Delving into Computer-assisted Coding

Save to myBoK

This practice brief has been updated. See the latest version here. This version is made available for historical purposes only.

This practice brief discusses computerized tools available to automate the assignment of certain medical or surgical codes (ICD-9-CM and CPT/HCPCS) from clinical documentation that are traditionally assigned by coding or HIM professionals as well as clinical providers. It also outlines the driving forces that are shaping the current and future applications of this technology, examines application of the technology, and provides guidance about the steps necessary to position coding professionals for the coming coding revolution. AHIMA chartered the computer-assisted coding e-HIM™ work group to help healthcare organizations navigate and understand how to prepare for and thrive in a profoundly changing work environment.

Background

The healthcare industry is creating powerful tools to transform clinical data input into useful clinical data output. Clinical coding is approaching a tipping point where an increasing amount of work is done by machine, saving precious time and human resources for more complex coding and much needed data analysis tasks.

Many factors directly influence this change, including advances in natural language processing and informatics, adoption of electronic health records (EHRs), compliance issues, and a mandate for reducing labor-intensive administrative reporting processes. In addition, as epidemiological classification systems such as ICD-9-CM have been utilized increasingly for reimbursement purposes, greater attention has been placed on productivity and compliance. The work process for coding has changed over the past 25 years, with data collection going from manual indices and logs to computerized databases. Use of ICD-9-CM alone for statistical data capture has been replaced by the use of both ICD-9-CM and CPT/HCPCS codes. Manual coding is now facilitated through the use of encoding systems that contain various edits and references.

Automation in the form of computer-assisted drafting and computer-assisted manufacturing, for example, has revolutionized many industrial processes and allowed humans to build structures and machines not previously possible (e.g., computerized axial tomography [or CAT] scan). The same process is on the horizon for clinical coding.

In the coding workflow, clinical documentation (paper or electronic) is analyzed by a person and translated into ICD-9-CM or CPT/HCPCS codes (using a book or a software program) and entered into a database. New automation tools for coding allow the translation process to be assisted by computer software instead of manual review and translation alone. These new tools are not dependent on a fully implemented EHR, but as EHRs proliferate, adoption of these tools is expected to accelerate. EHRs with an embedded clinical terminology, such as SNOMED CT, will be a catalyst for significant change. A granular clinical terminology used for data capture in an EHR greatly simplifies the task of generating automated codes in a classification system. As the US adopts ICD-10-CM and ICD-10-PCS and automated maps become available, these automated tools for coding will become even more practical and valuable.

Current State of the Technology

What Is Computer-assisted Coding?

There are many tools to assist coding professionals in the code assignment process, including bar codes, pick or look-up lists, automated super-bills, logic or rules-based encoders, groupers, imaged and remote coding applications, and hard coding via chargemaster tables.

Advances in computer technology have resulted in computer applications that go a step further and actually suggest potentially applicable medical codes. Various terms are used for such systems, including automated coding, automated documentation, autocoding, computer-generated coding, and computer-assisted coding, each of which has various implied meanings. For the purposes of this practice brief, we define computer-assisted coding (CAC) as the use of computer software that automatically generates a set of medical codes for review, validation, and use based upon clinical documentation provided by healthcare practitioners.

The technology that enables CAC tools, particularly natural language processing (NLP), started years ago, as early as the 1950s with formal language theory. In earlier years, progress was slow, but since the late 1990s, technology has progressed more rapidly and is currently advancing at a furious pace. Many factors within the healthcare industry are driving this technology, including the movement to adopt EHRs and create a national health information infrastructure. This document addresses the various industry forces more in-depth in a later section. Figure 1 is a timeline depicting key advancements in the evolution of CAC technology.

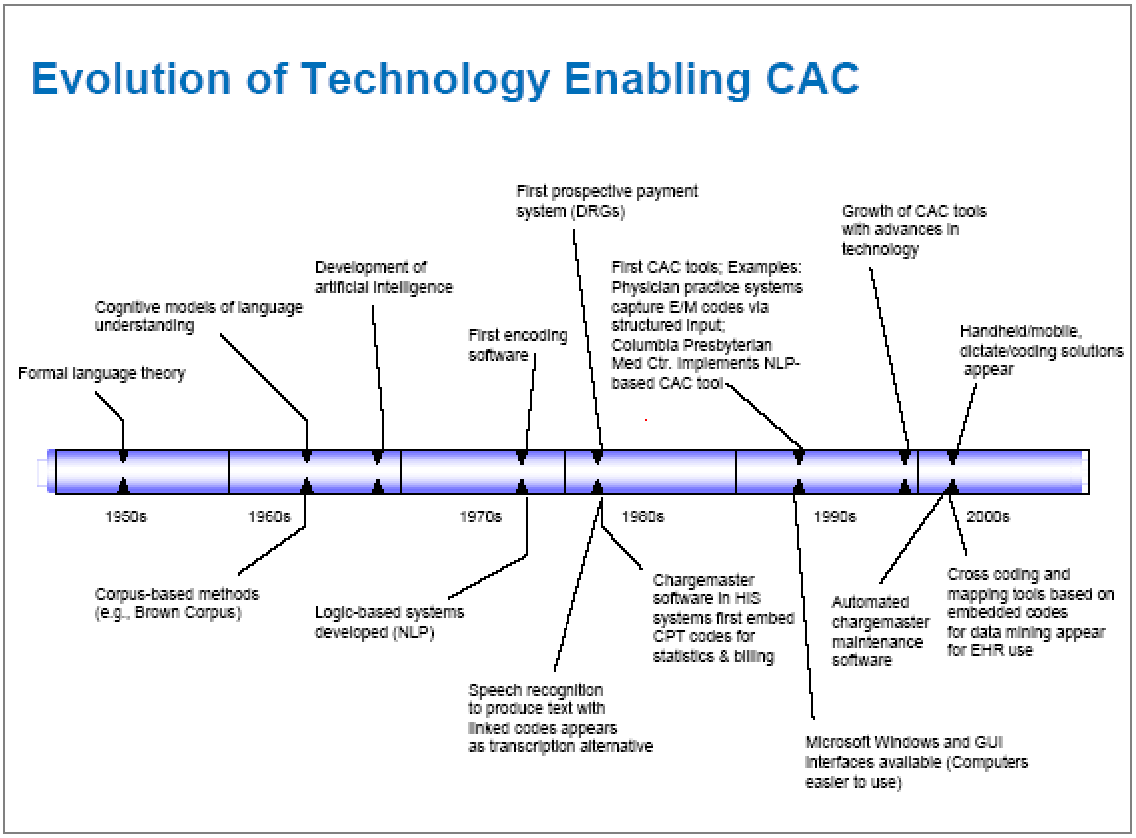


Figure 1

How Does Computer-assisted Coding Work?

CAC can be accomplished using either NLP or structured input. In simple terms, NLP is a software technology that uses artificial intelligence to extract pertinent data and terms from a text-based document and convert them into a set of medical codes to be used or edited by a coding professional. NLP is also known as computational linguistics, in which the study of linguistics, semantics, and computer science is used to abstract information from free text. For example, a natural language processor would determine if the phrase "history of cancer" means the patient does or does not have a personal or family history of cancer by analyzing the context and semantics of the rest of the sentence. With this method of CAC physicians can document health record information using their preferred terms. See appendix A for more information on how NLP uses artificial intelligence to emulate human understanding of natural language in free text. More detailed information on computer linguistic competence can also be found in appendix B.

Structured input, also known as codified input, is based upon the use of menus that contain clinical terms. As an individual menu item is chosen, a narrative text phrase is produced and becomes part of the health record documentation. Each menu item that affects coding is directly mapped to its relevant code. For example, the pre-op diagnosis menu item of "acute tear lateral anterior horn of the meniscus" is directly mapped to the applicable ICD-9-CM diagnosis code (836.1). The physician chooses the applicable clinical menu item, and the ICD-9-CM code is automatically produced to be used or edited by the coding professional. In contrast to NLP, this method records "history of cancer" as "family" history when it is entered in the specific data field for family history.

Structured input is differentiated from a pick list because it does not require human intervention to select the code. For example, a physician documenting a polypectomy would be prompted to select the specific technique used to remove the polyp. The applicable medical codes for each technique available in the menu are embedded within the system. Advantages of this method of CAC include reduction in the cost of medical transcription and improved documentation.

Industry Forces Affecting Development of CAC

Why Does the Healthcare Industry Need CAC Tools?

Since the 1980s clinical coding has become increasingly complex. Prospective payment systems (PPSs) have expanded to multiple healthcare settings. As this occurred, each PPS brought specific reporting requirements that a coder must understand and recall. Other reporting requirements, such as the correct coding initiative and payer-specific coverage policies, have also expanded the various rules that a coder must apply correctly. At the same time, the compliance liability for erroneous or fraudulent claims has increased, leaving little tolerance for coding errors. In addition, healthcare financial pressure to send (drop) the bill or claim to the insurance company as efficiently as possible has increased dramatically, and the physical time to code a record has significant impact on an organization's accounts receivables, so there is also an increased emphasis on productivity. Meanwhile, medical care continues to advance and increase in complexity, requiring coding professionals to increase their understanding of pathophysiology and even pharmacology. And this is occurring in an industry where there is already a shortage of skilled HIM-educated and certified coding professionals.

The current coding workflow is expensive and inefficient. The coding process requires that coders know more and code with greater accuracy and speed than ever before. This has created a demand to further improve the process. For example, much of the coding in the outpatient arena is repetitive and well suited to computerized tools that will reduce the workload on the professional coder, freeing these individuals for more complex coding tasks. The industry needs automated solutions to allow the coding process to become more productive, efficient, accurate, and consistent.

In addition to these industry-wide forces, there are factors related to the technology itself that affect the development of CAC. There are many advantages to CAC, which drives technology advancement. However, there are currently disadvantages as well, which present barriers to implementing CAC technology. Below is a brief summary of the key advantages and barriers to use of CAC tools. Refer to appendix C for a full discussion of the advantages and disadvantages.

Advantages of CAC include:

- Increased coding productivity
- Increased efficiency; frees professional from mundane tasks
- Comprehensive code assignment
- Consistent application of rules
- Electronic coding audit trail

Barriers to CAC include:

- Cost of CAC hardware and software
- Complexity, quality, and format of health record documentation
- User resistance to change
- Technological limitations
- Potential increase in errors in the coding process
- Lack of industry standards

Application of CAC Technology

CAC in Use Today

Computer-assisted coding, as defined for the purposes of this practice brief, is currently in use or under development for specific pockets of outpatient reporting including, but not limited to, the following "best scenario" applications:

- Radiology
- Gastroenterology procedures

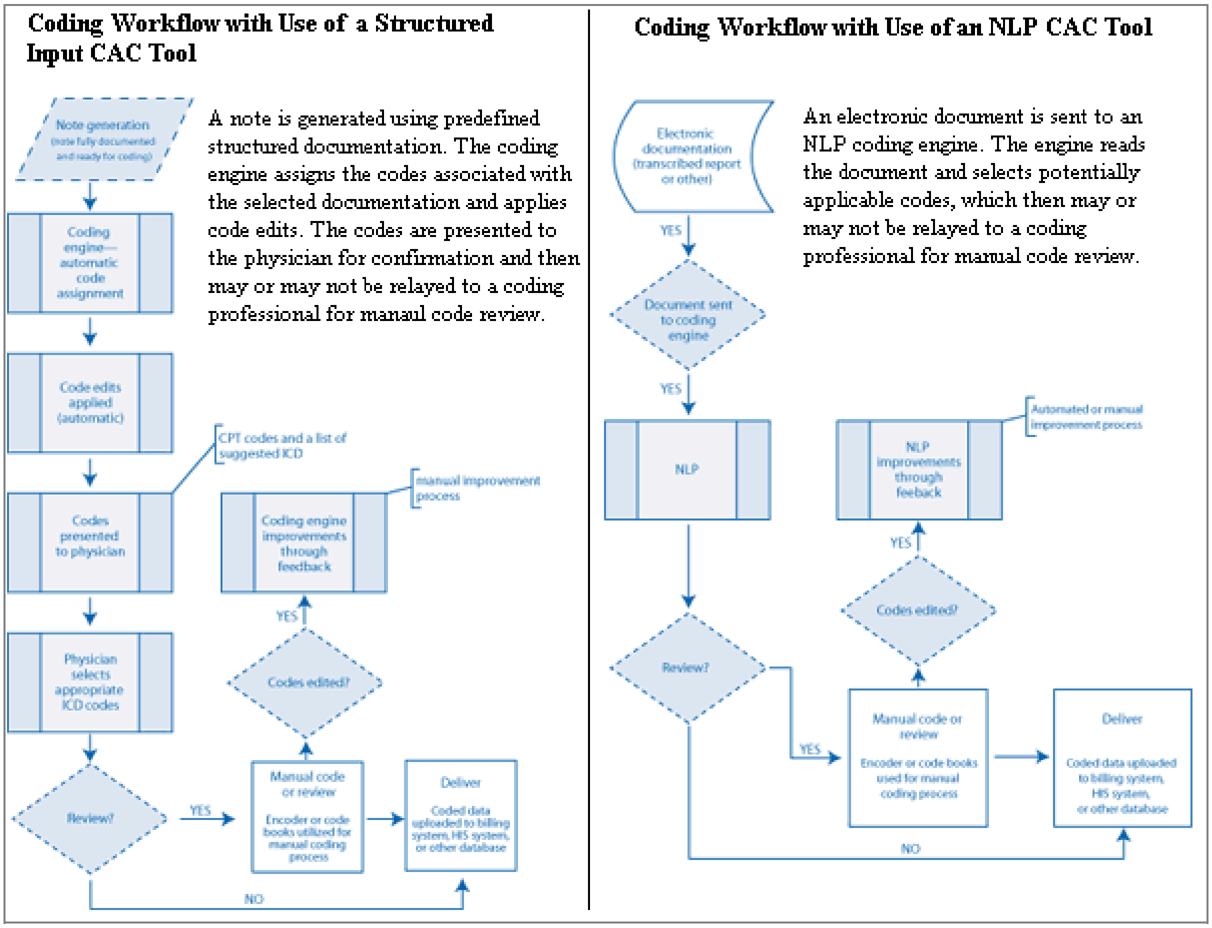
- Pathology
- Emergency medicine
- Interventional cardiology
- Orthopedics
- Podiatry
- Pulmonary medicine
- Urology procedures
- General medicine, primary care
- Other medicine subspecialties

The use of CAC depends on the availability of an electronic text-based document (whether that text is produced using structured input, free-style dictation and transcription, speech recognition, or canned template text). Electronic documentation is most typically found in the ambulatory environment at this time, such as physician office, radiology, pathology, emergency departments, and other hospital outpatient departments. In addition, CAC works best within medical domains that have a limited vocabulary. NLP-based tools in particular work best where there is a limited number of source documents that must be analyzed for code selection and less extensive coding guidelines. Development of a CAC tool for hospital inpatient use is much more complex because it involves multiple forms and formats created by multiple healthcare providers. At the time of this report, the work group identified only one CAC application to facilitate the code assignment process for inpatient acute care reporting for reimbursement purposes.

CAC is clearly making headway in both accuracy and consistency of code selection and productivity gains for clinical indexing and claims processing in the specific domains included in the bulleted list above. There are demonstrated cases in the physician coding and billing domain, as well as in the hospital outpatient environment, where CAC is improving both accuracy and speed of coding. There are many potential uses of coded data beyond administrative reporting. See appendix D for a discussion of potential uses. A description of several CAC use cases can be found in appendix E.

How Does CAC Affect the Coding Workflow?

Computer-assisted coding tools, whether utilizing NLP or clinician-friendly structured input, have great potential and have begun revolutionizing traditional workflow in certain domains (see the workflow diagrams below). As these diagrams illustrate, the traditional coding workflow is significantly altered with the use of either an NLP or structured input coding tool. In both workflows, the coding engine will improve based on feedback from the coding professional. This feedback loop may be automated (as with a statistics-based NLP tool) or manual (as with rules-based NLP or structured input-based tools).



Presently, CAC tools are typically implemented as a best practice via the code-assist model. This model utilizes software that does an initial screening against well-defined terms and produces a preliminary set of draft codes, which are reviewed, edited, and revised by a human coder to generate the final set of codes. The final assessment of codes remains the responsibility of coding professionals who can edit and correct these codes using their expert knowledge along with other tools and references. Currently, the best practice is to review 100 percent of the cases, but as CAC systems mature, this will be done more commonly via exception audits, with the system indicating which cases require review.

It must be noted that there will be situations where the complexity of care or newness of terminology or technique may result in inaccurate or missed code assignments. The reliability and validity of the CAC tool should be audited routinely to maintain coding integrity. In addition, continuous assessment is helpful in determining efficiency gains and quality improvement results. Ongoing assessment can also identify workflow problems or bottlenecks in the coding process. An important consideration in the adoption of a CAC tool is continual evaluation of the point at which the time to edit becomes less than the original time to code. This is a prime indicator of return on investment.

Can a Computer Code as Well or Better Than a Human Coder?
The limited research available suggests that NLP-based CAC tools have improved since 2000. A summary of available NLP research in appendix F is available online. Studies assessing the accuracy rate of NLP-based CAC tools, within limited domains, have reported accuracy rates ranging from 57 percent to 98 percent.[1-9] However, when assessing the quality of the code output from NLP-based CAC tools, researchers struggle with defining coding accuracy. A major problem encountered is the variability of the codes assigned by human coders against which the NLP output is compared. Research on the accuracy and consistency of human coders shows inherent variability. One study performed to evaluate reported levels of agreement between code selection by physicians and coders showed variation as high as 20 percent.[10]

In general, the work group found that CAC tools, though much faster, are not necessarily more accurate and may be a bit less accurate, depending on the domain and technique, than human coders. However, CAC technologies are improving and evolving rapidly and must continue to be monitored for applicability in the coding process across different practice domains.

Will Computers Replace Human Coders?

As CAC technology becomes increasingly sophisticated there will be less demand for coders to perform manual coding tasks. Computers will not replace all of the people who are currently working as clinical coders, but computers will begin to reduce the number of hours spent manually assigning codes. Computers are not capable of taking on the new roles and responsibilities and performing the review, validation, and oversight tasks that will be created as a result of computerization of clinical coding.

Just as software applications have continued to slowly evolve over the past several decades to create tools that assist transcriptionists (versus replacing them), CAC technology should be viewed as a tool to assist coding staff rather than as a replacement for coding staff. Though it is anticipated that computers will take over some coding tasks, computers are not expected to replace human coders. Just as transcriptionists who work with the latest technology (e.g., speech recognition) have modified their role to become "expert editors," automation tools for coding will likely result in a role change for coding professionals and will result in the better use of such staff for complex decision support tasks.

It should be noted that there may be some circumstances where CAC can be applied without human intervention today. Users reported to the work group limited instances where confidence in the CAC code output was high and only random editing was performed. An example of this is code assignment for normal mammogram reports. As these systems advance, the range of which situations will be acceptable for direct computer-generated coding is expected to increase. Overall, however, CAC, without human review, is not to the point where large displacement of the coding work force can occur to any significant degree.

Preparation for CAC

CAC software is fast and efficient, but machines are not yet capable of all of the aspects of interpretation and analysis that human coding professionals provide. Coding professionals are still needed, but it is predicted that they will move from "production" coders to knowledge workers through expert use and adaption of CAC tools. The competencies and skill sets of the knowledge-worker coder are different than those of the production coder. With the use of computer-assisted coding tools, coding professionals will no longer be tasked with the time-consuming, repetitive code assignment that can be accurately performed by a computer. Instead, knowledge workers will concentrate on tasks involving critical thinking skills, such as interpretation and analysis of documentation or aggregate data—in short, the tasks a computer cannot perform.

Computers can do many tasks faster or more efficiently than human coding professionals. However, computers cannot do everything that coding professionals can do. Coding professionals should concentrate on perfecting their skills related to the tasks that computers cannot do to ensure long-term success. "Migration of Coding Tasks" includes tasks performed by clinical coders today and shows the migration of coding tasks to knowledge-worker tasks as well. This illustrates where the professional coder's role may expand when the computer performs the routine task of manually assigning codes. "Suggested Activities for Developing Knowledge-Worker Skills" provides suggestions for developing the skills that will be necessary to fill these expanded roles.

With the use of computer-assisted coding tools, coding professionals will challenge themselves to further develop their skills and competencies in the clarification and scrutiny of data. In the future, a computer will do simple tasks that do not require critical thinking. Coding professionals should begin to evaluate the tasks they currently perform now. Identify the simple tasks, the repetitive, mundane things that you do by memory. Expect that, eventually, a computer will do these tasks faster and more efficiently. Also identify the tasks that require your judgment and intellect. These are your strengths, the skills that make you invaluable. Concentrate on building your skills and expertise in these areas. Build your unique expertise so that you are positioned to capitalize on the advantages offered by computer-assisted coding tools.

Migration of Coding Tasks		
	Could be performed by:	
Task	**Human Coding Professional**	**Computer-assisted Coding Tool**

Straightforward assignment of diagnosis codes, procedure codes, modifiers. Example: chronic otitis media with myringotomy including tube insertion is something the computer can assign accurate codes to and production coders assign accurate codes from memory.	X	X
Apply reporting guidelines (e.g., NCDs, NCCI edits, LMRPs)	X	X
Interpet documentation for correct code assignment; that is, extrapolate correct meaning from context on specific cases. Example: review and edit codes suggested by a CAC tool; determine if "postoperative anemia" indicates a condition occurring in a defined time period, after surgery, or if it is a postoperative complication.	X	
Request clarification in ambiguous documentation, whether nonspecific or inconsistent. Example: a structured input CAC system can prompt the physician for clarification at the point of input to avoid ambiguous documentation; a coder may need to review the entire record to identify the principal diagnosis or may need to initiate a physician query to clarify a diagnostic statement of "urosepsis" which could be a UTI or septicemia.	X	X
Participate on documentation improvement teams, serving as a resource on specific documentation elements needed to assign codes to the highest degree of specificity. Example: documentation to support time-based codes such as hospital discharge day management, CPT codes 99238–39; documentation of aspiration pneumonia; documentation delineating a comprehensive examination.	X	
Validate accuracy of codes assigned; recognize inappropriate application of rules. Example: application of rules, such as E codes cannot be listed first; inpatient coding guidelines applied to a rehab patient type; correct application of context-specific coding guidelines such as sequencing of respiratory failure or use of late-effect codes.	X	X
Interpret coded data to obtain information. Example: assist a physician in identifying individual cases of community-acquired pneumonia, not separately classified in ICD, by using other types of abstracted data such as core measures data or data for patient safety goals such as prophylactic antibiotics.	X	
Ensure data integrity within multiple internal systems and reporting integrity issues. Example: all systems fed in, no omissions; verify charges on accounts are accurate such as combining outpatient and inpatient charges to comply with 72-hour window rule.	X	

Educate others in the area of data retrieval, data analysis, internal data systems, and data integrity. Example: annual code changes and associated documentation requirements.	X	
Use multiple databases including, but not limited to, clinical, health plan, and national and state comparative systems for data retrieval using various report-writing tools. Example: assist in the interpretation of databases such as Leapfrog, Hedis, OSHPD in California.	X	
Aggregate data and identify patterns. Example: respiratory cases with high-dollar charges and no ventilator management reported.	X	X
Interpret aggregate data on comparative or benchmarking data and create reports of the analysis. Example: investigate a statistically significant variation on the OIG report related to specific DRGs 14/15 or 79/89, verify that variation is valid; analyze physician practice patterns of complication rates and collaborate with physician to validate patterns.	X	
Assist in the development of complex integrated database design, development, or implementation. Example: data dictionary integration and crosswalks between disparate results reporting information systems.	X	
Provide input on coding guidelines, seek to obtain guidelines where there is no clarification. Example: send questions to AHA's Coding Clinic.	X	

Suggested Activities for Developing Knowledge-Worker Skills	
Skills to Perfect	**Activities to Achieve Skills**
Become a documentation expert	Participate in concurrent documentation improvement processes Obtain formal education in health information management Pursue professional development (e.g., enroll in Web-based training)
Strengthen skills beyond general, straightforward code assignments	Become intimately familiar with coding guidelines and how they are determined Focus on more difficult, specialty coding and applying guidelines that vary based on context Take a proactive attitude toward learning and understanding payer-specific coding interpretation
Develop effective audit techniques	Look for opportunities to cross-train with individuals who perform audits within the HIM department or other departments such as billing, compliance, or risk

	management or other areas where auditing is performed
Be comfortable with technology, information systems, and statistical applications such as spreadsheets and databases	Team up with IT/IS staff; take an active role in development and testing of new applications, software upgrades, coding updates, computer input screens, or other health record documentation tools Obtain training in statistical applications (e.g., Microsoft Excel and Access) View demonstrations and visit with CAC vendors at state and national conventions
Develop interpersonal skills (e.g., effective communication skills, consulting skills, and critical decision making)	Get involved on multidisciplinary committees; work with medical and administrative staff on health record documentation standards Offer in-service educational programs related to clinical coding, documentation, coding, and abstracting software for interdepartmental staff

Practice Guidance

Building Blocks to Prepare an Organization for CAC

Evaluate existing clinical documentation. CAC tools require electronic clinical documentation. Determine what portions of the health record are used for code assignment and what portions of this are available in or could be converted to electronic form. Assess how current systems could be used to capture original data in a structured format using standards (such as Health Level Seven's clinical document architecture) to the extent possible. Consider what adjustments would be necessary to work with a CAC tool. Is existing clinical documentation, in whatever form, sufficient for accurate code assignment to the highest degree of specificity available in the coding system? If not, where can improvements be made, and can CAC facilitate this process? Evaluation of clinical documentation must be performed for each unique treatment setting (e.g., outpatient, physician office, inpatient, ED) with input not only from HIM but also from a diverse provider and user work force.

Assess current coding workflow. Assess what is being done currently, step by step, and identify how use of a CAC tool would alter the current workflow. Also identify processes in the current workflow that may be improved by CAC, made superfluous by CAC, or may hinder the utility or acceptance of a CAC tool.

Define expectations for balancing productivity and accuracy. Identify your "gold standard" for translating clinical data into medical codes. Define current productivity and accuracy rates for code assignment and the organization's tolerance level for coding variances. Will the organization expect the same accuracy rate from a CAC tool? What level is acceptable? What level of productivity does the organization expect from the CAC tool? Define the expectations for balancing productivity and accuracy to achieve a return on investment.

Define organizational goals and objectives. Determine what the organization wants to accomplish and evaluate whether or not a CAC tool can help achieve this. For example, a CAC tool may be helpful for a radiology practice that currently employs no professional coding staff and wants to improve compliance. A CAC tool may be helpful for an organization that is chronically short staffed in the coding division and desires improved productivity for existing staff. However, while an NLP-based tool can facilitate improved documentation through feedback, it will not necessarily generate better documentation.

Broaden coder skill sets. Equip staff with the required skills to capitalize on advantages offered by CAC tools. Support and encourage all coding professionals to pursue personal professional development to move up the coding and clinical data management career ladder.

Plan carefully to successfully manage the change. Clarify exactly what change needs to occur. Outline the steps to implementation. Is it a major transformation of work processes or an adaptation of existing practices? Does it affect a sole unit, or will it cut across multiple functions? Who is and is not involved? Will customers be affected, and in what way? The clearer you are about the change and expected behavior, the more likely people will be to respond. Communication is key to preparing for successful change. People need time to prepare. Communicate early and often.

Guidance in Evaluating CAC Tools

Understand the available technology. Become familiar with structured input and NLP technologies relating to CAC, how they work, and the advantages and disadvantages of each. Attend vendor presentations and evaluate the tools available for your clinical domain. Remain informed on advances in this technology, especially in your practice area, so you can help your organization make an informed decision.

Determine the best form of data input. How patient clinical documentation is captured and stored is a primary consideration for determining which main type of CAC tool to evaluate (structured input versus NLP). If physicians are already using a template for documenting patient care, they may be able to convert to a structured input tool fairly easily. If physicians insist on free text (i.e., unstructured documentation) or a combination of structured input and free text, investigate NLP tools. In addition, a system already in place may determine which CAC software application is best based on compatibility.

Consider the desired output. If the code-assist model is adopted, what information (clinical and nonclinical) will be presented to the coder and in what format? Are suggested codes linked to the documentation that supports the code assignment? Is an interface with an encoder, abstract database, or billing system desirable? Will the coding output be shared with other clinical users? Can reports be generated off the CAC data? To what extent does a tool accommodate data represented in standard format, such as Health Level Seven's clinical document architecture?

Identify specific criteria for evaluating code assignment functionality. Define minimum coding accuracy and productivity levels and consider how this may be validated. Address expectations for version control, including what version of the code system, NCCI file, or E&M documentation guidelines is used; the mechanism and timeliness of implementing updated versions; and what mechanism exists for creating a history of code assignment for compliance (e.g., are individual cases stamped with the software version?). Are mapping techniques or decision pathways appropriately driving code assignment? What ongoing quality controls are in place to assess this? Does the CAC tool suggest modifiers on CPT codes? Are specific payer requirements considered? Request that the vendor provide evidence of reliability.

Suggestions for Coding Professionals Working with CAC Tools

Develop a testing and audit plan to validate the results of the software application. Consider development of a "golden document set" (fully coded and validated) that can be used to test and compare initial software integrity, subsequent updates, and machine logic. Document the findings and, as necessary, create a project management plan to facilitate rapid reconciliation of issues, revisions, necessary upgrades, or refinements. Define acceptable confidence thresholds for various coding systems (e.g., ICD-9-CM, CPT, E&M, HCPCS) to optimize the advantages of using the tool and to provide a baseline for applicable specialty use. For example, is it acceptable if the CAC tool correctly extrapolates procedure codes for a mammogram 98 percent of the time?

Use the software for its intended purpose. Once you have tested the system and validated it, use the software as intended. Coding staff should function as editors and validators and should resist the temptation to recode the entire record on every case. Use your knowledge of the system's strengths and weaknesses to maximize your efficiency. In time it is likely that you will learn to trust the system's logic and will focus your attention on the areas and cases you know are weak or have been selected for more detailed review.

CAC is currently available in the outpatient or physician practice domains and will continue to evolve and be adapted. As the transition to EHRs and the adoption of ICD-10-CM and ICD-10-PCS occur in the US, the detailed and logical structure of these systems will increase the use of CAC tools across many different domains. In addition, as CAC technology becomes increasingly sophisticated, there will be less demand for coding professionals to perform traditional clinical coding tasks. CAC software applications that assist coding professionals in their workflow by allowing them to review and edit a draft set of codes will require coding professionals to further develop skills and competencies in the clarification and scrutiny of data. Computer-assisted coding is a budding technology whose time has come, and it heralds a new era for coding professionals.

Web-based Training to Prepare the Upcoming Knowledge Worker

Building skills in the following areas can help position professionals to capitalize on the advantages of CAC tools:

- Clinical data management
- Healthcare data analytics
- Clinical documentation improvement methods
- Conversational information technology (IT)
- Project management for IT
- SNOMED CT basics

AHIMA offers online courses in these and other topics. Go to http://campus.ahima.org for more information on Web-based training or to http://imis.ahima.org/orders for other professional development opportunities.

Appendix A: Primer on NLP for Medical Coding

This appendix addresses the question "How exactly does NLP computer-assisted coding work?" The information here is, admittedly, a gross simplification of natural language processing (NLP). The intent is not to accurately describe NLP theory, but to provide a simple explanation in lay terms. Although basic, this appendix provides a rudimentary introduction to the science of NLP. The reader should keep in mind that NLP is indeed a true marriage of linguistics, computation statistics, and computer engineering. Various applications of NLP will have different nuances in logic and process.

Much like machine language translation software that can automatically translate from one language into another (e.g., from German into English), sophisticated NLP software can translate from the language of medical English into the language of CPT and ICD-9-CM codes. Just as machine translation software can be used to meet some translation needs (and such software gets better every day), NLP computer-assisted coding software is also being used to meet some medical coding needs. It, too, gets better every day.

Two Theoretical Approaches

There are two basic approaches to using NLP software to read and understand regular, normal printed language (such as that found in physician procedure reports). One approach is to build up an understanding of the words and phrases that are used in medical documents and the codes that should be used to report these various words and phrases by way of a complex (and large) series of rules. Logically, this is called a rules-based approach. It is also called a knowledge-driven approach, as the software's coding ability is based on a knowledgeable person providing the complex rules (and exceptions to the rules and exceptions to the exceptions) to properly assign code numbers to words and phrases.

The second approach is to build up an understanding of the words and phrases that are used in medical documents and the codes that should be used to report these various words and phrases by way of a large body (or corpus) of reports. Statistics are then derived from the corpus of reports. A corpus is merely a collection of texts selected on some principled basis (including opportunistically). This statistics-based approach is also sometimes called a data-driven approach, as the software's coding ability is based on a body of statistics and the software simply makes predictions of what a proper code should be for a given word or phrase based on what statistics indicate were the codes that were assigned to such words or phrases in the software's large corpus of data.

Parallels to Language Acquisition

Consider for a moment the way secondary languages are traditionally taught. Often, the student sits in a class, generally in high school, and is taught grammar and vocabulary. By learning the rules of the language (whether something is the subject or the direct object, whether it is present or past tense, whether the subject in question is a masculine or feminine or neuter noun, and so on), the student can noodle through the rules and vocabulary, making sure proper endings are placed on words and that subjects and verbs agree. The student can then translate an English sentence into this foreign language. More often than not, the sentence is correct (or at least understandable), and greater familiarity with grammar rules and vocabulary will ultimately result in better and more accurate translations. In this regard, secondary language acquisition is much like a rules-based approach to NLP.

However, primary language is not typically acquired via this type of explicit rule-based instruction. For primary (native) languages, the speaker comes prewired for linguistic readiness and is simply constantly bombarded with enormous amounts of native speech in a linguistically rich environment. Over time, this enables the user's brain to establish a connection between linguistic structures and meaning, resulting in an ability to communicate. There is no explicit focus on the rules of grammar; the speaker simply emulates the language he or she hears. The rules are indirectly acquired. Right or wrong, the speaker will end up speaking like a Roman when growing up around Roman speakers. In this regard, primary language acquisition is much like a statistics-based approach to NLP; the software simply does statistical calculations and codes the way other coders code.

Statistics-based NLP: As Good or As Bad as the Coders It Emulates

So how does statistics-based NLP emulate coders from a statistical standpoint? Such technology is often based on proprietary programming, but as complicated as it is, it's not rocket science.

Software Development

Development Step 1: Use Previously Coded Reports to Create a Baseline Pool or Body of Data (Corpus is the official NLP jargon for this large pool of data)

The larger this corpus of data is, the better statistical guesses the software can make. Also, the more accurately all the reports in this corpus of data were coded, the better guesses the software can make. If the reports in the corpus have the average kinds of errors, the software will learn to be an average coder. If the corpus is extraordinarily accurate, the software will probably be extraordinarily accurate, too. Often, there is a tradeoff to be made: a large baseline corpus can be obtained (or may be even created if the resources are available to code a lot of reports), but it may not be coded well. Or, perhaps some very accurately coded reports are available, but not very many of them. Having a corpus of reports that is both voluminous and highly accurately coded is best. Having a high volume of not as well coded data or a smaller volume of very well coded data is more common, unfortunately, and each has its own set of advantages and disadvantages in trying to teach software how to code.

Hopefully, this large corpus of data will be representative of the types of services the software will be expected to code. To the degree that it is representative, the software will be able to make guesses on how to code new reports the software encounters based on how similar reports were coded in the corpus. To the degree that it is not representative, the software may see things during production that it has never encountered before. In such instances, the software may be stymied—at least until enough similar reports have been amassed in its ever-increasing corpus to allow it to make an informed statistical prediction.

Development Step 2: Normalize the Data

There are numerous ways of saying the same thing. "Heel fracture," "fracture of heel," "calcaneal fracture," and "fractured calcaneus" are all nearly identical (but subtly different) ways of expressing the same concept. For the purposes of making statistical guesses, the software would make better predictions if it could consider all these slightly different expressions as the same thing. So, data are normalized. That is, expressions that mean the same thing are distilled down into some computer-ese phrase (perhaps something like "calcaneus-fracture") in order to get higher statistical counts of similar conditions. In this regard, normalization is reducing words or expressions down to a least common denominator, distilling down that which is key and eliminating that which is extraneous. It is in this normalization or distillation process that some NLP errors can occur. Perhaps a phrase gets distilled down into a phrase that is actually fundamentally different from the initial expression. Coding and linguistic experts can help reduce or prevent problems related to normalizing the data.

Coding a Report

Having a large corpus of well-coded reports that have all been normalized, or reduced down into linguistic pieces to which statistical calculations can be applied, the software is now able to code a new report.

Process Step 1: Submit the Document to the Software for Reading

A report is submitted to the NLP software for coding. Reports must be in some kind of electronic format so that the document can be read by the software, much like a spell-checking program will read a document. As the software reads the document, it will identify those words and phrases it recognizes from the large corpus of words and phrases and associated codes it has already been exposed to during the development process. It also reduces or normalizes any recognized language into those computer-ese phrases into which the software distills everything. If the original corpus was representative, the software should see key words or phrases it recognizes in this report. If this report has language that is new to the software, the software (like any coder exposed to something for the first time) will not be sure how to code it.

Process Step 2: NLP Software Performs Statistical Number Crunching

After identifying language it recognizes and normalizing that language into the computer-ese phrases it works with, the NLP software will check its large corpus to see if it has ever encountered such phrases before and, if so, how those phrases were coded. Readers may remember "Family Feud," in which the studio audience was polled and the top responses to some survey question (ranked by their frequency) were displayed. NLP software performs much the same step. Imagine a report the NLP software encounters with the phrase "fracture of heel." The software normalizes this expression to "calcaneus-fracture" and then checks the large developmental corpus of data it uses to make its coding predictions to see if "calcaneus-fracture" also occurred in any of the previously encountered reports. For example, we will postulate that there were 100 other instances of "calcaneus-fracture" in the software's corpus and that those coders who coded that original batch of training reports were largely correct and in agreement on how to code such heel fractures. Here is (hypothetically) what the NLP software came up with in its "Family Feud" number crunching:

First place	91 times	Code 825.0
Second place	6 times	Code 825.1
Third place	2 times	Code 829.0
Fourth place	1 time	Code 852.0

In this case, there seemed to be great agreement among the original coders that "fracture of the heel" is coded as 825.0. The software is able to make a pretty good guess and also knows to ignore the codes that are clearly outliers. It looks like one coder transposed two digits and came up with a bad code. This happened only once, but indeed, if it had happened a lot of the time, the software would be compelled to consider it a candidate for the proper code. In this case, however, the evidence is clear, and the software predicts 825.0 as the proper code. When coders give a (generally) uniform coding message, the software's prediction becomes easy.

However, imagine if it this distribution had occurred:

First place	55 times	Code 825.0
Second place	40 times	Code 825.1
Third place	3 times	Code 829.0
Fourth place	2 times	Code 852.0

In this case, the first-place winner is not as much of a runaway. The software would be a bit more hesitant to use the 825.0 code. In the end, it probably would use that code, but instead of using the code with confidence, it might use the code not very confidently.

Or, imagine that the corpus actually had only two previous examples of "calcaneus-fracture," with this coding distribution:

First place	2 times	Code 825.0

There were no second or third places; the only two examples were both coded with 825.0. Although "calcaneus-fracture" resulted in code 825.0 100 percent of the time, the statistics (with a sample size of only two) are not that compelling. The software again would likely report 825.0 but not very confidently.

Ultimately, however, the software will simply statistically predict the best code to report for a given word or phrase based on how such a word or phrase was coded in the past. NLP software that emulates a pool of human coders has its advantages and its disadvantages, of course, based largely on the quality and volume of the coding done by the coders whom the software is emulating. Although "garbage in, garbage out" can be a fairly easily surmountable problem, it may be a comfort to know that, in the end, NLP coding software simply emulates human coders.

Process Step 3: Route the Report

Once the code is assigned, the coded report can be routed for final disposition. Such routing can be done based on a number of factors, not the least of which is how confident the software was in its coding prediction. A provider may want those reports that the software coded confidently routed to one work queue and the reports that were coded not very confidently to another work queue. Or, perhaps all the reports that resulted in a particular ICD-9-CM code are routed to a special work queue. Or, perhaps all the reports with some type of predefined CPT/ICD-9-CM mismatch are routed to another special queue. Such special routing is a standard feature of nearly all NLP coding products.

Process Step 4: Feedback for Machine Learning

Once a report is provided to a human for review, the coder reviewing that report will either accept or reject the codes that were provided by the software. Regardless of what is done to the codes, however, the coder's decision on that particular report will ultimately be fed back into the software's large corpus and be used for future coding predictions. In our example from above, the software predicted 825.0 to be the correct code. Will the coder let this code stand? Or will the coder change it? Either way, the next time the software attempts to code a "calcaneus-fracture" report, there won't be 100 reports contributing to the statistics; there will be 101 reports and an ever so slightly revised distribution of codes. The software will be able to make an even more informed decision than it did before because with each note that is reviewed, the new coding results are added to the corpus. The software's code prediction is either statistically reinforced or statistically refuted by the coder as the software continues to emulate those coders responsible for the software's coding quality. Like any coder, with experience, the NLP software makes better and better coding decisions.

Certainly, there are other aspects of NLP than are outlined in this appendix, but perhaps some of the mystery has been removed.

Appendix B: Continuum of Linguistic Competence

The goal of any system that processes natural language text for use in healthcare, regardless of the degree of human oversight or effort, is a representation of that text in an unambiguous way so that it can be used for a variety of purposes, including clinical practice, quality assurance, clinical research, clinical decision support, and reimbursement. Typically, this is accomplished through medical coding—the assignment of unique codes from classification systems or terminologies that express the meaning of the text, including observations, diagnoses, procedures, and elements of reimbursement. While this may be the end result of a computer-assisted coding (CAC) system, the underlying knowledge of the language and domain that inform the process and that are reflected in the resulting output can vary widely. In other words, the linguistic competence of the CAC system may be seen as a continuum. This continuum may range from a simple list of codes that are the summary of observations and events in a document to a complex data structure that incorporates a rich representation of all the fine shades of meaning that occur in a document.

At the relatively unsophisticated end of the continuum is simple **key word look-up**. While this may be easy to implement, its accuracy is not great because meaning often is expressed through combinations of words and because this method does not address issues such as negation, qualifiers, and conjunctions. Also, some systems at this level may not be able to process even trivial linguistic variants, such as word order and pluralization, among words that express the same meaning.

An example of key word look-up is processing an operative report word by word, comparing each word to a list of possible operations. When the processor finds the word "tonsillectomy," it flags the report as a description of a surgical procedure to remove the patient's tonsils, using a code from some classification or vocabulary with this meaning.

The next level of linguistic competence of these systems is represented by **corpus-based approaches** that can employ statistical techniques. In this approach, a body or corpus of natural language texts representative of the domain of interest is marked up in such a way as to identify single words as well as combinations of words that express a particular meaning. Statistics derived from the corpus may be used to characterize the probability with which certain combinations of words signify particular meanings. Pattern matching is then used to identify these combinations of words in the text of interest so that the codes associated with them in the original corpus may be applied.

For example, a natural language processing (NLP) system for interpreting imaging reports might use a body of previous imaging reports marked up in such a way as to identify the meaningful words and phrases in them. In the mark-up process, it may be seen that every time the phrase "air-fluid level" appears in a chest x-ray report, it is associated with the diagnosis "pleural effusion." Thus, when the processor detects the phrase "air-fluid level" in subsequent reports, it will assign a diagnostic code designating a pleural effusion.

A refinement that can be used with other techniques such as key word look-up is a **script-based method**. A script is a formal representation of possible sequences of events in a given type of scenario, such as an operative procedure. Typically, a script is linked to one or a combination of key words, and when scanning identifies those words in a document, the linked script is evoked in order to characterize the meaning of the document in additional detail. In this setting, the processor fills in the blanks of the standard script with the information contained in the text being processed. Logical (if-then) rules may be used as part of this process to refine the level of linguistic competence even further.

For example, if the processor detects the sentence, "The patient was brought to the operating room," it will invoke a script for an operative procedure. The sequence of events in that scenario may include introduction of intravenous access, prepping and draping of the skin, initiation of general anesthesia, and insertion of the endotracheal tube. Because the processor knows that it is processing an operative report, it can look for words signifying the expected events, capturing patient-specific details in the process. It may suffice to code the sentence, "The skin was prepped and draped in the usual sterile fashion," with an appropriate procedure code, whereas the code for the procedure of inserting an IV line might be a pair of codes consisting of the location of actual insertion in that particular patient as well as the procedure code for establishing IV access. The processor will continue going through the report until it reaches the end, attempting to fill in the blanks in the script with words in the report.

A still more complex level of linguistic competence is afforded by **semantic pattern matching**. This is a variant of a key word match in which the narrative text is scanned for linear combinations of words from particular classes of terms that signify a particular meaning. This technique employs linguistic knowledge of how different kinds of terms combine to represent meaning (e.g., <finding> of <position> <body part>). These classes of terms may be defined in a dictionary or vocabulary associated with a system. This differs from simple key word matching because the patterns used in the matching contain classes (e.g., <body part>) that can be instantiated using a data dictionary and not the literal words (e.g., "left arm"). The processor attempts to match every phrase in a document to its library of patterns, using a data dictionary to determine if a particular word is a member of the class that appears in pattern (e.g., that an "arm" is a <body part>).

This level of linguistic competence may be refined further by constructing a formal **semantic grammar** for the domain of interest. Such a grammar might include a collection of semantic classes (e.g., finding, location, and so on) that may be decomposed into combinations of other classes, which may be further decomposed, and so on. This decomposition and formal expression helps to distinguish this level of competence from mere semantic pattern matching. Specific words may then be assigned to specific semantic classes, and rules may be identified that specify how classes may be combined to generate well-formed expressions in narrative text.

For example, the processor can try to match the phrase "fracture of left femur" to the semantic pattern {<finding> of <position> <body part>} by confirming in a data dictionary that "fracture" is a type of finding; by matching to the literal word "of"; by confirming that "left" represents a position; and finally by determining that "femur" is a body part. Once the match has been made, the processor can use its internal rules to determine how to assign a code to the overall phrase based on the codes that represent the individual components of the phrase. In some vocabularies, this might be a combination of two codes: "fracture of femur" and "left."

Further linguistic competence may be incorporated into a CAC system by including a **syntactic grammar**. This may yield further knowledge of the meaning of a text by taking advantage of the parts of speech and the way that they appear structurally in a document. A system incorporating a syntactic grammar will include the patterns of combinations of words that represent different parts of speech (i.e., noun phrase, verb phrase), and which patterns may be decomposed, as in a semantic grammar, into simpler patterns (i.e., nouns and modifiers).

For example, the phrase, "I explained the risks and benefits," in a procedure note might be matched to the syntactic pattern <sentence> = <noun phrase> + <verb phrase>. This is further refined by mapping "I" to <noun phrase> and decomposing

<verb phrase> into {<verb> + <direct object>}, in turn mapping "explained" to <verb> and "risks and benefits" to <direct object>, and so on.

Application of analysis with both semantic and syntactic grammars, in combination with standard terminologies that use unambiguous identifiers or codes to represent meaning, can produce a linguistically rich structure that is normalized across documents and expresses all the meaning in a narrative text. This canonical form may be refined further in knowledge-based systems that incorporate common sense and practical information about a particular domain. This rich canonical form, complete with expressions of the semantic content, syntactic structure, and practical domain knowledge contained in the original narrative text, represents the relatively sophisticated end of the continuum of linguistic competence.

For example, "fracture of left femur" might be represented using this hypothetical structure (with nesting and modifiers indicated by punctuation): <observation>= [<finding> = "fracture"^<part of speech> = noun^<code> = 123^<location>= [<body part> = "femur"^<part of speech> = noun^<code> = 456^<position> = [<laterality> = "left" ^<part of speech> = adjective^<code> = 987]]].

References

Allen, James. *Natural Language Understanding.* Redwood City, CA: Pearson Addison Wesley, 1994.

Brill, Eric, and Raymond J. Mooney. "An Overview of Empirical Natural Language Processing." *AI Magazine* 18, no. 4 (1997): 13–24.

Friedman, Carol, and Stephen B. Johnson. "Medical Text Processing: Past Achievements, Future Directions." In *Aspects of the Computer-based Patient Record,* edited by M. J. Ball and M. F. Collen. New York: Springer, 1996: 212–28.

Jurafsky, Daniel, and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition.* New York: Prentice-Hall, 2000.

Manning, Christopher D., and Hinrich Schuetze. *Foundations of Statistical Natural Language Processing.* Boston: MIT Press, 1999.

Appendix C: Advantages and Disadvantages of CAC Technology

As with any technology, there are both advantages and disadvantages to the use of either structured input-based or natural language processing (NLP)-based computer-assisted coding (CAC) tools. To complicate matters somewhat, different methodologies of CAC may offer a slightly different set of advantages and disadvantages. This appendix presents both advantages and disadvantages for CAC tools in general, regardless of the methodology, as well as advantages and disadvantages unique to either structured input-based or NLP-based tools.

Advantages

Advantages of CAC Tools

Increase in coding productivity. Early studies have shown that, for some coding applications, there is a net increase in coding speed (see appendix H). This is largely because the key point of CAC software is that codes have already been selected by the software and the coding professional need only review the preselected codes and make whatever changes are necessary. Certainly some medical domains lend themselves to higher speeds than others. Domains where the documentation tends to be highly repetitive (e.g., screening mammogram radiology reports) or where procedural techniques are fairly predictable (e.g., gastroenterology endoscopies) realize the greatest speeds.

Of course, the high speeds are offset when a code is found to be incorrect and editing needs to be done, so there may be a tradeoff between accuracy and speed (which is always the dilemma in coding). There are different costs (i.e., amount of effort) associated with reviewing the set of candidate codes, selecting one or more codes provided by the system, modifying or correcting candidate codes, and identifying missing codes. But given a well-designed coding interface with which a coding professional might work, the edit function can be ergonomically designed to facilitate the process and maximize editing speed.

Productivity may also be affected by increased efficiency in the coding workflow itself. Using the CAC tool, different medical report types can be routed to particular work queues so that coding professionals can get into a rhythm by coding all the reports of one type, physician, or location or any other attribute the tool is designed to recognize. An added consideration, unique to the structured input-based CAC method, is that the system prompts more complete documentation so that delays caused by missing details needed to assign a precise code may be minimized.

Increase in coding consistency. It is difficult to make organizational improvements in coding when coding is done inconsistently from one day to the next or from one coder to the next. CAC is very consistent—even when it is not right. This consistency makes for a compelling case. Codes will be assigned in the same manner each time. In a structured input-based tool, codes linked to structured input are assigned once at the time the input is created. NLP rules-based software does not forget a rule one day and remember it another. Even mistakes would be consistently generated.

Availability of coding audit trail. Because coding decisions made by CAC software are based on programming and on rules and statistical calculations, the reason a particular code was selected at any given time can be reconstructed and analyzed if necessary. System designers or developers may reconstruct such audit trails as needed.

Data query ability. The use of CAC data for such purposes as Joint Commission auditing, quality assurance measures, performance studies, credentialing, and research is an attractive feature of this technology. Many CAC systems offer different ways to query data from their systems, including prewritten "canned" reports, ad-hoc queries, and the use of structured query language (SQL) to access the data.

Potential for more comprehensive code assignment. In the case of outpatient claims, CMS-1500 forms have a limited number of fields for ICD-9-CM codes. Because of production demands, often only the codes necessary for reporting to third-party payers are captured. An advantage of CAC software is that it can code a report with ten diagnoses as quickly and easily as it can code a report with only two diagnoses. To the degree that a provider would allocate resources to capture all applicable codes, NLP could probably do it faster; this advantage simply falls under the category of productivity. But to the degree that a provider may elect to not be as thorough and simply report only the diagnoses required for reimbursement, the CAC advantage is that it could provide a more complete clinical picture, with *all* diagnoses successfully captured (or "recalled," in computer science language), not merely the lucrative ones.

Potential increase in coding accuracy. Coding rules are a moving target, with clarifications offered quarterly in the case of ICD-9-CM and HCPCS Level II codes and monthly in the case of CPT codes. Revisions to the CPT, HCPCS, and ICD-9-CM code sets are made twice a year, and payer rules such as NCCI edits are frequently updated. Official clarification on proper use of the ICD-9-CM code set comes from the Central Office for ICD-9-CM (either via its quarterly *Coding Clinic for ICD-9-CM* publication or via questions posed directly to the Central Office for ICD-9-CM). Similarly, official clarification on proper use of the CPT code set comes from the American Medical Association (AMA) by way of:

- *The CPT Assistant* monthly newsletter
- *The CPT Assistant: Clinical Examples* bimonthly newsletter
- *Principles of CPT Coding (1st, 2nd, and 3rd editions)*
- *The CPT Companion*
- *CPT Changes: An Insider's View*
- Questions posed directly to the AMA's CPT Information Services Department

Because of these changes, as well as the intricacy of coding rules, CAC software is possibly better equipped to code compliantly than even the most skilled coding professional.

Furthermore, the axiom, "If it wasn't documented, it wasn't done," may apply to CAC software better than to human coders. Both structured input-based and NLP-based CAC applications are incapable of assuming, leaping to conclusions, or even "reading between the lines"; the software tends to be more purely accurate (for better or worse), as it is based solely on the documentation.

Potential decrease in coding costs. Because CAC software does not need vacations or health insurance, it can code less expensively than human coders. It can work straight though lunch hour, and it can work in the middle of the night without a night-shift differential. By being available at times when humans are more expensive, turnaround time for code assignment can be shortened, resulting in improved accounts receivable management. As with any major change, the return on investment should take into consideration all relevant factors, including the initial investment in the system, how the tool is implemented, and ongoing costs.

Advantages Unique to Structured Input-based CAC Tools

Improved documentation. Many times, the individual who is generating clinical documentation has a clinical background and does not appreciate the information needed to code the document completely and compliantly. The difference between clinical language and coding language can create ambiguities and omissions. A structured approach to creating documentation can ensure that the clinical details that affect coding are recorded. In this way, the documenter is creating a detailed, coding-ready document.

Decreased documentation costs. When a structured input system is used, the need for dictation and transcription is eliminated. Thus, the costs associated are removed.

Creation of ancillary documentation. The primary document that is produced via structured input is a procedure report. However, many other documents can be produced automatically once the procedure report has been generated. Ancillary documents include letters to referring physicians, patient instructions, letters to patients, image-only reports, coding reports, pathology reports, and letters to other physicians. The creation of these ancillary documents greatly improves the efficiency for the clinician who is documenting the procedures and improves communication with referring physicians.

Advantages Unique to NLP-based CAC Tools

Use of free text for recording documentation. Physicians can continue to document health record information using their preferred terms.

System improvements through feedback. NLP systems can learn and improve with time, just as any coding professional would. For example, a statistics-based NLP program is generally based on an initial training corpus, or sample of previously coded documents (coded by humans). When the software encounters a report that resembles previously coded reports, it will predict the likely code(s) based statistically on how such similar reports were coded by the humans who had coded those other similar reports. As the corpus, or pool of reports, grows and as coding errors are caught and corrected by coding professionals and fed back into the corpus, the software can make better statistical predictions on how a report is to be coded. Some NLP systems can learn in an automated fashion; others learn through supervised learning.

Disadvantages

Disadvantages of CAC Tools

User-specific integration. The workflow and integration of the CAC tool are critical to the success of CAC. This can be a substantial undertaking, given the many existing interfaces that may be needed.

Currently, coding systems may include:

- Abstracting applications
- Encoder integration
- Integration with reimbursement applications (e.g., DRG or APC groupers)
- ADT interface for demographics

- Interface with HIS or practice management systems
- Imaged records
- Electronic health records

CAC will add yet another point of potential system failure and potentially an additional vendor. Diagnosing and supporting systems with this level of integration can be resource intensive and complex, depending upon the sophistication of an organization's information technology staff and systems. In some cases, however, the CAC system is not used directly by coding staff. Instead, system outputs (documentation and preliminary codes) are sent electronically to the coders, who then review that data in their own systems.

User acceptance and change management. This technology may be seen as threatening to coders, and therefore considerable effort (education, change management) will be required for a successful implementation. There is a possibility that coding productivity will decrease, at least initially, for the following reasons:

- Natural learning process
- Investment in training (materials, time)
- Increased effort by coders to prove the system wrong

Cost (initial purchase and ongoing maintenance). The cost of a CAC system may include a PC, the operating system, and perhaps a browser. In addition, there may be the cost of a LAN, WAN, Internet connectivity, and system administration overhead. If the CAC application is running at the customer's site rather than via an ASP, there will be additional systems administration costs for servers and their care and feeding. Also, CAC software must be updated whenever new coding rules are promulgated or whenever code set changes are made. This requires ongoing maintenance to continually meet compliance and regulatory standards. The costs of implementing these changes must be borne by someone.

Potential for coding errors or fraudulent claims. CAC tools may result in coding errors, especially if the user is not performing review and audit functions as recommended.

Disadvantages Unique to Structured Input-based CAC Tools

Use of structured input. Physicians can be quite resistant to new technologies and may not want to use structured input software to generate their procedure documentation. Also, a physician may either not agree with the exact wording of the structured input narrative reports or may view the use of structured input software as too cumbersome to compete with dictation.

Disadvantages Unique to NLP-based CAC Tools

Reliance on electronic documents. An important premise of NLP coding is that an electronic document must be run through the NLP software so that the software can read the document, much like a spell-checking program reads a document. Necessarily, NLP requires documents be in an electronic format. This means that other documents that might affect code assignment (whether they are handwritten, scanned type, pictures, or graphs) cannot be used by the NLP software.

The existence of multiple formats for documents negatively affects not only the ability of the NLP system to accurately identify codes, but also the coder's workflow. Nonelectronic documents must be aggregated and reviewed in order to ensure completeness, meaning that the coder may have to work between the computer and printouts of various kinds. Furthermore, the documentation (even electronic) does not always come at the same time, especially for inpatient or extended-stay encounters. This can have a big impact on productivity.

Software development efforts. NLP software may require considerable software development and improvement efforts to get the software to code extremely well. That is, the software may be a victim of the Pareto principle (80/20 rule): 20 percent of the software development efforts will succeed in getting about 80 percent of notes coded successfully; but to code the remaining 20 percent of notes will require the other 80 percent of software development and improvement efforts. The greater the coding accuracy required, the greater the amount of time needed to achieve such high levels of accuracy. There may be a point of diminishing returns.

Potential NLP coding mistakes. NLP coding software does make coding errors. What is particularly frustrating is that many of these errors are unlike any coding errors a human would ever make, and they look particularly "boneheaded." Often, such coding errors occur when software has detected an apparent pattern but extrapolated that pattern to apply more widely than is appropriate. A typical example might occur when software was trained on a limited corpus (sample) of reports and only saw the word "screening" in the context of "screening mammograms." Statistically, the correlation between the word "screening" and the code V76.12 may be extremely high, but when presented with the expression "screening colonoscopy," it may report the code V76.12, Other screening mammogram. This is the wrong code for a screening colonoscopy, but statistically it may appear to be the most appropriate code to report.

"Machine learning" presents problems if coders teach the software their errors. As the saying goes, "garbage in, garbage out." Machine learning systems learn what you give them and will therefore learn mistakes with alacrity and indifference. NLP systems that emulate human coding are vulnerable to all the advantages and disadvantage that reliance on a human standard would cause. Statistics can "water down" the effect of outlier coders. But NLP systems are unable to discern the difference between codes reported correctly consistently and those that are reported incorrectly consistently.

NLP coding software may tend to assign less specific (.9) unspecified-type codes than a human might. That is, a human might know where to look to help make a coding decision that the software could not make. Of course, notes that get consistently changed from a .9 to some other code will eventually be handled better by NLP software as it machine learns from the corrections the coders and auditors are making to the codes. This likely will not happen, though, if the coder is basing his or her coding decisions on data from locations in the chart to which the NLP software does not have access.

The fact that a condition happened in the patient's past and is not current sometimes confuses the software. Although a "history of breast cancer" is easily identified by software, a "history of vomiting for three days" becomes more confusing to the software. To avoid such confusion, it may be necessary for documentation to be dictated "just right."

NLP software may pick up duplicate conditions if the condition is simply worded in slightly different ways in different points in the document, and it may not be able to determine correctly the proper rule set to invoke based on different modalities or settings. For example, when a report is provided to the software for coding, it may be unclear whether to apply physician practice coding rules or hospital-based outpatient coding rules.

Appendix D: Potential Uses of Structured Code Output

Structure in healthcare data has at least two important dimensions: format and content. Examples of the former include standardized messages and relational database tables. A structured format allows the interpreter of data to locate a specific kind of data or to know that a particular data element belongs to a particular type. As a specific example, the Health Level Seven (HL7) messaging standard specifies a structure for transmitted messages in which particular types of data (e.g., personal identifiers, test results) are stored in specific fields of a text message. Another example of structured format is the HL7 Clinical Document Architecture (CDA), in which the content of particular documents, such as discharge summaries or imaging reports, is subdivided into sections with standardized labels.

In distinction from structured format, the structure of data content is principally the domain of controlled terminologies. In their simplest form, terminologies provide an agreed-upon collection of codes for tagging or identifying healthcare data. A coding scheme may provide high-level representations (e.g., a code for "laboratory test result"), granular representations (e.g., "magnetic resonance imaging of left upper extremity without contrast"), or representations that fall between these extremes. Meaning may be assigned to data by coding individual elements with single, highly granular concepts ("precoordination") or combinations of less granular elements ("postcoordination"). In either case, the codification provides structure by abstracting mostly narrative data into agreed-upon, distinct concepts that represent the meaning of the narrative data.

The structure of healthcare data is a continuum. Even a relatively small degree of structure in data offers some value to those who need to use or interpret the data. The value of structure lies in the reduction of ambiguity that occurs when agreed-upon representations are used to transmit and interpret data. By overcoming the challenges of synonymy (different people using different natural language names for the same concept) and the inherent ambiguity of narrative text, the codification of data renders them increasingly useful for various applications, with the utility increasing as the degree of structure or level of codification increases.

Important uses of coded data include the following:

• **Clinical care.** Coded data may be imported into clinical software such as electronic health records, thus allowing data to be shared easily among practitioners—an important consideration when many patients do not have a single medical home and frequently must change providers for insurance reasons.

• **Decision support.** Unambiguous representation of medications, diagnoses, allergies, and vital signs may be used by computer-based clinical decision support systems to apply decision rules in order to generate alerts and reminders to help prevent errors or to help practitioners adhere more closely to standards of care.

• **Public health.** Aggregating coded data over an entire population can be used to track the health status of the population and intervene when appropriate. Important public health applications that take advantage of coded data are epidemiology (including detection of potential bioterrorism events), health services utilization studies, and immunization registries.

• **Research.** Many healthcare organizations maintain general data warehouses that can be mined in order to discover new knowledge (e.g., linking a disease to its potential causes or risk factors) or track patients enrolled in clinical trials.

• **Quality assurance.** Clear representation of hospital admission data, medications, potential complications of care, and the like can be used to assess adherence to practice guidelines, compare the performance of healthcare professionals and organizations, and assist interventions that will improve patient care.

• **Administration.** Coded data enable administrators to perform resource allocation, budgeting, marketing, contract negotiations, material management, and other functions that are important to the operation of healthcare organizations.

• **Reimbursement.** Levels of service, procedures performed, and supporting diagnoses all may be used to prepare requests for reimbursement of performed services.

By providing a high level of structure in healthcare data, we can increase the degree to which computers can be used as intelligent assistants to human decision makers to help improve patient care and the operation of healthcare organizations.

Appendix E: Summary of Use Cases

Computer-assisted coding (CAC) products are currently in use in a number of varied healthcare settings. As noted in this practice brief, there are tools that use natural language processing (NLP) that can, in principle, be used with any medical text provided the medical domain knowledge is available. The NLP tools most commonly used today have been optimized for specific domain areas, including radiology, general medical, and emergency medicine. At the other end of the spectrum are CAC tools that are applied to structured input. These tools use menus to guide the provider's clinical documentation and generate medical codes based on the structured input. Thus far, such systems have been deployed in such procedurally driven specialties as gastroenterology and pulmonary medicine.

The CAC tools seem to enjoy a high degree of effectiveness and acceptance when HIM professionals have been actively involved in the implementation. To date, the NLP tools have had the greatest impact in the area of radiology, where customers report large increases in coding productivity, in some cases as much as three-fold. In situations where coding had not been done by professional coders, customers also report improved compliance. Similar to the radiology experience, emergency medicine also seems to benefit from increased coding productivity, with some sites reporting gains of nearly two-fold. However, in the case of structured input systems, analysis of the impact is more complicated, as there are many factors to consider. For example, the efficiency of the coding staff is improved, but some physicians may take longer to document using the menus. Some of the efficiencies gained may in part be related to operational improvements made to accommodate the transition from a paper-based system to an electronic workflow and electronic medical record (EMR). The obvious benefits and challenges combined still reflect that the overall experience is positive.

In an effort to obtain real-world experience with CAC tools, several vendors were contacted to help identify healthcare organizations that would share their recent experience in the application of these technologies. Interviews were subsequently conducted and are summarized in this appendix. The use cases are organized by type of CAC application-NLP based or structured input-for commonality.

NLP-based CAC Tools

Site 1. The emergency department of a small health system comprises acute care and rehabilitation beds as well as assisted-living units. The emergency department treats 40,000 patients per year, with an admission rate of 60 percent.

Goal: Offset coder shortage and improve revenue reimbursement.

Experience: Nearly two years ago the emergency department implemented an EMR, which eliminated the need for traditional medical transcription services. An NLP application had been used to assign codes from the transcribed notes, so it was a natural progression to send the EMR documents through the same coding process. Vendor-employed coders validate the baseline code(s) assigned by the NLP product and pass the codes back to the hospital billing system. The management reporting component facilitates real-time monitoring of outstanding records to be coded. Frequently, records are coded within six hours or less of submission, and overall, the unbilled days are down to two. The pairing of an EMR and the NLP product has achieved the desired goals of the organization. An outside consultant recently reviewed 100 charts and found the coding to be 100 percent accurate.

Site 2. A large radiology service performing 80,000 exams per month and using dictated text notes to manually generate ICD and CPT coding.

Goal: To replace the radiologist's role in coding and improve overall coding integrity and compliance.

Experience: The service experienced a three-fold increase in efficiency with CAC replacing the manual physician process. Fewer than 5 percent of the NLP-coded exams are changed following manual audit review.

Site 3. A multisite radiology group practice whose largest location performs 200,000 exams annually.

Goal: To improve productivity and reduce the demand to outsource coding support.

Experience: The customer transitioned from a purely paper-based process in October 2003. Human coders did not change the "confident" category of CAC-generated codes for tens of thousands of reports. Because this customer transitioned from a purely paper-based process, they were able to reduce personnel by eliminating much of the data entry. Although unable to quantify the increase in productivity, they "wouldn't go back."

Site 4. An emergency department in a large metropolitan hospital treating 77,000 patients annually. The department has been using an NLP application for two years for professional fee coding and one year for facility coding.

Goal: Reduce the huge coding backlog created by record logistics and access issues.

Experience: The HIM department was involved in the selection, planning, and implementation of the application from the outset. After using the vendor's coders for one year, the code validation/auditing function was brought in house. Now, three remote coders review 100 percent of the CAC-suggested codes. The coders adapted to the application in less than a month, while it took another month to get used to working with an EMR rather than a paper record. Now, the coders and management could not imagine going back to the old way. The initial goals have been achieved for the emergency department, with a dramatic reduction in the unbilled days. Based on this positive experience, there has been some discussion about expanding the application to other domains.

Structured Input-based CAC Tools

Site 1. A 550-bed teaching hospital installed a structured input documentation system approximately four years ago for endoscopy and is now expanding into cardiology and pulmonary medicine.

Goal: Create a paperless workflow.

Experience: The endoscopy area is satisfied with the paperless features of the system. Currently, hospital coding is done from the endoscopy procedure report without any reference to the computer-generated codes. The HIM department was not involved in the initial implementation in endoscopy or the recent expansion to cardiology. The application has met the endoscopy unit's goal of automating the workflow and creating an EMR. However, the HIM department is not actively evaluating the computer-generated coding feature of the product.

Site 2. A multispecialty hospital implemented a structured input application in November 2002 for gastroenterology, pulmonary, and urology. The combined procedure volumes for these departments are approximately 200 to 300 cases per month.

Goal: The decision to procure the software was the result of a new physician, who required that an automated documentation tool be available for real-time capture of procedure notes and images, joining the staff.

Experience: Software flexibility in menu design and customization by the physician influenced the selection of this particular product. Many qualitative rather than quantitative benefits were realized. Medical transcription resources were reallocated to other medical specialties, thereby improving turnaround time in areas where production was an issue. Immediate access to the report information has reduced contention for report access, improved communications with the referral community, and significantly reduced claims denials and requests for information. The quality of documentation is improved and more consistent, although there are some issues with physician selections from pick lists. Gastroenterology physicians initially believed they would assign codes throughout structured input but discovered that billing and coding rules were more complex, requiring professional coder involvement and review. Having HIM assume greater coding responsibility enabled the organization to bill for procedures based upon definitive diagnoses and findings rather than presenting signs and symptoms used on charge slips. HIM acknowledged that shortages of skilled coders and pressure to reduce billing lag make this product appealing in the long term. HIM also noted that their coding roles are changing from "heads down" coding to editors and educators for accurate documentation and reimbursement. They believe that there may be an elevation of the coding professional among clinicians as they recognize their value as collaborators, educators, and technical resources.

Site 3. Two outpatient gastroenterology endoscopy sites that perform approximately 850 procedures per month served as an original beta test site for the application in 1997.

Goal: To promote timely clinical documentation and the assignment of accurate codes for billing and reimbursement purposes.

Experience: The key selection factors of this software were based upon customizable menus and content, and the fact that the developers were all medical professionals who defined content and scope. The ability to automate existing forms and content within the application was appealing.

The product evaluation included two months of validating coding assignments both prior to and after implementation. The initial audited results showed significant underbilling because of missing information and secondary diagnoses. Improved coding accuracy and capture were noted with use of the new system.

First-year productivity savings were significant, and the organization was able to expand its procedure volume capacity from 250 to 700 procedures per month. Clerical time to record and bill patient procedural information was significantly reduced because of increased network access and real-time reporting. Productivity gains offset the creation of a new pre-admit position and saved .5 FTE in HIM. Dropping a bill used to take a week and now only takes one day. Late charges and lost or incomplete documentation do not delay billing anymore. Rebilling and insurance queries are decreasing, and response times have been improved through online access to information. A coding audit committee meets each month with representatives from compliance, billing, medical staff, and administration. Formal audits are performed quarterly.

Appendix F

Annotated Bibliography

AHIMA. "Clinical Data Specialist." In *Evolving HIM Careers: Seven Roles for the Future*. Chicago: AHIMA, 1998. .
As technology progressively automates more processes, what will HIM jobs look like in the future? *Evolving HIM Careers* offers an in-depth look at seven new roles for HIM professionals, including the role of the clinical data specialist.

AHIMA. "Natural Language Processing as a Means to Increase Productivity." Audio seminar, May 13, 2004. Available online at http://campus.ahima.org/audio/2004seminars.html.
This seminar provides an itinerary to follow in getting ready for 21st century coding systems. The program discusses what healthcare organizations and HIM services must do to ensure a smooth transition to full implementation. The program also addresses how natural language processing (NLP) is being used in the industry and how coders would interact with NLP programs, as well as how NLP will increase coder productivity.

Beinborn, Julie. "Automated Coding: The Next Step." *Journal of AHIMA* 70, no. 7 (1999): 38–43.
This article discusses the fact that the coding process has been on its way to automation for a long time, for example, by the use of encoders and groupers. There is a contrast of computer-assisted encoders versus the future of automated coding as well as an overview of how automated coding processes work.

Boelle, Pierre-Yves, Antoine Flahault, Laurent Letrilliart, and Cecile Viboud. "Automatic Coding of Reasons for Hospital Referral from General Medicine Free-text Reports." *Proceedings of the 2000 AMIA Annual Symposium*, 487–91.
A simple string matching system has been designed for the automatic encoding of medical data written as free-text sentences. For each free-text sentence concerning the reason for referral, the program has been designed to retrieve, from the look-up table file, a sentence that could be matched for character string sequence and thus be automatically coded based on the match. The accuracy rate is estimated at 80 percent at code level. Although the automatic encoding system does rely on a conventional morphological approach, its efficacy as compared to manual processing, when applied to real

clinical data from a large sample of practitioners, suggests that it is a reliable alternative—in terms of both time and money—to manual encoding for hospital referrals in general practice.

Bowman, Jim, and Mary Stanfill. "Physicians Cast Wary Eye at Computer-assisted Coding." *Journal of AHIMA* 75, no. 8 (2004): 76–77.

This "Coding Notes" article discusses computer-assisted coding (CAC) from the physician's perspective. It addresses the following questions: Is it likely that CAC technology will be integrated into physician offices? If so, what will coders do? And what should HIM professionals do now to prepare? The author relays that physicians will adopt CAC when it makes good business sense to do so. An insert bulleted list includes financial barriers to adoption of new healthcare IT by physicians. When physicians adopt CAC, coders' jobs will include more complexity and responsibility. To prepare, HIM professionals should embrace technological advances and learn more about CAC technology.

Evans, David, John Holbrook, Douglas Stetson, and Homer Warner, Jr. "Has Natural Language Processing Finally Arrived? Autocoding and Data Mining Examined." HIMSS Panel Presentation, February 2001.

The panel discussion for this topic described the pros and cons of the natural language processing (NLP) movement. The presentation reveals the different levels of the NLP hierarchy, the different studies that were used and their accuracy, as well as limitations of NLP use for medical coding. The capabilities of NLP are also outlined to provide the audience with an understanding of what can be done and the limitations that exist.

Garvin, Jennifer, and Valerie Watzlaf. "Current Coding Competency Compared to Projected Competency." *Perspectives in Health Information Management* 1, no. 2 (2004).

Coding competency is extremely important to the HIM profession and healthcare in general. The research presented in this article evaluates coding skill and competency using practice-based research. The projected skill set for the clinical data specialist, the future coding role set forth in the publication *Evolving HIM Careers*, was used to determine how prepared current coders are in terms of projected competencies. To conduct this investigation, a random sample of coders and noncoders were surveyed to determine how well the current level of skills relate to the skills described for the clinical data specialist. In addition to evaluating the skills of current coders, noncoders were used to determine whether there was a statistically significant difference between coders' and noncoders' skills relative to the future competencies. If the coders and noncoders had similar self-assessed skills, the validity of the skill set would be questionable. If, however, the self-assessed skill was significantly different, the assertion that the skill set is specific to coders would be more credible.

The findings from the research suggest that there are many skills projected for the clinical data specialist that are shared by both coders and noncoders. Also, neither coders nor noncoders reflected the level of competence in their self-assessed skills in many areas, such as understanding coding and classification systems other than ICD-9-CM and CPT, designing audit tools, performing quality audits, and selecting statistical software applications appropriate to the data to be captured. The research also suggests that coding professionals who wish to prepare for the future should acquire more communication, research, and management skills. Furthermore, because only a few skills were found to be significantly different between the two groups, the noncoding health information professionals can prepare to become coding professionals by gaining skills in coding systems and reimbursement software. Moreover, the implication is that the skill set projected for coders applies to all HIM professionals.

Hagland, Mark. "Revolution in Progress: How Technology Is Reshaping the Coding World." *Journal of AHIMA* 73, no. 7 (2002): 32–35.

This article supplies an overview of the revolution in progress, including the current use of technology to support home-based coding through continued uses of autocoding technology. This article also provides some key pieces of advice for coding professionals as they embrace the technology-driven changes. This advice includes recommendations to read and learn as much as possible about new technology; understand the implications of those technologies; realize and accept that trends toward innovations like autocoding will inevitably mean lower volumes of work for current coding professionals, and that those coding professionals who remain will see the level of their work rise—toward editing and quality assurance; upgrade and enhance skills as much as possible to be able to move up to that higher level as technology changes and improves; upgrade knowledge in areas like Medicare and Medicaid reimbursement and private insurer reimbursement, and learn what kinds of supportive knowledge will be needed to work with physicians and changing technology; become part of the solution by learning about the new technology options; and become a knowledge adviser to colleagues and a change agent within the organization.

Hieb, Barry. "NLP Basics for Healthcare." Gartner Research, August 16, 2002. Available online at www4.gartner.com/1_researchanalysis/vendor_rating/vr_landc.html.

Gartner, Inc., provides research and analysis on the global IT industry. Its goal in providing in-depth analysis and actionable advice on all aspects of technology is to assist enterprises to make informed technology and business decisions. This white paper explains the basics of NLP, including a look at the basic processes and requirements that are involved. The paper concludes that NLP has long represented the "missing link" in the automated processing of medical text. Healthcare information processing requires basic NLP attributes.

Johns, Merida. "A Crystal Ball for Coding." *Journal of AHIMA* 71, no. 1 (2000): 26–33.

This article provides an anticipated realistic snapshot of coding in the year 2010. The article also discusses the AHIMA Coding Future Task Force's evaluation of forces that will move coding in the future. The key factors discussed are rapid technology change, standards development, cost-driven environment, demand for information and quality, and the importance of HIM professionals to be prepared by taking the lead and capitalizing on opportunities.

Schnitzer, Gregory L. "Natural Language Processing: A Coding Professional's Perspective." *Journal of AHIMA* 71, no. 9 (2000): 95–98.

This article offers a brief overview of NLP from a coding professional's perspective that may help to explain why medical coding is one of the few places where NLP technology may enjoy a great deal of success. The article discusses that if a service is not appropriately documented, there is no way for NLP to assign a code; NLP technology is consistent in code assignment, will combine accuracy with speed, and will get smarter each time. Additionally, the article addresses the new roles for HIM professionals, indicating that those professionals who excel at coding, are knowledgeable on nosology and nomenclature issues, and are well-versed in all the major code sets will be best prepared and always have a role to play as HIM moves into the 21st century.

Schnitzer, Gregory L., and Mary H. Stanfill. "Outwit, Outlast, Outcode: Surviving in the Autocoding Era." *Journal of AHIMA* 72, no. 9 (2001): 102–4.

The authors of this "Coding Notes" article speak of the changing dynamic of the coding world. The new nomenclature for the coding world will soon be technologically defined, and the authors describe how coders can be ready for this new era, the autocoding era. This era will include the introduction of NLP software into the healthcare arena for hospitals to use on a broad scale. Coders may perceive the introduction of NLP technologies to be threatening to their jobs. The article gives several approaches to alleviate these fears and describes how a coder can prevail in the new era. These traits include perseverance, adaptability, and expertise. The work also defines what characteristics would be useful to master during this era, including auditing coding quality, communicating and educating, and computer literacy. In the autocoding era, HIM professionals will have to learn these abilities and characteristics to survive.

Stanfill, Mary H. "Electronic Antidotes to Coding Ailments." *Journal of AHIMA* 72, no. 6 (2001): 71–73.

Coding and claims processing software may be the antidote to medical practice business challenges. There are basically three types of software programs available for coding: database files, encoders, and charge-capturing programs. Steps in evaluating the type of software that may be helpful to a facility are outlined. Organizations are advised to begin by evaluating the need for coding software. Take advantage of any opportunity to try out a software application for free for a specified time frame, and thoroughly test these applications for accuracy and reliability. Always request a list of references and take the time to contact these customers.

Warner, Homer, Jr. "Can Natural Language Processing Aid Outpatient Coders?" *Journal of AHIMA* 71, no. 8 (2000): 78–81.

This article summarizes the results of a study performed by 3M Health Systems on ICD-9-CM and CPT evaluation and management coding in emergency room charts. The study revealed that NLP programs have the advantage in being predictable, programmable, and fast. While automated coding systems still need improvement before they will meet hospital standards on coding accuracy, they may be able to provide an immediate benefit as decision support tools for emergency medicine coding. The performance results from this study provide support for using NLP technology in coding because coding speed, like coding accuracy, affects expeditious reimbursement. For now, with the shortage of qualified outpatient coders to meet the APC-related increased demand for professional services coding, HIM directors may want to consider NLP technology to improve coding productivity.

Warner, Homer, Jr. "Good Isn't Enough." *Health Management Technology* 22, no. 6 (2001): 30–31.

Studies suggest that NLP can improve medical record coding productivity and consistency without sacrificing quality. NLP extracts facts, such as ICD-9-CM codes, from narrative text that is typically created by a transcriptionist working from physician dictation. 3M-sponsored studies conducted in laboratory settings have demonstrated 30 percent to 50 percent improvement in coder productivity, reduction in workload reflected in the number of charts that can be coded without human intervention (40 percent to 60 percent), and improved intercoder consistency with no reduction in coding accuracy. Among those who study the potential of NLP as an autocoding or coding-assist tool for billing, debate exists over the size of the market opportunity, in light of the variation in prerequisite use of dictation/transcription. If NLP vendors can demonstrate their technology's ability to relieve medical record coding staffing pressures and provide a cost-effective end-to-end billing solution for healthcare providers, mainstream adoption may be just around the corner.

Warner, Homer, Jr. "Will Natural Language Processing Help Coders Any Time Soon?" *Proceedings of the 2001 AHIMA Convention.*

NLP systems that aid the process of coding medical records for billing purposes are beginning to appear on the commercial market. NLP systems have the potential to favorably affect the whole medical record coding cost and workflow equation, promising improved productivity and coding consistency without sacrificing quality. Several NLP vendors have invested heavily to bring their commercial offerings to market, and their efforts vary in terms of scope, methodology, delivery, and application. The 3M-sponsored study results presented herein will shed some light on the technology readiness of the NLP products from three vendors. The formative results presented suggest that NLP software is commercially viable, at least in some markets like radiology and emergency medicine. However, technology viability is only part of the challenge confronting these emerging ventures.

Zender, Anne. "From Coder to Knowledge Engineer." *Journal of AHIMA* 74, no. 7 (2003): 104.

This brief article summarizes one individual's change from a coding position to a nontraditional knowledge engineer. Knowledge engineer responsibilities include writing rules that identify medical terms and attaching corresponding ICD-9-CM codes. The software recognizes the language in the dictation and assigns codes based on the information in the rules. Therefore, the rules must represent all the possible terms a physician could use in describing a condition or diagnosis (e.g., "diabetes mellitus," "diabetes," "DM").

Available Research Testing NLP-based CAC Tools

Research: Year, Objective, Source	Findings	Conclusions

1995 Evaluation of the automated detection (using MedLEE) of six clinical conditions in narrative chest radiograph reports.[1]	• Natural language processing (NLP) had a sensitivity of 81 percent and a specificity of 98 percent. • Physicians had an average sensitivity of 85 percent and an average specificity of 98 percent.	• The natural language processor was not distinguishable from physicians and was significantly superior to all other comparison subjects. • NLP seems to have the potential to extract clinical information from narrative reports in a manner that will support automated decision support and clinical research.
1998 An evaluation that measures performance of different methods by comparing results of variations on one NLP coding engine (MedLEE).[2]	• NLP systems containing simpler pattern-matching algorithms using limited linguistic knowledge performed well (higher sensitivity by 5 percent to 6 percent) compared to those containing more complex linguistic knowledge. • The method based on recognizing a complete sentence had a significantly worse sensitivity than the others by 7 percent and a better specificity by .2 percent. • None of the methods had a significantly worse specificity.	• The methods based on analysis of sentence segments rather than complete sentences showed substantial increases in sensitivity while incurring only a small loss in specificity. • It will be important to perform other studies to see if the results of this study are generalizable.
1999 Comparison of the accuracy of automated (MedLEE) and manual coding in the data acquisition tasks of an ongoing clinical research study.[3]	• The overall accuracy of the automated system was 84 percent. • The overall accuracy of manual coding was 86 percent.	• The difference in accuracy between the two methods was small but statistically significant. • When the necessary information is available in electronic format, NLP is a highly efficient and appropriate method for clinical research applications. • Although there is a small decrease in the accuracy of the data as compared to traditional methods, automated systems may greatly expand the power of chart review in clinical research design and implementation.
2000 Study to test the accuracy and speed of one NLP coding engine (LifeCode) against both production coders and expert consultants in emergency medicine coding.[4,5]	• NLP agreed at least as often as other study participants in ICD-9-CM coding of emergency department charts. • There was a 48 percent reduction in human coders' coding time when using LifeCode as an assist. • Only moderate agreement was observed between any of the participants in E/M code assignment.	• Accuracy in E/M coding is relative. • LifeCode is as accurate as the human coders used in this study. • Automated programs like LifeCode have a significant advantage over human coders in being predictable, repeatable, and fast. • Given the shortage of experienced human coders, the tedious nature of medical record coding, and the inherent variability in human coding, one can

		argue that the future of automated coding systems looks bright.
2000 Comparison of the accuracy and productivity in assigning ICD-9-CM and CPT E/M codes in emergency room charts for three NLP companies (A-Life, CodeRyte, and Paradigm) to experienced and inexperienced human coders.[5-8]	• Surprising variability among experienced coders. • 91 percent to 93 percent perfect match between NLP system and 3M coders on CPT codes. • 57 percent to 61 percent perfect match between NLP system and 3M coders on ICD-9-CM diagnosis codes. • Codes derived by the NLP system agreed with at least one experienced coder nine out of ten times. • 33 percent reduction in coder workload is possible. • 48 percent increase in productivity with autocoding.	• NLP systems have the potential to affect the whole coding workflow andsystem. • NLP readiness: the technology is commercially viable in some markets (radiology, emergency department). • This was a laboratory test; it is uncertain if productivity gains would be real in the production environment. • A field test to validate laboratory findings in this study is needed. • Studies suggest NLP can improve medical record coding productivity and consistency without sacrificing quality.[3]
2000 Study to examine and compare the accuracy of three methods for automated coding of German-language free-text diagnosis phrases.[9]	• Correct ICD diagnosis found 40 percent for three-digit codes and 30 percent for four-digit codes. • Performance was significantly better when mapping free text to SNOMED.	• A satisfactory quality of automated encoding of free-text diagnoses into ICD has not yet been reached.
2001 Evaluation of NLP coding (using MedLEE) on ten years of narrative chest x-ray reports with comparison, on a subset of the reports, to manual coding by a physician, as well as historical HIM coding from the financial system.[10]	• NLP sensitivity was .81 and specificity was .99. • This was comparable to that previously reported for NLP and for expert coders. • A review of cases with a diagnosis of pneumothorax showed the NLP database (sensitivity 1.00, specificity .996) was more accurate than financial discharge coding (sensitivity .17, specificity .996).	• Accuracy of the NLP coding was comparable to that of manual coding and superior to accuracy of financial discharge coding for the diagnosis of pneumothorax. • Internal and external validation in this study confirmed the accuracy of NLP for translating chest x-ray narrative reports into a large database of information. • NLP can produce huge databases of coded clinical information and offers the potential to become an important tool for clinical research. • NLP automated coding of clinical reports is a better source of information than manual financial discharge coding for answering clinical questions, at least for some conditions. • The full potential of NLP will not be realized until it can be extended to complex narrative reports such as admission notes and discharge

		summaries. Although preliminary work shows promise, further research is needed to confirm accuracy.
2001 One discharge summary was coded manually, using MedLEE's (an NLP engine) coding output, and again using MedLEE coding to SNOMED RT. Outputs were then compared to assess SNOMED RT's capacity to represent clinically significant information and MedLEE's capacity to code into the SNOMED RT system.[11]	• Results for MedLEE's coding output compared to the "gold standard" (i.e., manual coding by the expert) showed 83 percent sensitivity, with a positive predictive value of 92 percent. • Results for MedLEE's coding in SNOMED RT showed 81 percent sensitivity, with a positive predictive value of 84 percent.	• SNOMED RT was shown to code with adequate completeness and comprehensiveness the information categories for which it was designed. Yet, 15 percent of the clinically significant information required coding beyond the scope of SNOMED RT's design. • MedLEE's coding to a standardized nomenclature offers, at best, a partial interoperability solution. • Larger and targeted evaluation studies are required.
2002 MedLEE extraction of comorbidities from discharge summaries and chest x-ray reports was compared to those manually reported in administrative data (ICD-9-CM codes).[12]	• Thirteen of the 19 comorbidities studied were underreported in the administrative data (ICD-9-CM). • The NLP system could be tuned to distinguish comorbidities from complications, while 30 percent of medical complications identified by ICD-9-CM codes lacked any documented evidence.	• The NLP system detected more comorbidities and was more accurate than ICD-9-CM codes in administrative data. • Comorbidities derived from NLP of medical records can improve ICD-9-CM–based approaches.
2003 Evaluation of the recall and precision of one NLP coding engine (LifeCode) in extracting findings from cancer-related radiology reports.[13]	• LifeCode had a recall of 84.5 percent and precision of 95.7 percent in the coding of cancer-related chest x-ray report findings.	• Despite the use of a modest-sized training set and minimal training iterations, when applied to cancer-related reports the system achieved recall and precision measures comparable to other reputable natural language processors in this domain, and precision was comparable to the human coder.
2003 Preliminary evaluation study of a method that extracts structured information from free-text surgical pathology reports.[14]	• 91 percent of 275 reports reviewed were coded at least so that all specimens and their critical pathological findings were represented in codes available in the UMLS.	• There is cautious optimism that this approach can be useful. • There is a need to better adapt the UMLS to the specific needs of surgical pathology reports.
2004 Evaluation of the recall and precision of one NLP coding	• Recall for extracting all terms: the NLP system was 84 percent; the experts ranged from 69 percent to 91 percent.	• The NLP method for extraction of relevant clinical information appeared to be comparable to or better than six experts.

engine (MedLEE) in automatically assigning codes to clinical sentences.[15]	• Precision: the NLP system was 89 percent; the experts ranged from 61 percent to 91 percent.	

Notes

1. Hripcsak, George, Carol Friedman, Philip O. Alderson, William DuMouchel, Stephen B. Johnson, and Paul D. Clayton. "Unlocking Clinical Data from Narrative Reports: A Study of Natural Language Processing." *Annals of Internal Medicine* 122, no. 9 (1995): 681–88.
2. Friedman, Carol, George Hripcsak, and Irina Shablinsky. "An Evaluation of Natural Language Processing Methodologies." *Proceedings of the 1998 AMIA Annual Symposium,* 855–59.
3. Elkins, Jacob S., Carol Friedman, Bernadette Boden-Albala, Ralph L. Sacco, and George Hripcsak. "Coding Neuroradiology Reports for the Northern Manhattan Stroke Study: A Comparison of Natural Language Processing and Manual Review." *Computers and Biomedical Research* 33, no. 1 (2000): 1–10.
4. Morris, William C., Daniel T. Heinze, Homer Warner, Jr., Aron Primack, Amy E. W. Morsch, Ronald E. Sheffer, et al. "Assessing the Accuracy of an Automated Coding System in Emergency Medicine" *Proceedings of the 2000 AMIA Annual Symposium,* 595–99.
5. Warner, Homer, Jr. "Will Natural Language Processing Help Coders Any Time Soon?" *Proceedings of the 2001 AHIMA Convention.*
6. Warner, Homer, Jr. "Good Isn't Enough." *Health Management Technology* 22, no. 6 (2001): 30–31.
7. Warner, Homer, Jr. "Can Natural Language Processing Aid Outpatient Coders?" *Journal of AHIMA* 71, no. 8 (2000): 78–81.
8. Warner, Homer, Jr., John Holbrook, David Evans, and Douglas Stetson. "Has Natural Language Processing Finally Arrived? Autocoding and Data Mining Examined." HIMSS Panel Presentation, session 49, February 2001.
9. Franz, Pius, Udo Hahn, Rudiger Klar, Stefan Schulz, and Albrecht Zaiss. "Automated Coding of Diagnoses—Three Methods Compared." *Proceedings of the 2000 AMIA Annual Symposium,* 250–54.
10. Hripcsak, George, John H. M. Austin, Philip O. Alderson, and Carol Friedman. "Use of Natural Language Processing to Translate Clinical Information from a Database of 889,921 Chest Radiographic Reports." *Radiology* 224, no. 1 (2002): 157–63.
11. Lussier, Yves A., Lyudmila Shagina, and Carol Friedman. "Automating SNOMED Coding Using Medical Language Understanding: A Feasibility Study." *Proceedings of the 2001 AMIA Annual Symposium,* 418–22.
12. Chuang, Jen-Hsiang, Carol Friedman, and George Hripcsak. "A Comparison of the Charlson Comorbidities Derived from Medical Language Processing and Administrative Data." *Proceedings of the 2002 AMIA Annual Symposium,* 160–64.
13. Mamlin, Burke W., Daniel T. Heinze, and Clement J. McDonald. "Automated Extraction and Normalization of Findings from Cancer-related Free-Text Radiology Reports." *Proceedings of the 2003 AMIA Annual Symposium,* 420–24.
14. Schadow, Gunther, and Clement J. McDonald. "Extracting Structured Information from Free Text Pathology Reports." *Proceedings of the 2003 AMIA Annual Symposium,* 584–88.
15. Friedman, Carol, Lyudmila Shagina, Yves Lussier, and George Hripcsak. "Automated Encoding of Clinical Documents Based on Natural Language Processing." *Journal of the American Medical Informatics Association* 11, June 7 (2004): 392–402.

CAC Web Resources

The following Web sites were discovered by the work group members in surveying the landscape for computer-assisted coding (CAC) use or products. This list is not intended to reflect all possible companies or resources associated with the process or software products that support it, but it will be helpful in a quest for more information on the topic.

AHIMA: www.ahima.org

A-Life Medical: www.alifemedical.com

American Academy of Family Physicians, PDA software: www.aafp.org/fpm/20020500/33codi.html

AssistMed: www.AssistMed.com

Chargemaster management:
www.chargemasters.com
www.craneware.com
www.codecorrect.com

CodeRyte: www.coderyte.com

Conceptual Health Solutions (OptiCode) now LingLogix (website under construction at www.lingologix.com).

Dictaphone: www.dictaphone.com/products/healthcare/publication.asp?buid=1

e-MDs (Evaluation and Management Coding for Physicians): www.e-mds.com or www.e-mds.com/emds/prodserv/emcoder.html

gMed: www.gmed.com

Healthcare Free Ware: www.healthcarefreeware.com/bill.htm

Health Language: www.healthlanguage.com

HSS: http://web1.hssweb.com/products/solutions_Providers.html

Inference Systems International, Inc. (Coding Pilot): www.tyeriver.com or www.midwestinformatics.com

KINETX Healthcare's PLATOCode coding engine: www.kinetx.com

Kiwi-Tek (CodeReady): www.kiwi-tek.com

Language and Computing:
www.landcglobal.com (home page)
www.landcglobal.com/pages/whitepapers.php (white papers)

MedQuist (CodeRunner): www.medquist.com

PLATO: www.platohealth.com, www.platocode.com

Practice Velocity: www.practicevelocity.com/cv_features.htm

ProVation Medical:
www.provationmedical.com (home page)
www.provationmedical.com/company/cc_news.asp (articles page)

QuadraMed: www.quadramed.com/web/solutions/him/

Textomy (Bibliography of Natural Language Processing in Biomedicine; National Research Council Canada): http://textomy.iit.nrc.ca/cgi-bin/BNLPB_ix.cgi

3M Health Information Systems: www.3m.com/us/healthcare/his/products/index.jhtml or www.3m.com/us/healthcare/his/products/records/data_dictionary.jhtml

Note: Links provided were accurate as of 5/1/05. They do not represent an endorsement of organizations, products, or services by the American Health Information Management Association.

Appendix G: Glossary of Terms

The terms defined here are meant to elucidate those used within the context of the computer-assisted coding practice brief and appendices. Online glossary resources are also provided to assist the reader in locating and understanding terms contained within additional articles and papers referenced in this practice brief.

Artificial intelligence (AI): Computational techniques to automate tasks that require human intelligence and the ability to reason.

Automated NLP engine: The part of a natural language processing application that performs the syntactic (structure, morphology) and semantic (meaning) processing of the text or speech.

Clinical coding engine: The part of a clinical coding application that contains the logic (algorithms) and knowledge (relationships) necessary to perform the encoding of concepts.

Code-assist model: A model of applying a computer-assisted coding tool that involves an initial screen against clinical documentation by the software tool, producing a preliminary set of draft codes, which are then reviewed, edited, and revised by a human coder to generate the final set of codes.

Cognitive linguistics: The study of the relationship between language and the human mind. Workers in this field seek to understand language as it relates to models of human thinking, interpreting language in light of the social and psychological contexts in which it is generated and understood. Cognitive linguistics can be contrasted with computational linguistics, in which workers use algorithmic or computer-based approaches to interpret language. Natural language processing typically employs the techniques of computational linguistics, although cognitive linguistics may also inform the process.

Computer-assisted coding (CAC): The use of computer software that automatically generates a set of medical codes for review/validation and/or use based upon clinical documentation provided by healthcare practitioners.

Computer linguistic competence: A useful concept for organizing a description of different types of applications, ranging from simple key word look-up (on the end of low competence) to semantic and contextual interpretation using a grammar and vocabularies (on the end of high competence).

Corpus: A large body of natural language text used for accumulating statistics on natural language text. The plural is *corpora*.

Electronic health record (EHR): The current term used to refer to computerization of health record content and associated processes.

Electronic medical record (EMR): A term that may be treated synonymously with computer-based patient record and/or electronic health record; often used in the US to refer to an electronic health record in a physician office setting or a computerized system of files (often scanned via a document imaging system) rather than individual data elements.

Free text: Alphanumeric data that are unstructured, typically in narrative form. Unstructured data are not processed uniquely by the computer system without application of natural language processing tools. Free text provides the benefit of expressivity and flexibility. However, information that is recorded as free text is significantly more difficult to use for data analysis, aggregation, and comparison. (See also *structured data*.)

Granularity: The level of detail.

Health Level Seven (HL7): An ANSI-accredited standards development organization created in the 1980s to develop standards for healthcare computer applications to share data. (See www.hl7.org.)

Informatics: A field of study that focuses on the use of technology for improving access to and utilization of information. Health informatics is the systematic study of information in the healthcare delivery system—how it is captured, retrieved, and used in making decisions—as well as the tools and methods used to manage this information and support decisions.

Knowledge management: Capturing, organizing, and storing knowledge and experiences of individual workers and groups within an organization and making this information available to others in the organization.

Levels of knowledge (in NLP): Natural language is described as encompassing the various levels of knowledge as follows:

- Phonetic: constructing words from basic sounds
- Morphological: constructing words from subunits (e.g., friend + ly = friendly)
- Syntactic: constructing sentences from words
- Semantic: deriving meaning from sentences
- Pragmatic: adding meaning from a sentence's context
- World: background, cultural, common sense information that adds meaning to a sentence

Linguistics: The scientific study of language, which may be undertaken from many different aspects, for example, sounds (phonetics) or structures of words (morphology) or meanings (semantics).

Machine learning: Subspecialty of artificial intelligence concerned with developing methods for software to learn from experience or extract knowledge from examples in a database.

Natural language processing (NLP): A range of computational techniques for analyzing and representing naturally occurring text (free text) at one or more levels of linguistic analysis (e.g., morphological, syntactic, semantic, pragmatic) for the purpose of achieving human-like language processing for knowledge-intensive applications. (See also *Levels of knowledge in NLP*.)

Normalization: A formal process to standardize various representational forms so that expressions that have the same meaning will be recognized by computer software as synonymous in a data search. This may involve elimination of various kinds of punctuation signs and coordinating conjunctions, conversion of letters from lower to upper case, and so on. The process must be formalized with consistent rules applied systematically to maintain data integrity. For example, "fracture of heel" may be normalized as "heel fracture."

Semantic grammar: A formal definition of a language that uses concepts from a particular domain of discourse to specify acceptable expressions in that language. This is distinct from a syntactic grammar, in which a language is defined in terms of the parts of speech that comprise it. For example, a semantic grammar intended for use in the interpretation of chest x-ray reports may specify that valid expressions involving the word "lung" may include reference to laterality (e.g., left, right, or both).

Semantics: The meaning of a word or term. (See also *Levels of knowledge in NLP*.)

Statistical NLP: A group of techniques relying on mathematical statistics and used in natural language processing, for example, to find the most likely lexical categories or parses for a sentence. Often, the techniques are based on frequency information collected by analyzing large corpora of sentences in a single language, to find out, for example, how many times a particular word ("dog," perhaps) has been used with a particular part of speech. The sentences in the corpus have usually been tagged in some way (sometimes manually) so that the information about the part of speech, each time each word is used, is known. Statistical NLP may also be referred to as Boolean NLP.

Structured data: Documentation of discrete data using controlled vocabulary rather than narrative text.

Structured input: A form of data entry that captures data in a structured manner (e.g., point-and-click fields, pull-down menus, structured templates, macros).

Syntax: The format or structure of data. (See also *Levels of knowledge in NLP*.)

Unstructured data: See *free text*.

References

Amatayakul, Margaret K. *Electronic Health Records: A Practical Guide for Professionals and Organizations*, *2nd Edition.* Chicago: AHIMA, 2004.

Coiera, Enrico. "Health Informatics Glossary," In *Guide to Health Informatics*, *2nd Edition,* 1997. Available online at www.coiera.com/glossary.htm.

Open Clinical. Available online at www.openclinical.org.

Wilson, Bill. "The Natural Language Processing Dictionary." Updated June 25, 2004. Available online at www.cse.unsw.edu.au/%7Ebillw/nlpdict.html.

Additional Online Resources

Health Level Seven. "Glossary of Terms." January 2002. Available online at www.hl7.org.

Partnership for Health Information Standards, Canadian Institute for Health Information. "Glossary of Terms." Available online at www.cihi.ca or http://secure.cihi.ca/cihiweb/en/partner_glossary_e.html

Tabar, Pamela. "The Latest Word." Healthcare Informatics Online, January 1998. Available online at www.healthcare-informatics.com/issues/1998/01_98/glossary.htm.

van Bemmel, J. H., and Mark A. Musen, Editors. "Handbook of Medical Informatics." Web Site Version 3.3, 1999. Available online at www.mieur.nl/mihandbook/r_3_3/glossary/us/glossary.htm.

Wilson, Bill. "The AI Dictionary." Updated June 15, 2004. Available online at www.cse.unsw.edu.au/%7Ebillw/aidict.html.

Appendix H [figure 1]: CAC Timeline

Notes

1. Hripcsak, George, Carol Friedman, Philip O. Alderson, William DuMouchel, Stephen B. Johnson, and Paul D. Clayton. “Unlocking Clinical Data from Narrative Reports: A Study of Natural Language Processing.” *Annals of Internal Medicine* 122, no. 9 (1995): 681–88.

2. Elkins, Jacob S., Carol Friedman, Bernadette Boden-Albala, Ralph L. Sacco, and George Hripcsak. "Coding Neuroradiology Reports for the Northern Manhattan Stroke Study: A Comparison of Natural Language Processing and Manual Review." *Computers and Biomedical Research* 33, no. 1 (2000): 1–10.
3. Warner, Homer, Jr. "Will Natural Language Processing Help Coders Any Time Soon?" AHIMA National Convention proceedings, October 2001. Available online at www.ahima.org.
4. Warner, Homer, Jr. "Good Isn't Enough." *Health Management Technology* 22, no. 6 (2001): 30–31.
5. Warner, Homer, Jr. "Can Natural Language Processing Aid Outpatient Coders?" *Journal of AHIMA* 71, no. 8 (2000): 78–81.
6. Warner, Homer, Jr., John Holbrook, David Evans, and Douglas Stetson. "Has Natural Language Processing Finally Arrived? Autocoding and Data Mining Examined." HIMSS panel presentation, session 49, February 2001.
7. Mamlin, Burke W., Daniel T. Heinze, and Clement J. McDonald. "Automated Extraction and Normalization of Findings from Cancer-Related Free-Text Radiology Reports." *Proceedings of the 2003 American Medical Informatics Association (AMIA) Annual Symposium*, 420–24.
8. Schadow, Gunther, and Clement J. McDonald. "Extracting Structured Information from Free Text Pathology Reports." *Proceedings of the 2003 AMIA Annual Symposium*, 584–88.
9. Friedman, Carol, Lyudmila Shagina, Yves Lussier, and George Hripcsak. "Automated Encoding of Clinical Documents Based on Natural Language Processing." *Journal of the American Medical Informatics Association* 11, no. 5 (2004): 392–402.
10. Lorence, Daniel P., and Awad Ibrahim. "Disparity in Coding Concordance: Do Physicians and Coders Agree?" *Journal of Health Care Finance* 29, no. 4 (2003): 43.

References

"Clinical Data Specialist." In *Evolving HIM Careers: Seven Roles for the Future*. Chicago: AHIMA, 1999.

AHIMA. "Natural Language Processing as a Means to Increase Productivity." Audio seminar, May 13, 2004. Available online at http://campus.ahima.org/audio/2004seminars.html.

Beinborn, Julie. "Automated Coding: the Next Step." *Journal of AHIMA* 70, no. 7 (1999): 38–43.

Boelle, Pierre-Yves, Antoine Flahault, Laurent Letrilliart, and Cecile Viboud. "Automatic Coding of Reasons for Hospital Referral from General Medicine Free-text Reports." *Proceedings of the 2000 AMIA Annual Symposium*, 487–91.

Bowman, Jim, and Mary Stanfill. "Physicians Cast Wary Eye at Computer-assisted Coding." *Journal of AHIMA* 75, no. 8 (2004): 76–77.

Chuang, Jen-Hsiang, Carol Friedman, and George Hripcsak. "A Comparison of the Charlson Comorbidities Derived from Medical Language Processing and Administrative Data." *Proceedings of the 2002 AMIA Annual Symposium*, 160–64.

Evans, David, John Holbrook, Douglas Stetson, and Homer Warner Jr. "Has Natural Language Processing Finally Arrived? Autocoding and Data Mining Examined." HIMSS panel presentation, February 2001.

Franz, Pius, Udo Hahn, Rudiger Klar, Stefan Schulz, and Albrecht Zaiss. "Automated Coding of Diagnoses—Three Methods Compared." *Proceedings of the 2000 AMIA Annual Symposium*, 250–54.

Friedman, Carol, George Hripcsak, and Irina Shablinsky. "An Evaluation of Natural Language Processing Methodologies." *Proceedings of the 1998 AMIA Annual Symposium*, 855–59.

Garvin, Jennifer, and Valerie Watzlaf. "Current Coding Competency Compared to Projected Competency." *Perspectives in Health Information Management* 1, no. 2 (2004). Available online at www.ahima.org.

Hagland, Mark. "Revolution in Progress: How Technology Is Reshaping the Coding World." *Journal of AHIMA* 73, no. 7 (2002): 32–35.

Hieb, Barry. "NLP Basics for Healthcare." Gartner Research, August 16, 2002.

Hripcsak, George, John H. M. Austin, Philip O. Alderson, and Carol Friedman. "Use of Natural Language Processing to Translate Clinical Information from a Database of 889,921 Chest Radiographic Reports." *Radiology* 224, no. 1 (2002): 157–63.

Johns, Merida. "A Crystal Ball for Coding." *Journal of AHIMA* 71, no. 1 (2000): 26–33.

Lussier, Yves A., Lyudmila Shagina, and Carol Friedman. "Automating SNOMED Coding Using Medical Language Understanding: A Feasibility Study." *Proceedings of the 2001 AMIA Annual Symposium*, 418–22.

Morris, William C., Daniel T. Heinze, Homer R. Warner Jr., Aron Primack, Amy E. W. Morsch, Ronald E. Sheffer, et al. "Assessing the Accuracy of an Automated Coding System in Emergency Medicine." *Proceedings of the 2000 AMIA Annual Symposium*, 595–99.

Schnitzer, Gregory L. "Natural Language Processing: A Coding Professional's Perspective." *Journal of AHIMA* 71, no. 9 (2000): 95–98.

Schnitzer, Gregory L., and Mary H. Stanfill. "Outwit, Outlast, Outcode: Surviving in the Autocoding Era." *Journal of AHIMA* 72, no. 9 (2001): 102–4.

Stollman, Neil, and Matthews, Kathleen. "Positive Productivity, Better Billing." *Health Management Technology*, August 2002: 22–26.

Zender, Anne. "From Coder to Knowledge Engineer." *Journal of AHIMA* 74, no. 7 (2003): 104.

Prepared by

AHIMA computer-assisted coding e-HIM work group:

Sean Benson
Jim Bowman, MD, MSM, FACS
Gloryanne Bryant, BS, RHIT, CCS
Win Carus, PhD
Ray Chien, MS
Colleen Deighan, RHIT, CCS
Christy DeLellis, RHIT
Melissa Ferron, RHIA, CCS
James Flanagan, MD
Margaret M. Foley, MBA, RHIA, CCS
Gail Garrett, RHIT
Darice M. Grzybowski, MA, RHIA, FAHIMA
Robert A. Jenders, MD, MS, FACP
Kathy M. Johnson, RHIA
Karen Karban, RHIT, CCS
Diana McWaid-Harrah, MS, RHIA, CCS, CPC
Janice Redden, CCS, CPC-H
Gregory L. Schnitzer, RN, CCS, CCS-P, CPC, CPC-H, RCC, CHC

Vickie Schraudner, RHIA
Rita Scichilone, MHSA, RHIA, CCS, CCS-P
Mary H. Stanfill, RHIA, CCS, CCS-P
Robin K. Stults, RHIA
David Sweet, MLS
Adriana Van Der Graaf, RHIA, CHP, CCS
Joe Weber, MS, MBA
Martha Weiner, RHIA

Acknowledgments

Assistance from the following individuals is gratefully acknowledged:
Peter L. Elkin, MD
Kenneth Macklem
Senthil Nachimuthu, MD
Aldo Tinoco, MD
Pat S. Wilson, RT(R), CPC

The computer-assisted coding e-HIM work group was supported by a grant to the Foundation of Research and Education from MedQuist.

Article citation:
AHIMA e-HIM™ Work Group on Computer-Assisted Coding. "Delving into Computer-assisted Coding" (AHIMA Practice Brief). *Journal of AHIMA* 75, no.10 (Nov-Dec 2004): 48A-H (with web extras).

>
