Computer-assisted coding and natural language processing

Without changes to current coding technology and processes, ICD-10 adoption will be very difficult for providers to absorb, due to the added complexity and coding overhead of ICD-10.

Computer-assisted coding, based on NLP, can help reduce the coding burden.

Richard Wolniewicz, PhD
Division Scientist,
Natural Language Processing
3M Health Information Systems

Executive summary

Natural language processing (NLP) promises to reduce costs and improve quality for healthcare providers. By processing text directly with computer applications, an organization can leverage the wealth of available patient information in clinical documentation to improve communication between caregivers, reduce the cost of working with clinical documentation, and automate the coding and documentation improvement processes. Where other applications of technology often require caregivers to change their existing, proven processes to accommodate the technology, NLP allows applications to work with the most valuable form of clinical communication: **the clinical narrative.**

This paper introduces and describes NLP—also referred to as computational linguistics or "text mining"—from the healthcare perspective, and particularly addresses the technology in the context of **computer-assisted coding**. There is a need to demystify NLP and improve expectations for it, because today's healthcare organization can clearly benefit from this powerful tool.

Without changes to current coding technology and processes, ICD-10 adoption will be very difficult for providers to absorb, due to the added complexity and coding overhead of ICD-10. Computer-assisted coding, based on NLP, can help reduce the coding burden. However, confusion still exists about what NLP is and what it can and cannot do, so it is vital that healthcare organizations and their technology leaders understand the NLP questions—and the answers—that other industries have addressed.

At the core of many NLP conversations is the choice between rule-based and statistical NLP models. NLP engines consist of multiple components, each performing a specific operation to enhance the understanding of the text. Choosing between rule-based and statistical NLP models drives important application workflow considerations, which in turn leads directly to the need for informative accuracy measurements for NLP systems.

3M Health Information Systems approaches NLP and computer-assisted coding with a clear goal: Incorporate the best features of multiple NLP approaches to deliver strong solutions for auto-suggested coding and clinical documentation improvement. 3M solutions emphasize client productivity, return on investment (ROI), and solid preparation for the increased documentation specificity demanded for ICD-10 coding. As this paper will demonstrate, it's time for healthcare leaders to focus less on jargon and technical details and instead seriously consider the accuracy of results and the benefits an NLP application can provide, especially for the implementation of ICD-10.

3M

Applications of NLP in everyday life

Although researchers have been studying NLP in clinical settings since the 1960s, the U.S. healthcare industry is a late adopter of NLP. Other industries have successfully applied NLP to their business processes, and health care can learn from their experiences. Examples of familiar NLP applications are helpful in introducing both the concepts and the technologies at play in NLP.

NLP is widely used in modern computer systems, but in the most successful cases, it has disappeared entirely into the background, where it is adding significant business value, but is largely invisible to the user. Most of us are amazed as we watch IBM® Watson™ appear to "listen" to "JEOPARDY!" questions and "reply" with the right answer in seconds. But many of us do not realize that NLP is also "behind the scenes" in more mundane technology, such as voice-mail phone trees, educational essay test-scoring systems, and even e-mail spam detection software.

E-mail spam detection

One common NLP example that has evolved quickly during the past decadeis e-mail spam detection, and it offers several useful lessons we can apply to the challenge of computer-assisted coding in health care.

Many of us remember early e-mail spam filtering, which started as little more than a "black list filter" for our e-mail. Whenever a spam e-mail was received, we would go through the process of tagging the e-mail as spam, and selecting how we wanted to filter similar messages in the future: by the specific e-mail address, all e-mail from a given domain, etc.

Today, few of us would tolerate such a manual process against the flood of spam in our inboxes. Instead, a modern e-mail system such as Google Mail™ webmail service looks not only at fixed fields such as the send-to address, but also at the text of the e-mail itself, seeking patterns that distinguish spam vs. legitimate e-mail. In that sense, the spam classifier reads the e-mail.

More importantly, a modern spam classifier can learn how to spot spam. The system observes millions of features in the text, such as appearances of particular phrases, and how often those features are associated with spam. If many users tag e-mail with the phrase "Nigerian bank account" as spam, the system will start to predict future e-mail as spam when it sees that phrase (in this case, a trigram or three-word phrase).

The e-mail spam example is simple and familiar, but is also closely related to computer-assisted coding: The decision to assign a specific code to a patient record is similar (but much more complex) to the problem of assigning a spam flag to an e-mail, and many of the same considerations and techniques apply.

Some valuable NLP "take-aways" from the spam example to consider in the context of computer-assisted coding include:

- There is a natural progression from rule-based systems to more complex capabilities that combine rules and statistics.
- The spam classifier truly starts to learn when statistics come into the picture; until then it is encoding the knowledge of an expert (you, the e-mail user) as rules. However, even the statistical system still uses rules when available.
- The rate of learning increases with more data. While it took months for early spam classifiers to learn from a single user's e-mails, Google Mail™ webmail service's hundreds of millions of users mean that new spam can be classified in seconds. We see significant benefits when we share our data to train the spam classifier, though it is important that none of us be forced to allow everyone to read our e-mail. Sharing data to train NLP differs significantly from sharing data with other users.
- As the NLP becomes more sophisticated, the technology integrates naturally and is less visible.

Major types of NLP applications

Many people look at NLP technology and see a "black box" they feel is beyond their understanding. Quite contrary to that perception, all NLP solutions use multiple components that work together—and the NLP solutions on the market today do not differ as greatly on the technical level as many vendors would have you believe.

Keep in mind that NLP is a tool that can reside inside almost any text processing software application, and virtually every NLP technology on the market today is built according to the same essential principles and fundamental functionality. When considering the NLP choices before you, evaluate the results of the overall application first, because software vendors today often assign commercial or even trademarked names to what is simply a basic component that exists in every NLP solution.

There are many categories of NLP; these five cover many of the most common applications:

- **Machine translation.** This type of application today receives significant amounts of research funding and emphasis. Example: Google Translate™ translation service, which translates the contents of foreign language web pages (and can detect and identify the language on the page even if you cannot).
- **Speech recognition.** Familiar to the healthcare industry for its use in automated transcription and dictation software, this type of NLP application processes plain speech input and is widely used to service clients on telephone voice response systems.
- **Question answering.** This type of NLP application also works with plain speech input and uses it as the basis for an information search. The most famous example: IBM® Watson™, the reigning JEOPARDY! champion. Watson does not literally "understand" JEOPARDY! questions or the texts it uses to answer them; rather, it extracts relevant information from the text, queries multiple evidence sources, and integrates the results to identify the most confident response.
- **Knowledge extraction.** A more sophisticated and complex use of NLP, this type of application must understand and interpret input in order to produce valid information as its output. Two examples: **computer-assisted coding of medical record data** and online market research software. In the medical record world, a code contains the meaning of what happened to a patient, so the NLP engine must be able to analyze patient data and extract knowledge from whatever related and relevant material it can access (lab reports, physician notes, discharge summary reports, EHR, etc.). Online market research software also requires understanding: To answer a marketer's question, "What are people thinking and saying about Brand X Pizza today," the application must be able to read the text of blogs and other social media, understand the context of statements ("Brand X Pizza tastes good" vs. "Brand X Pizza built two new stores last month"), and determine sentiment. Sentiment is one of the more complicated problems in NLP today.
- **Classification.** This type of NLP application neatly sorts and organizes information for us into relevant categories. Examples: the e-mail spam filters described above as well as the Google News™ news service, which sorts content into logical "buckets," such as sports headlines, baseball scores, book reviews, oil supply news from the Middle East, and so on.

"The best way to evaluate the performance of a language model is to embed it in an application and measure the total performance of the application."

— Daniel Jurafsky and James H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, Second Edition*

NLP techniques

The other common denominators of all NLP applications are the steps that NLP takes to derive its output. Most NLP systems include common steps such as the following:

- **Tokenization.** What are the words and punctuation signs that the system needs to pay attention to? For example, is "U.S.A." one token? Three tokens? Or six? How do we identify these tokens? This step is usually rule-based and fairly straightforward.
- **Sentence and structure detection.** In this step, NLP in a computer-assisted coding application needs to identify the sections of the clinical narrative that represent the patient history, present diagnosis, etc. The system must also figure out when the occurrence of a period means the end of a sentence and when it means the end of an abbreviation like "Dr." A period following "ER" does not necessarily mean the end of a sentence. There are a myriad of things to detect, ranging from the paragraph breaks to the section headings of a radiology report. Special rules have to be identified and although this seems very easy to us humans, this is surprisingly difficult for a computer to consistently get right.
- **Part-of-speech (POS) tagging.** English is often a very frustrating language to study when it is not your first language. Even identifying the "part of speech" for a word is complicated. For example, most verbs in English can also be nouns; you can "run" to the store, but you can also go out for a run, or ride a bobsled down a run. For NLP to understand a word like "run" it must be able to understand and interpret the surrounding terms and understand the context in which the word is used.
- **Normalization.** In the world of health care, this is an absolutely essential step in NLP development and one in which ontology—the ability to define what something is—plays a vital role.[1]

  Consider how many ways a term like "COLD" can be interpreted in a clinical environment:

  – COLD can be an acronym for Chronic Obstructive Lung Disease

  – A patient in shock can tell an ER nurse that he feels cold (physical temperature)

  – A mother can call the family doctor and describe her child's symptoms as indicative of a very bad "cold"

  There are many ways of looking up all the meanings for one term or considering all the different ways people can say the same thing (as in using "CHF" for congestive heart failure). One approach is to maintain huge lists of synonyms and huge lists of words that look and sound the same but have very different meanings. Not only is context "King" in this case, but ontology driven by a comprehensive data dictionary is essential. Such a dictionary should also include all possible abbreviations, variants, alternative expressions, and even misspellings and slang terminology used to describe a single concept.
- **Named entity resolution.** This step calls for the NLP engine to look up all important words (mostly nouns) in a dictionary/ontology resource and resolve the meanings into a concrete concept, such as the name of a drug, an anatomical part, etc.
- **Parsing.** This step basically answers the question, "What is the structure of this sentence?" Students struggle with "parsing" or diagramming sentence structure in an English class, and it is not any easier for NLP, especially in clinical narratives that often involve dealing with a very complicated context. For example, "heart attack" can appear in a clinical narrative, but does it refer to the patient's own experience or her father's medical history—the only way a human or a computer can figure that out is through the parsing step.

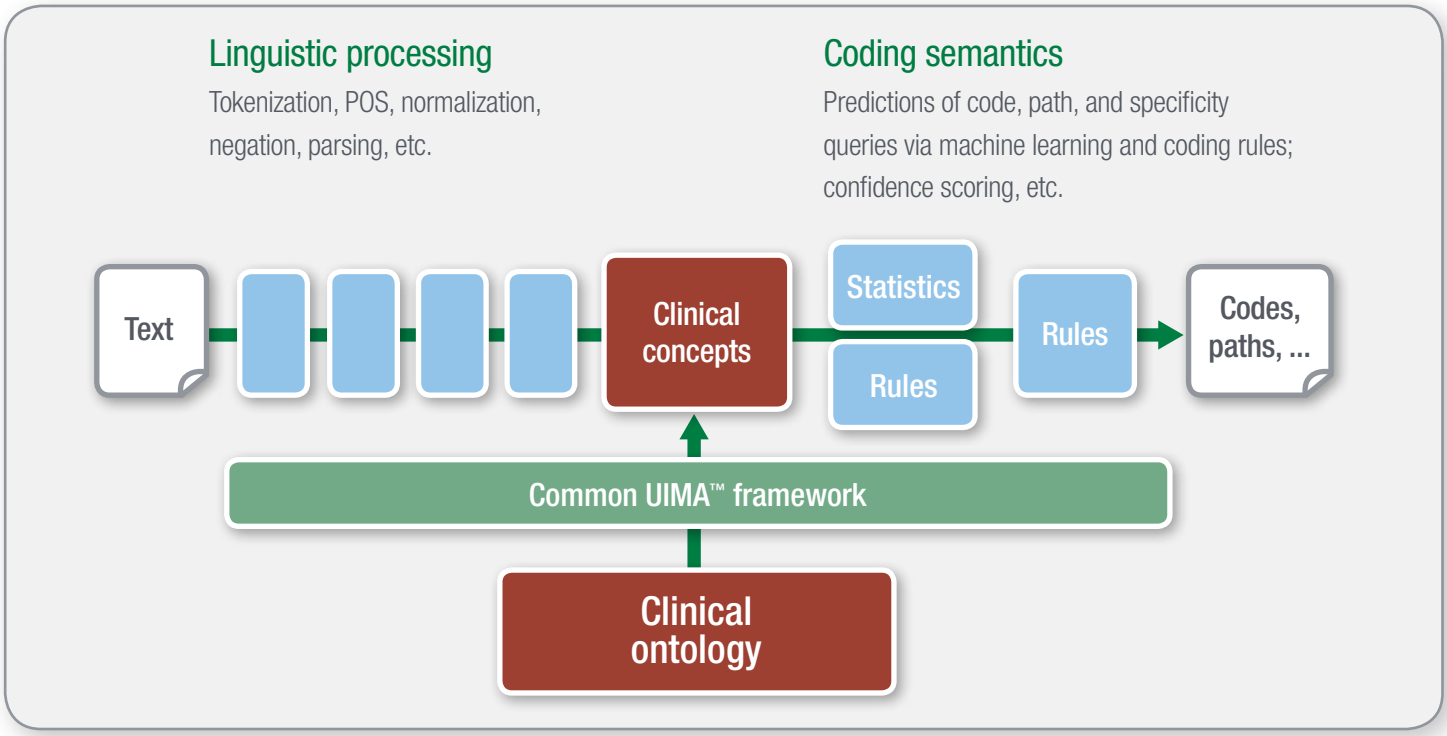

Figure 1 visually summarizes the multiple components of an NLP engine and the steps in which unstructured text is handled.

- **Negation and ambiguity detection.** How often do we encounter a phrase and wonder if it is negating an idea or introducing doubt and ambiguity? Consider the phrase, "The patient has pneumonia" vs. "Tests results preclude pneumonia" vs. "Possible indications of pneumonia." English has thousands of ways of saying something did not happen, so this adds complexity to NLP development. Ambiguity is similar to negation when it comes to coding healthcare data–a physician may suspect a patient has pneumonia but does not have enough data to declare a pneumonia diagnosis. The result? "Possible pneumonia" appears in the patient's chart and the NLP engine must be sure not to erroneously code pneumonia.

- **Semantics.** In many respects, all the previous steps culminate in this one: understanding the meaning of words (semantics) where the previous steps deal with combining them (syntax). An NLP computer-assisted coding application must examine linguistic evidence and arrive at a final output in the form of a code, a classification, etc. And that output must make sense to the human being on the receiving end of the information. In computer-assisted coding, it is reasonable to assume that the NLP developed for inpatient coding should have a different set of "competencies" than the NLP developed for outpatient coding—the problems of each are very different.

  It is also at this step where we see the most divergence in implementation: Here actual clinical content is applied as coding rules (and the quality of that content will determine accuracy), and some systems also incorporate statistical models for coding. 3M Health Information Systems employs appropriate technology for both semantics and syntax in its computer-assisted coding NLP solution.

Searching for a “standard” in NLP: UIMA™

UIMA—Unstructured Information Management Architecture—is a standardized and integrated NLP solution available as open source. UIMA is a technical platform that runs inside a computer process and serves to integrate a pipeline of software components, each of which executes a single NLP step such as those discussed above.

Developed by IBM beginning in the 1990s, UIMA was used to implement IBM® Watson™, and is also the foundation of Mayo Clinic’s cTAKES (clinical Text Analysis and Knowledge Extraction System), an open-source clinical NLP system. Released as open-source under an Apache license in 2006, UIMA has also been adopted as a standard by OASIS (Organization for the Advancement of Structured Information Standards). Because of its prominence, UIMA is a technology familiar to most researchers in the field, and as such is a safe choice for applying NLP in health care when you need to integrate capabilities from multiple sources.

3M Health Information Systems has adopted UIMA as the standard for its computer-assisted coding NLP solution. Many modern clinical NLP innovations are based on UIMA, such as the SHARP Area 4 research sponsored by the ONC and building on capabilities from the Open Health Natural Language Processing (OHNLP) Consortium.[2] Systems built on UIMA can benefit directly from this rapidly-evolving research.

The challenge: Mining unstructured data

In the Apache *UIMA Conceptual Overview*, the authors summarize what healthcare researchers and industry thought leaders have known—and been frustrated by—for decades: "Unstructured information represents the largest, most current and fastest growing source of information available to businesses and governments."*

So how do we extract the gold and silver from a seemingly bottomless data mine? The same authors point to the value of developing applications that can exploit a variety of technologies: "In analyzing unstructured content, UIM [unstructured information management] applications make use of a variety of analysis technologies including:

- Statistical and rule-based natural language processing (NLP)
- Information retrieval (IR)
- Machine learning
- Ontologies
- Automated reasoning and
- Knowledge sources (e.g., CYC, WordNet, FrameNet, etc.)"*

*Both quotes are taken from Section 2.1. of the *UIMA Conceptual Overview* found at this URL:

http://uima.apache.org/downloads/releaseDocs/2.2.0-incubating/docs/html/overview_and_setup/overview_and_setup.html#ugr.ovv.conceptual

Rule-based vs. statistical models in NLP

Historically, "rule-based NLP vs. statistical NLP" was a major debate in the 1990s, and one of the big "divides" in the NLP research and development communities. After the dust settled, the importance of statistical approaches was clear, and indeed, the most effective NLP applications today combine both rules and statistics.

Rules

Rule-based NLP essentially means a group of experts write deterministic rules to implement the mappings in the NLP components. In some cases, a set of rules is all you need. For example, most NLP systems use a rule-based approach for tokenization, because it works so well for that step in the NLP process.

Rules are an excellent way to encapsulate certain types of expert knowledge, such as linguistic expertise for expressing negation. However, parsing is difficult to codify, so most modern NLP parsers are statistical. For computer-assisted coding NLP semantics, deep expertise in informatics is essential to performing the named entity resolution step, as well as nosologists for addressing coding rules.[3]

An important example of where rule-based NLP can be advantageous occurs whenever new regulations are introduced and coded examples and documents are not available for "training" a statistical system. In this case, a timely solution employs an appropriately trained and experienced human expert to intervene and update the system with the new or revised rules.

Perhaps the most important advantage to rule-based NLP is that it can be faster to get a system launched covering the most common cases and then improve coverage over time. Moreover, rules are written in ways experts understand, so it is easier to diagnose errors.

Continued on next page >

< Continued from previous page

Statistics

Statistical NLP means the system learns the mappings for the NLP components as statistical relationships by processing many examples. Examples of statistical-based content include the results you see from the e-mail spam filter examples cited earlier. A spam filter is looking at statistics across a huge group of customer input, so there is minimal expert involvement.

The accuracy of a statistical model goes up along with the volume of data available for learning. A statistical model will even learn in a production environment, so as coders use an NLP computer-assisted coding system, it learns the codes most often selected. One consequence is that the initial performance of the deployed system will be lower than the performance 6-12 months after deployment, which is important to consider when evaluating the results.

Statistical methods require a very large annotated data set to train on and become proficient. They can continue learning with lower costs compared to rule-based systems. But the requirement for a large training set makes them inappropriate for situations with few data examples. A rare code which a hospital sees once a year is probably not a good candidate for statistical prediction. On the other hand, in an outpatient environment with a narrower code set and a higher coding volume, the statistical approach is a natural fit.

Admittedly, statistical engines are more of a "black box," and the root cause of errors can be harder to diagnose. Correcting errors requires finding new linguistic features to help the statistical engine discern relationships, which is a more complex task than looking at edge cases in a rule. Ultimately, however, the ability of a system to be more accurate for common cases and to learn in production is a huge advantage in dynamic environments such as health care.

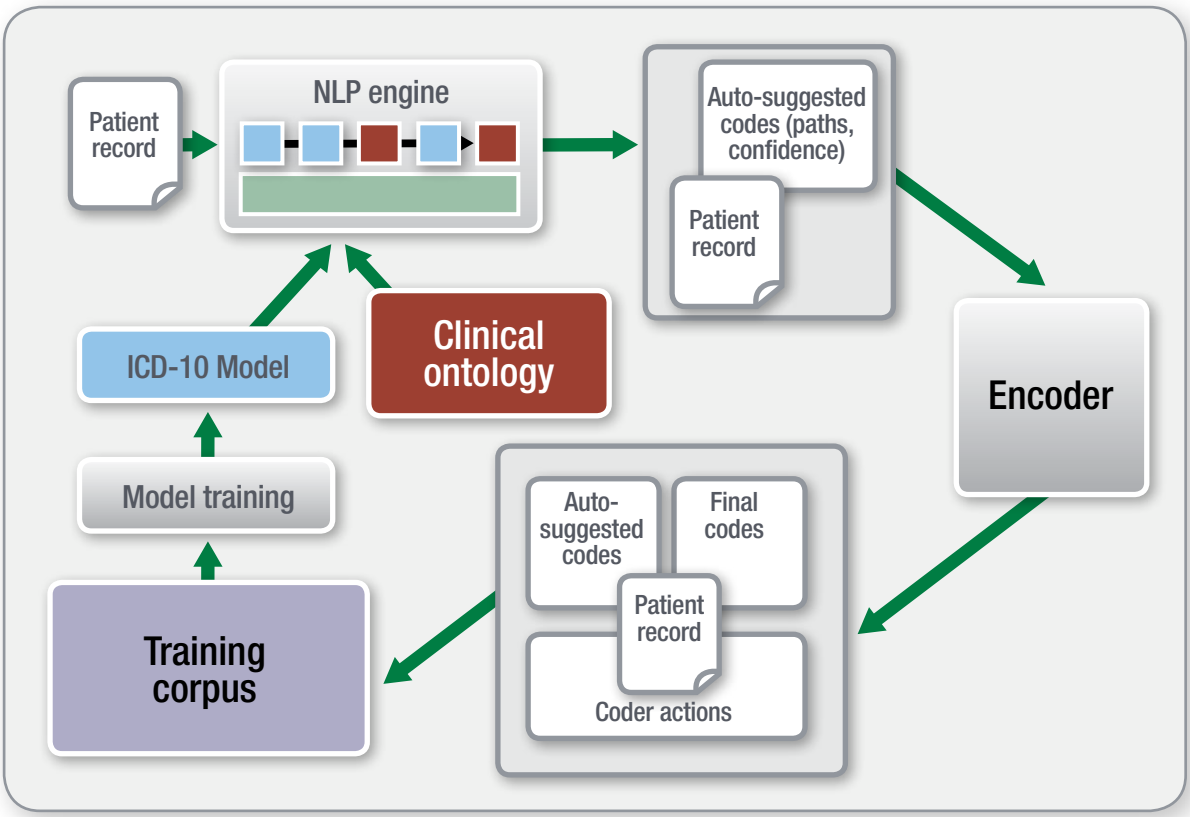

Figure 2 Feedback loop for effective, ongoing "training" in 3M's computer-assisted coding NLP system

Figure 2 shows a "computer-assisted coding feedback loop" and how NLP technology can fit within the larger context of a computer-assisted coding solution. Appreciating the power of machine learning and the ability of a statistical model to learn and improve over time, 3M Health Information Systems is applying this approach in its NLP computer-assisted coding solution.

The hybrid approach

Today, most NLP systems are a combination of the two models, thus adopting a "hybrid" approach. For example, as cited earlier, tokenizers are almost always rule-based, but where data volumes number in the tens-of-thousands of examples, statistical semantics tends to outperform for a given level of effort. On the flip side, small sample sizes encourage rule-based techniques.

It is not uncommon to see rule-based tokenization, parts of speech, and negation combined with statistical parsing and semantics.

A hybrid solution can open up the best of both worlds: the compressed time-to-market feature of a rule-based approach with the improved scalability across a broad user base of statistical methods. At the very least, many clinical researchers and medical informaticists today agree that rule-based and statistical approaches "are complementary."[4]

Such details don't generally matter to an end user of an NLP application, with the possible exception of learning: Can a system learn from its mistakes in production and self-correct? As a practical matter, only statistical systems can do that, and this characteristic may matter to a healthcare organization when evaluating a computer-assisted coding project.

There is no one-size-fits-all solution. Instead, the best combination of NLP techniques vary from application to application, and even from site to site based on variations in language, structure of documents, etc. Remember to focus on accuracy in the context of your specific application workflow.

Accuracy measurements

Many people want to try to reduce NLP comparisons to a single "accuracy" number, but this is usually misleading. Not all errors are created equal, and when looking at an automated classifier, it is vital to consider type-I and type-II errors. (Other common terms, precision and recall, are identified based on type-I and type-II errors.) For computer-assisted coding, assume that the "default" is not to assign a code (in statistics-speak, this is the "null hypothesis"):

- A type-I error occurs when a code is assigned that should not be present. The cost of a type-I error may show up as failed audits.
- A type-II error occurs when a code is missed that should have been assigned. The cost of a type-II error is under-billing.

Depending on your organization's workflow, it may be easier to detect and correct a type-I or type-II error. At the very least, the business workflow you build will differ for each type of error.

For a good discussion of errors and statistical accuracy, including the important related concepts of specificity and sensitivity, refer to http://en.wikipedia.org/wiki/Sensitivity_and_specificity. Another useful description of all these terms can be found at http://en.wikipedia.org/wiki/Confusion_matrix.

Conclusion

NLP is a multi-faceted technology that precludes the notion of a "best" NLP. "Best" for NLP depends on the application it is used in and the context in which it is applied. When considering an NLP solution for your organization's computer-assisted coding needs, consider the following:

- Evaluate the NLP based on its accuracy measures rather than the specific NLP algorithm it employs
- Know your application before evaluating NLP
- Ask yourself how a computer-assisted coding NLP application will fit into the overall coding workflow at your organization
- What is the tradeoff between type-I and type-II errors that can maximize the ROI?
- What is the existing level of human performance in your coding personnel? Does the NLP need to match it to deliver value?
- Is the workflow structured to capture feedback for learning? If so, will the system's NLP learn and improve over time, and if so, how does that improve the ROI for your organization?

When it comes to the question of statistical vs. rule-based NLP, avoid the false dichotomy: Virtually every NLP engine has some rules, and statistical techniques are clearly better in high-data-volume applications.

Rules will outperform statistics when data volumes are low (think rarely-used ICD-10 codes). But statistics will out-perform when data volumes are high (think most commonly used ICD-10 codes). Finally, think about whether and how the NLP will learn and adapt itself in your organization's production environment.

To learn more

If you would like additional information on 3M's approach to NLP and computer-assisted coding technology, as well as our **3M™ 360 Encompass™ System**, please contact your 3M representative today. You may also call us toll-free at 800-367-2447 or explore our website at www.3Mhis.com.

Footnotes

[1] Ontology for 3M Health Information Systems calls for the use of the 3M™ Healthcare Data Dictionary (HDD) to "normalize" healthcare data. The 3M HDD is an asset that will prove essential in the 3M solution for ICD-10 computer-assisted coding. The modular nature of NLP means that 3M solutions are able to leverage the best solution for each of the components—such as the 3M HDD for ontology.

[2] For more information on the SHARP (Strategic Health IT Advanced Research Projects) Area 4 research from the Mayo Clinic and the Office of National Coordinator for Health Information Technology (ONC), see http://healthit.hhs.gov/portal/server.pt/community/healthit_hhs_gov__home/1204 and http://informatics.mayo.edu/sharp/index.php/Main_Page.

[3] The 3M Nosology Team along with the 3M Healthcare Data Dictionary team of medical informaticists bring expertise and over 30 years of experience to the development of 3M's NLP computer-assisted coding solution. The advantage of integrating 3M's proprietary coding logic is a significant draw for other software developers working on computer-assisted coding NLP solutions.

[4] For additional articles and discussions of NLP in medical applications, see the September 2011 issue of *JAMIA (Journal of American Medical Informatics Association)*. This issue also includes a more detailed introduction to NLP entitled "Natural language processing: an introduction" *(J Am Med Inform Assoc* 2011;18:544-551) by Prakash M. Nadkami, Lucila Ohno-Machado, and Wendy W. Chapman); the cited phrase appears on p. 545.

3M Health Information Systems

3M Health Information Systems works with providers, payers, and government agencies to anticipate and navigate a changing healthcare landscape. 3M provides healthcare data aggregation, analysis, and strategic services that help clients move from volume to value-based health care, resulting in millions of dollars in savings, improved provider performance, and higher quality care. 3M's innovative software is designed to raise the bar for computer-assisted coding, clinical documentation improvement, performance monitoring, quality outcomes reporting, and terminology management.

For more information, visit www.3Mhis.com or follow **@3MHISNews** on Twitter.

3M

Health Information Systems

575 West Murray Boulevard
Salt Lake City, UT 84123
U.S.A.
800 367 2447
www.3Mhis.com

Published 04/15
70-2011-6545-6