LifeCode ä - A Natural Language Processing System for Medical Coding and Data Mining

Daniel T. Heinze Ph.D., Mark L. Morsch, Ronald E. Sheffer, Jr., Michelle A. Jimmink, Mark A. Jennings, Willam C. Morris Ph.D., Amy E. W. Morsch Ph.D.

A-Life Medical, Inc.
9555 Chesapeake Drive – Suite 101
San Diego, California 92123
dheinze, mmorsch, rsheffer, mjimmink, mjennings, bmorris , amorsch@alifemedical.com

Abstract

LifeCode™ (patent pending) is a Natural Language Processing 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 is now in beta-test. A pilot program for performing data mining on acute care clinical records has been completed. The LifeCode NLP engine uses a large number of specialist readers whose particular outputs 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 billing 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 are wrapped as DCOM servers and reside in various delivery packages including On-Line Transaction Processing (OLTP), a web-browser interface and an Automated Speech Recognition (ASR) interface.

Problem and Task Description

LifeCode™ (patent pending) 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 per year business and generates in excess of seven hundred million clinical documents in transcribed free-text form. Viewing medicine as a business, the clinical information in the free-text records has a necessary application in producing a bill for services and facility utilization. This is the realm of medical coding and billing. Another desirable business application of the information is tracking physician performance and resource utilization. From the clinical perspective, the information in the clinical notes can be used to improve communications between multiple providers for the same patient, to monitor the efficacy of alternate courses of treatment and to provide feedback and alerts relative to the course of care for a particular patient.

Although the Electronic Medical Record (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 due to 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 due to 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.

Other programs for NLP of 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; Aronow and Feng, 1997; Croft, Callan and Aronow, 1995; Hirsch and Aronow, 1995; Lenhert, et. al., 1994; Sonderland, et.al., 1995). 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 has the patient been ill) or timing (how frequent are the bouts of pain). They do not discern negation (the patient was diagnosed not to have the illness under discussion). The list goes on, but these examples should be sufficient. Medical word and phrase tagging systems operate at a much more granular level to apply tags that disambiguate semantic senses (Sager, et.al., 1994a; Sager, et.al, 1996). They would discern, for example, between the verbal use of "burning" (e.g. the

flame was burning the patients finger) and the adjectival use (e.g. the patient had a burning substernal chest pain). Tagging does not in 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, etc.) typically have not dealt with the issues of role, negation, timing, etc. (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, et.al. 1994b), 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 either in specific medical specialties such as radiology (Ranum, 1988; Zingmond and Lenert, 1993) or in general medical texts (Sneiderman, et. al., 1995; Sneiderman, Rindflesch and Aronson, 1996) typically lack both the specificity (in terms of roles, temporality, etc.) 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 per 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 (e.g. 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 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 in order to maintain the appropriate quality level. We refer to this ability as "self-awareness".

LifeCode provides both linguistic competence and medical knowledge and logic to:

- Use NLP to extract from a free-text clinical note…
  - the patient demographics (name, age, gender, etc),
  - the patient's chief complaint,
  - 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, etc.),
  - the medical history of the patient and (as applicable) the patient's family,
  - relevant social history (use of tobacco, alcohol and drugs, living arrangements, etc.)
  - the nature and extent of the physical examination performed by the physician,
  - the nature and extent of old records consulted, professional consultations and medical tests performed by the physician,
  - the final diagnoses, potentially also including possible and ruled-out diagnoses,
  - the course of treatment including surgical procedures, drug therapy and monitoring levels, and
  - the disposition of the patient at the end of the clinical encounter with the physician.
- Use an Expert System to determine from the extracted information…
  - the most specific version of each diagnosis and procedure,
  - the level of documentation of the history and physical examination,
  - the risk to the patient presented by the medical condition and treatment,
  - the complexity of the medical decision making for the physician,
  - the level of service provided by the physician, and
  - the appropriate manner 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++ together with several finite-state and context sensitive processors. This top layer contains two modules, the NLP extraction engine and 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 an SQL 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 contains the medical vocabulary; definitions covering anatomy, microbiology, medications, signs, symptoms, diagnoses, and procedures; and rules for medical coding. This data (and more) comprises the

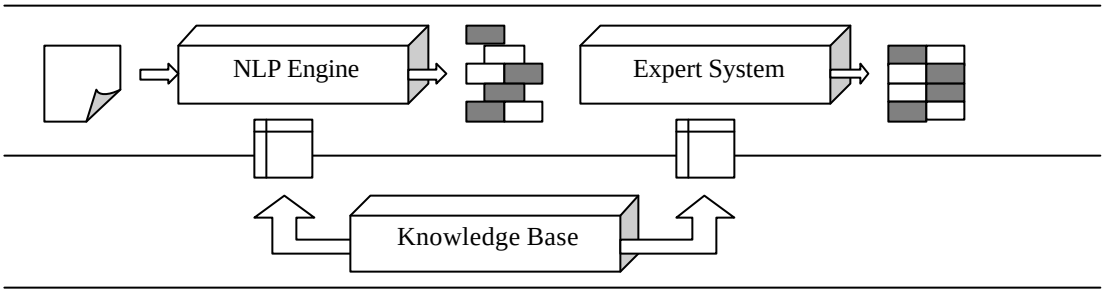

Figure 1. LifeCode Architecture

knowledge base and is 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 executable modules. 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 techniques - 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: document segmenter, lexical analyzer, phrase parser, and 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. These include 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 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 appear in the original note. As we will discuss later on, knowing the context of each data element is required in order 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. This functionality includes 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. This 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 all of the tokens have been generated by the lexical analyzer, the phrase parser performs a bottom-up syntactic analysis. The parser is highly resilient and tolerant of the incorrect grammar that characterizes clinical documents and unknown words. The information needed for medical coding is expressed primarily 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 up to a complete sentence, that 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 upon 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. Consider a dictionary with four ICD-9 codes:

786.50 Chest pain unspecified
786.51 Substernal chest pain
786.52 Chest wall pain
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): chest, pain, unspecified, substernal, wall, and 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 that 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 above, 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. This includes using anatomy, medication, and microbiology concept hierarchies and synonym lists to improve chances of a match. Also, syntactic heuristics may be applied. These heuristics join and redistribute words from two or more consecutive phrases that were divided by the phrase parser and compute the meaning for the newly combined phrase. With meanings assigned to individual chunks of text, the extracted data elements are formed by collecting all of 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 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 may 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) may 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 via ftp or dial-up connections. The notes are held for a period of time until payor demographics and addenda have been received. From there, LifeCode runs on the documents with the results stored in an SQL database. The document, medical codes, and payor demographics are returned to the client electronically, and their 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 brain-child 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) 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, e.g. the assessment of duration, ability to shift attention to different narrators and events, a sense of urgency, etc. In terms of the core AI components, there is little in LifeCode that has not been available in NLP work for some time. This includes 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, rather it is 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 fifty thousand per sentence. In addition to techniques such as caching, LifeCode employs a novel dynamic programming technique that is, to the best of our knowledge, on the order of ten 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 500MHz Pentium PC running Windows NT) is able to run a knowledge base with well more than three million entries against a four hundred word document in ten to twenty 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. This would require that a human expert review all of 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 semi-knowledge. That is, LifeCode has, beyond it's core knowledge, a very broad but shallow knowledge of application relevant aspects of medicine. This semi-knowledge enables LifeCode to distinguish between relevant and irrelevant information, and further between medical information that is within its expertise versus that which is outside its expertise.

The core LifeCode engine is wrapped in an industrial strength data center that controls local and remote I/O, document reformatting, database storage and archival, version control, QA review, user interfaces and accounting. 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. And, at the algorithm level, the high-speed dynamic programming and the semi-knowledge algorithms are patent pending.

Application Use and Payoff

A-Life has over 400 hospitals under contract of which 26 hospitals have been implemented as of the date of writing (January 2000). A-Life completed a successful testing program of the first application for the coding of emergency medicine at two billing company sites in early 1998. Full commercial operations started in July 1998. A-Life's solutions for Emergency Medicine and Radiology are used by billing companies and providers (hospitals and health centers) to completely automate daily coding operations for the vast majority of medical charts. LifeCode codes approximately 70% of documents with no human intervention. The remaining documents are coded as completely as possible and categorized as either requiring additional QA review or as incomplete charts due to documentation deficiencies. Of the charts sent for additional QA review, about half are already coded correctly and completely and require no further changes. From a statistical standpoint, this seemingly high review level is needed in order to keep LifeCode's false-positive rate below 1% for billable procedures (observed false-positive rates for human coders in production settings is 2% to 4%).

The payoffs and benefits for using LifeCode can be summarized as:

- significant overall reduction in medical coding costs via enhanced productivity;
- far more accurate, consistent and complete assignment of codes than is humanly possible;
- more efficient operations by reducing a large labor force that is difficult to recruit and retain;

- greatly increased uniformity and validity of codes assigned and data produced;
- elimination of coding inconsistency typically found with manual processes;
- a major asset in developing in-house compliance programs;
- reduction of accounts receivable cycle due to faster turnaround, decreased error rate and fewer payer rejections;
- an audit trail showing coding logic matched with coding results, stored for use during a payer audit;
- compliance guaranteed – HCFA-compliant coding reduces risk of fines for fraud and abuse; and
- a competitive advantage for customers allowing them to expand their sales.

Other benefits that will accrue in time from the use of LifeCode are:

- Electronic data availability/retrieval allows for utilization review, clinical protocol analysis and process enhancement for billing and claims submission.
- 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:

- By automating the medical coding task, human coders are able to focus on tasks that require human expertise such as quality control, review of difficult documents, and physician education.
- Optimization of existing staff, overall reduction of staff and reduced costs for hiring and training.
- Reduction of paper flow and reduced storage costs.
- Operational, statistical and clinical reports assist customers 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 R&D department started with two part-time and has grown to now be seven full-time individuals. The group is composed of three AI software experts, three 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 R&D group has also been aided greatly by our beta-customers. The application infrastructure was developed by our Information Systems department that currently consists of six software engineers, two systems administrators and two installation engineers. Finally, our marketing staff has contributed in terms of market driven requirements and expectations. The development time, to date, from the R&D department has been close to twenty person-years. The time contributed by other departments within A-Life and by our beta-customers would easily exceed that number. The development methodology for R&D has been iterative thus 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 R&D to IS where they will be reimplemented 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 as to how to respond to these changes. Both the changes and the growing complexity of the regulations (driven primarily by the Health Care Financing Administration (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 market place. It can be expected that this duality, with the attendant banes and blessings, 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 staying power are both necessary to penetrate such a market. As of the time of writing, LifeCode is beginning to enjoy the rewards of widespread market acceptance. The pathway to acceptance led through small, enterprising billing companies such as Applied Medical Systems of Durham, North Carolina to large, prestigious clients such as Louisiana State University Health Sciences Center in Shreveport, Louisiana and MedAmerica in Oakland, California. But 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 range to full OEM relationships with health information systems and services giants such as Dictaphone and MedQuist. Even larger partnerships are in the negotiation stage. It is this diversity of both direct customers and OEMs that will ensure the acceptance and success of LifeCode.

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 this data is analysis of the free-text, physician-dictated medical record. The second, and equally important, source of data is QA 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 very much in a state of flux and LifeCode must be regularly updated to reflect the dynamic nature of this. This includes changes in the practice of medicine and the effects this has on medical coding, 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, knowledge engineers, and software engineers with differing areas of expertise may contribute to improving the system without being limited by their individually varying knowledge of programming, linguistics, or the intricacies of coding.

Currently LifeCode 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 be precisely audited in terms of why and how a particular decision was made. We believe, however, that in the future, automated learning techniques could be applied as an aid to dealing with variations in language use between physicians.

Conclusion

LifeCode advances the state-of-the-art in NLP along several boundaries. 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 medical records management. Given the current growth in direct sales and partnerships, the future for LifeCode is bright. Automation of medical coding and data mining will soon move from nicety to necessity.

References

AMA. 1999. *Current Procedural Terminology: CPT 2000*. American Medical Association.

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

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

Aronow, D. B., Shmueli, A. 1996. A PC Classifier of Clinical Text Documents: Advanced Information Retrieval Technology Transfer. *Proceedings – American Medical Informatics Association Fall Symposium.* 932ff.

Aronow, D. B., Feng, F. 1997. Ad-Hoc Classification of Electronic Clinical Documents. *D-Lib Magazine*, January.

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

Bentley, J. 1996. The Impossible Takes a Little Longer. *Unix Review*, December, 75-79.

Croft, W. B., Callan, J. P., Aronow, D. B. 1995. Effective Access to Distributed Heterogeneous Medical Text Databases. *Proceedings – MEDINFO 95*. 1719ff.

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

Langacker, R. W. 1999. Explanation in Cognitive Linguistics and Cognitive Grammar – Seminar hand-out

UCSD Department of Linguistics Conference on The Nature of Explanation in Linguistic Theory. University of California at San Diego. December 3-5, 1999.

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

Lenert, L. A., Tovar, M. 1993. Automated Linkage of Free-text Descriptions of Patients with Practice Guidelines. *Proceedings – Symposium on Computer Applications in Medical Care.* 274-278. New York: Institute of Electrical and Electronics Engineers.

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

Medicode. 1999. *Physician ICD-9-CM: International Classification of Diseases, 9th Revision, Clinical Modification*. Fifth Edition.

Ranum, D. L. 1988. Knowledge Based Understanding of Radiology Text. *Proceedings – Symposium on Computer Applications in Medical Care*. 141-145. New York: Institute of Electrical and Electronics Engineers.

Sager, N., Lyman, M., Nhan, N. T., Tick, L. J. 1994. Automatic Encoding into SNOMED III: A Preliminary Investigation. *Proceedings – Symposium on Computer Applications in Medical Care.* 230-234. New York: Institute of Electrical and Electronics Engineers.

Sager, N., Lyman, M., 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., Nhan, N.T., Lyman, M.S., Tick, L.J. 1996. Medical Language Processing with SGML Display. *Proceedings of the 1996 AMIA Annual Fall Symposium.* 547-551. Hanley & Belfus.

Sneiderman, C. A., Rindflesch, T. C., Aronson, A. R., Browne, A. C. 1995. Extracting Physical Findings from Free-Text Patient Records. *Proceedings – American Medical Informatics Association Spring Congress*.

Sneiderman, C. A., Rindflesch, T. C., Aronson, A. R. 1996. Finding the Findings: Identification of Findings in Medical Literature Using Restricted Natural Language Processing. *Proceedings – American Medical Informatics Association Fall Symposium.* 239-243.

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

Yang, Y., Chute, C.G. 1992. An Application of Least Squares Fit Mapping To Clinical Classification. *Proceedings – Symposium on Computer Applications in Medical Care.* 460-464. New York: Institute of Electrical and Electronics Engineers.

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