AI Magazine Volume 22 Number 2 (2001) (© AAAI)

LIFECODE
A Deployed Application for Automated Medical Coding

Daniel T. Heinze, Mark L. Morsch, Ronald E. Sheffer, Michelle A. Jimmink, Mark A. Jennings, William C. Morris, and Amy E. W. Morsch

■ LIFECODE is a natural language processing (NLP) and expert system that extracts demographic and clinical information from free-text clinical records. The initial application of LIFECODE is for the emergency medicine clinical specialty. An application for diagnostic radiology went into production in October 2000. The LIFECODE NLP engine uses a large number of specialist readers whose particular output are combined at various levels to form an integrated picture of the patient's medical condition(s), course of treatment, and disposition. The LIFECODE expert system performs the tasks of combining complementary information, deleting redundant information, assessing the level of medical risk and level of service represented in the clinical record, and producing an output that is appropriate for input to an electronic medical record (EMR) system or a hospital information system. Because of the critical nature of the tasks, LIFECODE has a unique "self-awareness" feature that enables it to recognize the limits of its competence and, thus, ask for assistance from a human expert when faced with information that is beyond the bounds of its competence. The LIFECODE NLP and expert systems reside in various delivery packages, including online transaction processing, a web browser interface, and an automated speech recognition (ASR) interface.

LIFECODE is a natural language processing (NLP) system that extracts clinical information from free-text medical records. In the United States alone, medicine is a trillion dollar a year business and generates in excess of 700 million clinical documents in transcribed free-text form. With the view of medicine as a business, the clinical information in the free-text records is coded to produce a bill for services and facility use. Another desirable business application of the information is to track physician performance and resource use. From the clinical perspective, the information in the clinical notes can be used to improve communications between multiple providers for the same patient, monitor the efficacy of alternate courses of treatment and provide feedback, and provide alerts relative to the course of care for a particular patient.

During a medical encounter, the physician (or other medical practitioner) elicits from the patient a medical complaint, the history of the condition, and a review of the patient's medical status based on the patient's answers to the physician's questions or a review of old medical records. The physician will perform a physical examination of one or more organ systems or body areas and optionally order or perform and review the results of tests, consult with other healthcare practitioners, perform medical or surgical procedures, counsel the patient regarding current and follow-up care, make arrangements for further work-up or continuing care, and arrive at the final diagnosis(es) for the visit. As a matter of law and as a requirement for payment, the physician must thoroughly document all aspects of the rendered care. The document can be handwritten, typed, dictated and transcribed, or created with a paper or computer check-off system (invariably augmented with free text). Voice-only recordings are not currently acceptable as a legal medical record.

After the document is created, it must be reviewed and signed by the physician, the information in the document must be mapped onto a large and complex set of medical codes for purposes of statistical abstracting and billing, and the document must be stored as a

Copyright © 2001, American Association for Artificial Intelligence. All rights reserved. 0738-4602-2001 / $2.00

part of the patient's permanent medical record.

Currently, virtually all medical coding is done manually using either reference books or computerized code look-up systems. The system is at best a computer-aided human. For storage, paper documents have been the mainstay of medical records, but there is increasing pressure to move to the *electronic medical record* (EMR), also referred to as the computerized patient record. Although the EMR has been a major goal in health information management (HIM) for more than two decades, the success of such systems has been seriously limited because of the relative inaccessibility of the information in free-text clinical documentation. Attempts to change the documentation habits of physicians have not had significant success largely because of the increased time and inconvenience associated with using computer interfaces that require formatted input. Further, numerous consultations with practicing physicians have taught us that there is a basic inability of fully structured systems to represent many of the nuances that make each case unique.

The broad, although not universal, belief is that the enabling technologies for the success of the EMR are high-accuracy automated speech recognition for the creation of the medical document and NLP for the coding and abstraction of the document. LIFECODE fills the coding and abstraction niche using a unique blend of NLP techniques.

Other programs for NLP on medical free text differ substantially from LIFECODE. Medical document retrieval and classification systems determine only if a particular subject is discussed within a document (Aronow and Shmueli 1996; Aronow, Cooley and Sonderland 1995; Aronow, et al. 1995; Croft, Callan, and Aronow 1995; Hirsch and Aronow 1995; Lehnert et al. 1994; Sonderland et al. 1995).[1] The system developed by Sonderland et al. (1995) used a *K*–nearest-neighbor approach and would broadly be classified as a statistical system. Although it was also used in an attempt to perform medical coding and abstraction, such approaches do not distinguish typical roles such as agent (who performed the surgery) or patient (who had the illness). They do not discern temporal information such as duration (how long the patient has been ill) or timing (how frequent the bouts of pain are). They do not discern negation (the patient was diagnosed as not having the illness under discussion). The list goes on, but these examples should be sufficient. Research in statistical text processing has advanced considerably in the last five years, with techniques such as naive Bayesian, support vectors, and boosting currently in vogue (Friedman, Geiger, and Goldszmidt 1997). The application of statistical methods to medical coding and abstracting is best made in areas that can be painted in broad, coarse strokes, most notably classification of paragraphs into various categories of subjective information (what the patient reports about his/her own condition) and objective information (what the physician reports based on physical examination and testing of the patient).

Compared to statistical methods, medical word and phrase tagging systems operate at a much more granular level to apply tags that disambiguate semantic senses (Sager et al. 1996, 1994). They would discern, for example, the verb form of *burning* (for example, the flame was burning the patient's finger) from the adjectival form (for example, the patient had a burning substernal chest pain). Tagging does not in and of itself solve issues such as roles, negation, and temporality. Attempts to do medical coding (assignment of predefined medical codes that identify diseases, injuries, medical procedures, and so on) typically have not dealt with the issues of role, negation, timing, and so on (Larkey and Croft 1995; Lenert and Tovar 1993; Yang and Chute 1992). Some, however, use very complex linguistic processing and achieve very high accuracy (Sager, Lyman, and Bucknall 1994), but such systems require many years of development and have not been able to move easily into the commercial marketplace. Systems that use a less rigorous linguistic approach, whether to specific medical specialties such as radiology (Ranum 1988; Zingmond and Lenert 1993) or to general medical texts (Sneiderman, Rindfleisch and Aronson 1996; Sneiderman et. al. 1995) typically lack both the specificity (in terms of roles, temporality, and so on) and the accuracy (in terms of precision and recall) to be used in critical tasks such as medical billing or populating an EMR from free text. None of the systems and projects discussed thus far incorporate the inference and logic capabilities necessary to refine medical diagnosis and procedure codes according to the extensive medical and legal guidelines, nor do they have the knowledge required to use coded information for reporting purposes.

Further, by way of comparison, commercial products that advertise medical NLP (for example, HBOC's AUTOCODER or Medicode's ENCODER PRO) are essentially keyword-recognition systems for searching online versions of paper reference manuals. They lack NLP competence but do have some level of knowledge regarding

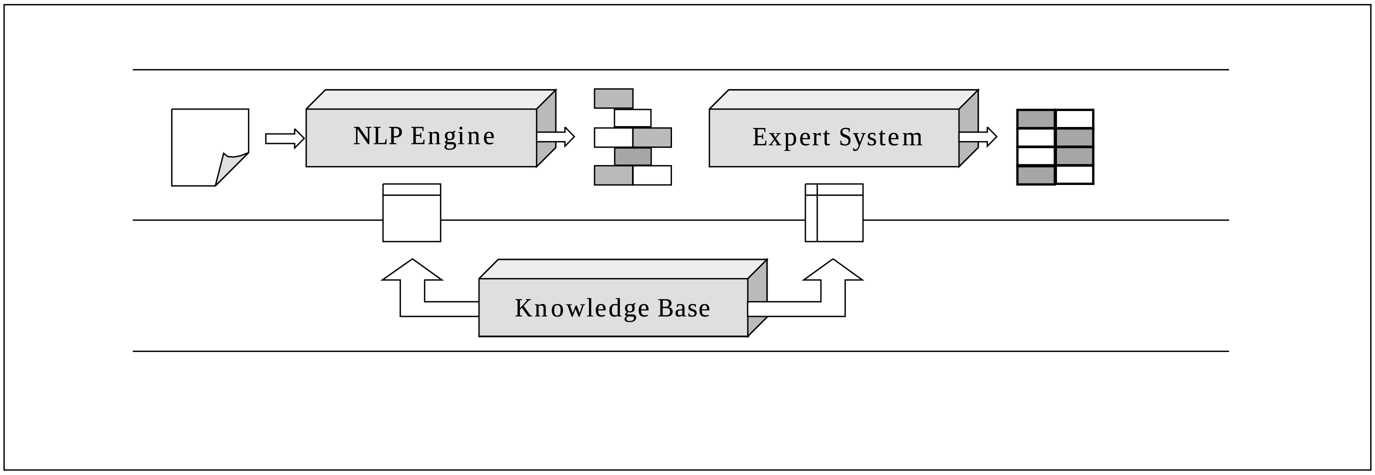


Figure 1. LIFECODE *Architecture.*

the proper use and reporting of user-selected codes.

Aside from the issues already discussed, a major drawback of all these systems is that they are unable to discern the presence of information that is beyond the scope of their competency. To be useful in a real-world application, a medical NLP system must be able to discern when it is able to operate unassisted and when it needs to seek human intervention to maintain the appropriate quality level. We refer to this quality as *self-awareness.*

LIFECODE provides both linguistic competence and medical knowledge and logic to use NLP to extract from a free-text clinical note (1) the patient demographics (name, age, gender, and so on); (2) the patient's chief complaint; (3) facts about the history of the present illness (duration, severity, time of onset, circumstances of medical relevance, related signs and symptoms, location of the injury-illness, context of onset, and so on); (4) facts about the medical history of the patient and (as applicable) patient's family; (5) facts about the relevant social history (use of tobacco, alcohol and drugs, living arrangements, and so on); (6) facts concerning the nature and extent of the physical examination performed by the physician; (7) facts concerning the nature and extent of old records consulted, professional consultations, and medical tests performed by the physician; (8) the final diagnoses, potentially also including possible and ruled-out diagnoses; (9) the course of treatment including surgical procedures, drug therapy, and monitoring levels; and (10) facts about the disposition of the patient at the end of the clinical encounter with the physician. LIFECODE is also an expert system that determines from the extracted information (1) the most specific version of each diagnosis and procedure; (2) the level of documentation of the history and physical examination; (3) the risk to the patient presented by the medical condition and treatment; (4) the complexity of the medical decision making for the physician; (5) the level of service provided by the physician; and (6) the appropriate manner in which to report the event for billing purposes based on the type of medical provider, the place of medical care, and the particular requirements of the insurance carrier.

Application Description

The LIFECODE system is organized into two layers, as seen in figure 1. The top layer is the executable portion, implemented largely in C++ with several finite-state and context-sensitive processors. This top layer contains two modules: (1) the NLP extraction engine and (2) the expert system. As shown in figure 1, documents flow into the NLP extraction engine and are transformed into a collection of discrete data elements. These data elements are represented in figure 1 as a poorly aligned group of shaded and unshaded blocks, signifying the unfinished nature of the information at this stage. The expert system module takes this collection as input and applies rules that filter, combine, and restructure the data elements into the data records that are then saved in a master database. The bottom layer represents the system knowledge base. In an effort to abstract the domain knowledge away from the source code, the knowledge bases contain the medical vocabulary; definitions covering anatomy, microbiology, medications, signs, symptoms, diagnoses, and procedures; and rules for medical coding. These data (and

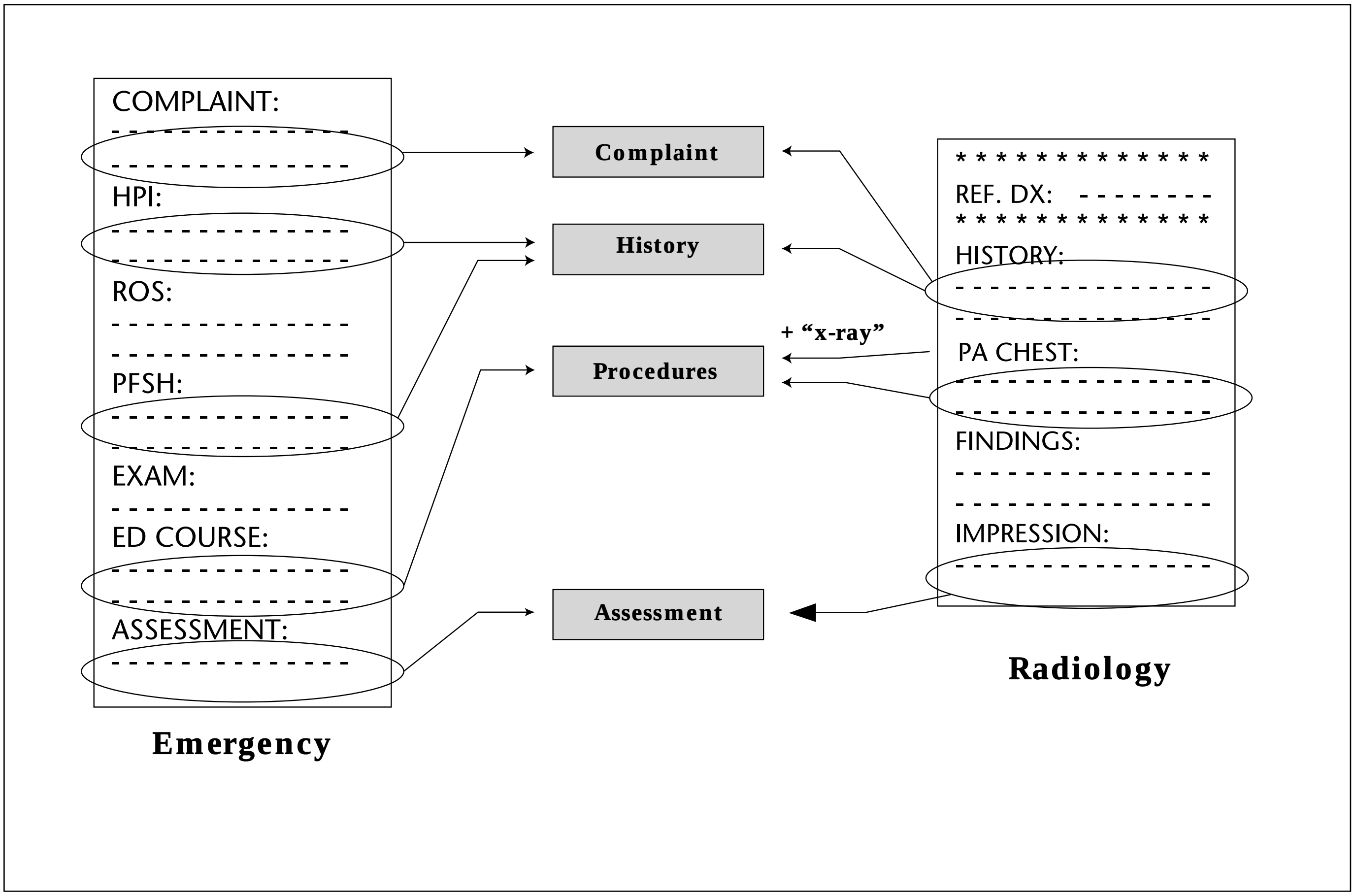


Figure 2. Document Segmentation.

more) make up the knowledge base and are written using proprietary specification languages that are compiled, using custom utility programs, into C++ data objects. These data objects are read in at initialization of the top layer that can be executed. In figure 1, these data objects are illustrated by the icons that are shown flowing from the knowledge base to the NLP engine and expert system modules. This design allows system upgrades through modification of the knowledge bases, without requiring recompilation of the C++ source code for the NLP engine or expert system.

Looking more closely at the executable layer, the NLP module blends multiple types of text-processing technique, including morphological reduction, pattern matching, bottom-up parsing, synonym substitution, and vector analysis, to recognize, extract, and categorize the key information in a clinical note. There are four components that make up the NLP module: (1) document segmenter, (2) lexical analyzer, (3) phrase parser, and (4) concept matcher. These components execute in sequence, accepting the note as ASCII text and producing a list of discrete data elements that are organized by type with each assigned a semantic label. The types broadly categorize the extracted information according to the main themes of the note, including procedures, diagnoses, symptoms, current history, past history, physical examination, and medications. The semantic labels assign a meaning to each element that corresponds to a definition in the system's external knowledge base.

Clinical notes are typically composed of multiple paragraphs, divided into blocks of text by headings. The *document segmenter,* as illustrated in figure 2, identifies and categorizes the text based on the meaning of the heading that precedes each block. The meanings of the headings are determined by comparing, using a flexible pattern-matching scheme, against a set of possible heading definitions specified in the knowledge base. This process places each portion of a note in a broad context as defined by the meaning of an associated section heading. Examples of section headings are "History of Present Illness," "Review of Systems," "Physical Examination," "Medical Decision Making," and "Final Diagnosis." As the text is processed by subsequent modules, this context is preserved and is used later on to compute the type of each extracted data element. The output of the document segmenter is a linked list of text sections, stored in the order they appeared in the original note. As we discuss later, knowing the context of each data element is required to reach the level of precision required of medical coding.

The *lexical analyzer module* is a series of processors designed to transform the text into a string of symbols that are consistent with the vocabulary of the knowledge base specifications. These functions include acronym expansion and morphological reduction. In the acronym expansion, each unambiguous acronym in the text is converted into its full definition. Acronyms considered ambiguous, having more than one potential meaning, are either left unchanged, allowing the concept matcher to resolve conflicts, or an appropriate default definition is selected. Morphological reduction transforms multiple morphological variants of a word into a base form, which is done for words where morphological variation does not affect the underlying concept being expressed. In addition to text transformation, scalar values representing temporal information, vital signs, and laboratory test results such as body temperature and oxygen saturation are extracted and stored. Cardinal and ordinal numbers are replaced by tokens that uniquely encode their values.

After the lexical analyzer has generated all the tokens, the phrase parser performs a bottom-up syntactic analysis. Figure 3 shows the architecture of the NLP engine, including the phrase parser and concept matcher. The parser is highly resilient and tolerant of the incorrect grammar that characterizes clinical documents and unknown words. Primarily, the information needed for medical coding is expressed in the noun phrases of a text. The boundaries of a noun phrase are typically defined by prepositions, verbs, or some type of punctuation. The phrase parser uses these delimiters to form chunks of text of a size from two or three words to a complete sentence, which roughly corresponds to the granularity of the definitions within the knowledge base. Although nouns and noun phrases are the focus, verbs are not ignored in this process. Verbs can be key terms in the definitions of medical procedures. Therefore, the phrase parser preserves verbs and most other modifying words as it forms chunks of text.

The concept matcher uses vector analysis to assign meanings to each phrase. These meanings are represented as labels and can correspond to one or more chunks of texts, depending on the scope of the definition in the knowledge base. In vector analysis, meanings are assigned by modeling the knowledge base as a vector space. Each word is a separate dimension. Every definition in the knowledge base is represented by a vector in this vector space. To find the meaning of a phrase, the concept matcher plots a vector representing the phrase into the knowledge base vector space to determine the closest meaning.

The following example illustrates the vector analysis performed by the concept matcher for a simple ICD-9 dictionary (Medicode 1999). Consider a dictionary with four ICD-9 codes: (1) 786.50, chest pain unspecified; (2) 786.51, substernal chest pain; (3) 786.52, chest wall pain; and (4) 786.59, musculoskeletal chest pain.

These four codes cover the chest pain category within the ICD-9 coding guidelines. Codes 786.53 through 786.58 are not defined but are available for future expansion of the guidelines. In these four definitions, there are six unique words (ignoring case): (1) chest, (2) pain, (3) unspecified, (4) substernal, (5) wall, and (6) musculoskeletal. For the purposes of vector analysis, these six unique words can be treated as six dimensions. Thus, the four definitions in the example dictionary can be represented as four unit vectors within a six-dimensional space. The concept matcher assigns meaning to a phrase by identifying the vector from the dictionary, and thereby the definition, that most closely matches the vector formed from the words in the phrase. The closest match is determined by computing the angular difference between the vector from the phrase and each vector from the dictionary. The angular difference is computed using a simple inverse cosine formula. The vector from the dictionary with the smallest angular difference, as long as the difference is below a defined threshold, is the best match. A threshold is required to ensure that the best match from the dictionary has significant similarity with the words in the phrase. Typically, this threshold is set between 0° and 45°. To obtain a perfect match, an angular difference of 0°, a phrase must contain every word in a definition but no more. For the simple ICD-9 dictionary defined earlier, the phrase *chest wall pain* is a perfect match for the definition of the ICD-9 code 786.52.

A second evaluation phase after the initial vector difference computation is used to refine the matches, which includes using anatomy, medication, and microbiology concept hierarchies and synonym lists to improve chances of a match. Also, syntactic heuristics can be applied. These heuristics join and redistribute

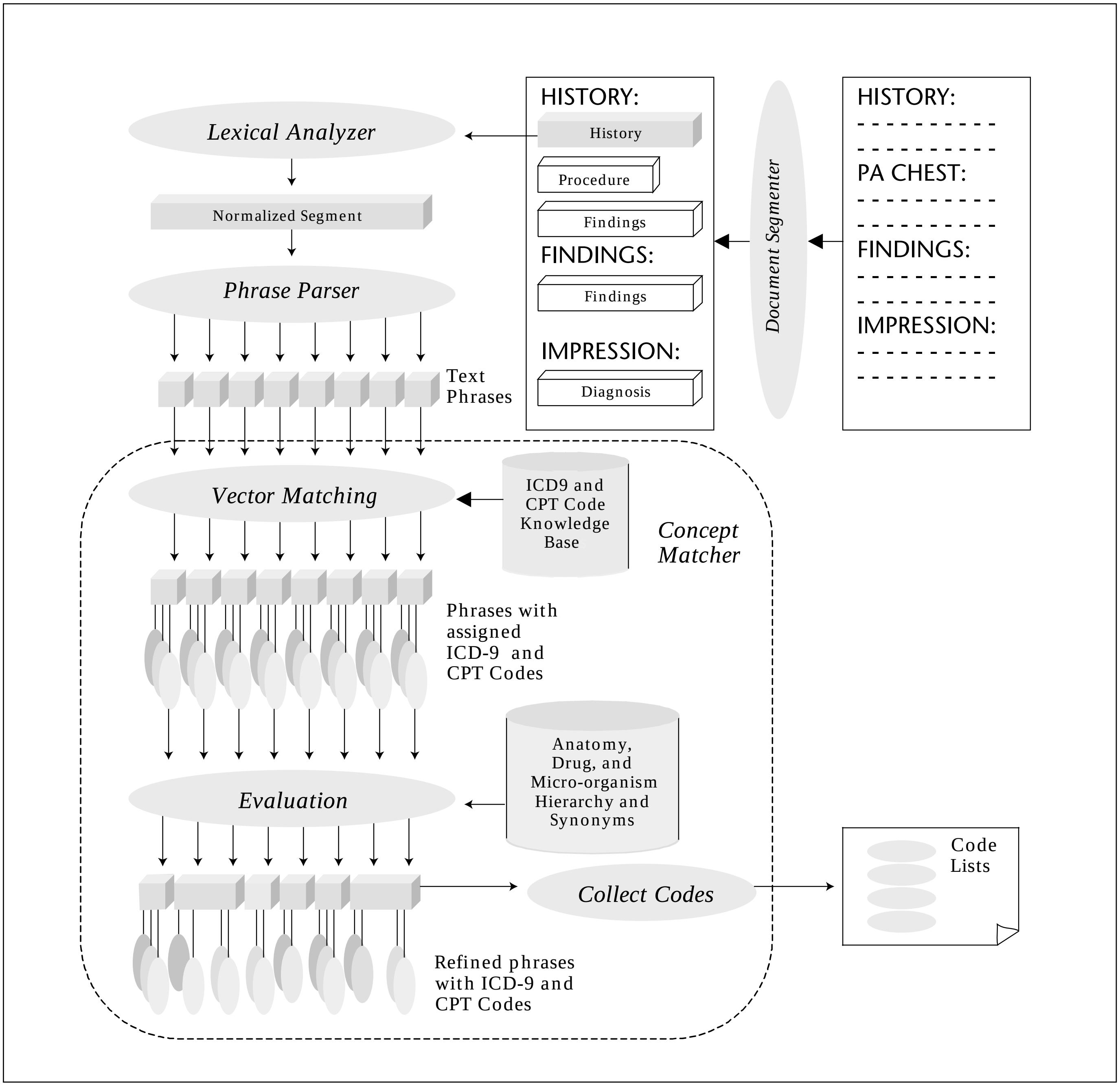


Figure 3. Natural Language Processing Engine Architecture.

words from two or more consecutive phrases that were divided by the phrase parser and compute the meaning for the new combined phrase. With meanings assigned to individual chunks of text, the extracted data elements are formed by collecting all the semantic labels and forming a list. The labels are grouped on this list according to their context in the note.

The expert system module applies specialty-specific rules of medical coding, assigning a final set of diagnosis and procedure codes. The codes are derived from the semantic labels; in fact in many cases, the actual ICD-9 (diagnosis) (Medicode 1999) or current procedural terminology (CPT) (medical procedure) (AMA 1999) codes are used as labels. This module consists of specialized algorithms and business rules that have been developed through analysis of published guidelines and consultation with medical coding experts. The context is important at this stage because elements with similar definitions can have different roles in different contexts. For example, in emergency medicine, the review of systems (a subjective inventory of symptoms from the patient) and the physical examination (an objective report of findings made by the physician) can have similar language and, therefore, similar concepts. However, they serve different roles in assigning an overall level of service code to the encounter. Data elements from these two contexts cannot be intermingled. In addition to computing the final codes, the expert system assesses the quality of the coding, flagging notes that should be reviewed by a human expert. The criteria for this assessment are the consistency of the data extracted, the complexity of the diagnoses and procedures, and incomplete information.

The entire LIFECODE system runs at the core of a continuously operating (24/7) data center. Our business operates as a service bureau, receiving electronic notes by ftp or dial-up connections. The notes are held for a period of time until insurer demographics and addenda have been received. From there, LIFECODE runs on the documents with the results stored in a master transaction database. The document, medical codes, and insurer demographics are returned to the client electronically, and the client's staff reviews the results using a coding review workstation. The data center operates within a windows nt environment on high-end Intel pentium platforms.

Uses of AI Technology

In the sense that LIFECODE is the brainchild of its inventors and developers, it is in the lineage of cognitive linguistics. We cannot, however, claim that LIFECODE is a truly cognitive system. "Cognitive linguistics and cognitive grammar are 'cognitive' in the sense that, insofar as possible, language is seen as drawing on other, more basic systems of abilities (e.g. perception, attention, categorization) from which it cannot be dissociated" (Langacker 1999, p. 1). LIFECODE, of course, does not have "more basic systems of abilities," as listed by Langacker. It is, however, designed to operate as if it did possess these basic systems and, more importantly, the corresponding mental capacities, for example, the assessment of duration, ability to shift attention to different narrators and events, and a sense of urgency. In terms of the core AI components, there is little in LIFECODE that has not been available in NLP work for some time, including such basic functions as lexical, syntactic, and semantic analysis. What makes LIFECODE unique is the organization of basic components in a manner that reduces each of the functions into a myriad of agents that work either independently or cooperatively. At this level of reduction, the lines between lexical, syntactic, and semantic analysis begin to blur. However, for the sake of illustration, there are nearly three dozen agents that operate primarily at the lexical and syntactic level. It is, then, not so much the advances in AI techniques that have made LIFECODE possible but, rather, the particular reduction that we have applied to the top-level functions and the system-level organization that has been imposed to synthesize a domain-specific level of natural language understanding.

At the algorithm or technique level, there are two noteworthy advances in LIFECODE. LIFECODE represents an advance in the sheer amount of knowledge that it is able to apply to NLP within a reasonable amount of time. The computationally intensive nature of NLP is well known. In dealing with a single sentence, LIFECODE's core engine will reference the linguistic and medical knowledge bases from several thousand to several million times. The average number of references is about 50,000 a sentence. In addition to techniques such as caching, LIFECODE uses a novel dynamic programming technique that is, to the best of our knowledge, on the order of 10 times faster than other algorithms. Typical of dynamic programming techniques (Bentley 1996), this algorithm utilizes a large table to store partial results during the vector analysis. As a result of this technique, LIFECODE (on a 500-megahertz PENTIUM PC running WINDOWS NT) is able to run a knowledge base with well more than 3 million entries against a 400-word document in 10 to 20 seconds.

The second noteworthy technique is LIFECODE's self-awareness. For the medical applications against which LIFECODE is applied, it is unrealistic to think that a computer could at this time reach a level of understanding that would enable it to work unsupervised and unaided. In fact, human professionals frequently find themselves resorting to reference materials or consulting experts. In this respect, humans are largely aware of the limits of their mental abilities and are able to determine when consultation is required. For our applications, a computer would not be particularly useful if it did not know when it was at the limits of its knowledge or abilities, which would require that a human expert review all the computer's output, thus negating the computer's usefulness. In one sense, the ability to know when to ask for help can be construed as the ability to recognize the difference between those unknowns that matter and those that do not matter. To achieve this ability in LIFECODE, we have developed a technique that we call

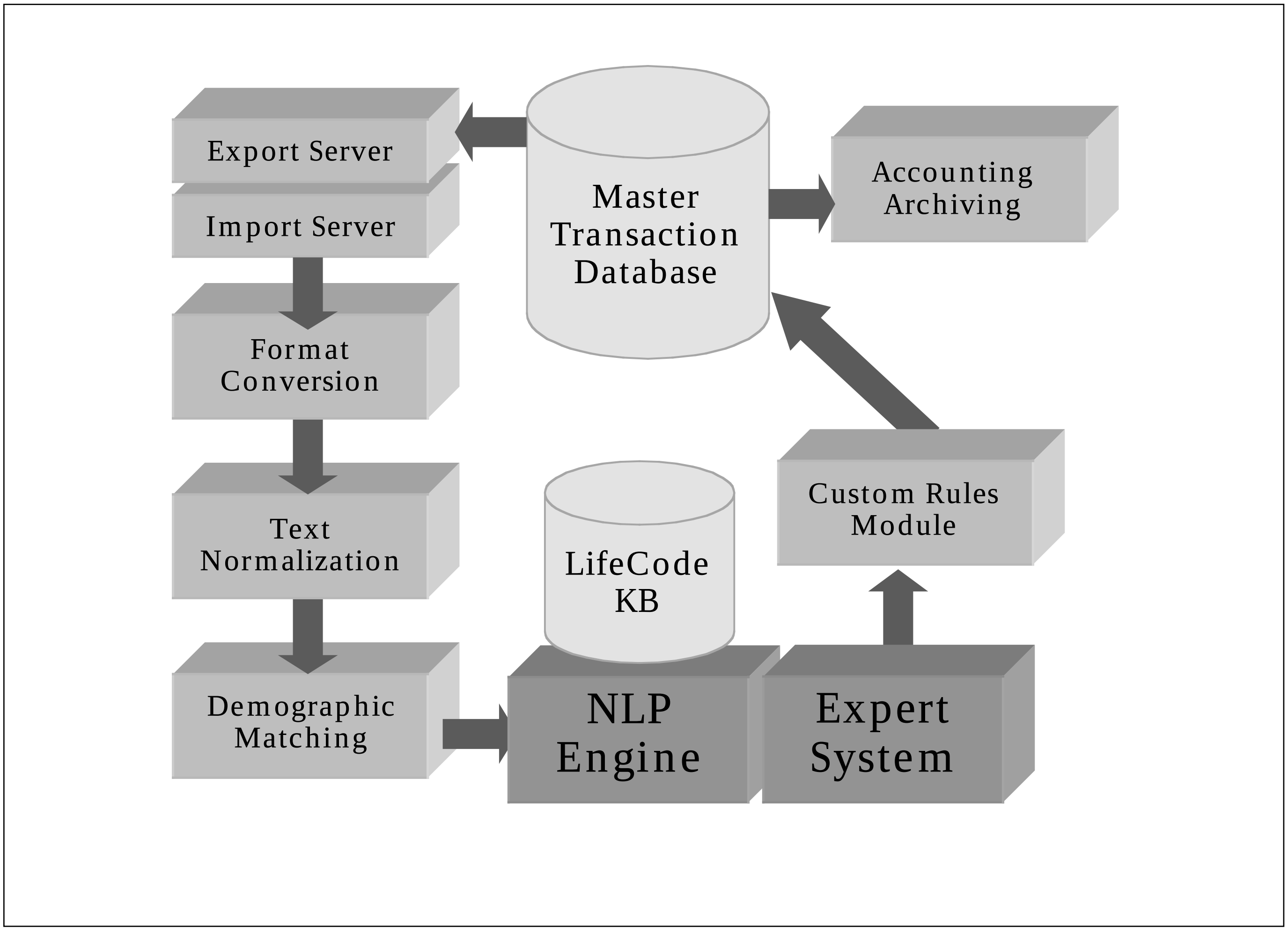


Figure 4. Data Center Work Flow.

semiknowledge; that is, LifeCode has, beyond its core knowledge, a broad but shallow knowledge of application-relevant aspects of medicine. This semiknowledge enables LifeCode to distinguish between relevant and irrelevant information and further distinguish medical information that is within its expertise from that which is outside its expertise.

The core LifeCode engine is wrapped in an industrial-strength data center that controls local and remote input-output, document reformatting, database storage and archival, version control, question-answer review, user interfaces, and accounting. The flow of a transaction through the data center is shown in figure 4. Within the medical applications that we have approached, LifeCode is patent pending as a top-level business-process method. At the NLP system level, it is patent pending in terms of its organization and approach to NLP. Finally, at the algorithm level, the high-speed dynamic programming and the semiknowledge algorithms are patent pending.

Test and Validation

As expected, the market is skeptical of disruptive technologies such as LifeCode. The medical establishment has been particularly resistant to technology (Christensen, Bohmer, and Kengy 2000), so it is particularly incumbent upon us to validate the accuracy of LifeCode. Because of the nature of medicine and the long and often convoluted history of development behind the common medical-coding systems and the vast body of legislative interpretation and regulation imposed by the government and private payers (that is, the Health Care Finance Administration [HCFA], the regulatory body behind Medicare, and the private health insurance companies), medical coding is a strange mixture of science, art, and folklore. As such, there is no gold standard for the coding of any particular medical document. In this light, we have developed a test and validation method that is based on a panel of experts. For discrete events that can be coded simply as the presence of

									Consensus	Consensus					
	A	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**Agreement**	**Kappa**	**-2**	**-1**	**+1**	**+2**	**RVU**
A (Expert)	*	0.35	0.24	0.31	0.34	0.43	0.51	0.34	*0.73*	0.57	3	19	4	0	2.00
B (LifeCode)	*0.58*	*	0.24	0.30	0.36	0.33	0.36	0.28	*0.71*	0.51	0	13	12	0	2.13
C (Standard Billing)	*0.51*	*0.55*	*	0.28	0.57	0.37	0.27	0.30	*0.72*	0.57	3	18	4	0	2.17
D (Expert)	*0.51*	*0.52*	*0.49*	*	0.30	0.45	0.40	0.37	*0.68*	0.46	3	12	11	0	2.14
E (Expert)	*0.55*	*0.63*	*0.72*	*0.50*	*	0.44	0.38	0.30	*0.78*	0.63	0	7	11	0	2.18
F (Premium Billing)	*0.58*	*0.53*	*0.55*	*0.57*	*0.65*	*	0.33	0.34	*0.69*	0.54	0	2	18	1	2.36
G (Standard Billing)	*0.67*	*0.58*	*0.54*	*0.58*	*0.59*	*0.53*	*	0.38	*0.71*	0.54	3	18	2	0	1.95
H (Expert/Auditor)	*0.51*	*0.49*	*0.52*	*0.55*	*0.48*	*0.53*	*0.54*	*	*0.59*	0.42	3	21	8	0	1.99

Table 1. A–H Interrater Agreement (Italics) and Interrater Kappa (Roman), Agreement with the consensus, Kappa with the Consensus, Number of Charts Deviating –2/–1/+1/+2 from Consensus and average relative value units for E/M levels.

some health condition or the performance of some medical procedure, we rate code differences between LIFECODE and the expert humans (broadly speaking and in descending order of severity) as false positives, false negatives, and specificity (that is, the code is for the right medical condition or procedure but is either a more or less specific version as compared to the "correct" code). For composite events, such as evaluation and management codes, where the level of agreement between even the experts dips seriously and where the concepts of false positive-negative do not directly apply, we resort to Kappa-statistics and *F*-statistics. The evaluation and management codes (a five-level scale of codes) account for about 80 percent of the reimbursement of emergency and primary care physicians and are a prime target for fraud and abuse investigations. However, these codes are assigned on an array of as many as 100 discrete and often subjective assessments that must be made about each medical document, for example, what the risk to the patient was based on the presenting condition(s). The Kappa-statistic is calculated by dividing the difference between observed and random agreement by the maximum possible difference between observed and random agreement:

$$K = \frac{P_0 - P_r}{1 - P_r}$$

where P_0 is the observed agreement, P_r is the random agreement, and 1 is the maximum observed agreement. A Kappa value less than 0.40 indicates poor agreement. Kappa between 0.40 and 0.70 indicates fair to good agreement. Kappa above 0.70 indicates strong agreement. The *F-distribution* is a measure of the difference in degree of diversity between two randomly sampled sets of data (Freund 1980). The *F*-statistic is an estimate of this distribution and is computed as the ratio of variances:

$$F = \frac{S_1^2}{S_2^2}$$

where S_1^2 and S_2^2 are the variances of the respective samples. Separate collections of documents coded according to similar evaluation and management coding policies exhibit similar variances in their evaluation and management levels. An *F*-statistic much greater or much lesser than 1 might indicate a shift in coding policy.

To validate LIFECODE's performance on coding, we participated in a seven-way blind study against human coders (Morris et al. 2000). The coders represented various levels of capability and various regions of the country. All coded the same set of documents to a specified standard under rigid time and resource control. As can be noted from table 1, the pairwise agreement between coders is very poor for evaluation and management coding, and the Kappa rarely exceeds a fair correlation. The agreement between each coder and the consensus evaluation and management code for each document is considerably better but is still not high. However, when the agreement criteria is relaxed from exact match to within one level (that is, change the scale from 1, 2, 3, 4, 5 to 1/2, 2/3, 3/4, 4/5), all participants (including LIFECODE) had a nearly perfect (.98) correlation with the consensus codes. Considering that determining the evaluation and management level is based on a series of single-dimension absolute judgments that are combined according to a set of rules (Wickens 1992), the observed agreement is as good as can be expected for a cognitive task of this nature. In this test, the NLP coding was statistically indistinguishable from human coders. A human auditor, who knew only that one of the coders was a computer, was also not able to distinguish LifeCode from the human coders.

This level of coding quality must be maintained on a daily basis for a large customer base. For production purposes, sequential sampling and correlation of the LifeCode distribution of assigned codes to the payers' statistically predicted distribution of codes using the F-statistic are used as part of the overall quality control system. This attention to test and validation has demonstrated that LifeCode is equal or superior to human coders in terms of accuracy and is far more predictable and reliable. Another advantage of LifeCode is that because of the "glass box" (that is, a system for which the rationale for selections can be observed and understood—as contrasted with "black box," for example, a neural net) nature of the system, errors and omissions (subject only to repayment of inaccurate charges) can be distinguished from fraud and abuse (subject to repayment and a $10,000 penalty for each miscoded record). The ability to understand the rationale for code choices is a major consideration in an atmosphere of extensive fraud-abuse investigation and prosecution by the Office of the Inspector General (OIG) and the HCFA.

Application Use and Payoff

A-Life completed a successful testing program of the first application for the coding of emergency medicine at two billing company sites in 1998. Full commercial operations using the LifeCode system started in July 1998. In October 2000, a version of LifeCode for radiology report coding completed beta testing and entered production at three beta test sites. A-Life's solutions for emergency medicine and diagnostic radiology are used by billing companies and providers (hospitals and health centers) to completely automate daily coding operations for the majority of medical charts. LifeCode codes 60 to 70 percent of documents with no human intervention. The remaining documents are coded as completely as possible and categorized as either requiring additional quality assurance review or as incomplete charts because of documentation deficiencies. Of the charts sent for additional quality assurance review, about two-thirds are already coded correctly and completely and require no further changes. From a statistical standpoint, this seemingly high review level is needed to keep LifeCode's false-positive rate below 1 percent for billable procedures (observed false-positive rates for human coders in production settings is 2 to 4 percent).

The payoffs and benefits for using LifeCode can be summarized as (1) significant overall reduction in medical coding costs because of enhanced productivity; (2) far more accurate, consistent, and complete assignment of codes than is humanly possible; (3) more efficient operations by reducing a large labor force that is difficult to recruit and retain; (4) greatly increased uniformity and validity of codes assigned and data produced; (5) elimination of coding inconsistency typically found with manual processes; (6) a major asset in developing in-house compliance programs; (7) reduction of accounts receivable cycle because of faster turnaround, decreased error rate, and fewer payer rejections; (8) an audit trail showing coding logic matched with coding results, stored for use during a payer audit; (9) compliance guaranteed—HCFA-compliant coding reduces risk of fines for fraud and abuse; and (10) a competitive advantage for customers allowing them to expand their sales.

Other benefits that will accrue in time from the use of LifeCode are (1) electronic data availability-retrieval, which allows for utilization review, clinical protocol analysis, and process enhancement for billing and claims submission, and (2) instant feedback to physicians on the quality of documentation, thus improving patient care and optimization of accurate, allowable reimbursement.

Positive operational effects for the users of LifeCode include (1) by automating the medical coding task, the ability of the human coders to focus on tasks that require human expertise, such as quality control, review of difficult documents, and physician education; (2) optimization of existing staff, overall reduction of staff, and reduced costs for hiring and training; (3) reduction of paper flow and reduced storage costs; (4) assistance for customers with operational, statistical, and clinical reports in better managing operations.

Application Development and Deployment

The development of LifeCode began with the founding of A-Life Medical, Inc., in February 1996. The research and development department started with 2 part-time employees and has grown to 11 full-time individuals and occasional student interns. The group is composed of five AI software experts, five linguists (all computationally oriented), and one knowledge engineer. Additionally, the company has grown to include medical specialty experts both as employees and as regular consultants. The research and development group has also been aided greatly by our beta customers. The application infrastructure was developed by our engineering department that currently consists of six software engineers, three systems administrators, and four installation engineers. Additionally, the client services department provides both software and product quality control as well as domain expertise. Finally, our marketing staff has contributed in terms of market-driven requirements and expectations. The development time, to date, in the research and development department has been close to 40 person-years. The time contributed by other departments within A-Life and by our beta customers would easily exceed this number. The development methodology for research and development has been iterative, leading to an organic growth of the core product. The application infrastructure was developed with a standard design-build-test approach with version control. We are now at the point where mature portions of the core technology are being transferred from research and development to engineering, where they will be refined based on lessons learned in the initial development phase.

During the initial development phase, the two greatest difficulties were the rapidly changing regulations governing clinical documentation and the widespread uncertainty within the medical community about how to respond to these changes. Both the changes and the growing complexity of the regulations (driven primarily by HCFA and secondarily by private insurers) have been both a bane and a blessing: a bane in that they have made it far more difficult to produce a product that can deal with the complexity and a blessing in that it is increasingly difficult for humans to deal with the regulations and so automation has become very appealing in the marketplace. It can be expected that this duality will exist in any highly regulated market. The lesson is to be prepared for the unavoidable drain on capital and time as well as the risk of being regulated out of business.

A further deployment issue has been market acceptance. LIFECODE is significantly different from anything else that has been in use in the medical-coding marketplace, and users are predictably skeptical. A quality product that meets a real need and has staying power is necessary to penetrate such a market. As of the time of writing, LIFECODE is gaining market acceptance. The pathway to acceptance led through small, enterprising billing companies such as Applied Medical Systems in Durham, North Carolina, to large, prestigious clients such as Louisiana State University Health Sciences Center in Shreveport, Louisiana, and MedAmerica in Oakland, California. However, direct sales alone do not make up the whole story. In the long run, industry partners will make up the largest part of the business for a specialty product such as LIFECODE. As with the direct sales, these partnerships began with joint selling agreements with small medical records companies such as ER Records in Irving, Texas, and go to full original equipment manufacturer (OEM) relationships with health information systems and service vendors such as MedQuist, Siemens/Shared Medical Systems, and L&H/Dictaphone. The goal of A-Life Medical is to provide the medical NLP component for coding and abstraction in all major HIM systems. At the time of writing, we have contractual relations to this end with about 75 percent of our target HIM vendors. It is the acceptance by, and diversity of, both direct customers and OEMs that ensures the success of LIFECODE.

System Maintenance

After the initial deployment of the LIFECODE NLP engine in a production environment, the maintenance and subsequent development of the core knowledge bases is real-world data driven. A cycle of feedback and maintenance is an integral part of the system. The first source of these data is analysis of the free-text, physician-dictated medical record. The second, and equally important, source of data is quality assurance and customer use of the system. LIFECODE's self-awareness feature routes certain medical records to human experts who "fix" the coding of the record. Targeted comparison analyses allow linguists and software engineers to iteratively improve the accuracy of the system. Knowledge bases and software algorithms are continually refined to better match the language used by the physician and the domain knowledge elicited from professional medical coders.

As medical specialties are added, knowledge bases are created and a cycle of maintenance and "natural language adaptation" is used to adjust to phrasings employed by physicians in these specialties. Within specialties, coding knowledge is currently in a state of flux, and LIFECODE must be regularly updated to reflect this dynamic environment. Medical coding is affected by changes in the practice of medicine; yearly updates of codes; and major, but less frequent, changes in coding guidelines. LIFECODE's unique design permits independent editing of source code, knowledge bases, and the expert coding system. Linguists and software engineers with differing areas of expertise can contribute to improving the system without being limited by their individually varying knowledge of programming, linguistics, or the intricacies of coding.

We are currently using or developing learning techniques for paragraph classification, factor analysis, and vocabulary acquisition. For coding-specific knowledge, LIFECODE currently does not use learning techniques because changes in medical codes and policies must be imparted to the system prior to the existence of any real-world data by which learning could be driven. Also, for purposes of compliance, it is necessary to have a system that can precisely be audited in terms of why and how a particular decision was made. We believe, however, that automated learning techniques are rightly applied as an aid to dealing with variations in language use between physicians and the canonical definitions of diagnoses and medical procedures that are published in the various medical coding and vocabulary standards. We are currently developing a knowledge base that will complete LIFECODE's knowledge of canonical definitions for the diagnosis and procedure codes across all medical disciplines. We are using learning techniques to flesh out the canonical skeleton with the language that is actually used by physicians in clinical practice.

Conclusion

LIFECODE advances the state of the art in NLP along several lines. Its architecture brings together a number of NLP and expert systems technologies in a coherent commercial product. At the algorithm level, it represents a step forward in terms of high processing speed with very large linguistic knowledge bases. Also, its self-awareness capability is a necessity for system output to be used without human intervention on every decision and is, to our knowledge, unique among NLP applications. Finally, as a method for doing business, LIFECODE has the potential to significantly influence the future course of health information management. Given the current growth in direct sales and partnerships, the future for LIFECODE is bright. Automation of medical coding will soon move from nicety to necessity.

Note

1. D. B. Aronow and F. Feng, 1997. Ad Hoc Classification of Electronic Clinical Documents. D-Lib Magazine, January. Available at www.dlib.org/dlib/january97/medical/olaronow.html.

References

AMA. 1999. Current Procedural Terminology: CPT 2000. American Medical Association, Chicago, Illinois.

Aronow, D. B., and Shmueli, A. 1996. A PC Classifier of Clinical Text Documents: Advanced Information Retrieval Technology Transfer. In *Proceedings of the American Medical Informatics Association Fall Symposium*, 932ff. Philadelphia, Pa.: Hanley and Belfus.

Aronow, D. B.; Cooley, J. R.; and Sonderland, S. 1995. Automated Identification of Episodes of Asthma Exacerbation for Quality Measurement in a Computer-Based Medical Record. Technical Report, IR-61, Center for Intelligent Information Retrieval, University of Massachusetts at Amherst.

Aronow, D. B.; Feng, F.; and Croft, W. B. 1999. Ad Hoc Classification of Radiology Reports. *Journal of the American Medical Informatics Association* 6(5): 393–411.

Aronow, D. B.; Sonderland, S.; Ponte, J. M.; Feng, F.; Croft, W. B.; and Lehnert, W. G. 1995. Automated Classification of Encounter Notes in a Computer-Based Medical Record. Technical Report, IR-67, Center for Intelligent Information Retrieval, University of Massachusetts at Amherst.

Bentley, J. 1996. *The Impossible Takes a Little Longer. Unix Review* 14(12): 75–79.

Christensen, C. M.; Bohmer, R.; and Kengy, J. 2000. Will Disruptive Innovations Cure Health Care? *Harvard Business Review* 78(6): 102ff.

Croft, W. B.; Callan, J. P.; and Aronow, D. B. 1995. Effective Access to Distributed Heterogeneous Medical Text Databases. In *Proceedings of MEDINFO 95*, 1719ff. Amsterdam: IOS.

Freund, J. E., and Walpole, R. E. 1980. *Mathematical Statistics*. Englewood Cliffs, N.J.: Prentice Hall.

Friedman, N.; Geiger, D.; and Goldszmidt, M. 1997. Bayesian Network Classifiers. *Machine Learning* 29:131–163.

Hirsch, M., and Aronow, D. B. 1995. Suggesting Terms for Query Expansion in a Medical Information Retrieval System. Technical Report, IR-63, Center for Intelligent Information Retrieval., University of Massachusetts at Amherst.

Langacker, R. W. 1999. Explanation in Cognitive Linguistics and Cognitive Grammar. Paper presented at the Conference on the Nature of Explanation in Linguistic Theory, 3–5 December, San Diego, California.

Larkey, L. S., and Croft, W. B. 1995. Automatic Assignment of ICD9 Codes to Discharge Summaries. Technical Report, IR-64, Center for Intelligent Information Retrieval, University of Massachusetts at Amherst.

Lehnert, W.; Sonderland, S.; Aronow, D. B.; Feng, F.; and Smith, A. 1994. Inductive Text Classification for Medical Applications. Technical Report, TC-32, Center for Intelligent Information Retrieval, University of Massachusetts at Amherst.

Lenert, L. A., and Tovar, M. 1993. Automated Linkage of Free-Text Descriptions of Patients with Practice Guidelines. In Proceedings of the Symposium on Computer Applications in Medical Care, 274–278.

Washington, D.C.: IEEE Computer Society.

Medicode. 1999. *Physician ICD-9-CM: International Classification of Diseases. 9th rev. Clinical Modification.* 5th ed. Salt Lake City, Utah: Medicode.

Morris, W. C.; Heinze, D. T.; Warner Jr., H. R.; Primack, A.; Morsch, A. E. W.; Sheffer, R. E.; Jennings, M. A.; Morsch, M. L.; and Jimmink, M. A. 2000. Assessing the Accuracy of an Automated Coding System in Emergency Medicine. In CD-ROM Proceedings of the American Medical Informatics Association 2000. Bethesda, Md.: The American Medical Informatics Association.

Ranum, D. L. 1988. Knowledge-Based Understanding of Radiology Text. In Proceedings of the Symposium on Computer Applications in Medical Care, 141–145. Washington, D.C.: IEEE Computer Society.

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

Sager, N.; Lyman, M.; Nhan, N. T.; and Tick, L. J. 1994. Automatic Encoding into SNOMED III: A Preliminary Investigation. In Proceedings of the Symposium on Computer Applications in Medical Care, 230–234. Washington, D.C.: IEEE Computer Society.

Sager, N.; Nhan, N. T.; Lyman, M. S.; and Tick, L. J. 1996. Medical Language Processing with SGML Display. In Proceedings of the 1996 AMIA Annual Fall Symposium, 547–551. Philadelphia, Pa.: Hanley and Belfus.

Sneiderman, C. A.; Rindfleisch, T. C.; and Aronson, A. R. 1996. Finding the Findings: Identification of Findings in Medical Literature Using Restricted Natural Language Processing. In Proceedings of the American Medical Informatics Association Fall Symposium, 239–243. Philadelphia, Pa.: Hanley and Belfus.

Sneiderman, C. A.; Rindfleisch, T. C.; Aronson, A. R.; and Browne, A. C. 1995. Extracting Physical Findings from Free-Text Patient Records. In CD-ROM Proceedings of the American Medical Informatics Association Spring Congress. Philadelphia, Pa.: Hanley and Belfus.

Sonderland, S.; Aronow, D. B.; Fisher, D.; Aseltine, J.; and Lehnert, W. 1995. Machine Learning of Text Analysis Rules for Clinical Records. Technical Report, TC-39, Center for Intelligent Information Retrieval, University of Massachusetts at Amherst.

Wickens, C. D. 1992. *Engineering Psychology and Human Performance.* New York: Harper Collins.

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

Zingmond, D., and Lenert, L. A. 1993. Monitoring Free-Text Data Using Medical Language Processing. *Computers and Biomedical Research* 26:467–481.

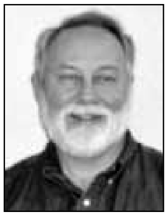
Daniel T. Heinze is cofounder and chief technology officer of A-Life Medical, Inc. He was previously co-director of the Raytheon/ Pennsylvania State University Center for Intelligent Information Processing. He holds a Ph.D. from Pennsylvania State University and a B.S. and an M.S. in computer science from the New Jersey Institute of Technology. He also holds a B.A. in humanities and language from Bob Jones University and an M.Div. from the Biblical Theological Seminary. His e-mail address is dheinze@alifemedical.com.

Mark L. Morsch is the director of engineering and information systems for A-Life Medical, Inc. He was previously on staff at the Raytheon/ Pennsylvania State University Center for Intelligent Information Processing. He holds an M.S. and a B.S. in electrical engineering from Clarkson University. His research interests include text mining, knowledge representation, and the blending of symbolic and connectionist models of AI. His e-mail address is mmorsch@alifemedical.com.

Ronald E. Sheffer, Jr., is a senior research linguist for A-Life Medical, Inc. He received his B.A. in linguistics, with a minor in German, from the University of California at Berkeley and his M.A. in linguistics from the University of California at San Diego. Previously, he worked on the Cognitive Modalities Project at International Neural Machines in Waterloo, Canada. His research interests include cognitive grammar and cognitive linguistics in general as well as computational application of these theories. His e-mail address is rsheffer@ alifemedical.com.

Michelle Jimmink is a research linguist for A-Life Medical, Inc. She received her B.A. in linguistics from the University of California at San Diego in 1998. Her research interests include discourse analysis and natural language. Her e-mail address is mjimmink@alifemedical.com.

Mark A. Jennings is a senior research engineer at A-Life Medical, Inc. He was in the Cognitive Science Program at Indiana University and received his M.S. in computer science in 1994. He received a B.A. in psychology in 1992 from Grove City College, Pennsylvania. Before A-Life, he worked at the Raytheon/Pennsylvania State University Center for Intelligent Information Processing developing natural language processing technologies for web searching and information extraction. His e-mail address is mjennings@alifemedical.com.



William Morris is a senior research linguist at A-Life Medical Inc. He received a Ph.D. in cognitive science and linguistics from the University of California at San Diego. His research interests include natural language processing, connectionist networks, first-language acquisition, and the properties of grammatical relations in syntactically ergative and semiergative languages. His e-mail address is bmorris@alifemedical.com.

Amy E. Walter Morsch is a senior knowledge engineer at A-Life Medical, Inc. She received her Ph.D. in biophysical chemistry from the University of Rochester in 1994. She has worked for A-Life Medical since 1996. Her interests include the areas of medical informatics and bioinformatics. Her e-mail address is amorsch@alifemedical.com.