Artificial Intelligence and the SAS® System: Why You Have To Teach the SAS® System about Sex!
David B. Malkovsky, SAS Institute Inc., Cary, NC

ABSTRACT
This paper presents a very brief survey of the areas that comprise artificial intelligence (AI) research (knowledge representation, natural language understanding, expert systems, and so on).

The majority of the paper describes AI research done at SAS Institute and a new product, SAS/ENGLISH® software, which is a result of that research effort. SAS/ENGLISH software provides users with a means of asking for information about their data by typing English questions using a user-friendly interface that avoids a computer-specific language or syntax. The importance of knowledge acquisition to make this interface work is discussed. The subtitle of the paper is used to draw attention to the need for teaching AI programs the words and concepts that humans take for granted when they query each other in normal conversation.

INTRODUCTION
There is disagreement in the field of artificial intelligence (AI) as to what constitutes the discipline. It is not unusual to find people who believe they are working in the field of AI, but are not considered to be doing so by any of their colleagues. Conversely, there are other individuals working in areas that are traditionally considered to be part of AI who refuse to apply that label to their work. In reading announcements of new products or advertisements in leading computer periodicals, you are led to believe that almost everyone in the computer field today is working on, or in, the AI field.

Answers to the question of what constitutes AI often center on defining the borders or limits of the field. One possible (and overly simplistic) test for receiving the AI classification is that any program written in LISP or Prolog belongs to the field of AI. Although in the first half of the history of AI this was very true, it is no longer a necessary or sufficient test.

Other litmus tests of AI-ness identify systems that use knowledge, employ search techniques, have an inference engine, or modify their behavior over time.

One humorous definition of AI states that "AI is any program or system that never gets to market." The basis of this view is that AI is magic and complex. If a marketable product is produced, the problem it solves was never in the field of AI. Rather, the task solved by such a system was only a complex software engineering task that was earlier misclassified as an AI problem.

Obviously there is no one correct answer to the question "What is artificial intelligence?" One acceptable answer is that AI is "the part of computer science concerned with designing intelligent computer systems, that is, systems that exhibit the characteristics we associate with intelligence in human behavior—understanding language, learning, reasoning, solving problems, and so on" (Barr, Cohen, and Feigenbaum 1989, p. 3).

MAJOR AI APPLICATIONS
The kinds of intelligent human behavior that AI investigators have tried to simulate include deduction, vision, learning, planning, problem solving, and natural language understanding. In addition, there has been research in basic AI techniques, such as ways to represent knowledge, development of special programming languages (for example, LISP, logic programming languages like Prolog, and object-oriented programming languages), tools (expert systems shells, for example), and architectures for AI applications.

To date, the following kinds of AI applications have been developed:

- Natural language interfaces to databases—interfaces that do not require users to learn special command languages
- Expert systems—systems that use the knowledge of specialists to perform tasks such as medical diagnosis, chemical analysis, geological exploration, and computer system configuration
- Automatic programming systems—systems to create programs based on descriptions of what the programs should accomplish
- Vision and robotic systems with sensory apparatus that respond to changes in their environment
- Neural network approaches to pattern recognition and knowledge acquisition
- Fuzzy logic approaches for performing inexact reasoning.

Such systems were developed for medicine, business, engineering, science, and manufacturing applications.

The remainder of this paper will examine one specific AI application from SAS Institute. SAS/ENGLISH software, which draws on a long tradition of language processing and AI research.

SAS/ENGLISH SOFTWARE
The function of a natural language interface such as SAS/ENGLISH software is to provide easy access to data without the user’s having to learn the syntax of a retrieval language or know the structure and field names associated with individual data values.

Using the full-screen interactive facility illustrated in Display 1, you type your question and SAS/ENGLISH software provides your answer. Once information is accessed, a user can use the full power of the SAS® System to manage, analyze, and present the data without ever leaving the menu-driven environment.

SAS/ENGLISH software works by translating a question into Structured Query Language (SQL) commands and presents the retrieved data or answers in the SAS/ASSIST® software environment.

SAS/ENGLISH software is a quick and easy way to get information and generate reports. Users can find, order, and summarize data by asking simple who, what, when, where, how many, and yes/no questions.
To understand a question or query, SAS/ENGLISH software uses a Knowledge Base (KB). The KB describes the data in terms of real-world concepts, relationships, and vocabulary. This type of information is called domain-specific information. It includes the context, concepts, and content of the data that do not exist in the built-in Knowledge Base of SAS/ENGLISH software. The activity of acquiring domain-specific information is known as knowledge base definition. The person who performs this task is the KB Administrator (KBA). The successful completion of this task requires an understanding both of the purpose of the application and of how a typical user interacts with that application.

The effort required to provide a level of detail and coverage to successfully implement a complete SAS/ENGLISH KB is not trivial. There are three major tasks of KB administration:

- creating the physical data model
- creating the conceptual data model
- defining the vocabulary.

Creating the Physical Data Model

The first task of KB administration is to specify the name and location of the data files associated with the application. The application-specific data are stored in one or more data sets that can be accessed by the SAS System. The physical location and storage method are unimportant to the SAS/ENGLISH product, as long as the data are accessible by a native SAS engine or through a SAS/ACCESS software product.

Creating the Conceptual Data Model

Having identified the particular data components of the application, the second task is to specify what concept or object each data element represents. For example, if a particular employee data set is a component of a human resource application, the employee data set can be modeled by an employee object. If the user enterprise is related to the sale of a product, a particular group of data may represent the concept of a sales transaction.

Part of identifying the objects or the concepts of an enterprise is specifying the relationships between the various objects. Continuing with the human resource example, a second data source may contain the job descriptions for the various positions within the company. The data file might not only include the job title, but the salary range, job responsibilities, payroll status, and other job-related accounting information. The relationship between employee object and the job information must be defined to the SAS/ENGLISH KB in order for questions about an employee's job to be answered.

Defining the Vocabulary

The last function of KB administration is to define the vocabulary and specify idioms that describe the data. Each application has a specific vocabulary that is customarily used with the data accessed by the application. In addition, the same data in different industries or applications are often known by different names. For example, people are commonly known as students in a school environment, as employees in a corporate setting, and as subjects in medical experiments.

Objects, Attributes, and Relationships

SAS/ENGLISH Knowledge Base administration makes some very important and fundamental assumptions about the data that it can model. The data model is oriented toward recognizing objects (also called entities), attributes of objects, and relationships between objects.

An object corresponds to a concept or thing in the real world and is represented by data in a data set. It can be a person, an event, a physical object, or an organizational entity, such as a department in a company. Objects or entities have attributes or characteristics. For example, people have names, heights, and weights; and events have times and places.

SAS/ENGLISH Knowledge Base Definition

The SAS/ENGLISH Knowledge Base Definition facility is a full-screen SAS/ACCESS software application that is used to interactively define a Knowledge Base describing the data for a specific application. Before defining a KB, the data must be identified and made accessible to SAS/ENGLISH software. This step may require defining SAS views or using a SAS/ACCESS product if the data are not in native SAS data sets.

Defining a Knowledge Base is usually an iterative process. A KBA begins with an initial model of the data, builds a Knowledge Base, and then tests some sample queries. If the queries are not understood, additional vocabulary may be needed or objects and their attributes may need to be defined differently. In some cases, it may be necessary to structure the data differently and redefine the Knowledge Base.

The first attempt at defining a KB is usually not the last because it is nearly impossible to think of everything during the first session of KB definition. Modifications and improvements of the KB are always possible over the life of the application. Most decisions made during the definition process are easily altered or reversed.

Sample SAS/ENGLISH Knowledge Base Definition

This section of the paper will discuss various components of the Knowledge Base definition process and present some of the interactive screens associated with each component.

Once a name and location for the Knowledge Base is chosen, the Select Data Set screen, illustrated in Display 2, is the central point for building the KB.
Knowledge Base: SAS/ENGLISH
EX Description: SAS/ENGLISH
< Choose a New Data Set > < Set Other KB Options >
Select a data set that has already been defined.
Data set name: Reference word

Display 2 SAS/ENGLISH Select Data Set Screen

This screen has options for adding new data sets to the Knowledge Base (Choose a New Data Set button) and performing specialized and advanced KB operations (Set Other KB Options button). The Query option allows you to test queries to see the effects of adding or deleting information from the KB. The Browser option lets you examine the KB using the Knowledge Base Browser.

As new data sets are added to the Knowledge Base, they are listed as test data sets. This word is used by SAS/ENGLISH software instead of the data set name to specify the object. A complex KB for a business application may look like the one shown in Display 3 below.

Display 3 SAS/ENGLISH Select Data Set Screen

The Select Data Set screen is the central point for specifying the contents of the KB. The Define a Data Set screen is the central point for defining the contents of each data file to SAS/ENGLISH software. The screen example in Display 4 below shows that much of the information is similar to that produced by running the CONTENTS procedure on the data set. Information unique to SAS/ENGLISH software includes specifying vocabulary, defining special attributes, and setting object options.

Display 4 SAS/ENGLISH Define a Data Set Screen

The next level of specification is the definition of the type and use of each variable in the data set. Fortunately, there are only two groups into which attributes fall. They are based on the data type of the attribute, the data are either numeric or character. For the purpose of this discussion, we will look at one example of each type. SALARY and SEX are the attributes used here to represent numeric and character type attributes, respectively.

After selecting the SALARY attribute on the Define a Data Set screen, the next screen seen by the Knowledge Base Administrator is the Define Numeric Attribute screen, as shown in Display 5 below.

Display 5 SAS/ENGLISH Define Numeric Attribute Screen

In the middle of the screen, there are several general types of numeric attributes from which the KBA must choose the best description of the attribute. The selection of the description or type adds information to the Knowledge Base as to how or where in a query this data can be specified. The various numeric types are the following:

- A number that identifies the object.
  This type is chosen if the attribute uniquely identifies an object in the KB or identifies the observation in the data set. For those of you familiar with relational databases, SAS/ENGLISH software assumes that such an attribute forms a primary singleton key.

- A quantity
  This type is chosen if the attribute counts something.
Money is usually the purchase price of some object or service, or a form of compensation received by some object or person. If the KBA selects price as the specialized type of money attribute, the K3 is automatically modified to add the words PRICE, COST, and VALUE as synonyms for the attribute salary. The verb COST is also added so that queries of the form "Which sales cost more than $100?" are supported. Finally, the adjectives EXPENSIVE, CHEAP, and COSTLY are incorporated so the command "List the cheapest sale" is now possible.

If the form of compensation selection is made instead, as would be the case in this example, then SALARY, COMPENSATION, PAY, and REMUNERATION are added to the Knowledge Base as vocabulary associated with salary. Two verbs, EARN and MAKE, are also included so queries like "How much did Bob earn?" are recognized.

There is a third option, Some other kind of money attribute, for which SAS/ENGLISH software makes no extra assumptions about this attribute.

As shown above, default adjectives like CHEAP are provided for a price monetary attribute, but none are given for a compensation monetary attribute. A KBA can define meaningful adjectives for this attribute by selecting Specify adjectives describing EMPLOYEE by SALARY on the Define Numeric Attribute screen. Making such a selection results in the screen shown in Display 7:

Display 7 SAS/ENGLISH Define Adjectives Screen

This example Define Adjectives screen is already filled with information specified by the KBA. For example, the words EXPENSIVE, HOT, and FAT are equated with high salary values, and the words CHEAP and SLIM are associated with small data values. The KBA specified the root form of the adjectives because SAS/ENGLISH software automatically recognizes both the comparative and superlative forms when they are used in queries. Thus the KBA specified FAT rather than the comparative form, FATTER, or the superlative form, FATTEST.

The direction field (Plus, None, and Minus) is needed to indicate the direction associated with a comparative data value. In this example, the larger the data value, the higher (or fatter, or hotter) the salary. If the attribute were age and the data values were represented by a person's date of birth, then the oldest person would be the person with the smallest (or earliest) birth date (direction of minus).

This screen provides a method of specifying values for numeric attributes that can be associated with adjectives with subjective meanings. This is because words like rich and poor or young and old can have very different meanings, depending on the audience or situation. For example, RICH is associated with salaries over 70,000 dollars.

The second major classification of attributes is those attributes that are represented by character data values. The Define Character Attribute screen, illustrated in Display 8, is used to define such attributes to the Knowledge Base.
In the middle of the screen there are several general types of character attributes from which the KBA must choose the best description of the attribute. As was the case with the numeric attributes, the selection of the description or type adds information to the Knowledge Base. The various character types are the following:

- **a name**
  - This type is chosen if the values of the attribute are names of the data set object being defined. These may be names of persons or things; they may be complete names or parts of names.

- **an identifier**
  - This type is chosen if the attribute uniquely identifies the object or data set observation. SAS/ENGLISH software assumes that such an attribute forms a primary singleton key.

- **a value that classifies**
  - This type is chosen if the attribute classifies or categorizes the object in some manner. The data values may classify as boolean attributes, or into any number of categories. Typical classifiers might be job titles or gender of employees (as in this example), or a status label for a project (active, inactive, completed, and so on).

- **a location**
  - This type is chosen if the data values for the attribute represent locations or addresses. This attribute may be part of a typical address or some other variety of geographical location.

- **an identifier of another object**
  - This type is chosen if the values of the attribute identify some data set object other than the current object.

- **some other property**
  - This type is chosen if none of the other types fit the attribute. SAS/ENGLISH software will make no assumptions about this type of attribute and will not add any default information to the Knowledge Base for it. This type should rarely, if ever, be chosen.

Continuing with the screen example, the obvious choice for the best description of the attribute SEX is that the data values for this variable classify employees into one of two groups, based on gender. An important concept to understand is that the SAS/ENGLISH product does not (and cannot) make any assumptions about the data and how they are used until the KBA provides this type of information. The original creator of the employee data set did a good job in providing descriptive names for the various variables in the data set. The variable names impart meaning and implied use for the data values to a human but not to a computer program (at least not yet). Although SAS/ENGLISH software understands the concepts associated with numeric and character attributes outlined above, the system does not make any inference as to the use of any object or attribute without explicit instructions from the Knowledge Base Administrator.

For the current character attribute SEX, most people would draw the logical conclusion that the data values are likely to be F and M and represent FEMALE and MALE, respectively. Although this inference is indeed correct, in this example it need not be always true. SAS/ENGLISH software uses the Classifier Attribute screen, shown in Display 9, to define the relationships between concept or vocabulary and data value.

On this screen, the vocabulary that can be recognized in a query is found in the Word column and the data values that are equated with each word are found in the Value column. More than one word can be associated with a particular data value, as multiple data values can be associated with a particular word. This screen shows examples of the first type of association.

In the second type of association, one word refers to a list of possible values if a job attribute exists for an employee object. The classifier attribute JOB could have the data values MANAGER, CIVIL, and MECH. The words manager, civil engineer, and mechanical engineer could be associated with each value. Furthermore, you can define engineer as a synonym for someone who is either a civil or a mechanical engineer.

**CONCLUSION**

There are other aspects of Knowledge Base definition that are not covered here. These include how to specify which attributes are components of composite locations or names, how to define new attributes that are mathematically computed from the values of other attributes, how to collect attributes into a group so they can be accessed collectively as a single attribute, how to specify units of measure for numeric quantities, how to
specify more specialized object options, and how to define additional verbs for use in queries.

SAS/ENGLISH software, like any other AI application of this type, is only as powerful and as useful to users as the Knowledge Base that defines the application.

DISCLAIMER

SAS/ENGLISH software is an experimental product (the first distribution of the product requires Release 6.07 of the SAS System), and the final implementation appearance of the product may differ from what is presented in this paper.

LIST OF ARTIFICIAL INTELLIGENCE SOURCES


SAS, SAS/ACCESS, SAS/AF, SAS/ASSIST, and SAS/ENGLISH are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.