocol. Only facts that are in the social knowledge base can be added to the coded protocol. Only that part of the social definition that is not currently in the coded protocol can be added. If there is a fact in the coded protocol that contradicts a fact in the social knowledge base, and that fact is not erroneous, then the fact in the coded protocol takes precedence over the fact in the social knowledge base. In this way, individual differences are retained.

The principle of immediate comprehension states that for an individual, new knowledge must be related to previous knowledge. The social expert in explicating the implicit knowledge cannot add new topics, that is, new maps. Further, since facts are related by sharing concepts there are only two ways in which the social expert can add facts to the coded protocol without violating the principle of immediate comprehension—by adding a new concept and the relationship to that base. The social expert uses social knowledge to determine which of these acts to perform. Further, different levels of validity are used to define explication meta-rules. Validity is used to segregate facts into three categories—definitives, logical connectives, and simple connectives.

Rule 1: Add concepts only if their presence is socially valid. Definitives have the highest level of social validity. Consequently, if the anterior concept is present the posterior is always, at least, implied. For each definitive fact in the social knowledge base, if the anterior concept is in coded protocol but the posterior concept is not,
### Social Knowledge Base

<table>
<thead>
<tr>
<th>Definitives</th>
<th>Logical Connectives</th>
<th>Simple Connectives</th>
</tr>
</thead>
<tbody>
<tr>
<td>a → b</td>
<td>a → e</td>
<td>a → h</td>
</tr>
<tr>
<td>c → d</td>
<td>f → q</td>
<td>f → h</td>
</tr>
<tr>
<td>b → e</td>
<td>b → c</td>
<td></td>
</tr>
<tr>
<td>e → h</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Generate Definition of a**

1. a → b
   
2. a → b → e
   
3. e → a
   
3. b

**Novice Coded Knowledge Base**

1. b
   
2. a → b → c
   
3. e → a
   
3. b

**Modified Novice Coded Knowledge Base**

1. h → e
   
2. a → b
   
3. e → a
   
3. b

**Figure 4: Generative Definition**

NOTE: In the lower left-hand corner there is an example of the process of generating a definition. This process requires the use of information in the social knowledge base (top). This knowledge base is composed of three types of facts: definitives (top left), logical connectives (top middle), and simple connectives (top right). Only definitives can be used to add concepts. The arrow indicates the directionality of the fact and thus which concept can be added. Logical connectives can be used to change the sign of a relationship. Dotted lines indicate a negative sign, solid lines a positive sign. Simple connectives can be used only to add a relationship. In the lower left the definition of a is generated in a forward chaining fact inclusion fashion. Step 1, based on the definitives we see that there is only one fact that has an anterior concept. This fact is added. Then the connectives are checked to determine sign changes or additions. Since there are only two concepts and they are already related, this process ends. Step 2, based on the definitives all facts with b as the anterior concept are added. Step 3, the connectives are checked and the relationship between a and e is added. Step 4, based on the definitives all facts with e as the anterior concepts are added. Step 5, the connectives are checked. Since b is not an anterior concept for any social definitive the network at Step 5 is the generated definition of a.

In modifying the coded protocol (middle right) SKI is effectively adding the generated definitions. In the modified coded protocol (bottom right) we see that SKI has added the definition of a, as well as the definition of the other concepts. SKI has also used the logical connectives to alter the sign of the relationship between b and c. SKI does not delete information in the original coded protocol that is not in the social knowledge base; for example, the fact a→m. Because the original and the modified coded protocol are knowledge bases associated with student interviews the directionality of all facts is assumed to be identical.
ADDSOC adds the posterior concept and the relationship to that concept.

Rule 2: Change sign on a relationship for a fact if there is consensus to the sign for that fact. Logical connectives are socially valid in the sense that if a fact is present, the sign of the relationship is always the agreed upon sign. For each fact in the coded protocol, if the sign of the relationship is not the sign of the relationship of the corresponding logical connective fact in the social knowledge base, the sign of the fact in the coded protocol is switched. The corresponding fact is the one that matches on both concepts.

Rule 3: Add a relationship between two concepts if there is consensus to the existence of the corresponding fact. Simple connectives are socially valid in the sense that if both the anterior and posterior concepts are present, the relationship between them is always at least implied. For each pair of concepts in the coded protocol, if there is not a relationship between these concepts and there is a simple connective fact composed of these two concepts, then the relationship between these two concepts is added to the coded protocol.

Rule 4: ADDSOC only explicates implicit knowledge. ADDSOC cannot determine whether the facts present in the coded protocol were explicit or implicit in the verbal protocol. ADDSOC makes the assumption that all facts initially present in the coded protocols were explicit in the verbal protocol. Consequently, ADDSOC does not delete concepts or facts from the coded protocol.

Inference Net: There is an inference net across the three types of facts. That is, all definitives are logical connectives, and all logical connectives are simple connectives.

The directionality of facts in the social knowledge base does not transfer to the coded protocol. The directionality of the facts in the social knowledge base is used solely for determining whether or not a new concept should be added; however, once the concept and fact has been added to the coded protocol, the fact is treated as a simple bidirectional relationship between the two concepts just like the other facts in the coded protocol. Thus bidirectionality of the facts in the coded protocol is preserved.

Individual differences are maintained in several ways. None of these meta-rules deletes facts from the coded protocol. Only rule 2
can be used to modify facts in the coded protocol. And, this rule can change only the sign of the fact. Sign is changed only if the logical connective indicates the need. Logical connectives are those facts for which within that society there is a logical and highly agreed to relation between the concepts such as *current hall members are Third Easters*. Due to the high social validity of the sign in logical connectives it was felt that if a fact in the naive coded protocol had the opposite sign as that of the logical connective the coder had made a mistake. Note, this rule was rarely applied and every time it was applied it was found that the coder had indeed made a mistake. And, finally, all of those facts that are added to the coded protocol are those facts that are connected to facts already coded and that have a high level of social validity. An implicit assumption is that information unique to the individual is explicit in the protocol. If the coder misses explicit information SKI can't make up for it.

**AN ILLUSTRATION OF THE METHODS**

In this section, the raw and the coded data are described. First, the raw data is presented. Then there is a description of the way the social expert's knowledge was extracted. This is followed by a description of the vocabulary and knowledge base used by CODEF and SKI. Then CODEF is used to code a set of verbal protocols. And, finally, SKI is used to diagnose and correct errors in these codings.

CODEF is an interactive user-oriented computer-assisted procedure for coding verbal protocols as a set of facts that is a knowledge base. CODEF can be used to code any form of articulable data; for example, free-form interviews, stories, and historical documents. CODEF uses the vocabulary of the social expert to aid the coder. CODEF also uses the social expert's coding decisions to restrict the types of facts, concepts, and relationships allowed. For additional details refer to Appendix A and the CODEF manual.
RAW DATA

The raw data includes student data—interviews with the individual students on the hall as to their perceptions of the tutor selection process—and sociocultural data—historical and cultural background information drawn from interviews, personal observation, letters from previous residents, and stories told at the Third East reunion (Carley, 1984).

The student data is a set of verbal protocols—transcriptions of guided free-form interviews with MIT undergraduates on Third East during the tutor selection process. Each interview was with a single student. The interviews were open ended, informal, and chatty. Each student, in a particular interview round, was asked the same core set of questions; however, the questions were not always asked in the same order or in the same way. Sometimes the questions were repeated during the interview to try and evoke a more complete response. Some of the questions were repeated during each round of interviews—for example, What is it that you want in a tutor?—whereas other questions were unique to a particular interview round—for example, Who was your first choice for tutor? The tutor selection process lasted around 3 months, during which each of the consenting students was interviewed, alone, in three different rounds of interviews. These three rounds of interviews correspond to three different phases in the tutor selection process: TIME 1 (at the beginning of the process, prior to interviewing any of the candidates), TIME 2 (after the first round of interviews with tutor candidates but before the last interview), and TIME 3 (at the end of the process, after the vote for tutor candidates had been taken, and the results were known).

The interviews with the students ranged from 15 minutes to a little over an hour; typically, they were between 30 and 45 minutes. The tape-recorded dialogue for these interviews was then transcribed. Each interview resulted in a single verbal protocol. If a student was interviewed during all three rounds, then there would be three verbal protocols associated with that student. Due to students not being interviewed, illegibility of transcription,
mechanical failure of the tape recorder, and so on, only 80 of the 135 potential interviews were codable. These 80 codable interviews are referred to as the student interviews.

Of the 80 codable verbal protocols (student interviews), only a sample of 48 were coded. Different student interviews were coded at different stages (refer to the next section). There were four coders, each of whom coded a subset of these 48 interviews. My husband Rick and myself, as the "social experts" on the sociocultural environment of Third East, coded only a few of the interviews, 8 and 4, respectively. The resultant knowledge bases were used as prototypes against which to test the accuracy of the knowledge bases produced by the novices and the modified knowledge bases produced by SKI. The two other coders, D and C, were relative novices on the particular sociocultural environment. Coder D was an ex-hall resident and so had previous acquaintance with the living group, but little acquaintance with the current residents. Coder C was a Harvard student with no previous knowledge of the living group. Coder D coded all 48 interviews—20 TIME 1, 9 TIME 2, 19 TIME 3. Coder C coded 27 of the 48 interviews—6 TIME 1, 2 TIME 2, 19 TIME 3. Only the knowledge bases produced by the novices, C and D were modified by SKI.

The sociocultural data includes a reconstruction of the social history of the hall, an overview of Third East and MIT traditions and culture, and a description of the current living group. This information was drawn from the general oral history, specific discussions with ex-hall members, and tape recordings of hall legends produced at the Third East reunion. The reconstructed history includes a description of each of the tutors, the role they played on the hall, and the students' reaction to those tutors. The cultural overview includes a description of the physical environment, definitions, and examples of slang, and a discussion replete with examples of traditions. The current living group description includes a description of students, hall functions, and politics at the time of the study. The reconstructed history and cultural overview was sent to a selection of 24 ex-Third Easters chosen across the spectrum of years in order to solicit their comments.
Their comments were then incorporated into both documents. The history goes back to the first tutor on Third East, and covers a span of almost 15 years. The current hall description was distributed to the students currently in the living group for their comments and was revised accordingly. This process produced 3 protocols: (1) a history of the living group, (2) a cultural overview, and (3) a description of the current living group. A version of CODEF was used to code these protocols. The eventual result is the reduced knowledge base, 3EKB.

**EXTRACTING THE SOCIAL EXPERT'S KNOWLEDGE**

One of the first tasks the knowledge engineer faces is to extract from the expert his or her vocabulary and his or her knowledge (the rules he or she uses to analyze and solve problems in this limited domain). Being both the knowledge engineer and “expert” on the project, I did not need to go through a process of interviewing and error checking but instead could simply write down the vocabulary and knowledge base and then check it for errors.

Since I was treating the vocabulary and knowledge base as formalizations of social knowledge, I did not rely entirely on my own judgment. Recall that social knowledge is a function of social validity that is operationalized in terms of the degree of agreement among the “relevant” majority. Both 100% and 80% validity levels were used. The relevant majority was defined as a sample of 5 residents, current and previous. After the social vocabulary and the social knowledge base were constructed, this group judged each fact for its level of validity. This process is described below.

**Constructing the Vocabulary**

The social vocabulary containing the set of generalized concepts that relate to the concept tutor was constructed. This step involved collating the specific terms used by current and past residents when talking about tutors into a set of general concepts. The process began by using Bales's list of personality traits as a set
of base concepts on which to map other concepts (Bales and Cohen, 1979). Using these concepts, an attempt was made to code 5% of the codable student interviews and the historical and cultural documents previously produced. It quickly became apparent that most of the concepts in describing the concept tutor could not be mapped into the Bales traits. Consequently, in the process of coding these protocols, the number of concepts rose to 208. Many of the added concepts are specific to the sociocultural environment, for example, hacker, Third Easter, and tool. As the novices coded the first set of interviews, this list of concepts increased to 217. Most of the additional 9 concepts were added after having coded only two of these interviews. This suggested that there would be closure over the number of concepts necessary to describe students’ perceptions of the concept tutor in this sociocultural environment.

The 217-concept social vocabulary is not complete in terms of actual words used by the students. It is complete in terms of the set of general concepts needed to represent the perceptions about tutors forwarded by current and past residents. Note, although the coders were not allowed to add concepts to the vocabulary after the first phase of coding, in discussions with the coders during latter stages, no additional concepts were even suggested. That is, even if they had been allowed to add concepts to the vocabulary, there were no additional concepts that they would have added. Thus it appears that there was closure over the concept list.

Constructing the Expert’s Knowledge Base

Prior to having the novices code the student interviews the expert’s social knowledge base relating to the concept tutor was constructed. This knowledge base is referred to as 3EKB. The procedure used in coding this information was similar to that used by the novices to code the student interviews; however, there were a few differences. First, multiple sources of raw data were used to construct the knowledge base. These sources (identified
above) were processed as were the student interviews and then coded using CODEF. Second, the types of facts and relationships allowed were different. The allowable fact types are described below. Third, the knowledge base, 3EKB, was “tuned” by running it through two different validation checks (see the following subsection).

The resultant knowledge base, 3EKB, is composed of 1,462 facts. These 1,462 facts serve as the domain-dependent rules, that is, the knowledge base used by SKI to determine what implicit knowledge is present in a verbal protocol vis-à-vis the concept tutor. Of these facts, 382 are definitives, 927 are logical connectives, and 153 are simple connectives.

A sample of current and past Third Easters were asked to rate each of these facts on two criteria: (1) are the concepts always related?—yes/no (consensus to existence) and (2) is the relationship between the concepts always as shown?—yes/no (consensus to sign). The set of facts are divided into 3 categories according to the response of these coders. If a fact received a mixed response on question 1 and question 2 the fact was dropped from the general knowledge base. The remaining facts were categorized as either definitives, logical connectives, or simple connectives.

A fact is defined to be a definitive if it has a high level of social validity. A fact was classified as a definitive only if 100% of the sample group responded with a yes to question 1 and a yes to question 2. If a fact is a definitive and if a member of the society states the anterior concept then the relationship and the posterior concept are always implied and the sign of the relationship is agreed upon.

A fact is a logical connective if there is less consensus to its existence, but given its existence, high consensus to its sign. A fact was coded as a logical connective if 80% or more of the sample group agreed that the fact existed (yes to question 1) and all of the sample group agreed that if it existed it would have the stated sign (yes to question 2). If a fact is a logical connective and if a member of the society states both concepts then the relationship between the concepts is always implied, and the sign of the relationship is agreed upon.
A fact is a simple connective if there is both less consensus to existence and less consensus to sign than there is for the definitives. A fact was coded as a simple connective if at least 80% of the sample group agreed that the fact existed (yes on question 1) and if at least 80% of the sample group agreed that if it existed it would have the stated sign (yes on question 2). If a fact is a simple connective and if a member of the society states both concepts then the relationship between the concepts is always implied; however, the sign of the relationship is not agreed upon.

Tuning the Knowledge Base

The knowledge bases of expert systems are often tuned to improve performance. The tuning process differs from system to system, but usually involves a series of steps in which the expert watches how the system performs, notices that it reaches an erroneous conclusion, or fails to reach a conclusion, and then backtracks through the expert system's reasoning process to discover whether a particular piece of information led to the mistake or if there is a missing piece of information. To tune the knowledge base two different error-checking procedures were used—informant observer and frame explication.

Informant observer error checking. Three ex-Third Easters went through the entire knowledge base extracting those facts that they felt were miscoded; that is, not true for all Third Easters. Less than 10% of the facts were in this miscoded category. The list of miscoded facts was then given to two current Third Easters and the three ex-Third Easters who were asked to evaluate the social validity of these facts in terms of (1) everyone agrees, (2) most people agree, and (3) ambiguous (few people agree). Those facts felt to be ambiguous by at least one of the five informants were dropped from the knowledge base as having little social validity. All others were moved down one level in social validity.

Frame explication error checking. This tuning method is based on the assumption that the knowledge base contains the social expert's rules for making implicit knowledge explicit. Consequently, SKI, when using this base for correcting the novice-
coded protocol, should explicate, that is, add, only that information that the social expert would. Thus the modified knowledge base produced by SKI should have fewer explication errors than does the unmodified knowledge base produced by the novice.

Three student interviews were selected at random and then coded by both the novices and myself (acting as the social expert). This resulted in 3 knowledge bases for each interview—2 coded by the novices and 1 coded by myself. The knowledge bases coded by myself (the expert knowledge bases) were much more elaborate, that is, contained many more facts, than those coded by the novices (the novice knowledge bases). The novice-produced knowledge bases were then modified by SKI using 3EKB as the social knowledge base. These modified bases were then compared with the corresponding expert bases. Any differences between the modified and expert bases were treated as possible explication errors. The source of each possible explication error was traced down (see Figure 5).

If a fact occurred in a modified base and it was not in the expert's base it was treated as a possible error of inclusion. If a fact occurred in the expert's base that was not in the modified base it was treated as a possible error of exclusion. If a fact occurred in both the modified base and the expert's base but with opposing directions on the relationship it was treated as an error of directionality. If a fact occurred in both the modified base and the expert's base but with opposing signs on the relationship it was treated as an error of sign.

If an error of inclusion occurred that fact was located in 3EKB. Then using a backward chaining procedure the modified base was searched using 3EKB to locate the initiating fact in the novice base that had caused the inappropriate fact to be included. If the initiating fact was not in the expert's base then the error was caused by the coder miscoding explicit information. This happened three times; that is, three times it was not an error of inclusion but an error due to the misinterpretation of the data by the novice. If the initiating fact was in the expert's base then the facts in 3EKB generated by ADDSOC from the initiating fact were located. In general, such generated facts were found to have been inappropriately coded; that is, the level of social validity attached to the facts in 3EKB was too high. The level of social
Figure 5: Tuning the Knowledge Base

NOTE: This is an illustration of the process of tuning the social knowledge base by frame explication. The original novice-coded knowledge base is passed through SKI to produce the modified novice-coded knowledge base. This is done using the original untuned, social knowledge base (top right). The modified base is not identical to the expert's coding of the protocol, that is, the expert-coded knowledge base. This indicates possible errors of explication due to the presence of incorrect data in the original social knowledge base are changed to produce the corrected social knowledge base.

Both the original untuned (top right) and the tuned social knowledge base (bottom right) have the same format. The D's are definitives, the L's are logical connectives, and the C's are simple connectives.

In the modified novice-coded knowledge base the fact e-g is missing; however, examination of the social knowledge base shows that it was not a generable fact. That is, a-g is not in the social knowledge base. Thus it is assumed that a-g was explicit in the protocol and the coder missed it. In the modified base the fact g-d is missing. Backward chaining on the original social knowledge base shows that there is only one fact that can lead to the inclusion of g, the fact, h-g, and no facts that lead to the inclusion of concept d. Further, backward chaining shows that there are no facts in the social knowledge base which lead to the inclusion of the concept l. Thus, the fact g-d can be included just in case of g or l and d are in the novice-coded knowledge base. Since these concepts are not in the novice-coded base SKI could not add the appropriate inferences. Thus the error is in the novice missing explicit information and not in the social knowledge base. In the original and modified coded protocols the fact b-d exists but is of opposite sign as that in the expert's coding of the protocol and the social knowledge base. This is a sign error; therefore, the fact b-d must be raised a level in the social knowledge base if in the future we are to create correct modified novice-coded protocols. In the modified novice-coded protocol the fact e-f is an error of inclusion. This fact was added to the coded protocol because the fact e-f was added because b-e was added and b was in the coded protocol. Of these b-e was correctly added. Therefore, the "offending fact" is e-f whose validity level should be decreased by one.

Imagine that the original novice-coded knowledge base is again passed through SKI to produce a new modified novice-coded knowledge base (no shown). This time the corrected, that is, tuned, social knowledge base is used. The fact b-e is added, the sign of b-d is changed, and a relationship between a and d is added. It is still the case that a-g and hence a-d are not added. Note: a-g cannot be added because it is a fact that is specific to the individual and is not agreed to by all members of the society. This is one way in which individual differences are maintained, however, at the cost of expecting those differences to be explicit in the protocol and expecting the coder to pick out these explicit facts.
validity for such facts was then decreased by one level. Less than 10% of the facts in 3EKB were modified using this tuning technique.

If an error of exclusion occurred, the noninluded fact was located in 3EKB and, using a backward chaining procedure, the set of possible initiating facts in 3EKB were located. An initiating fact is a fact that can force the noninluded fact to be included. If any of these initiating facts occurred in the expert’s base but not in the novice’s base then the novice coder failed to code an explicit fact. This happened fairly frequently, especially with coder C. If any of the initiating facts occurred in both the expert’s and novice’s base the facts in 3EKB generated by ADDSOC from this initiating fact were located. In general, such generated facts were found to have been inappropriately coded; that is, the level of social validity attached to these facts in 3EKB was too low. In such cases, the level of social validity was raised by one. Less than 1% of the facts were modified through this procedure.

The coding procedure followed by both novice and expert coder in coding the interviews restricted the output knowledge base. First, for each verbal protocol a single knowledge base was produced. Second, all facts in the knowledge base are bidirectional. Since CODEF ensured these outcomes, no errors of directionality could occur.

If an error of sign occurred the fact was located in 3EKB. First, that fact was checked to see if it was miscoded; that is, to see if the sign was the expected one. If it was miscoded, the sign was reversed. If the sign was not miscoded, the level of social validity for that fact was reduced so that the fact became a simple connec-tive. Less than 1% of the facts were modified in this manner.

After all of the modified knowledge bases had been processed in this manner the novice bases were then rerun through SKI. The new modified bases were again processed using the procedures described above. This process was carried out 3 times. At which point all remaining errors were found to be the result of the novice coder having incorrectly coded the explicit information.
CODING STUDENT INTERVIEWS

A coder takes a verbal protocol and, using CODEF, creates a coded version of the protocol in the form of a knowledge base—a coded protocol. In this case, the coders were given a modified transcript of a student interview. The coders were told to code the information in this transcript about tutors using CODEF to create a knowledge base. The resultant knowledge base contains the coded version of a particular student's perception of the concept tutor at a particular stage in the tutor selection process. Following are details on this procedure.

The representation scheme previously presented is for representing coded data; it is not a coding procedure. Many coding procedures can be used with this scheme. Typical coding procedures and their relative advantages and disadvantages are discussed in some detail by Ericsson and Simon (1984) in their book Protocol Analysis: Verbal Reports as Data. Which exact procedure is followed depends to some extent on the goal of the research. The coding procedure used for this study is described in the following section.

General Coding Procedure

First the coders were given a brief introduction to the sociocultural environment and the process of tutor selection. They were instructed that the only thing of interest in the protocols that they were going to be coding was the information pertaining to the concept tutor. This was broadly defined to include information such as the following: What the students wanted in a tutor or did not want, actions and behaviors of past tutors, and reviews of particular individuals as possible tutors. Next the coders were introduced to the representation scheme and the limitations placed on this scheme by CODEF. At the same time, the coders were given the vocabulary—3ECL. Ambiguous concepts were defined for the coders. The coders were told that they were to take specific terms in the protocol and map these onto the correspond-
ing, more general, concepts in the vocabulary. Specific examples for each concept were provided in an annotated vocabulary list (Carley, 1984: Appendix 2). The difference between negative and positive relationships was described. They were then given the CODEF manual to read (Carley, 1983). This manual contains examples of three verbal protocols and the resultant coded knowledge base. The first two of these protocols are marked to show how they were coded. The third example is unmarked so that the coders can train using this protocol. The coders were given a demonstration of CODEF and were allowed to use it. All protocols given the coders had been previously prepared (see Appendix B).

In order to extract the facts from the verbal protocol, the coders were taught the following procedure:

(1) Read through the transcript and using a marker, cross out all sections that do not apply to the concept tutor. Since topics corresponded to paragraphs, this typically involves crossing out entire paragraphs.

(2) Reread the protocol, circling specific concepts used to talk about the concept tutor in the text and then on the right hand side of the transcript or on another piece of paper list the general concept(s) from the vocabulary that are implied by the specific concepts in that text segment. Single nouns, as well as full subject and object phrases, can be mapped into general concepts.

(3) Reread the protocol a third time and, using CODEF, code the set of relationships that exist between all possible pairs of concepts in the set of concepts previously listed.

(4) Assign an absolute strength to each relationship using the following rules:
   (a) Relation is weakly inferred—if two concepts occurred in different topic segments and a relationship was implied, place a relationship between them with an absolute strength of 1.
   (b) Relation is explicit but weak—if two concepts occurred in the same semantic segment place a relationship between them with an absolute strength of 2.
   (c) Relation is strongly inferred—if two concepts occurred in the same topic segment and a relationship was strongly implied,
place a relationship between them with an absolute strength of 2.

(d) Relation is explicit and strong—if the relationship between two concepts was mentioned more than once, raise the absolute strength to a 3.

(5) Assign a negative strength to a relationship if one of the concepts is used in a negative way; that is, in the opposite way to which it is worded. For example, if the concepts are gnerd and friendly and the relationship has a strength of 2, but you wish to denote a negative relationship a gnerd is not friendly, then code the relationship as −2.

For an example of an annotated transcript with the facts denoted refer to Figure 6.

The goal in coding the student interviews was to produce, for each interview, a single knowledge base containing all of the information on tutor for a particular student at a specific point in time. Since the coders were told to code only that information relating to tutor, the resultant knowledge base is a map that can be interpreted as the student's definition of the concept tutor. Turning to Figure 7, you will see an example of such a map. This map is the coded representation of an interview with the student Johann as coded by myself. It represents Johann's perception of tutor at TIME 1. The underlying knowledge base contains 51 facts.

CODEF Reduces Coding Time

The use of CODEF by novices tended to decrease the coding time by 50% to 75%. Despite instructions, the coders (B, C, and D) established a pattern of initially reading the interview and marking it by hand with which concepts in the printed list of the social vocabulary they considered relevant. This part of the process took about 2 to 3 hours. Then they would reread the interview, this time using CODEF to code in the actual facts using the previously marked concepts. This took between 30 minutes and an hour and a half. Before CODEF was developed, an interview
well let's see,
there are various considerations
capable of doing the job
how wide a majority of the hall
# students can tutor
the person would actually be able to tutor
academic counseling
Academic
comments about self
broad Engineering background
interacts
um
fits in
how wide a group of people
# students share interest
they’d would (be) likely to share some interest with
sports
I would hate to make athletic ability
a particularly high point for a tutor
Social
but on the other hand,
interacts
a lot of people do tend to interact
Social
with the tutors in that way
since he's been fairly active that way
you know
someone who says they like disco music
musical tastes
would probably,
musical tastes
it's certainly a personal bias
musical tastes
that I don't like disco music
# students share interest
but I also think I have a reasonable idea
Social
tolerant
of what percentage of the hall likes it
interesting
and it's not too high
inspiring, unusual
so someone that was into playing it regularly
musical tastes
would not be as impressive to me
open door
as someone who was into mainstream rock
friendly
you know
friendly
zeppelin, stones, beatles, sort of thing
musical tastes
you know
that would just be likely
interacts
to draw people into their room
due to that
open door
and
goes into rooms
drop by other peoples' room more frequently
friendly
due to that
leisure activities
you know there are plenty of other things
musical tastes
that would get people together besides music
interacts
but that springs to mind
Figure 6: Annotated Protocol
NOTE: On the left-hand side is a segment from a prepared verbal protocol. On the
right-hand side, directly across from the key concept, is the correlated term from the
vocabulary list—SECL.
was coded by two of the coders (B and C). They took over 8 hours, and the accuracy in terms of their similarity of the coding was low. Using CODEF, the coding of a comparable interview by the same codes took less than half the time, and the accuracy was dramatically increased. Without CODEF, it takes less than 30 minutes. This is due to the stylized fashion in which the data are stored and printed.

Part of the time savings came from the fact that the coders no longer had to rely on a printed version of the concepts but rather could look up the concepts on-line. Thus the time searching for a particular social concept decreased. Another saving is attributable to the fact that the coders could now enter the data using English and did not have to deal with the coded representation of the data. Dynamic error checking, limited vocabulary, and on-line help all also helped to decrease the amount of time it took to code an interview and to increase the accuracy of the result (Carley, 1984: chap. 8).
Phases in Coding
the Student Interviews

The student interviews were coded in three phases: (1) initial protocol set, (2) final protocol set, and (3) modification by SKI. A training session preceded each of the first two phases. This three-phase approach was taken so as to compare the value of minimal training of the coders with the value of simply using an expert system. The coders were given only minimal training for details (see Carley, 1984). There were three training sessions during a two-week period. Two of these training sessions occurred prior to phase 1 of coding where 9 protocols were coded. These 9 protocols are the initial protocol set. The third and final training session followed. After this session, the final 17 protocols were coded and the coders rechecked the coding of the first 9 protocols. After all 26 protocols were coded they were processed by SKI.

*HOW WELL DOES SKI WORK*

SKI was evaluated by contrasting the level of reliability and validity in the coded representation of the verbal protocols both before and after SKI was used to explicate the implicit information in the student interviews (knowledge base comparisons). Reliability is defined in terms of two novices producing the same coded representation of a particular protocol as would a social expert. Measures of similarity based on both concepts and facts are used.

First, protocols coded by both of the novices (C and D) are compared in the section on reliability. Then, in the section on validity, four protocols coded by myself, acting as the social expert, are compared with the coded versions of these protocols produced by the novices. In both sections, the impact of both minimal training and SKI are explored. The effect of minimal training on coding reliability and validity is ascertained by comparing novice-coded knowledge bases during the middle of the training period (PHASE 1) and after the training (PHASE 2) with knowledge bases produced by other novices and with knowl-
edge bases produced by the expert. The impact of SKI on coding reliability and validity is ascertained by comparing the coded knowledge bases after having been modified by SKI (PHASE 3) with modified knowledge bases produced by other novices and with unmodified knowledge bases produced by the expert.

Measuring Similarity

Measures of knowledge base similarity were defined. Some of these measures are based on the degree to which the two knowledge bases are the same in terms of concept or fact inclusion, and others are based on the degree to which the two bases are the same in terms of overall pattern (both inclusion and exclusion). The reason for using measures of similarity in terms of concepts, rather than just measures of similarity in terms of facts, is that it was important to be able to discern the degree to which two networks were similar both at the overall structural level (facts) and the node level (concepts). Following is a list and description of these measures.

*Percentage of Shared Concepts (SC):* The number of concepts that the two knowledge bases have in common divided by the total number of concepts in the union of the two knowledge bases times 100.

*Percentage of Shared Facts (SF):* The number of facts that the two knowledge bases have in common divided by the total number of facts in the union of the two knowledge bases times 100.

*Percentage of Shared Potential Facts (SF/SC):* The number of facts that the knowledge bases have in common divided by the number of potentially sharable facts times 100. Potentially sharable facts are facts such that both concepts in the fact are in both knowledge bases whether or not the relationship is present. For both knowledge bases, that portion of the knowledge base that contained only those concepts that were in both knowledge bases was used (subknowledge bases). The percentage of shared potential facts is thus the percentage of shared facts conditioned on the shared concepts. The percentage of shared potential facts is based on the number of facts in both sub-
knowledge bases divided by the total number of facts in the union of the two subknowledge bases.

Concept Consensus (PHI.C): The degree of similarity in concept inclusion and exclusion. This is the phi coefficient calculated for a 2-by-2 table, knowledge base 1 by knowledge base 2, included and excluded concepts. The total number of concepts for the matrix is the number of concepts in the vocabulary, 217.

Fact Consensus (PHI.F): The degree of similarity in fact inclusion and exclusion. This is the phi coefficient calculated for a 2-by-2 table, knowledge base 1 by knowledge base 2, included and excluded facts. The total number of facts for the matrix is the number of potential facts, 22, 722.

Proportionate Reduction in Errors for Concepts (Q.C): This can be thought of as the proportionate reduction in errors in predicting whether or not one knowledge base has the concept based on knowledge that that concept is included in the other knowledge base. This is Yules Q based on the table for concepts described above.

Proportionate Reduction in Errors for Facts (Q.F): This can be thought of as the proportionate reduction in error in predicting whether or not one knowledge base has the fact based on knowledge that that fact is included in the other knowledge base. This is Yules Q based on the table for facts described above.

RELIABILITY—COMPARABILITY ACROSS NOVICES

In general, novices with only minimal training tend to code verbal protocols very differently (see Table 1). Novices are better at locating isolated concepts in protocols than locating facts; they are more likely to include the same concepts than they are to include the same facts. On average, when coding the same verbal protocol, only 34.14% of the concepts and 13.59% of the facts used by both coders are the same. Even given that both coders incorporated the same concepts, on average, they coded the same facts only 58.37% of the time (SF/SC). For both concepts and facts the level of consensus is quite low—PHI.C ranges from .31 to .76 and PHI.F from .10 to .51. But the proportionate reduction
in error for both is fairly high—both Q.C and Q.F tend to range over .9.  

It is interesting to note that for the same verbal protocol coder D's representation is more elaborate, that is, contains more concepts and more facts, than does coder C's. The point of interest is that C is an ex-Third Easter, with limited knowledge of the current hall residents, whereas D is a Harvard student with no personal knowledge of Third East or tutors. One would expect that C would be more of a social expert on the sociocultural environment, better acquainted with the 3E slang, and also able to code the interviews more completely than D. And yet, consistently, D produced the more elaborate, and more accurate, coded data. One reason for the counterintuitive result is the amount of effort put into the job by the two coders. Although both were

<table>
<thead>
<tr>
<th>VERBAL PROTOCOL</th>
<th>CONCEPTS</th>
<th>FACTS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SC</td>
<td>PHI.C</td>
</tr>
<tr>
<td>INITIAL 9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ERNEST 1</td>
<td>63.64</td>
<td>0.76</td>
</tr>
<tr>
<td>LOWELL 1</td>
<td>30.43</td>
<td>0.45</td>
</tr>
<tr>
<td>CASTOR 3</td>
<td>29.27</td>
<td>0.41</td>
</tr>
<tr>
<td>ERNEST 3</td>
<td>31.43</td>
<td>0.44</td>
</tr>
<tr>
<td>EUNICE 3</td>
<td>39.13</td>
<td>0.56</td>
</tr>
<tr>
<td>JUBAL 3</td>
<td>26.92</td>
<td>0.41</td>
</tr>
<tr>
<td>MANNI 3</td>
<td>27.87</td>
<td>0.33</td>
</tr>
<tr>
<td>ZACCUR 3</td>
<td>21.62</td>
<td>0.31</td>
</tr>
<tr>
<td>ZEBADIAH 3</td>
<td>36.96</td>
<td>0.49</td>
</tr>
<tr>
<td>MEAN</td>
<td>34.14</td>
<td></td>
</tr>
<tr>
<td>STD</td>
<td>10.87</td>
<td></td>
</tr>
</tbody>
</table>
given information on the current hall, Third East slang, and so on, only coder D, the Harvard student, carefully read the material; on the other hand, C, having been a member of the hall, only cursorily looked at this information. Another reason just simply has to do with the fact that D was willing to generalize from specific statements made by the interviewee to the concepts listed by the researcher—it's just part of the job; whereas C basically refused to generalize, claiming that to do so was to put words into their mouths.

Discussions with the coders revealed a greater level of agreement than that displayed in the PHASE 1 codings; for example, sometimes the coders did not use the same name for a node but had the same concept in mind, such as mellow atmosphere and California atmosphere. Based on such disagreement, the names of the more confusing concepts were modified. In stepping through the first four interviews and the coded maps with the coders, there was no point at which they actually disagreed verbally as to the appropriate coding for the interview.

Training Increases Reliability

Additional training did increase the coder's reliability. Referring to Table 2 we see that the discussions and training from the third training period increased the degree of reliability in the coding. As part of PHASE 2 the novices were instructed to recode the initial 9 protocols given the information they had just learned and then to code the remaining protocols. The recoding of the original 9 (top half of table) shows a dramatic increase in reliability. On average, both coders tended to include almost 43% of the same nodes and a little over 17% of the same facts (initial 9 protocols).

Referring to the bottom half of the table, we see that the additional training also increased coder reliability for new material. When all knowledge bases are taken together, almost 40.73% of the coded concepts are jointly used and 15.82% of the facts. Further, given that the two coders coded the same concepts they shared 52.63% of the same facts (SF/SC).

Contrasting Tables 1 and 2, we see that the level of consensus, as well as the proportionate reduction in error, has increased.
<table>
<thead>
<tr>
<th>VERBAL PROTOCOL</th>
<th>CONCEPTS SC</th>
<th>PHI.C</th>
<th>Q.C</th>
<th>FACTS SFSC</th>
<th>SF</th>
<th>PHI.F</th>
<th>Q.F</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>INITIAL 9</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ERNEST 1</td>
<td>76.19</td>
<td>0.86</td>
<td>1.00</td>
<td>68.05</td>
<td>48.33</td>
<td>0.65</td>
<td>1.00</td>
</tr>
<tr>
<td>LOWELL 1</td>
<td>51.85</td>
<td>0.66</td>
<td>0.97</td>
<td>57.14</td>
<td>20.83</td>
<td>0.35</td>
<td>1.00</td>
</tr>
<tr>
<td>CASTOR 3</td>
<td>51.06</td>
<td>0.62</td>
<td>0.95</td>
<td>59.46</td>
<td>18.03</td>
<td>0.31</td>
<td>0.99</td>
</tr>
<tr>
<td>ERNEST 3</td>
<td>36.11</td>
<td>0.48</td>
<td>0.91</td>
<td>50.00</td>
<td>17.50</td>
<td>0.30</td>
<td>0.99</td>
</tr>
<tr>
<td>EUNICE 3</td>
<td>44.44</td>
<td>0.61</td>
<td>0.98</td>
<td>70.00</td>
<td>15.38</td>
<td>0.29</td>
<td>1.00</td>
</tr>
<tr>
<td>JUBAL 3</td>
<td>18.52</td>
<td>0.27</td>
<td>0.80</td>
<td>75.00</td>
<td>7.32</td>
<td>0.16</td>
<td>0.99</td>
</tr>
<tr>
<td>MANNI 3</td>
<td>31.15</td>
<td>0.37</td>
<td>0.77</td>
<td>40.00</td>
<td>6.08</td>
<td>0.11</td>
<td>0.96</td>
</tr>
<tr>
<td>ZACCUR 3</td>
<td>26.32</td>
<td>0.39</td>
<td>0.88</td>
<td>63.64</td>
<td>9.72</td>
<td>0.21</td>
<td>0.99</td>
</tr>
<tr>
<td>ZEBADIAH 3</td>
<td>51.11</td>
<td>0.83</td>
<td>0.95</td>
<td>72.00</td>
<td>16.35</td>
<td>0.31</td>
<td>0.99</td>
</tr>
<tr>
<td><strong>FINAL 17</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EUNICE 1</td>
<td>32.43</td>
<td>0.43</td>
<td>0.87</td>
<td>50.00</td>
<td>13.43</td>
<td>0.24</td>
<td>0.99</td>
</tr>
<tr>
<td>HAZEL 1</td>
<td>41.86</td>
<td>0.53</td>
<td>0.91</td>
<td>39.29</td>
<td>23.53</td>
<td>0.38</td>
<td>1.00</td>
</tr>
<tr>
<td>JOHANN 1</td>
<td>80.71</td>
<td>0.73</td>
<td>0.98</td>
<td>50.00</td>
<td>23.64</td>
<td>0.36</td>
<td>1.00</td>
</tr>
<tr>
<td>HAZEL 2</td>
<td>35.71</td>
<td>0.46</td>
<td>0.88</td>
<td>66.67</td>
<td>14.71</td>
<td>0.26</td>
<td>0.99</td>
</tr>
<tr>
<td>JACQUES 2</td>
<td>40.00</td>
<td>0.55</td>
<td>0.95</td>
<td>47.82</td>
<td>18.18</td>
<td>0.31</td>
<td>1.00</td>
</tr>
<tr>
<td>AARON 3</td>
<td>48.72</td>
<td>0.53</td>
<td>0.86</td>
<td>38.10</td>
<td>16.67</td>
<td>0.26</td>
<td>0.98</td>
</tr>
<tr>
<td>DEEY 3</td>
<td>31.03</td>
<td>0.39</td>
<td>0.81</td>
<td>65.00</td>
<td>14.13</td>
<td>0.27</td>
<td>0.99</td>
</tr>
<tr>
<td>EDITH 3</td>
<td>46.67</td>
<td>0.61</td>
<td>0.98</td>
<td>39.47</td>
<td>18.99</td>
<td>0.34</td>
<td>1.00</td>
</tr>
<tr>
<td>HAZEL 3</td>
<td>27.03</td>
<td>0.37</td>
<td>0.84</td>
<td>28.92</td>
<td>7.69</td>
<td>0.14</td>
<td>0.98</td>
</tr>
<tr>
<td>IAN 3</td>
<td>34.62</td>
<td>0.52</td>
<td>0.97</td>
<td>40.00</td>
<td>12.24</td>
<td>0.22</td>
<td>0.99</td>
</tr>
<tr>
<td>ISHTAR 3</td>
<td>30.77</td>
<td>0.43</td>
<td>0.90</td>
<td>66.67</td>
<td>12.50</td>
<td>0.22</td>
<td>1.00</td>
</tr>
<tr>
<td>JACQUES 3</td>
<td>42.86</td>
<td>0.53</td>
<td>0.80</td>
<td>48.15</td>
<td>12.00</td>
<td>0.21</td>
<td>0.99</td>
</tr>
<tr>
<td>JOHANN 3</td>
<td>43.64</td>
<td>0.56</td>
<td>0.94</td>
<td>57.58</td>
<td>12.32</td>
<td>0.25</td>
<td>0.99</td>
</tr>
<tr>
<td>LORENO 3</td>
<td>40.00</td>
<td>0.51</td>
<td>0.91</td>
<td>51.61</td>
<td>16.87</td>
<td>0.29</td>
<td>0.99</td>
</tr>
<tr>
<td>LOWELL 3</td>
<td>38.24</td>
<td>0.43</td>
<td>0.80</td>
<td>40.98</td>
<td>15.28</td>
<td>0.26</td>
<td>0.99</td>
</tr>
<tr>
<td>MAUREEN 3</td>
<td>42.00</td>
<td>0.52</td>
<td>0.90</td>
<td>34.15</td>
<td>10.17</td>
<td>0.21</td>
<td>0.99</td>
</tr>
<tr>
<td>WOODIE 3</td>
<td>36.00</td>
<td>0.48</td>
<td>0.90</td>
<td>50.00</td>
<td>9.71</td>
<td>0.19</td>
<td>0.98</td>
</tr>
<tr>
<td><strong>MEAN</strong></td>
<td>40.73</td>
<td></td>
<td></td>
<td>52.63</td>
<td>15.82</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>STD</strong></td>
<td>11.34</td>
<td></td>
<td></td>
<td>12.62</td>
<td>7.76</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Coder D is still producing more elaborate maps; however, the proportion of shared concepts and facts has increased. The level of reliability is still not particularly high across these two coders. This is due to the coding task being nontrivial, the minimal training, and the difference in coder competence and attention. Nevertheless, this exploration suggests that at least minimal training is necessary in order to increase cross-coder reliability.

Novices are better at picking out concepts than facts—in both Tables 1 and 2 the percentage of shared concepts is higher than the percentage of shared facts. As the coders are trained, their ability to pick out concepts increases much more rapidly than does their ability to pick out facts—there is on average an approximate 6% increase in the percentage of shared concepts and 2% increase in the percentage of shared facts. A possible explanation is that picking out facts, as opposed to concepts, requires knowledge of the sociocultural environment, that is, knowledge of which concepts can be potentially related. Whereas, picking out concepts, as opposed to facts, requires simple pattern-matching skills and limited generalization. To the extent that the set of concepts provided by the researcher in the vocabulary list utilizes the same words that are utilized in the protocols only pattern matching is required. In this case very little training should be needed. To the extent that the set of concepts provided by the researcher in the vocabulary list is more general or specialized terms more extensive training may be needed. In fact, this was the case. Even during PHASE 1, the coders always picked out concepts that could be found by pattern matching; for example, they always coded the phrase *Third Easter* when it appeared in a protocol using the vocabulary concept *Third Easter*. All of the concepts that were not coded, but should have been, were ones where generalization was required; for example, they were likely not to code the phrase *you always see this guy in the library so he won’t be any good* if it occurred in the protocol using the vocabulary concept *spends time on hall*. It is important that even the minimal training provided dramatically improved the coder’s ability to generalize, that is, pick out concepts, but not their ability to make the correct social inference, that is, pick out facts.
SKI Increases Reliability

After SKI has been used to modify all of the novice-coded maps, the degree of reliability across coders increases dramatically. When SKI is used on the novice-produced knowledge bases the resultant modified knowledge bases are much more elaborate than the originals, that is, contain more concepts and facts than do the originals. And, this elaboration is not random and has served to make the codings produced by the two novices more similar. Referring to Table 3, we see that after modification by SKI, 62.38% of the concepts and 49.02% of their facts are shared. Note that the level of consensus over concepts now averages at .702 (PHI.C) and over facts at .66 (PHI.F). And finally the percentage of shared potential facts, that is, the percentage of shared facts given shared concepts, has risen to 84.20% across all 26 maps (SF/SC). Thus one of the major effects of using social knowledge is to provide those relations between concepts that are socially known to a coder who has little current knowledge of the social unit.

To the extent that the training of novices is a process of making the novices experts in the sociocultural environment, this exploratory study suggests that SKI can be used instead of extensive training to increase the reliability in the coded output. Contrasting Tables 1 and 2, PHASE 1 and PHASE 2, we see that on average training served to increase slightly the overall reliability. Table 3 suggests that the use of SKI (PHASE 3) to modify the maps may actually be more effective than further training. Note the increase of 17.53% and 20.09%, respectively; whereas, going from PHASE 2 to PHASE 3 results in for these same 9 bases, an increase of 32.78% and 121.56%, respectively. Overall, SKI increase the comparability of coders' maps much more than several training sessions. SKI improves fact detection, whereas training improves concept detection.

VALIDITY—COMPARISON WITH SOCIAL EXPERT

After the initial training sessions, there is limited agreement between novice and expert codings (see Table 4). On average,
<table>
<thead>
<tr>
<th>VERBAL PROTOCOL</th>
<th>CONCEPTS SC</th>
<th>PHI.C</th>
<th>Q.C</th>
<th>FACTS SFSC</th>
<th>SF</th>
<th>PHI.F</th>
<th>Q.F</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>INITIAL 9</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ERNEST 1</td>
<td>83.87</td>
<td>0.90</td>
<td>1.00</td>
<td>90.48</td>
<td>68.47</td>
<td>0.62</td>
<td>1.00</td>
</tr>
<tr>
<td>LOWELL 1</td>
<td>66.67</td>
<td>0.78</td>
<td>1.00</td>
<td>77.87</td>
<td>54.19</td>
<td>0.73</td>
<td>1.00</td>
</tr>
<tr>
<td>CASTOR 3</td>
<td>92.76</td>
<td>0.89</td>
<td>1.00</td>
<td>93.87</td>
<td>60.59</td>
<td>0.82</td>
<td>1.00</td>
</tr>
<tr>
<td>ERNEST 2</td>
<td>58.00</td>
<td>0.87</td>
<td>1.00</td>
<td>79.31</td>
<td>35.20</td>
<td>0.540</td>
<td>1.00</td>
</tr>
<tr>
<td>EUNICE 3</td>
<td>80.77</td>
<td>0.88</td>
<td>1.00</td>
<td>92.42</td>
<td>68.33</td>
<td>0.80</td>
<td>1.00</td>
</tr>
<tr>
<td>JUBAL 3</td>
<td>36.36</td>
<td>0.46</td>
<td>1.00</td>
<td>82.69</td>
<td>28.87</td>
<td>0.45</td>
<td>1.00</td>
</tr>
<tr>
<td>MANNI 3</td>
<td>51.19</td>
<td>0.55</td>
<td>0.99</td>
<td>71.90</td>
<td>34.95</td>
<td>0.51</td>
<td>0.99</td>
</tr>
<tr>
<td>ZACCUR 3</td>
<td>43.64</td>
<td>0.56</td>
<td>1.00</td>
<td>88.90</td>
<td>29.80</td>
<td>0.51</td>
<td>1.00</td>
</tr>
<tr>
<td>ZEBADIAH 3</td>
<td>74.60</td>
<td>0.81</td>
<td>1.00</td>
<td>92.15</td>
<td>63.80</td>
<td>0.78</td>
<td>1.00</td>
</tr>
<tr>
<td><strong>FINAL 17</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EUNICE 1</td>
<td>54.39</td>
<td>0.65</td>
<td>1.00</td>
<td>66.45</td>
<td>44.78</td>
<td>0.63</td>
<td>1.00</td>
</tr>
<tr>
<td>HAZEL 1</td>
<td>55.56</td>
<td>0.62</td>
<td>0.99</td>
<td>77.24</td>
<td>43.60</td>
<td>0.61</td>
<td>0.99</td>
</tr>
<tr>
<td>JOHANN 1</td>
<td>70.83</td>
<td>0.79</td>
<td>1.00</td>
<td>81.75</td>
<td>60.59</td>
<td>0.75</td>
<td>1.00</td>
</tr>
<tr>
<td>HAZEL 2</td>
<td>51.65</td>
<td>0.62</td>
<td>1.00</td>
<td>88.24</td>
<td>41.98</td>
<td>0.59</td>
<td>1.00</td>
</tr>
<tr>
<td>JAQUES 2</td>
<td>55.07</td>
<td>0.65</td>
<td>1.00</td>
<td>69.67</td>
<td>36.95</td>
<td>0.57</td>
<td>1.00</td>
</tr>
<tr>
<td>AARON 3</td>
<td>70.09</td>
<td>0.70</td>
<td>1.00</td>
<td>82.50</td>
<td>57.36</td>
<td>0.72</td>
<td>1.00</td>
</tr>
<tr>
<td>DEETY 3</td>
<td>57.65</td>
<td>0.63</td>
<td>1.00</td>
<td>88.46</td>
<td>45.87</td>
<td>0.64</td>
<td>1.00</td>
</tr>
<tr>
<td>EDITH 3</td>
<td>76.80</td>
<td>0.84</td>
<td>1.00</td>
<td>76.05</td>
<td>53.70</td>
<td>0.71</td>
<td>1.00</td>
</tr>
<tr>
<td>HAZEL 3</td>
<td>55.56</td>
<td>0.65</td>
<td>1.00</td>
<td>78.86</td>
<td>40.00</td>
<td>0.58</td>
<td>1.00</td>
</tr>
<tr>
<td>IAN 3</td>
<td>54.29</td>
<td>0.68</td>
<td>1.00</td>
<td>80.88</td>
<td>41.98</td>
<td>0.60</td>
<td>1.00</td>
</tr>
<tr>
<td>ISHTAR 3</td>
<td>63.64</td>
<td>0.73</td>
<td>1.00</td>
<td>84.40</td>
<td>62.25</td>
<td>0.77</td>
<td>1.00</td>
</tr>
<tr>
<td>JAQUES 3</td>
<td>65.08</td>
<td>0.72</td>
<td>1.00</td>
<td>81.72</td>
<td>54.36</td>
<td>0.70</td>
<td>1.00</td>
</tr>
<tr>
<td>JOHANN 3</td>
<td>57.58</td>
<td>0.65</td>
<td>1.00</td>
<td>83.13</td>
<td>42.81</td>
<td>0.61</td>
<td>1.00</td>
</tr>
<tr>
<td>LORENZO 3</td>
<td>62.50</td>
<td>0.72</td>
<td>1.00</td>
<td>91.16</td>
<td>46.88</td>
<td>0.66</td>
<td>1.00</td>
</tr>
<tr>
<td>LOWELL 3</td>
<td>65.52</td>
<td>0.69</td>
<td>1.00</td>
<td>82.25</td>
<td>52.17</td>
<td>0.68</td>
<td>1.00</td>
</tr>
<tr>
<td>MAUREEN 3</td>
<td>61.11</td>
<td>0.67</td>
<td>1.00</td>
<td>71.58</td>
<td>48.46</td>
<td>0.65</td>
<td>1.00</td>
</tr>
<tr>
<td>WOODIE 3</td>
<td>68.66</td>
<td>0.76</td>
<td>1.00</td>
<td>90.43</td>
<td>58.54</td>
<td>0.73</td>
<td>1.00</td>
</tr>
<tr>
<td><strong>MEAN</strong></td>
<td>62.38</td>
<td></td>
<td></td>
<td>82.40</td>
<td>49.02</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>STD</strong></td>
<td>11.22</td>
<td></td>
<td></td>
<td>7.18</td>
<td>11.25</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### TABLE 4
Validity—Phase 1

<table>
<thead>
<tr>
<th>VERBAL PROTOCOL</th>
<th>CONCEPTS</th>
<th>FACTS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SC</td>
<td>PHIL.C</td>
</tr>
<tr>
<td>CODER C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>JOHANN</td>
<td>46.15</td>
<td>0.63</td>
</tr>
<tr>
<td>MINERVA</td>
<td>61.11</td>
<td>0.73</td>
</tr>
<tr>
<td>CODER D</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LORENZO 1</td>
<td>59.38</td>
<td>0.72</td>
</tr>
<tr>
<td>LORENZO 3</td>
<td>52.63</td>
<td>0.62</td>
</tr>
</tbody>
</table>

When coding the same verbal protocol, only 54.82% of the concepts and 20.45% of the facts were used by both novice and expert. Given that both novice and expert included the same concepts, there is only a 38.05% agreement on facts (SF/SC). 
Most of the mistakes by the novice coders are errors of exclusion.

Training Increases Validity

Referring to Table 5, we see that minimal training improved coder accuracy in including both concepts (average percentage of shared concepts is 62.49%) and facts (average percentage of shared facts is 23.48%). Contrasting Tables 4 and 5 we see that the level of consensus also increased for both concepts (PHIL.C) and facts (PHIL.F). Note, the level of improvement is greater for concepts than for facts.

Note, this training (training session 3) included giving the novices information on the hall and on concept types; but, not giving the coder any information on the knowledge bases that I had coded. Nor was information provided on what facts the coder had left out or should not have included in those specific protocols looked at in Tables 4 and 5.
<table>
<thead>
<tr>
<th>VERBAL PROTOCOL</th>
<th>CONCEPTS SC</th>
<th>CONCEPTS PHI.C</th>
<th>CONCEPTS Q.C</th>
<th>FACTS SFSC</th>
<th>FACTS SF</th>
<th>FACTS PHI.F</th>
<th>FACTS Q.F</th>
</tr>
</thead>
<tbody>
<tr>
<td>CODER C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JOHANN 1</td>
<td>60.71</td>
<td>0.73</td>
<td>0.98</td>
<td>40.00</td>
<td>22.22</td>
<td>0.37</td>
<td>1.00</td>
</tr>
<tr>
<td>LORENZO 3</td>
<td>37.74</td>
<td>0.49</td>
<td>0.91</td>
<td>43.59</td>
<td>17.14</td>
<td>0.31</td>
<td>0.99</td>
</tr>
<tr>
<td>CODER D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JOHANN 1</td>
<td>74.07</td>
<td>0.83</td>
<td>1.00</td>
<td>44.44</td>
<td>25.00</td>
<td>0.41</td>
<td>1.00</td>
</tr>
<tr>
<td>LORENZO 1</td>
<td>76.47</td>
<td>0.85</td>
<td>1.00</td>
<td>76.67</td>
<td>25.64</td>
<td>0.44</td>
<td>1.00</td>
</tr>
<tr>
<td>LORENZO 3</td>
<td>60.47</td>
<td>0.71</td>
<td>0.97</td>
<td>67.65</td>
<td>21.14</td>
<td>0.35</td>
<td>0.99</td>
</tr>
<tr>
<td>MINERVA 1</td>
<td>65.45</td>
<td>0.74</td>
<td>0.98</td>
<td>45.67</td>
<td>29.73</td>
<td>0.46</td>
<td>1.00</td>
</tr>
</tbody>
</table>

SKI Increases Validity

In Table 6 we see the results of comparing the modified versions of the maps coded by the novices with the nonmodified maps coded by myself. The results demonstrate that the use of SKI increases validity. The modified knowledge bases produced by SKI are much more elaborate than the original knowledge bases produced by the novices. This elaboration is not random and has served to make the codings produced by the novices and the social expert more similar. In Table 6 we see that after modification by SKI, 77.30% of the concepts and 62.94% of the facts are shared. Further, the degree of consensus is also high, for concepts this is .83 (PHI.C) and for facts .77 (PHI.F).

Contrasting Table 4 and 5, we see that training causes a percentage increase in similarity of concept inclusion of 13.99% and for facts of 14.81%, whereas in moving from Table 5 to Table 6 there is a percentage increase in similarity of concept inclusion of
23.72% and 168.07% for facts. SKI makes it possible to produce rich knowledge bases that reflect a social expert's perception of an interview without having to have a social expert do the coding or having to spend extensive time training coders.

As a final point, note that on average coder D produced much more accurate knowledge bases than did coder C (refer to Tables 4 and 5). After SKI was used, coder D’s knowledge bases were proportionally more accurate than coder C’s. Comparison of the verbal protocols with the knowledge base produced by coder C shows that coder C failed to include some of the information that is explicit in the protocol; for example, there were specific sentences relating to the concept tutor that were just not coded. SKI can’t catch big coding mistakes; that is, it cannot catch errors in coding explicit information. If the coder who extracts the explicit structure from verbal protocol is inaccurate, even the results produced by SKI won’t be that accurate. What SKI does catch are errors of explication. Consequently, if all explicit information is
included, SKI will correctly add the implicit information and thus produce accurate knowledge bases.

**A COMPARISON WITH OTHER APPROACHES FOR ANALYZING TEXT**

A wide variety of techniques for analyzing text have been used in the social sciences. These techniques include conversational analysis (Sacks, 1972), discourse analysis (Stubbs, 1983), computational hermeneutics (Alker et al., 1985; Mallery, 1985; Mallery and Duffy, 1986), script and plot analysis (Abelson, 1976; Abelson and Reich, 1969; Schank and Abelson, 1977; Lehnert, 1975, 1981; Wilensky, 1981; Lehnert and Ringle, 1982; Heise, 1987), content analysis (Stone, 1968a; Stone et al., 1968b; Markoff et al., 1974; Weber, 1984, 1985; Namenwirth and Weber, 1987), the extraction of affective meaning (Gollob, 1968; Heise, 1969, 1970), behavior-based protocol analysis (Newell and Simon, 1972; Ericsson and Simon, 1984), computer-based semantic analysis (Hildum, 1963; Danowski, 1986; Danowski and Andrews, 1985), and decision maps and logic (Axelrod, 1976; Heise, 1987). The approach presented in this article—an expert system to explicate social knowledge—differs in several ways from the approaches mentioned above:

- It uses a cognitively based knowledge representation scheme.
- It is based on a cognitive theory of the nature of knowledge, knowledge acquisition, and knowledge dissemination.
- It is concerned with the extraction of meaning as operational definition; rather than as implicit roles or plot-based behavior and motivation.

None of the other approaches has all of these characteristics.

The proposed approach and the script-based approaches (Abelson, 1976; Abelson and Reich, 1969; Schank and Abelson, 1977; Lehnert, 1975, 1981; Wiensky, 1981; Lehnert and Ringle,
1982; Heise, 1987) are based on cognitively predicated knowledge representation schemes. Conversational analysis (Sacks, 1972), discourse analysis (Stubbs, 1983), content analysis (Stone et al., 1968a, 1968b; Weber, 1984, 1985; Namenwirth and Weber, 1987), behavior-based protocol analysis (Newell and Simon, 1972; Ericsson and Simon, 1984), and computer-based semantic analysis (Hildum, 1963; Danowski, 1986; Danowski and Andrews, 1985), although sometimes used to discover cognitive aspects of behavior, are not cognitively based approaches. That is, not only are they not based on a cognitively predicated knowledge representation scheme, they are also not based on cognitive theories of knowledge acquisition and dissemination.

Many of the approaches to analyzing text use conceptual network knowledge representation schemes built out of relationships as concepts. These approaches can be divided into two sets dependent on whether or not they are based on a cognitive theory of the nature of knowledge, knowledge acquisition, and dissemination. Heise's (1969, 1970) and Gollob's (1968) method for extracting affective meaning, Danowski's (1986; Danowski and Andrews, 1986) and Hildum's (1963) computer-based techniques for extracting concept networks, and Axelrod's (1976) approach for extracting decision maps are not based on such cognitive theories. Whereas Heise's (1987) approach for showing decision logic, the script-based approaches, and the proposed approach are cognitively based. All of these methods can be used to represent meaning as implicit evaluation of the explicit text. Those methods that are not cognitively based cannot be used as a basis for determining what knowledge is implicit in the text. That is, these methods can be used to evaluate the words in a sentence or paragraph but not to tell you what words or sentences are implicit. Whereas cognitively based methods such as the proposed method and the script-based methods (e.g., Schank and Abelson, 1977; Heise, 1987) can determine what knowledge is implicit.

Most methods differ from the proposed method in their treatment of meaning. Austin (1962) argued that utterances are
actions. Conversational analysis (Sacks, 1972), discourse analysis (Stubbs, 1983), and computational hermeneutics (Alker et al., 1985; Mallery, 1985; Mallery and Duffy, 1986) follow from this perspective and seek textual understanding by analyzing the coherence, pattern, and nature of exchange in protocols. The transcript is treated as dialogue. Meaning derives from the organization, control, and sequencing of the conversation. The script and plot-based methods (Abelson, 1976; Abelson and Reich, 1969; Schank and Abelson, 1977; Lehnhert, 1975, 1981; Wilensky, 1981; Lehner and Ringle, 1982; Heise, 1987) and behavior-based protocol analysis (Newell and Simon, 1972; Ericsson and Simon, 1984) do not treat utterance as action but as representative of actions and goals. The transcript is treated as plot and sequences of action and reaction. It is also concerned with the sequencing of action in the transcript. The specific order of the sentences in the text matters to the analysis. Meaning is thought of in terms of behavior and motivation as it appears in the development of plot. Unlike the preceding methods the proposed method is not concerned with the action in the dialogue. The transcript is treated as a sample of the way a single individual represents a problem. The order of the sentences in some cases will not matter. Meaning is the operational definition rather than the implicit roles that appear in the organization of the conversation.

Like the proposed method, content analysis (Stone et al., 1968a, 1968b; Weber, 1984, 1985; Namenwirth and Weber, 1987), the extraction of affective meaning (Gollob, 1968; Heise, 1969, 1970), computer-based semantic analysis (Hildum, 1963; Danowski, 1986; Danowski and Andrews, 1985), and decision maps and logic (Axelrod, 1972, 1976; Heise, 1987) treat utterance as knowledge rather than action. The representation scheme typically associated with content analysis (Stone et al., 1968a, 1968b) focuses on node or word counts and relies exclusively on explicit information. The concern is with the categorization and quantification of the separate concepts. Meaning becomes the covariance between these counts. Whereas, for methods based on network representation schemes such as the proposed method, the sentence, paired words, fact (Minsky, 1975), or kernel assertion (Osgood, 1963) is the unit
of analysis. The concern is with the structure of linked information. Meaning is treated as a function of what concepts are related to what other concepts. For Gollub (1968) and Heise (1969, 1970) the meaning of a concept is numerically determined and is a function of the combined affective value of the related concepts. The relationship between concepts is used for doing affective evaluation. Concepts are treated as instantiations and are numerically evaluated. Only explicit concepts are used. Concepts in isolation can have an affective evaluation; that is, a meaning. For Danowski and Andrews (1985; Danowski, 1986) and Hildum (1963) meaning is inherent in the pattern of connected concepts and only explicit concepts are used. For Axelrod (1976) and Heise (1987) meaning is implicit in the logic of the entire network. The relationship between concepts is used for doing logical evaluation. In the proposed method the meaning of a concept is its symbolic definition and is thus a function of the set of concepts to which it is directly and indirectly related. Concepts are treated as place holders and the specific evaluation is not determined. In the proposed method both affective and logical evaluation are carried by the relationship. The focus is on the relationship between concepts and use is made of implicit as well as explicit information. As with Axelrod (1976) and Heise (1987) concepts are treated as place holders and are not numerically evaluated. 19

The proposed approach is distinct from other approaches used to analyze text because it is based on a cognitively motivated knowledge representation scheme and a cognitive theory of knowledge, and because it extracts meaning as operational definition. Although unique in these aspects, the proposed approach may not be appropriate for every text analysis task. For those tasks where the researcher is interested in extracting the "cognitive map" or "definition" or "view of the problem" it is an appropriate methodology. Moreover, the generality of the representation scheme is such that CODEF can be used to code information in ways that subsume many other approaches, such as decision maps (Axelrod, 1976). For those tasks that also require an understanding of vast amounts of social knowledge in order to interpret the protocols, SKI becomes an appropriate methodology. In fact, SKI
is the only method for text interpretation that utilizes social expertise to locate and incorporate implicit meaning. It is able to do this because it is based on a cognitive theory of knowledge.

GENERAL COMMENTS

Researchers in artificial intelligence have found that large databases of task specific knowledge allow the production of solutions that rival those of human experts. Such systems, expert systems, can be built just in case there is an expert. For the task coding protocols it was argued that two types of expertise are required. That is, social experts are experts because they know more than the novice and because they have collated knowledge so that they know "who knows what." This second type of expertise, consensual knowledge, allows the expert to interpret information in a protocol by reducing the number of possible inferences to just those that are congruent and shared social knowledge. For a specific society, social knowledge can be codified and used to explicate verbal protocols. By integrating theories on the nature of knowledge, social expertise, and coding it is possible to develop unified cognitive bases on which an expert system for coding verbal protocols can be developed. Such an expert system, acting as a lexical analyzer, can use social knowledge to diagnose explication errors in coded protocols. SKI is such an expert system. The knowledge base for SKI is the Third Easters' social knowledge about tutors. The use of SKI improves both the reliability and validity of the coding.

The maps coded by the novice, coder D, during PHASE 2 have an average of 33.021 concepts and 50.583 facts per knowledge base. These same knowledge bases, after modification by SKI, contain an average of 48.604 concepts and 202.188 facts. This is a 47.19% increase in concepts, and a 300% increase in facts. SKI as social expert elaborates the knowledge base not simply by increasing size (number of concepts), but more importantly by increasing interconnectedness (ratio of facts to concepts). In other words,
SKI is adding implicit knowledge, not just adding independent words.

The goal is to move toward a situation where coding is totally automated. Full automation requires that the machine both parse and translate the verbal protocol. Total automation is not achieved; however, SKI and CODEF go a long way in this direction as they remove many of the coder idiosyncrasies and assure consistent interpretation of a large set of available facts. The novice parses the protocol and translates explicit information into the general categories used by the researcher. CODEF assists in this process. SKI expects the novice-coded knowledge base to contain all of the information that is explicit in the verbal protocol. SKI is not able to include a set of concepts that are structurally distinct from those included. For example, SKI cannot add information on the importance of academics if there are no concepts related to the importance of academics to begin with in the knowledge base. Thus SKI cannot catch big coding mistakes. SKI can catch smaller mistakes, that is, mistakes of explication. Such mistakes are due to the human novice coder not knowing a particular facet of the sociocultural environment; for example, if the coder used the concept gnerd and antisocial but did not connect them then SKI would know to add that relationship because at MIT it is part of the social background information that gnerds are antisocial.

There are many advantages to the approach presented in this article. Having the social expert construct the social knowledge base and social vocabulary formalizes the knowledge thereby creating a potential for computer-assisted coding. Further, such formalized knowledge makes it possible to compare knowledge bases analytically on a fact-by-fact or concept-by-concept basis. Tools such as CODEF and SKI make it possible for the researcher to analyze large quantities of verbal data with relative ease, precision, and without completely foregoing the richness of verbal description. Among the quantitative advantages to using these procedures are increased reliability in the coded data, decreased training required for novices, increased utilization of novice
coders, decreased routine data processing by the social expert, and
decreased coding time. There are also qualitative advantages to
this approach. The process of articulating one's knowledge in
order to use SKI increases the social expert's awareness and under-
standing of this knowledge. This can in turn improve analysis.
Moreover, the social knowledge base as articulated by the social
expert serves as a compendium of his or her knowledge that can
then be used by other researchers who are not social experts in that
particular sociocultural environment.

SKI is more than just a computer-assisted approach for increas-
ing the reliability and validity of coded protocols. SKI is predi-
cated on cognitively based theories of the nature of knowledge,
social expertise, and coding. ADDSOC as the inference engine is a
model of the way the social expert uses social knowledge to
interpret protocols. At a theoretical level, extensions of this work
should seek to explicate further the underlying theories of knowl-
dge acquisition, knowledge representation, social validity, social
knowledge, and expert behavior.

At a functional level, SKI reduces the need for the coders to
have social expertise, thus minimizing the extent of training
required. One extension of this work would be to determine under
what conditions the use of SKI is to be preferred to extensive coder
training. I would suggest, on grounds of efficiency and system-
acticity, that either large numbers of protocols (greater than 10) or
very long protocols (greater than 3 pages) warrant an automated
approach such as that used herein. This research also suggests that
the process of training the novice coders involves teaching them
not only the mechanics of coding but also social knowledge.
Moreover, the acquisition of social knowledge is a complex pro-
cess involving the acquisition of both new concepts and new
definitions. Concepts appear to be more easily learned. Hence, if
the researcher wishes to rely exclusively on human coders, care
must be taken to teach them not only the words, but the legitimate
ways in which those words can be used in the sociocultural envi-
nronment. Exactly what the best method for teaching novices social
knowledge is a question for further research. And, finally, were an
appropriate explanation facility added, SKI could be used as a device for teaching students about that sociocultural environment.

This work is indicative of the way expert systems and ideas from artificial intelligence can be used in the social sciences. Verbal protocols are rich and often vivid windows on human behavior. Verbal protocols are also complex and difficult to analyze in a systematic and detailed fashion. As we develop more comprehensive theories of knowledge and expertise we will be better able to build intelligent systems for analyzing and interpreting verbal protocols. The expertise of such systems will lie not in their problem-solving skills but in their knowledge of the sociocultural environment. With the advent of such systems we will have a tool that will enable us to develop more accurate theories of human behavior.

APPENDIX A: CODEF DETAILS

CODEF helps the user code a protocol by asking a series of questions. These questions rely on the social expert's social vocabulary. If the vocabulary does not exist then CODEF asks the coder for a set of concepts used in the protocol, checks to see that these concepts are in the social vocabulary, and then asks the user on a pair-by-pair basis to identify whether or not a relationship exists between that pair of concepts. If a concept is not in the social vocabulary CODEF asks the user if he or she wants to add the new concept to the social vocabulary or to enter a different concept. The output of CODEF is a data file containing a knowledge base representation of the information in the verbal protocol that relates to the researcher's topic. The researcher may choose to create a separate knowledge base for each topic; thereby, creating more than one data file.

The CODEF algorithm is general and can be easily modified to accept the full functionality of the representation scheme previously described. Currently, however, CODEF is a prototype and thus it restricts the number of facts types, concept types, and the number of strength levels. These limitations were built into the current version of CODEF for two reasons—rapid development and rapid data entry. These limitations and other notes on operationalization are described in the following sections.
CONSTRAINTS ON CONCEPTS

CODEF can handle up to four types of concepts. The meaning attached to each type of concept is up to the researcher. The researcher must classify each concept in the social vocabulary as being of one of these types. If the researcher prefers not to classify concepts, this can be accomplished by having a single type.

CONSTRAINTS ON RELATIONSHIPS

Directionality: CODEF does not explicitly store information on directionality. If the researcher wishes to treat bidirectional facts differently than unidirectional facts two data files would need to be produced, one for each directionality. In this case, the coder must be instructed ahead of time to put bidirectional facts in file-a and unidirectional facts in file-b.

Strength: Strength ranges from 0 to 3. A 0 indicates that the relationship is not present. All uncoded facts are treated as having a strength of 0. The meaning of the values 1, 2, 3 is at the discretion of the researcher.

Sign: A positive relationship Chocolate is very good is denoted by using a strength that is greater than 0. A negative relationship Chocolate is not very good is denoted by using a strength that is less than 0. This gives an effective range of -3 to 3.

Meaning: Meaning is a function of concept type. Recall that the pairing of concept types can be used to limit or define the meaning of the relationship. Thus the current version of CODEF admits up to 16 different meanings for relationships. If the user has only one type of concept then all relationships have the same meaning. Note, an alternate way of encoding meaning using CODEF is to code each data file as continuing facts with different meanings. In this case, the researcher would set the number of concept types to 1 and could have as many meanings as desired.

In a data file there can be at most one relationship between two concepts. If there is more than one relationship between two concepts, the corresponding facts would be stored in separate files.

LIMITATIONS ON FACTS

CODEF does not explicitly code fact type. Currently, if the researcher wishes to code multiple types of facts, then each type of fact must be
stored in a different data file. For example, in coding the social expert’s knowledge base three different types of facts differing in the level of social validity are used (see Carley, 1984: chap. 6). Whereas, in coding the student interviews all extracted facts are treated as being of the same type; hence all facts extracted from a single interview are placed in the same data file.

Each fact must contain two distinct concepts. Note, this is not a severe limitation. The only type of fact that cannot be coded is one with the same concept in both positions, such as I am I. Other pieces of information can be broken down into a set of diadic relations.

APPENDIX B: TRANSCRIPT PROCESSING

Before the written transcripts of the verbal protocols were given to the coders they were preprocessed using the following rules.

(1) All student and tutor candidate names were replaced by aliases.
(2) Each topic segment began a new paragraph.
(3) Each semantic segment within that topic segment began a new physical line. A semantic segment refers to a clause, even if incomplete, that conveys a single piece of information. Typically, segments are clauses within complex sentences or complete short sentences.

PREPARED TRANSCRIPT

Following is an excerpt from a preprocessed transcript, that is, a protocol. This selection is from an interview with the student Johann at TIME 1. It contains some of Johann’s reasons for choosing possible candidates for tutor.

well let’s see,
there are various considerations
how wide a majority of the hall
the person would actually be able to tutor

um
how wide a group of people
they’d would (be) likely to share some interest with
I would hate to make athletic ability
a particularly high point for a tutor
but on the other hand,
a lot of people do tend to interact
with the tutors in that way
they have with Rick
since he's been fairly active that way
you know
someone who says they like disco music
would probably,
it's certainly a personal bias
that I don't like disco music
but I also think I have a reasonable idea
of what percentage of the hall likes it
and it's not too high
so someone that was into playing it regularly
would not be as impressive to me
as someone who was into mainstream rock
you know
zeplin, stones, beatles, sort of thing
you know
that would just be likely
to draw people into their room
and
drop by other peoples' room more frequently
due to that
you know there are plenty of other things
that would get people together besides music
but that springs to mind

NOTES

1. Since the topics of concern in the examples are social in nature and require an understanding of the sociocultural environment, the expert will be referred to as a social expert.

2. An expert system is a special type of computer program that "achieves high performance by using knowledge to make the best use of its time" (Hayes, 1983). An expert system uses heuristics to (a) narrow the search for an appropriate solution to the problem, (b) determine the important aspects of the problem, and (c) choose between competing solutions to a problem. Exemplar systems include R1 (McDermott, 1981), Prospector (Duda, 1978), and MYCIN (Buchanan, 1984).

3. CODEF and ADDSOC (the inference engine for SKI) are part of a set of programs collectively referred to as Frame Technology (Carley, 1984, 1986a), written in the high-level programming language C (Kernighan, 1978). These programs, along with the UNIX operating system (Kernighan, 1984), form the basis for a management information system for dealing with verbal network data. The language C was chosen for reasons of availability, portability, speed of data processing, and interfaceability with other software. To get copies of the programs and manuals, call or write the author. Both programs run under a Berkeley 4.2 operating system and can be compiled for an RT, SUN, or MicroVax. A PC version of CODEF is also available.
4. A possible alternate interpretation would result if the proposed representation scheme is used to code relationships as logical relations in much the same way that one would use a first-order predictable calculus. To code the problem-solving process that the individual goes through, Ericsson and Simon suggest that first-order predicate calculus can be used as a representation scheme. For examples and a discussion of this approach, refer to Ericsson (1984: chaps. 6 and 7). They use the functional notation $R(X, Y, \ldots)$ where the $R$'s are relations such as $more$ and $X, Y$, and so on are arguments such as $position 1, position 2$, and so on. The representation scheme suggested above can under certain conditions reduce to the first-order predicate calculus approach; for example, when there are exactly two arguments for the relationship each of which corresponds to one of the two concepts.

5. A concept's definition is elaborated if it contains the definitions of many supporting concepts.

6. What is measured as social knowledge and the consequent characteristics of that knowledge are dependent on the level of social validity used by the researcher (Carley, 1986b). The researcher will choose a level of social validity relative to some relevant majority. For example, the researcher might define the level of validity as 100% and the relevant majority as "all of the members of the social unit." In this case social knowledge would be exactly the set of facts known by every member of the social unit. Still another researcher might choose a validity level of 100% and define the relevant majority as a simple majority. In this case, social knowledge would be exactly the set of facts such that each fact is known by at least a simple majority of the members of the social unit. How social knowledge is measured affects the interpretation of the results.

7. Consensus to sign was previously referred to as the level of ambiguity (Carley, 1984).


9. Both the observer and the actors taking part in an exchange can use social knowledge to make explicit information that is not explicitly communicated. This explication act is an act of information processing. It is distinct from, and follows after, the communicative act. It may be impossible for the communicating actor, as Garfinkel (1968, 1981) suggests, to make all information explicit. This however, is not the point of this article. The concern here is with the ability of the individual being communicated to and the observer to make explicit the information implicit in the communication. Even simple cognitive models led to the conclusion that under time and effort stressed conditions the communicator will not make all information explicit. The question then is how does the observer or communicate make the implicit information explicit. It is argued that the observer or communicatee uses their knowledge of social knowledge to do this explication. The observer can make everything explicit; however, in a limited time the observer will be able only to make everything explicit for a limited domain. Thus efforts by the observer to make implicit knowledge explicit are not doomed to failure. On the contrary, SKI is proof that such efforts can, at least within limited domains, succeed.

10. Such background information has been referred to in a variety of ways; for example, social knowledge (Carley, 1986a, 1986b), that information that everyone knows, shared culture (Whorf, 1956), tacit knowledge (Polanyi, 1962), background knowledge (Sowa, 1984), and the universal encyclopedia (Bar-Hillel, 1960). In this article the term social knowledge will be used interchangeably with the term social background information.
11. A knowledge base contains coded verbal data as pieces of information or knowledge (Carley, 1986a, 1986b). A knowledge base is social if it contains information that is shared by members of the society.

12. In fact, social knowledge may lead to a mistaken interpretation (Carley, 1986b); that is, to adding information that is not in the speaker’s knowledge base.

13. Most of the students were interviewed during TIME 1 and TIME 3. Two students were never interviewed at all. Another student was interviewed only during TIME 1, and another one only during TIME 2, and one only during TIME 3. Some of the students were interviewed several times during the middle phase of the decision process (TIME 2), depending upon their willingness to be interviewed and their relevance to the process and the hall. For example, MEADE, who was the chairman of tutor selection, was interviewed a number of times throughout the process. These extra interviews were predominantly used to gather social background information and to provide information on the formal process (what was happening when).

14. This history is currently available in the MIT archives and in Carley (1984: Appendix 3). This time span was chosen for two reasons. First, by covering all tutors, it was possible to get the widest distribution about what it means to be a tutor on Third East. Second, major renovations effectively transformed the social environment by altering the physical structure. The living group went from being 3 semi-isolated entries filled with suites to a single hallway filled with single (1-person) rooms. A consequence was that the “hall” as a social unit did not exist prior to these renovations.

15. Backward chaining is a reasoning mechanism used in expert systems for recursively locating the antecedents of the goal. The user enters the goal (in this case the erroneous fact) and using a set of inference rules (the facts in 3EKB and the nature of the types of facts) locates the set of possible antecedents that could have produced this goal (in this case the fact in 3EKB and the modified base that made it possible to add the erroneous fact). Each of these antecedents is then treated as a goal and the process repeats until the stopping conditions are met.

16. The comparison of knowledge bases depends on the underlying assumptions about the nature of information. For a more detailed discussion of how to compare and combine maps refer to Carley (1984: chap. 10).

17. For this analysis, all facts were treated as having the same absolute strength. For a discussion of one way of dealing with strength refer to Carley (1984: chap. 10).

18. For the remaining analysis Q will not be used as its overall value is effectively swamped by the fact that the number of concepts/facts that are potentially available is immensely larger than the number of concepts/facts that are in fact used. For concepts there are 217 potential concepts and in the maps coded during this phase there were between only 11 and 47 concepts. For facts there are 37,060 potentially available facts, and yet the typical knowledge base vis-à-vis the concept tutor on Third East generally contains less than 300 facts. The number of facts in a typical knowledge base is small even when compared with the facts that are socially available; that is, all facts known by at least one person at some point during the selection process—1,462.

19. The proposed methodology is for coding a protocol, not evaluating or predicting the individual’s decision or logic given this protocol coding. When evaluation is done and additive approach somewhat similar to that employed by Gollub and Heise can be used (Carley, 1984, 1987b).

20. A semantic segment refers to a clause, even if incomplete, that conveys a single piece of information. Typically, segments are clauses within complex sentences or complete short sentences.
REFERENCES


Kathleen Carley is currently an Assistant Professor of Sociology in the Department of Social and Decision Sciences at Carnegie-Mellon University. Research interests include the relationship of social structure to cognitive structure, distributed decision making, sociology of science, and artificial intelligence. Recent articles include "An Approach for Relating Social Structure to Cognitive Structure" in the Journal of Mathematical Sociology, "Knowledge Acquisition as a Social Phenomena" in Instructional Science, and "Efficiency in a Garbage Can: Implication for Crisis Management" in March and Weissinger-Baylon (eds.) book Ambiguity and Command.