

MITA

An Information-Extraction Approach to the Analysis of Free-Form Text in Life Insurance Applications

Barry Glasgow, Alan Mandell, Dan Binney, Lila Ghemri, and David Fisher

■ MetLife processes over 260,000 life insurance applications a year. Underwriting of these applications is labor intensive. Automation is difficult because the applications include many free-form text fields. MetLife's intelligent text analyzer (MITA) uses the information-extraction technique of natural language processing to structure the extensive textual fields on a life insurance application. Knowledge engineering, with the help of underwriters as domain experts, was performed to elicit significant concepts for both medical and occupational textual fields. A corpus of 20,000 life insurance applications provided the syntactical and semantic patterns in which these underwriting concepts occur. These patterns, in conjunction with the concepts, formed the frameworks for information extraction. Extension of the information-extraction work developed by Wendy Lehnert was used to populate these frameworks with classes obtained from the systematized nomenclature of human and veterinary medicine and the *Dictionary of Occupational Titles* ontologies. These structured frameworks can then be analyzed by conventional knowledge-based systems. MITA is currently processing 20,000 life insurance applications a month. Eighty-nine percent of the textual fields processed by MITA exceed the established confidence-level threshold and are potentially available for further analysis by domain-specific analyzers.

MetLife's insurance application is designed to elicit the maximum amount of information relating to the client so that a fair contract can be reached between the client and MetLife. The application contains questions that can be answered by structured data fields (yes-no or pick lists) as well as questions that require free-form textual answers.

Currently, MetLife's Individual Business Personal Insurance unit employs over 120 underwriters and processes in excess of 260,000 life

insurance applications a year. MetLife's goal is to become more efficient and effective by allowing the underwriters to concentrate on the unusual and difficult aspects of a case and automate the more mundane and mechanical aspects. A 10-percent improvement in productivity for MetLife, while it still maintains the already existing high quality of the underwriting processing, or an increase in the consistency of the process will have sizable effects.

The use of expert systems to improve the insurance underwriting process has been the "holy grail" of the insurance industry, and many insurance companies have developed expert systems for this purpose with moderate success. A daunting problem has been the presence of textual fields.

MetLife's intelligent text analyzer (MITA) is an attempt to solve this problem using information extraction. Using this technique on the textual portion of the application allows the automation of underwriting review to a greater extent than previously possible.

Extracting information from free-form textual fields is a recurring problem in many information systems. Senator et al. (1995) discuss the need to analyze textual fields containing occupations and business types to detect financial crimes.

MITA Overview

Previous attempts have been made to understand the text fields on MetLife's insurance applications by means of keywords or simple parsing. These attempts have been inadequate. The application of full semantic natural language processing (NLP) was deemed too complex and unnecessary because the text often contains details that are not directly relevant to the decision making. A happy compro-

The use of expert systems to improve the insurance underwriting process has been the "holy grail" of the insurance industry.

mise—information extraction—was recently developed.

The MITA free-form text analyzers take in unstructured text, identify any concepts that might have underwriting significance, and return a categorization of the concepts for interpretation and analysis for risk assessment by subsequent domain-specific analyzers. By localizing the natural language processing of the input text in MITA, other domain-specific analyzers can focus on codifying underwriting domain knowledge.

The fields analyzed by MITA include a *Physician Reason field* that describes the reason a proposed insured last visited a personal physician, a *Family History field* that describes a proposed insured's family medical history, a *Major Treatments and Exams field* that describes any major medical event within the last five years, a *Not Revealed field* that includes any important medical information not provided in other questions, and an *Occupation Title and Duty field* that describes the proposed insured's employment.

AI in MITA

The information-extraction approach of NLP was chosen for use in MITA. The system was engineered based on a corpus of actual application texts. This approach was intended to provide an information-extraction system optimized for MetLife's insurance application text-processing needs.

Information Extraction

Analysis of free-form text has been pursued mainly from three viewpoints: (1) a keyword approach, (2) an in-depth natural language-analysis approach, and (3) an information-extraction approach.

The *keyword approach*, whereby the input text is scanned for words that are deemed highly relevant to the application at hand, has the advantage of being relatively easy to implement. However, it is of limited usefulness for accurate data extraction, which is a requirement of MITA.

An *in-depth natural language analysis approach*, in which the input text is fully and completely analyzed, is usually highly complex, costly, and still relatively brittle. Furthermore, it does not answer the need of an application in which the interest is focused on some parts of the text but not necessarily all (Sager 1980).

In an *information-extraction approach*, the input text is skimmed for specific information relevant to the particular application. Informa-

tion extraction (Cowie and Lehnert 1996; Lehnert and Sundheim 1991) is an ongoing area of natural language research, where the focus is on real working systems with periodic performance evaluations. Information extraction makes use of NLP tools (parsing, tagging, dictionaries) but also uses knowledge engineering or machine learning to identify the concepts contained in the texts and form a framework of these concepts.

To illustrate why information extraction provides a good match to the problem, consider the text field that describes why the client has last seen a physician. Frequently, this field describes a visit for a checkup with additional modifiers for a specific disease, a chronic condition, or the fulfillment of an occupation or athletic need. A simple keyword search-finding checkup would not provide sufficient information. Alternately, full NLP with deep semantics is not required because most Physician Reason texts could be analyzed in terms of four concepts: (1) *reason*, which usually describes the type of visit such as "regular visit," "school checkup," or "postpartum checkup"; (2) *procedure-treatment*, which describes a procedure or a treatment such as "prescribed antibiotics," "ear wax removed," or "HIV test"; (3) *result*, which describes the outcome of the exam or the procedure such as "nothing found," "all ok," or "no treatment"; and (4) *condition*, which describes a medical condition that the applicant has such as "high blood pressure," "ear infection," or "broken leg."

In building MITA, knowledge engineering was used to build a representation of the domain to be analyzed in terms of important concepts. NLP techniques are used to process text and extract these concepts when they are present in the text.

Advances in information extraction from text have been well documented in a series of rigorous performance evaluations sponsored by the Defense Advanced Research Projects Agency called the message-understanding conferences (MUC) (MUC3 1991; MUC4 1992; MUC5 1993; MUC6 1995).

A variety of techniques have been tested at MUCs, ranging from full-sentence syntactic parsers to machine-learning algorithms. The majority of the systems tested, such as CIRCUS (Lehnert 1990), BADGER (Fisher et. al. 1995; Soderland et. al. 1995), and the GE NLTOOLSET (Krupka et. al. 1992), used robust partial parsing followed by pattern matching to achieve information extraction. For both MUC 6 information-extraction tasks, the five top-performing systems were all pattern-matching systems (Aberdeen et. al. 1995; Appelt et. al.

Percent of Fields with Confidence Factor Greater than Threshold				
	Average Monthly Records	6/13 to 7/15/97	7/16 to 8/15/97	8/16 to 9/15/97
Medical 11A	3,692	0.70	0.70	0.70
Medical 11E	3,914	0.83	0.82	0.83
Medical Family History	4,671	0.85	0.83	0.85
Physician Reason	16,490	0.89	0.88	0.88
Occupation	18,603	0.97	0.97	0.97

Figure 1. MITA Production Results.

1995; Childs et. al. 1995; Grishman 1995; Krupka 1995; Lee 1995; Weischedel 1995). The pattern-matching approach clearly proved to be the state of the art with respect to information extraction. MITA was designed to use this approach, incorporating features from all these systems.

Description of the Corpus

The corpus used for this application contains some 20,000 applications, with each having at least 1 free-form text field. A basic analysis of the sample was performed to determine the range of syntactic forms commonly used. It was found that most sentences were cryptic, and major syntactic constituents such as subjects and verbs were freely omitted. Punctuation was terse and often incorrect. Instead, the text was basically a series of noun phrases clumped together, as in "C-section childbirth 1979 no complications."

Project History

The MITA system was built in three phases: (1) analysis, (2) design and prototyping, and (3) implementation.

Initial Analysis

During the spring of 1996, a team composed of MetLife underwriters and business analysts and Brightware consultants conducted an analysis of the free-form text fields in MetLife's new-business life insurance applications. The goal was to determine the best way to handle these fields in automating the underwriting initial review process. Factors considered in this analysis of each field included (1) the nature and strategy of the question being answered, (2) the potential underwriting significance and use of the information, and (3) the syntactic and semantic structure of the text.

The resulting recommendations for each field fell into two categories: (1) conversion of the free-form questions into structured questions (such as check boxes and drop-down lists) and (2) the development of intelligent text analyzers to process the free-form text and convert it into a form that can be used by downstream analyzers.

The results also included a list of the categories of potentially significant underwriting information provided by each field. These categories were called *concepts*. For example, for the occupation analyzer, three concepts were found to be of significance: (1) the job title, as in "waiter"; (2) the description of the duties, as in "serves food and beverages"; and (3) the work environment, as in "restaurant."

Design and Prototype

During the second phase of the project, a general system design was developed along with a prototype system.

The prototype was based on the Physician Reason field. Physician Reason was chosen because it is a field that is populated on 75 percent of insurance applications. It also presented a challenge because the medical terminology used was fairly complex.

The prototype was intended to show proof of concept and was to be expanded later into a fully implemented system for additional fields.

The design and prototype were developed by three Brightware consultants over a one-month period. Small lexical and medical dictionaries were built as part of the prototype.

Implementation

Based on the success of the prototype, an implementation phase was begun. Text analyzers were to be built for four medical fields and one occupational field.

Implementation began in late August 1996,

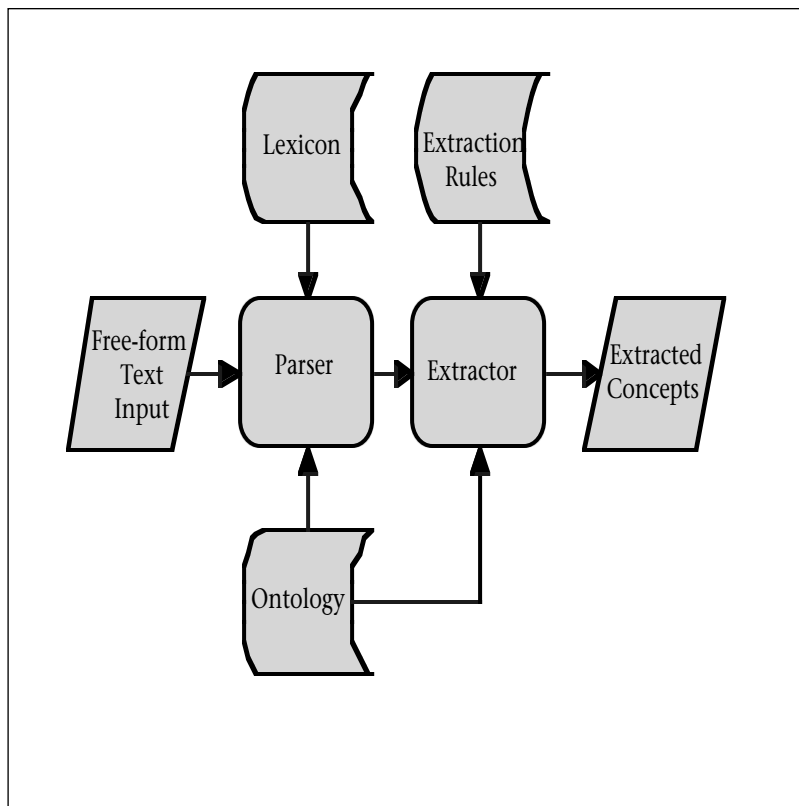


Figure 2. Overview of MITA's Process.

and the completed system was released to system test in January 1997. The core project team consisted of seven Brightware consultants working closely with four underwriters. Each underwriter represented expertise from a different area of life insurance underwriting. A number of MetLife system engineers participated in the project as well.

Production Deployment

MITA was placed into a production environment in June 1997. For the 3-month period from mid-July to mid-September, MITA processed an average of 20,400 insurance applications a month; each application contained an average of 2.3 textual fields (figure 1). Eighty-nine percent of the textual fields analyzed by MITA were successfully analyzed. Successful analysis is defined as having a confidence factor greater than the threshold value for that field. The threshold values were 50 for family history and 70 for the other 4 fields. Confidence factors are a mechanism developed for MITA that allow it to assess the quality of its own work. Details of this mechanism are described later in this article.

MITA Design Overview

Figure 2 shows the major components of MITA. A free-form text field is first processed by a *parser* that tags each word with parts of speech and semantic categories and then combines these words into larger structures called *phrases* (such as noun phrase and verb phrase) and *constituents* (such as subject and verb).

The parser uses relational tables to help accomplish its tasks. *Lexicon tables* contain the parts of speech. *Ontology tables* contain the semantic classification of words and phrases as well as the hierarchy of classes of which they are members.

Next, the *extractor* compares the parsed and tagged text to known extraction patterns. When a pattern applies, the extractor extracts important words and their classifications and categorizes them into what are called the text's concepts.

System Input and Output

The primary input to MITA is free-form text. An example from Physician Reason is the string "after childbirth applicant suffers from iron deficiency medication."

The exact output of MITA is specific to the particular field, but all analyzer output follow the same general format: The *concept* describes a particular type of information that can be found in this field. The *value* is the actual word(s) in the text that is associated with this particular instance of the concept. The *class* is a general category of values into which this particular value falls (classes are especially needed to allow automated processing of the extracted information). The output for this example is shown in figure 3.

Major MITA Modules

MITA accomplishes its work through three major modules, which, in turn, are supported by several other minor components. The major modules include (1) a parser, (2) a set of extraction rules, and (3) a set of dictionaries.

The Parsing Module The MITA parsing module is language specific (English). The module is domain and task independent and requires no code adjustments to move from one application domain to another. Parsing is performed in six stages:

The first stage of parsing is *part-of-speech tagging*. The input is read, and each word is tagged with its part(s) of speech. This first stage also interacts with specialized recognizers that were designed to handle uncommon constructs such as dates, measures, contractions, and mis-

spelled words, as described in Recognizers later in this section.

The second stage of parsing is *part-of-speech disambiguation*. In this phase, rules for part-of-speech disambiguation apply. These rules rely on the word's left and right contexts to assign it a correct part of speech.

The third stage of parsing is *bracketing*. At this stage, sentence constituents are grouped into simple noun phrases and verb phrases.

The fourth stage of parsing is *grouping*. In this stage, the simple noun and verb phrases are further combined by including conjoined elements, appositives, or adverbial phrases to form more complex noun and verb phrases.

The fifth stage is *buffering*. Next, the noun-phrase constituents are assigned the syntactic roles of object or subject, and modals are included into the verb-phrase constituents.

The sixth stage is *segmentation*. In this last stage, the parsed input is fragmented into logical segments, in which each segment represents a clause or a sentence. This parser is particularly robust and proved well suited to the task at hand. Because it does not include features such as tense, number, or case checking, it was able to handle incorrect or ungrammatical sentences. Its performance also degrades "gracefully" when it has to analyze sentences with unknown words in that it generates partial parses of what it successfully processed.

The Extraction Module Extraction rules are then applied to determine the presence of a given concept in the input text being processed and fill out slots for the value and the class of concept found. Each extraction rule is an if-then rule, which uses the pattern-matching mechanism of ART*ENTERPRISE. The left-hand side of the rule describes a set of constraints on words and word senses found in the syntactic buffers of the parser output. If these constraints are met, asserting the facts from the right-hand side (the then part) fills specified roles. A single sentence can generate an arbitrary number of concepts depending on the complexity of the sentence being processed.

Dictionaries MITA uses three dictionaries: (1) a lexical dictionary containing words in their various inflected forms along with their parts of speech; (2) a dictionary of medical terminology; and (3) a dictionary of occupational titles, duties, and environments.

The lexical dictionary has connections to the classifications in the two other ontology dictionaries so that words can be associated with semantic classes. MITA's dictionary tables contain about 75 megabytes of data. There are approximately 268,000 lexical entries and

Concept	Value	Class
Condition	"childbirth"	PARTURITION-NOS ^a
Condition	"iron deficiency"	IRON-DEFICIENCY-NOS ^a
Procedure-Treatment	"medication"	DRUG-NOS ^a

a. NOS = Not otherwise specified.

Figure 3. A Sample Output Frame.

approximately 135,000 classes, including all domains. There are approximately 4,000 composite classes (these are described later in this article).

Selecting Dictionaries

An important part of the implementation was the selection and acquisition of lexical, medical, and occupational dictionaries. The simple dictionaries that had been hand developed for the prototype were insufficient to support scaling up to a full production system. They contained only about 100 lexical entries and a few dozen classes. In the medical domain, much larger dictionaries were needed. In addition, a comprehensive dictionary of occupations was needed.

Several dictionaries of each domain were considered. Some criteria for each dictionary were adequate coverage for the domain, structure that could be adapted and enhanced to fit MITA's needs, accuracy, lack of licensing or copyright issues that would require extensive contract negotiations, and reasonable cost. The dictionary sources that were finally chosen are outlined in the following subsections.

Lexical Dictionary Merriam-Webster's *Collegiate Dictionary*, tenth edition, has more than 160,000 entries, including abbreviations, biographical and geographical names, and some foreign words and phrases. Each word has its part(s) of speech defined as well as its irregular forms.

Medical Dictionary The systematized nomenclature of human and veterinary medicine (SNOMED) is a classification system created for the indexing of medical records. It is developed and distributed by the College of American Pathologists and contains more than 144,000 entries. SNOMED is organized like a book and is divided into modules, chapters, sections, and subsections of data. It includes terms for referencing devices, diagnoses, drugs, living organisms, modifiers, morphology,

occupation, procedures, signs and symptoms, social context, and topology. SNOMED also provides the ICD-9-CM (Clinical Modifications of the ninth revision of the International Classification of Diseases) codes that most health insurance companies use for billing purposes. SNOMED has an elaborate and detailed glossary of medical terms organized hierarchically; for example, “a fracture” has the code M-12000, where M is the letter code for morphology, M-1 the code for traumatic abnormalities, M-12 the code for fractures, and M-12000 for (FRACTURE, NOS).¹

SNOMED, as well as ICD-9, classification schemes have been used to automatically extract medical concepts from medical records written by clinicians (Oliver and Altman 1994; Campbell and Mussen 1992; Yang and Chute 1992). These approaches relied on statistical methods or semantic nets-grammars to perform the classifications.

Occupational Dictionary The *Dictionary of Occupational Titles* (DOT) was developed by the U.S. Department of Labor, which also maintains it. It contains more than 12,000 occupation titles and organizes the data in a classification system. A title’s classification is identified by a 9-digit code. The first set of three digits identifies a particular occupational group, the second set of three digits identifies the worker functions associated with a title, and the third set serves to differentiate a particular occupation from all others. The DOT also provides the location of an occupation, types of duty, products manufactured, processes used, and raw materials used for some titles.

Customization

Each of these dictionaries required extensive reformatting and enhancement before it could be loaded into relational databases and integrated with MITA.

Lexical Dictionary Because the only information that was required from Webster’s dictionary was a word’s part of speech, the main purpose of the customization was to extract that information from a word’s definition. It was also necessary to generate the inflected forms of the words for them to be recognized by the parser. In addition, a sizable portion of the dictionary entries had multiple parts of speech (as many as four). These entries had to be customized manually to weed out the uncommon parts of speech and only leave those that the parser disambiguation rules could handle.

Medical Dictionary Customization of SNOMED required selecting the modules that would be useful for MITA. It was decided that topography, morphology, function, living

organisms, chemical-drugs and biologicals, physical agents forces and activities, general linkage-modifiers, disease-diagnosis, and procedures modules were required for MITA.

The first step was to manually create a class for each module and then automatically generate a class for each entry in the module. Determiners, prepositions such as *of*, and punctuation were removed from the original strings to make up a single unit; for example, “carbuncle of foot” became “carbuncle-foot.” This representation proved extremely useful for the string composite classifier, described later.

The next step was to link these classes in a parent-child relationship. The hierarchical organization of SNOMED helped in this task because modules became parents of chapters, which, in turn, became parents of sections, and so on. Occasionally, the same medical term was used in two different parts of SNOMED; in these cases, the name of the parent class was appended to the child class to uniquely identify it.

The text that MITA has to process is written by lay people describing their health condition rather than doctors; so, there is little medical terminology used. Thus, classes are generally situated two to three levels down in the hierarchy and are often too general for an accurate classification. For example, for the typical input “had broken leg,” the classes obtained from processing are (FRACTURE-NOS) (LEG-NOS). However, there is no link that associates these two concepts with their composite, more accurate, class (FRACTURES-LOWER-LIMB).

This gap between the representation scheme of SNOMED and concepts in the text has led to the design of a special module that links composite classes to their multiple components. This representation problem is further discussed in Do Amaral Marcio and Satomura (1995) and Sager (1994).

Occupational Dictionary The hierarchy of the occupational ontology was created from only the first three digits. Classes were added manually for each of the DOT classifications defined by the first two digits of the nine-digit code. The first-digit classifications became the top-level classes, and the two-digit classifications became the second level.

Next, new classes were created automatically at the root level in the ontology for classification pertaining to the third digit of the nine-digit code. For example, the occupation *aerodynamicist* with code 002.061-010 became a root-level class with parents aeronautical engineering occupations (code 002); occupations in architecture, engineering, and survey-

ing (code 00); and professional, technical, and managerial occupations (code 0).

For the DOT, a fair amount of customization was required because many of the lower-level titles shared the same name. To differentiate between these, information from either the worker functions associated with the title, the specific industry associated with the title (these data were imbedded in the DOT file for each title), or the title's parent had to be appended to the class name.

DOT provided two to three pieces of information for a small set of the titles: the location of an occupation, the types of duty, the products manufactured, the processes used, and the raw materials used. These data were used in the composite classification system, described later, and a class was created for each title that had these data in the ontology. For example, a composite classification was defined for the aerodynamicist class that looked for the occurrence of the analysis duties class in combination with an occurrence of the aerospace industry class.

Supportive Modules

An additional set of modules provides critical support to the major modules.

Recognizers The *recognizers* find and normalize nonstandard text contained in the input to make it easier to handle during formal parsing. Types of nonstandard text normalized include abbreviations; contractions; possessives; numbers; frequencies; dates; time periods; units of measure; symbols that mean the word *and*, such as & and +; telephone numbers; punctuation; and comparison symbols. In addition, special recognizers handle line concatenations where the break in text might have occurred within a word.

Spell-Correction Module The *spell-correction module* uses a dictionary of common misspellings to correct misspelled words in the input. This dictionary was custom built for MITA and is composed of the words that were commonly misspelled in the corpus.

This module is only a partial solution to the problem of misspellings. More sophisticated correction methods could have been applied but were not because the decrease in confidence resulting from these methods would cause most texts they handled to be processed manually (see the discussion about confidence factors later). Instead, more emphasis is being placed on the correction of spelling during entry of the text into the insurance application itself.

Multiword Lookup Module The *multiword lookup module* consists of a set of rules that look

at typical patterns of text where multiword phrases occur. Common phrases are stored in the dictionary. When a hit occurs, the parse of the individual words is replaced with a structure for the entire phrase. An example is the phrase "high blood pressure," a common medical phrase that carries a special meaning as opposed to the individual words *high*, *blood*, and *pressure*.

Composite Classifier Although multiword lookup handles phrases at the word level, the composite classifier handles them at the class level. The *composite classifier* allows multiclass relationships to be defined at the most abstract level possible, allowing the widest number of variations to be captured in the ontology. A class is designated as a composite class in the ontology. It is then associated with a list of *participant* classes. All participant classes must be instantiated in a phrase (at some level of abstraction) for the composite class to be assigned to a phrase.

An example of a composite class in MITA is *tonsillectomy*. This class is composed of (participant classes) *tonsils* and *excision*. When a concept containing the text "removal of tonsils" is encountered, the composite classifier recognizes that *excision* is a parent class of *removal* and thus assigns *tonsillectomy* as the classification for the concept. Because any phrase can end up instantiating multiple-candidate composite classes, the composite classifier contains heuristics to find the best-fitting composite class.

String-based composite classifier: Because many classes that are defined in the SNOMED ontology have names that are literally string concatenations of the names of other classes that compose these classes, MITA is able to cut short the complexity of the composite classifier by simply trying class-name string-concatenation combinations. This approach is efficient in execution and does not require the creation of multiword lexical dictionary entries or composite classes. It does not, however, handle any kind of abstraction of classes (although string matching of abstracted classes is a potential future enhancement).

Lexical coder: Each class within MITA's ontology, whether derived from SNOMED or DOT or created for MetLife's specific needs, has a code. These codes are used by *downstream analyzers* (analyzers that use MITA's output) to automate parts of the underwriting process. Because some of these analyzers do not have access to MITA's ontology and do not have abstraction capabilities, the *lexical coder module* was created. This module lists the hierarchical path of standard codes starting from the code of the extracted concept itself.

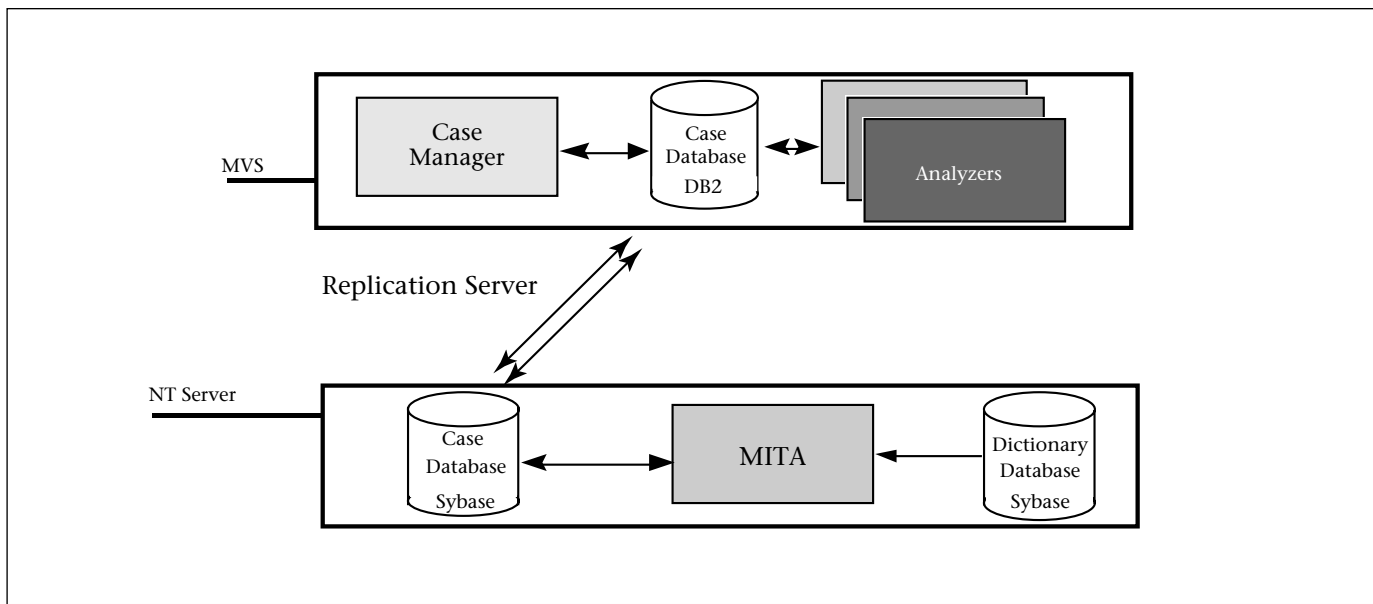


Figure 4. Deployment Environment.

Confidence Factor

MITA's NLP approach has been called *corpus-based information extraction*. The system is designed to perform well on texts that are similar to those occurring in the corpus (a sample of 20,000 cases). Unusual syntactic or semantic structures are less likely to be handled correctly. MITA achieves a high level of performance because it does its best work on the most common texts. However, in the life insurance business, a misinterpretation of text, although rare, can be costly to the insurer.

MITA achieves its reliability not because it analyzes text perfectly but because it can determine when it has done well and when it has not done well. This prediction of extraction quality is accomplished through confidence factors.

A *confidence factor* is a number between 0 and 100 that corresponds to the quality of MITA's output for an individual text. This number is derived primarily from the types of word that were not extracted from a text. Nonextracted words are given penalties based on part of speech and semantic class. Penalties also occur when a spelling correction was performed or when a composite classification was attempted but was unsuccessful.

The higher the confidence factor is, the more trustworthy the output. Downstream analyzers decide whether a particular output is reliable to process automatically based on the confidence factor. Each field has its own unique confidence-factor threshold and its own set of penalty values.

System Input

The input to MITA is provided through two tables: (1) the *work queue table*, which contains a queue of requests waiting to be processed by MITA, and (2) the *raw-text table*, which contains the actual text and field information.

System Output

The output from MITA is written to five separate tables, all linked by a common key: (1) the *results table*, which contains the extracted concepts; (2) the *confidence-factor table*, which contains the confidence factor for the concept extraction at a field level; (3) the *lex code table*, which lists all the SNOMED, DOT, or MetLife codes associated with a particular concept; and (4,5) two *audit tables*, which contain confidence-factor criteria and other useful statistics for continued maintenance of MITA.

MITA Platform

MITA runs on a MICROSOFT NT server platform (figure 4). It is coded in Brightware's ART*ENTERPRISE. The dictionaries, as well as the input and output tables, are all implemented in SYBASE tables.

MITA communicates with a case manager, which resides on a mainframe using the MVS operating system. The *case manager* coordinates all analyzers and uses DB2 tables. MITA's SYBASE input and output tables are replicated in DB2 using SYBASE REPSERVER. This approach makes all interplatform communication transparent to all processes.

ART*ENTERPRISE

MITA is implemented using Brightware's ART*ENTERPRISE tool set. ART*ENTERPRISE is a set of programming paradigms and tools that are focused on the development of efficient, flexible, and commercially deployable knowledge-based systems.

ART*ENTERPRISE rules contain a powerful pattern-matching language and an efficient inference engine. Rules are heavily used in the recognizers, the parser, the extractor, and every other phase of MITA processing. A rule orientation allows MITA to be heavily data driven, efficient, flexible, and easy to maintain.

Rule conditions often cause procedural code to be executed. Procedures are implemented as ART*ENTERPRISE functions and methods. They are coded in ART*ENTERPRISE'S ARTSCRIPT language.

Data structures within MITA are represented as *facts* (which are essentially lists of values and symbols) and as ART*ENTERPRISE *objects*. The object system allows full object-oriented programming within MITA and is fully integrated with rule and procedural components.

An especially useful feature of ART*ENTERPRISE is the *data integrator*. This component allows easy generation of the interface between MITA and relational databases. The data structures and the access specifics are represented as objects, and the database operations are invoked as methods on these objects.

Because MITA's extraction-rule patterns are expressed at abstract levels, and its input are specific, MITA spends a lot of time navigating class relationships. To make this class navigation efficient, MITA takes advantage of another ART*ENTERPRISE data structure—hash tables. MITA loads all class relationships into memory-resident *hash tables*. The hash tables consume a relatively small amount of memory, but the result is that queries and patterns accessing class relationships execute instantaneously.

Application Use and Payoff

By applying MITA, in conjunction with the downstream analyzers, to this initial review process, MetLife hopes to save as much as one-third the total underwriting time. In addition, it is expected that MetLife will achieve greater underwriting consistency. MITA currently extracts concepts that can be evaluated by the downstream analyzers 48 percent of the time.

Validation

There were two phases of MITA validation: The first was ongoing during the development phase, and its purpose was to focus develop-

ment on the areas that would maximize MITA's performance. The second was begun in the system test phase, and its purpose is to measure MITA output in the context in which they will be used to make underwriting decisions.

An important metric for MITA is its *recall*, which is the percentage of concepts that are successfully extracted from the text. Another metric is *precision*, which measures the accuracy of the concepts that are extracted.

MITA was built to extract all concepts in the text field but not to evaluate the underwriting significance of these concepts. Therefore, MITA must have a high recall to guarantee that the significant concepts are extracted.

Current MITA testing is focused on both attaining a high recall and identifying those concepts that are significant.

The corpus was segmented into a *training set* of cases, cases that knowledge engineers could analyze and use to develop extraction rules and dictionary entries, and a *performance set*, cases that were set aside to test system performance. The performance set was later used to compare MITA's results with annotated results supplied by the underwriters.

Confidence factors must have a high correlation with recall to be a reliable predictor of extraction quality. For this reason, validation tests also took the quality of confidence factors into account. The better the confidence factor (the more predictive it is of recall), the higher the confidence threshold can be set, allowing more cases with high-quality extractions to be considered reliable enough to allow automated underwriting.

At the writing of this article, performance statistics across the whole system were not known. However, preliminary statistics indicate overall recall around 85 percent and confidence factors with the ability to reliably predict at least 70 percent of the correct cases.

System-Test Phase Validation

Other validation approaches being used include *blind testing* to determine whether the output of MITA is sufficient to make underwriting decisions equivalent to those produced by an underwriter with access to full text. This test was conducted in conjunction with MetLife's actuarial staff. The goal was to determine the reliability and appropriate thresholds of confidence factors.

The blind-test set of 200 examples was selected from 3800 sample cases that the MITA development staff and the underwriters had not seen. For each of these cases, underwriters were presented first with only the MITA-extracted concepts and their classifications. They

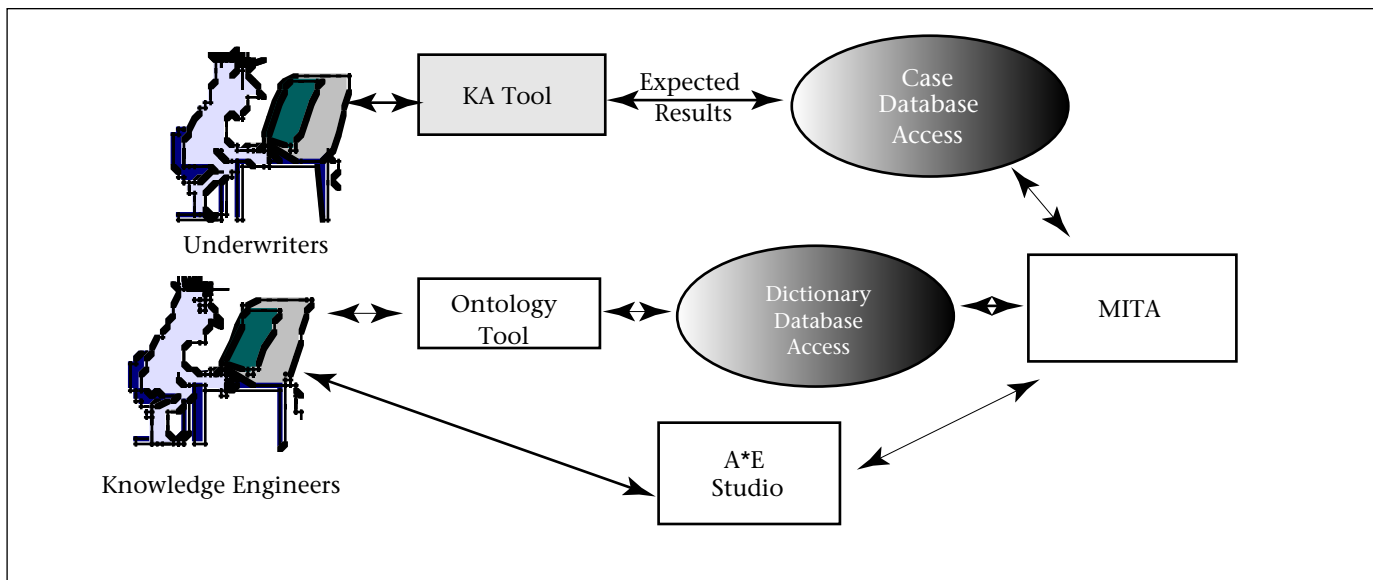


Figure 5. The MITA Development Environment and Tools.

were asked to draw and record underwriting conclusions from these. A week later they were given the actual text from which the extractions had been made and were asked again to draw and record their underwriting conclusions. The recorded results were then reviewed by actuaries to identify and characterize any differences in underwriting decisions.

Results show that two percent to seven percent of the extractions resulted in different underwriting conclusions. In most cases, the action led the underwriters to take extra review steps. This conservative approach, causing the underwriter to take extra review steps, is consistent with our design objective to maximize the recall of any significant underwriting event even if it results in the lowering of precision.

Another test conducted involved sorting the extractions from a randomly selected set of cases in descending order of confidence factor by field. The underwriters reviewed each extraction in the sorted order and identified where they saw the first incorrect extractions. The threshold was set at a level somewhat higher than the level where these inaccuracies were first encountered. As a result of this test, the threshold confidence factor was set to 50 for family history and 70 for the other four textual fields.

Knowledge Engineering

The system was engineered to imitate the ability of underwriters to read text and recognize the potentially important pieces of information. The focus of this engineering was to

achieve optimal performance for the five designated text fields. Furthermore, 20,000 sample cases that had previously been processed manually and that were stored in a relational database were made available to the team as a basis for testing and engineering the system.

The knowledge-engineering process, therefore, was focused on both underwriter knowledge and text characteristics. These objectives required that we process the corpus, looking for common texts and phrases, and focus our attention on the most common constructs in decreasing order of frequency.

Many hours were spent with the underwriters, looking at texts, asking about potential significance, hypothesizing patterns, confirming or rejecting these hypotheses by queries over the text samples, and asking questions of the underwriters.

As the rule base and lexicon grew, mechanization of the process began. A knowledge-acquisition tool was created that allowed underwriters to annotate cases with the correct results. This tool was coupled with a validation tool that could compare MITA's output with that specified by the underwriters. Later, the tool was enhanced so that rather than asking the underwriters to annotate from scratch, they were presented with MITA's output from a text and asked to confirm or correct MITA's results.

The underwriters' ability to recognize significant concepts was essential to the project. During every phase, the knowledge engineers consulted with them to determine what combinations of syntax, semantics, keywords, and

key phrases indicated the presence of a potentially significant concept. The underwriters also provided strong direction on how to classify words and phrases and what the relationships should be among classes.

Custom Development Tools

Three tools were developed to assist in MITA development (figure 5): (1) knowledge acquisition, (2) validator, and (3) ontology.

The *knowledge-acquisition tool* allows the underwriter to create annotated cases to be used in system development and testing. The knowledge-acquisition tool provides the underwriter with an interface showing a text and some preliminary extractions (concept, value, and class). The underwriter can change or add concept names with a drop-down list, change concept values (the extracted words) with cut and paste, and change or add classes through a link with the ontology tool that allows the underwriter to browse and navigate the class hierarchy.

The *validator* is a utility created in ART*ENTERPRISE that compares the output of MITA for a particular text to the annotated expected results. Reports show where and how the two sets of results differed.

The *ontology tool* assists the knowledge engineer in browsing and modifying the dictionary. It allows the user to navigate between words and their classes and even among the participants of a composite class. Entries and relationships can be added or changed at any point.

System Maintenance

In production, MITA has proved extremely robust. Maintenance of four of the analyzers has been limited to adding terms that did not appear in the original corpus or adjusting connections between classes. In the case of one analyzer, MITA 11A, however, there have been more extensive issues. Figure 1 shows that field 11A passes MITA 70 percent of the time, whereas the similar medical field, 11E, passes 83 percent of the time. We focused maintenance activities on field 11A to decrease this 13-percent difference.

To assist with the smooth maintenance of MITA, all unprocessed fields, along with their partially parsed results, are written to a log file. Analysis has shown four factors strongly contribute to the textual field falling below the confidence-level threshold: (1) previously unknown text patterns; (2) misspellings and unrecognized abbreviations; (3) medical terms, particularly drugs, that were not part of the

original corpus or in SNOMED; and tenses of words and/or word combinations that were not part of the original corpus but whose root terms are already in the dictionary.

Analysis of the 17,540 texts on the log file has indicated that there are three word patterns for which rules do not currently exist and which occur with sufficient frequency to justify inclusion. One such rule is of the following form:

(Vehicle type) (Accident), (Body-Part)
(And) (Body-Part) (Condition)
e.g. "Plane crash, arm and back broken"

Additional extraction rules will be written to handle these word patterns.

Within the corpus, the project team has found 4,669 words that are not in the MITA dictionary, the abbreviation table, or the *correct spell table* (which contains common misspellings and the related correct spelling). What is more significant is that these words occur 7,300 times throughout the texts, which means that of the 105,446 words in the texts, 6.92 percent are misspellings, unidentified words, or abbreviations. Of the 4,669 words that have been identified, only 118 occur more than 4 times in the texts; so, they would be candidates for inclusion in the MITA dictionary tables.

Of the 4,669 words that MITA could not identify, 87 were medical terms not found in the existing MITA dictionary. These terms occurred 199 times in the text.

Additional changes to be evaluated include adding a comprehensive drug list to the MITA ontology, (2) installing a spell checker during the data entry of the application, and (3) adding identified misspellings with a frequency greater than four to the correct spell table.

Technology Transfer

The earlier phases of the MITA project focused on rapid development over a short time period. Most of the work was performed by Brightware consultants with strong backgrounds in NLP, AI, and the engineering of knowledge-based systems. As the MITA implementation moved toward completion, a technology transfer effort was begun in conjunction with testing and deployment.

The goal of technology transfer was to transfer the everyday operations of MITA to MetLife staff members and prepare MetLife personnel as much as possible to support and enhance MITA in the future.

Initial technology transfer tasks included involving MetLife personnel in the analysis of potential dictionary and ontology changes to improve MITA's performance. Thus, staff mem-

The goal of technology transfer was to transfer the everyday operations of MITA to MetLife staff members and prepare MetLife personnel as much as possible to support and enhance MITA in the future.

bers were able to gain a familiarity with ontology structures and relationships and their impact on the performance of MITA.

Subsequent steps involved the participation of MetLife people in performing tests and analyzing MITA test results. Next, they became familiar with the ART*ENTERPRISE tool set and begin participating in making code changes, particularly to the extraction rules. Ultimate full support will require an understanding of the functioning of all MITA modules.

Because hands-on development is the only way to truly acquire an understanding of the technologies, tools, and structures underlying MITA, tentative plans call for a joint team of Brightware and MetLife people to develop the next set of text analyzers.

Summary

After six months of production, results indicate that MITA is able to successfully extract information from free-form text fields on life insurance applications that can be analyzed, providing a high degree of automation of the initial underwriting review. The same technique can be extended to other text fields in life insurance; other types of underwriting (property and casualty); and other insurance systems such as claims processing. Immediate extensions of MITA to handle underwriting correspondence and process electronic transmissions of requirements are contemplated.

The large number of applications, which will be enabled by information extraction, is indicative of the high percentage of business knowledge that appears as text, relative to structured information. Use of NLP to manipulate this body of knowledge is now becoming possible because of improved algorithms and increased computational power available for dedicated tasks.

Acknowledgments

We would like to acknowledge the contribution of the MetLife underwriters Eileen Kosiner, Bill Rowe, Mike Breen, and Joe Grecco, without whose enthusiastic participation MITA would not exist. We would also like to thank Linda Mattos for leading the underwriter effort, Joe Casalaina for assisting in numerous technical tasks, and Bob Riggio for providing guidance and support throughout the project. Thanks also to Ellen Riloff, who provided the benefit of her extensive knowledge of NLP and information extraction from both a theoretical and a practical point of view. Finally, thanks to the members of the Brightware team: Daniel Grudus, Fred Simkin, Michael Belofsky, David

Reel, Jay Runkel, Amy Rice, and Samir Rohatgi.

Note

1. NOS means "not otherwise specified."

References

- Aberdeen, J.; Burger, J.; Day, D.; Hirschman, L.; Robinson, P.; and Alembic, M. 1995. MITRE: Description of the Alembic System Used for MUC-6. In Proceedings of the Sixth Message-Understanding Conference (MUC-6), 141–155. San Francisco, Calif.: Morgan Kaufmann.
- Appelt, D.; Hobbs, J.; Bear, J.; Israel, D.; Kameyama, M.; Kehler, A.; Martin, D.; Myers, K.; and Tyson, M. 1995. SRA International FASTUS System: MUC-6 Test Results and Analysis. In Proceedings of the Sixth Message-Understanding Conference (MUC-6), 237–248. San Francisco, Calif.: Morgan Kaufmann.
- Campbell, K., and Mussen, M. 1992. Representation of Clinical Data Using SNOMED-III and Conceptual Graphs. In Proceedings of the Sixteenth Annual Symposium on Computer Applications in Medical Care, 354–358. New York: McGraw-Hill.
- Childs, L.; Brady, D.; Guthrie, L.; Franco, J.; Valdes-Dapena, D.; Reid, B.; Kieley, J.; Dierkes, G.; and Sider, I. 1995. Lockheed Martin: LOUELLA PARSING, An NL Toolset System for MUC-6. In Proceedings of the Sixth Message-Understanding Conference (MUC-6), 97–111. San Francisco, Calif.: Morgan Kaufmann.
- Cowie, J., and Lehnert, W. 1996. Information Extraction. In *Communications of the ACM* 39(1): 80–91.
- Do Amaral Marcio, B., and Satomura, Y. Associating Semantic Grammars with SNOMED: Processing Medical Language and Representing Clinical Facts into a Language-Independent Frame. Paper presented at MEDINFO 1995, 23–27 July, Vancouver, British Columbia.
- Fisher, D.; Soderland, S.; McCarthy, J.; Feng, F.; and Lehnert, W. 1995. Description of the UMass System as Used for MUC-6. In Proceedings of the Sixth Message-Understanding Conference (MUC-6), 127–140. San Francisco, Calif.: Morgan Kaufmann.
- Grishman, R. 1995. The NYU System for MUC-6, or Where's the Syntax? In Proceedings of the Sixth Message-Understanding Conference (MUC-6), 167–175. San Francisco, Calif.: Morgan Kaufmann.
- Krupka, G. 1995. SRI: Description of the SRA System as Used for MUC-6. In Proceedings of the Sixth Message-Understanding Conference (MUC-6), 221–235. San Francisco, Calif.: Morgan Kaufmann.
- Krupka, G.; Jacobs, P.; Rau, L.; Childs, L.; and Sider, I. 1992. GENLTOOLSET: Description of the System as Used for MUC-4. In Proceedings of the Fourth Message-Understanding Conference (MUC-4), 177–185. San Francisco, Calif.: Morgan Kaufmann.
- Lee, R. 1995. Sterling Software: An NL Toolset-Based System for MUC-6. In Proceedings of the Sixth Message-Understanding Conference (MUC-6), 249–261. San Francisco, Calif.: Morgan Kaufmann.
- Lehnert, W. 1990. Symbolic-Subsymbolic Sentence Analysis: Exploiting the Best of Two Worlds. In *Advances in Connectionist and Neural Computation Theory, Volume 1*, 151–158. Norwood, N.J.: Ablex.

Lehnert, W. G., and Sundheim, B. 1991. A Performance Evaluation of Text Analysis Technologies. *AI Magazine* 12(3): 81–94.

MUC3. 1991. *Proceedings of the Third Message-Understanding Conference*. San Francisco, Calif.: Morgan Kaufmann.

MUC4. 1992. *Proceedings of the Fourth Message-Understanding Conference*. San Francisco, Calif.: Morgan Kaufmann.

MUC5. 1993. *Proceedings of the Fifth Message-Understanding Conference*. San Francisco, Calif.: Morgan Kaufmann.

MUC6. 1995. *Proceedings of the Sixth Message-Understanding Conference*. San Francisco, Calif.: Morgan Kaufmann.

Oliver, D., and Altman, R. 1994. Extraction of SNOMED Concept from Medical Record Texts. In *Proceedings of the Eighteenth Annual Symposium on Computer Applications in Medical Care*, 179–183. Washington, D.C.: IEEE Computer Society.

Riloff, E. 1996. An Empirical Study of Automated Dictionary Construction for Information Extraction in Three Domains. *AI Journal* 85:101–134.

Riloff, E. 1993. Automatically Constructing a Dictionary for Information-Extraction Tasks. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, 811–816. Menlo Park, Calif.: American Association for Artificial Intelligence.

Sager, N. 1980. *Natural Language Information Processing: A Computer Grammar of English and Its Applications*. Reading, Mass.: Addison-Wesley.

Sager, N.; Lyman, M.; Bucknall, C.; Tick, I. 1994. Natural Language Processing and the Representation of Clinical Data. *Journal of American Medical Informatics Association* 1(2): 142–160.

Senator, T.; Goldberg, T.; Wooton, J.; Cottini, M.; Khan, U.; Klinger, C.; Llamas, W.; Marrone, M.; and Wong, R. 1995. The Financial Crimes Enforcement Network AI System (FAIS). *AI Magazine* 16(4): 21–31.

Soderland, S.; Fisher, D.; Aseltine, J.; and Lehnert, W. 1996. Issues in Inductive Learning of Domain-Specific Text-Extraction Rules. In *Connectionist, Statistical, and Symbolic Approaches to Learning for Natural Language Processing*, eds. S. Wermter, E. Riloff, and G. Scheler, 290–301. New York: Springer-Verlag.

Soderland, S.; Fisher, D.; Aseltine, J.; and Lehnert, W. 1995. CRYSTAL: Inducing a Conceptual Dictionary. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, 1314–1321. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence.

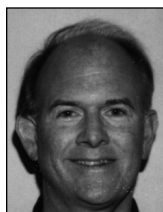
Weischedel, R. 1995. BBN: Description of the PLUM System as Used in MUC-6. In *Proceedings of the Sixth Message-Understanding Conference (MUC-6)*, 55–69. San Francisco, Calif.: Morgan Kaufmann.

Yang, Y., and Chute, C. 1992. An Application of Least Square Fit Mapping to Clinical Classification. In *Proceedings of the Sixteenth Annual Symposium on Computer Applications in Medical Care*, 460–464. Washington, D.C.: IEEE Computer Society.



Barry Glasgow received his Ph.D. in theoretical physics from St. Johns University in 1975. He has worked as an application project manager and advanced technology strategist. He is currently a director of Information Technology Services at MetLife responsible for business productivity and knowledge management. His e-mail address is bglasgow@metlife.com.

Alan Mandell received his B.S. in computer science from Baruch College in 1984. He is a manager of Information Technology Services at MetLife and has worked as a member of the New Business Systems Department in support of individual contracts. He is currently active in implementing various business process reengineering and systems techniques to improve the efficiency of individual operations.



Dan Binney is a managing consultant at Brightware, Inc. He was project manager and technical lead for the MITA Project. He has been involved with the design, development, and delivery of commercial knowledge-based systems for more than 11 years. Binney holds a Masters in computer science from the New Jersey Institute of Technology. His e-mail address is binney@brightware.com.



Lila Ghemri is a knowledge engineer at Brightware, Inc. She received her B.S. in computer science from the University of Bab Ezzouar, Algiers, and holds a Ph.D. from the University of Bristol, United Kingdom. She worked at MCC on the cyc Project. Her interest and work have been in

developing knowledge-based systems and natural language processing applications. Her e-mail address is ghemri@brightware.com.

David Fisher is a research associate and senior software engineer at the Natural Language Processing Laboratory of the University of Massachusetts at Amherst. His research focuses on the development and integration of information-extraction technology. He received his M.S. in computer science from the University of California at Berkeley (1994) and his B.S. in computer science from the University of Massachusetts at Amherst (1992). His e-mail address is dfisher@cs.umass.edu.