The practical application of NLP technology for Computer Assisted Coding: A White Paper

Stuart Covit, Andrew B. Covit, M.D, M. Elliott Familant, PhD

ARTIFICIAL
MEDICAL
INTELLIGENCE

EMscribe®

INTRODUCTION

To effectively understand the proprietary EMscribe® NLP technology, we must first define the existing nomenclatures that are prevalent when we discuss Computer Assisting Coding utilizing Natural Language Processing and remember to relate it to the NLP task at hand, mainly that of medical coding and medical abstraction of patient records.

The most widely used definition of Computer-Assisted Coding (CAC) is the use of computer software to generate a set of medical codes from documentation. The results are then presented to the coder for verification, validation and review.

Natural Language Processing (NLP) was a term first used in the 1950's encompassing the field of computer science and linguistics focused on the interactions between computers and human language. NLP is often considered a sub-field of artificial intelligence. The term "natural language" is used to distinguish human languages from computer languages and, in real life scenarios computer programming is used to process human language. NLP encompasses both text and speech and this is relevant as it relates to the industry offerings, many of which have emanated from the various speech recognition vendors.

With NLP-Natural Language Processing, newer terms have been spawned, NLU-Natural Language Understanding, NLC- Natural Language Comprehension, LP-Language Processing, and DLU-Deep Language Understanding and, though driven by vendor exclusivity, are all common acronyms and terms associated with the processing of human language. Many of these terms and the associated acronyms have been developed by industry vendors in the speech language processing space. AMI embraces the term NLP since it is the oldest descriptor and is used as a general association that classifies our use of human language to process ICD9/10 codes or any other coding system from free text for the patient encounter.

For the sake of this white paper we will associate the acronym NLP as Language Processing in whatever technique or form we describe. Types of Electronic NLP methodology are typically divided into four categories.

The first is a Statistics Approach which generates a set of electronic codes based on the probability that a particular medical term can be associated as correct with that particular code. Some of the nomenclature associates this as "confidence coding". The problem with this methodology is that it can become a huge undertaking to train any application to cover the entire medical corpus. Any workable statistics approach needs to first analyze huge amounts of records which is viewed as data in each specialty in order to securely derive high confidence levels for codes selected for every individual specialty. The methodology is useful for small pools of variable choices and becomes less suitable for large bodies of information such as the entire ICD9 coding system. This helps explain why those vendors who solely utilize a statistics approach have never been able to completely capture all specialties in medicine. Using the statistical NLP approach to create a commercialized product, vendors have been limited to processing specialties like radiology and cardiology as examples, or services such as labs or Emergency Department. In fact, several vendors have their origins in processing individual specialties relating to clinic-based coding. This has all been noted by the F.O.R.E. AHIMA report dated, July 11, 2005.

An Ontology Approach can be described as utilizing a set of definitions from a formal vocabulary, in this case, the formal body of medicine in its entirety and mapping that body of data to a representative set of standardized terminologies. The problem with this methodology stems from the practical application of NLP for coding and the effort required to extract codes accurately from a dynamic body of medical knowledge with its ever changing medical terminology. Although many vendors claim to be using their own proprietary ontologies as their language processing reference point, as of this date, there are no actual applications for ontology-based CAC/NLP in commercial use. Normally, large banks of data center computers are required with many years of programming to cover an entire body of knowledge and even though multiple programmers can be employed to cram huge amounts of algorithms into computers, using ontology-based NLP in Computer Assisted Coding has yet to see a practical or nimble application for the day to day tasking of coding and abstracting medical records.

Some vendors embrace a Logic-Tree Approach where users are guided through a series of questions to arrive at an end result. This methodology was popular in language programming in the 1980's, and this legacy approach can require a very time consuming effort to generate and approve suggested codes from a patient record.

The Rules Method utilizes software algorithms to guide a behavior or action. AMI believes that its proprietary technology loosely falls into this category and that there are no real weaknesses if properly implemented. Proper implementation requires time consideration and nimbleness of the architecture. Rules do effectively address large bodies of knowledge such as medicine and allow for ever evolving and necessary dynamic change. For the specifics of coding, the only weaknesses lie in how the rules software looks at language, the software's change or modification flexibility, and how the rules approach is applied in an effective and rapid manner.

THE AMI METHOD:

We first coordinated 19 practicing physician experts nationwide to help provide input on the specific grammars used by physicians when treating patients. All the subtlety of how medicine is practiced needed capturing, going beyond just medical terminology matching. *Medical terminology capture was the first step for our NLP.* To do this we had to include every major specialty in order to cover the entire medical corpus. Cardiology, Nephrology, Immunology and 17 other areas of expertise, were utilized. Second, we developed an efficient method to assimilate grammars, essentially ignoring existing traditional linguistics and NLP methodologies.

This technique that was developed employs the use of *Pivot Grammar architecture. The application of Pivot Grammars was the second step for our NLP.*

Coders were then contracted to provide coding guideline input based on payer reimbursement patterns. *Capturing outside coding guidelines was the third step for our NLP.* Once the solution was in Beta, at Robert Wood Johnson University Hospital, we devised the *NLP Overlay Approach* which addressed need to look at the patient chart in its entirety in addition to the specific document at hand. On October 17, 2005, we went live at the site and commercialized our NLP approach.

It was intrinsic that our approach should not be complicated or esoteric, while being nimble enough to make adjustments based on dynamic coding environments to satisfy hospital specific coding practices. On the other hand, our approach does not portray itself as overly simplistic such as word spotting or pattern matching, either of which would result in a degradation of the desired effective Natural Language Processing.

It is AMI's view that NLP use in healthcare does not mandate a focus on the individual grammatical sentence components, as other Rules-based approaches would suggest. Focusing on sentence structure, Noun, Verb, and Adjective, requires extensive syntactical overhead, the result of which increases computing resource consumption. Conversely, AMI's NLP proprietary coding engine does not focus on formal semantics, linguistics, lexical analysis, pragmatics, or discourse analysis.

Definition: *Pivot Grammar(s)* are a set of rules corresponding to the positioning of words or medical terms within a sentence or paragraph relative to a medical term (or token) and the effect of these; either pre- or post- clauses on those medical terms. For the sake of English, let's define a clause as a group of words made up of a subject and predicate, though the internal grammatical structure of the clause becomes irrelevant for the purpose of Pivot Grammars affecting a desired result. An example of an actual Pivot Grammar Rule may look like this:

Medical Term, Trigger, (token) corresponds to the ICD 9 or 10 code. If a found expression in the sentence corresponds to the Medical Term, Trigger (token) and a word or words in the pre or post clause relates to that trigger (token), then the Medical Term, Trigger (token) equals, "do not code".

Simple examples are,

1. (Cancer 199.10) (History)
2. (Exploratory laparotomy 54.11) (*is going for a*)

EMscribe® Engine Methodology

Chart Drill Down Methodology

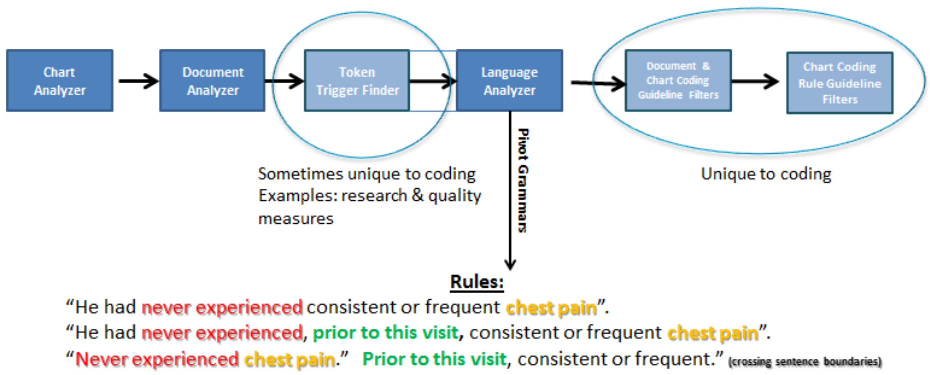

Fig.1 NLP Pivot Grammars and *NLP Overlay Approach*= Layered NLP processing

The NLP Overlay methodology addresses the importance of analyzing the grammar in its environment, taking into account the chart, document, paragraph and then the individual sentence. The tokens or trigger finder seen in figure 1 refer to the capture of the way physicians speak using medical terms. The AMI medical terms dictionary goes beyond the terms listed in any medical coding corpus, including slang, misspelled terms and words, and common acronyms. These extra entries can be both site specific and across installed platforms, capturing any physician language variant that may be associated with a disease state. As an example, there are 49 variants for the word Coumadin (warfarin), with those variants including every possible misspelling culled from live hospital or clinic environments. Another example, "decreased HIF" (Higher Integrated Functioning) associated with dementia, ICD9 code 290, a term commonly used by physicians only at a large academic hospital site in New York City. Further, document and chart coding rule and guideline filters include those external coding guidelines developed by CMS and other validated and reputable coding guideline sources, including payers for coding submission, also including referenced guidelines from sources like Coding Clinic, etc.; all represented by the two circled blue boxes on the right in figure 1 reflecting input from expert external coding sources.

To really understand the effectiveness of Pivot Grammars and the efficiency of how they change the behavior in a medical document, please refer to the bottom of figure 1 diagram. Under Rules here are three example sentences in the diagram. Note that the medical term associated with the ICD 9 code, known as the token or trigger as *chest pain*, 786.59. Example 1 points out the Pivot Grammar "*never experienced"* which tells the coding engine, "do not code". Example 2 demonstrates how the Pivot Grammar, "*prior to this visit"* alters the effect of "*never experienced"*, and therefore tells the coding engine, "code". In fact, this is a fairly complex concept because "*prior to this visit"* there's been no *chest pain* but, for this visit, there is, inferring that this is why he is being seen for the complaint of *chest pain* now. Example three, though not a typical example of how a physician might speak, shows how Pivot Grammars can cross sentence boundaries while still applying the need to code for *chest pain*.

Figures 2, 3, and 4, below, illustrate the direct impact of a Pivot Grammar on the coding of an actual patient record. Figure 2 illustrates how coding of the procedure, Exploratory Laparotomy (54.11) is modified by the pivot grammar phrase "*is going for a*" and not coded.

The coding engine would normally interpret *"She is going for an exploratory laparotomy for removal of this mass"* as a future event that hasn't taken place yet, therefore, "do not code".

To demonstrate the effect of Pivot Grammars on coding, Figure 3 shows a screen shot of the term "*is going for a*" in the "do not code" Pivot Grammar dictionary. Using a simple "*is going for a*" search within the Windows Operating System, we *find*, *backspace*, and *remove* the Pivot Grammar, "is going for a", which normally suppresses the medical terminology pre- or post- clause (or phrase). Figure 4 shows the resulting effect of removing the Pivot Grammar. Removing, "*is going for a*", from the, "do not code", Pivot Grammar dictionary causes that procedure, *exploratory laparotomy* to be underlined as a token or trigger and correlated with the associated code, 54.11. Of course, coding rules state that you can't code for a future event so this would never happen in a real coding environment. However, the example shows a specific Pivot Grammar example and the effect of the *pre-clause* in this record on the coding outcome.

The time that it takes process and remove the, "*is going for a*", Pivot Grammar in this illustration is under 2 minutes. This further demonstrates the nimbleness of using the Pivot Grammar approach to accommodate all coding including site specific coding guidelines. As can be seen, the coding engine does not require software programming language for changes but is written in clear English for ease of use. Access to these dictionaries containing the Pivot Grammars requires only typical operating system tools and no programs within programs or special environment for source code accessibility are needed.

IMPRESSION: The patient has a large pelvic mass, with omental inflammation and caking. She is going for a exploratory laparotomy for removal of this mass. The patient has agreed to have staging, with sampling of lymph nodes, biopsies and removal of appendix/omentum and only if necessary to remove part of the bowel and reconnect. However,

Fig.2 Pivot grammar example: "*is going for a*"= "Do not code".

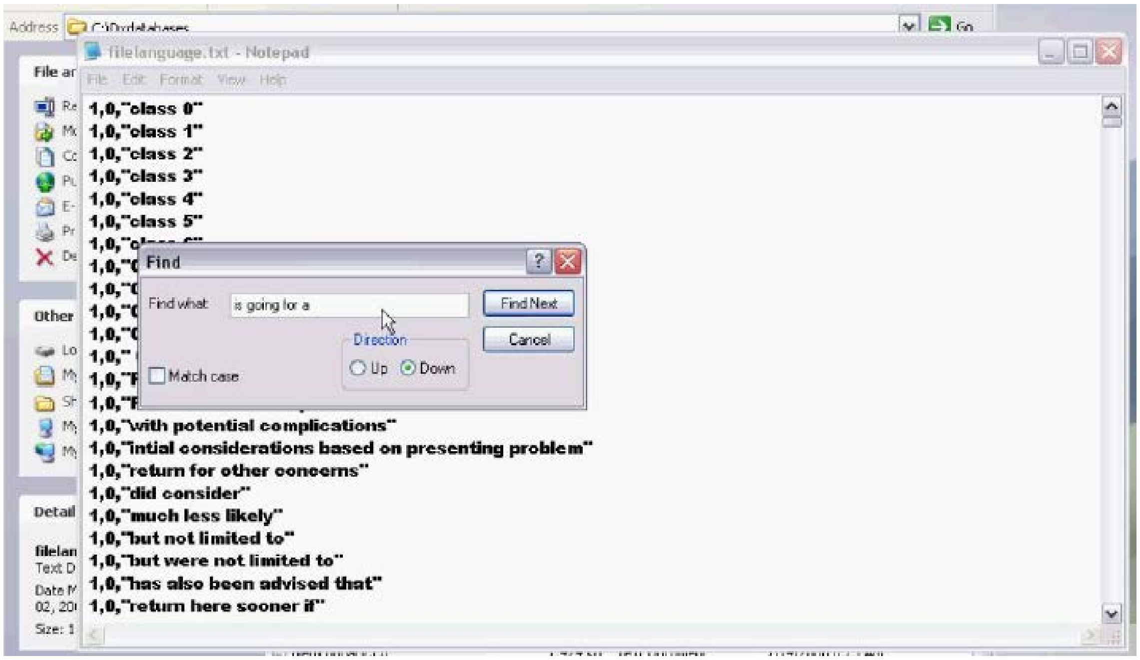

Fig.3 EMscribe® Do not code dictionary finding and removing "*is going for a*" Pivot Grammar.

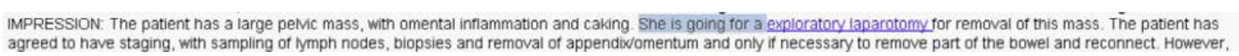
IMPRESSION: The patient has a large pelvic mass, with omental inflammation and caking. She is going for a exploratory laparotomy for removal of this mass. The patient has agreed to have staging, with sampling of lymph nodes, biopsies and removal of appendix/omentum and only if necessary to remove part of the bowel and reconnect. However,

Fig.4 Result: "*is going for a*" pivot grammar removed, exploratory laparotomy=coded

Another useful attribute of this proprietary technology is in the ability to offload and disseminate certain medical terminology attributes to a backend database which allows other vendor partners utilizing the EMscribe® core technology to provide support for the medical terminology portion. Tools can be applied that can give the appropriate administrative end user access to add locally relevant terminology to certain dictionaries. This empowers the site with a tool to augment and assist in rapid new terminology assimilation into the NLP. The goal of this off-loading methodology is to make NLP end user friendly and perhaps more importantly to de-mystify the current nomenclature with NLP being used in a practical setting.

ACCURACY: RECALL AND PRECISION

Engine validation and accuracy has been a looming comparative question that continues to affect all NLP vendors. The concepts of precision and recall as applied to the EMscribe® coding engine open a very compelling discussion. For the purposes of this white paper, we must associate coding accuracy with coder judgment. Unfortunately, since there is no coding answer key or primer, coders are identified as the expert reference point for what the engine produces as right or wrong. Obviously, this referencing method has flaws from both the training and human side, since the literature suggests that there is no real right answer key in coding. However variable or inaccurate, coders have operated in real coding environments with their supported credentials and education on *how to code* so we must use their expertise as our real life guide.

When technology gets introduced into common practice, terminology often becomes misused, inferring or implying aspects and characteristics that are misleading, not true or not relevant. An example of this involves the concept of automation. It is true that specific processes for any CAC application using NLP are automated, that is, tasks that would normally be manual or analog in nature become automatic, but it is a semantic stretch to imply that a given application using NLP is automated. So what does true NLP CAC automation really mean? CAC Automation defines the process of coding which self-operates robotically without the need for human intervention for immediate verification, validation and review of the coding process. The task of coding automation which relies solely on the NLP technology for coding a patient record while delivering a 100% favorable coding outcome is the only real useable definition. Simply, this means that the records do not touch coders but go direct to billing and that these records bypass human verification validation and review. CAC automation is a true robotic function. The NLP engine output provides the highest possible confidence, based on coder feedback, yielding 100% accuracy. No other NLP technology provides this level of confidence. This level of accuracy is site specific and well defined when applied in a commercial environment. Not all records are true automation candidates nor are there any generalizations for quality and quantity of the automation candidates from site to site. A CAC application using our NLP technology connected to a hospital system as the front end can always be audited by the user, reviewing those specified record types to check on the coding performance of this work pool. There are three fundamental pre-requisites to this true automation ability. The first is to create a pool or bucket containing certain record types that are automation candidates. These records originate from outpatient hospital or clinic areas, and include cases such as simple treat and release cases, radiology, emergency department and labs. The second is to carefully verify, validate and review with the assigned coding staff at the site, the output, make the necessary adjustments prior to go-live for automation output to achieve and mimic the output that coders would produce. Lastly, to accomplish this, the coding engine must have the necessary flexible architecture to provide the environment to rapidly create the required site specific medical terminology and Pivot Grammars for a true automation bucket. The effort requires some input for the given coding staff, usually a few weeks in duration; however the return on investment is direct and irrefutable similar to benefits and efficiencies seen with any robotic functionality seen in other industries.

Figure 5 is an example of a CAC automation candidate process robotically through our automation bucket.

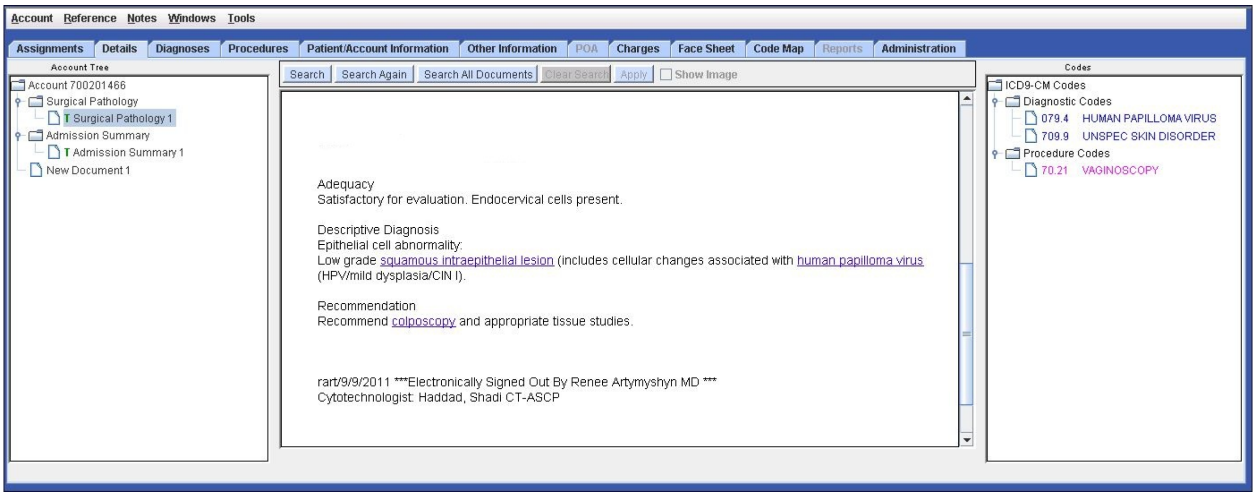

Fig.5 Real Automation Example 100% Accuracy with no human coder input.

There are currently a number of hospital sites which utilize automation. One 240 bed hospital site processes over 20,000 records per month, while at another hospital site with over 650 beds just 4400 records per month are processed for automation. The point is that there is typically a greater impact reflected in existing hospital processes than on the amount of automation records processed. Automation quantity is not driven by the size of the site or existing technology at a site, but rather by existing workflow and business practices of the individual customer site.

The CAC automation queue can help to boost the overall benchmark recall numbers for the CAC solution and, depending on the site, *accuracy* can be as high as 94% which is the highest the coding engine has attained so far. However, based on the structure and concept of our NLP, there is no reason why this number can't reflect even higher percentages. An "out of the box" NLP coding engine, un-tuned to hospital-specific coding practices, is 89.7% accurate*. These numbers have been generated from our experiences with our NLP engine performance since January 2005. The high recall parameter indicates that the NLP out of the box doesn't miss very much. However, the lower precision number of 74.9% is reflects the coding engine's pick up of some irrelevant data, such as names, acronyms, idiosyncratic sections of exclusion documentation such as headers. These are aspects that every application must be modified to handle when installing at a new site. To mitigate the lower precision number, all CAC/NLP applications require some level of adjustment, referred to as tuning or training for each site before the application is ready for go-live.

*Accuracy = coder-initiated code changes as the benchmark. 89.7% is the recall number, precision number is 74.9%

APPLICATION

Our NLP methodology, with its nimble approach allows for the relatively quick switching from one coding system to another since the Pivot Grammars and Medical Terminology and associated terms have already been captured. There is no need to apply General Equivalency Mapping components or cross-walking data to convert from one coding system to another. The work is simply one of association; the more specific the terminology the better the coding engine will perform with higher levels of code specificity. Essentially, from a technical standpoint, it is replacing one dictionary for another, such as replacing or swapping out the ICD9 coding system for the ICD10 coding system. To refine the engine from one coding system to another, we take our existing knowledge of the most frequently used Medical Terms and slang and associate the new coding system with these most frequently used medical terms; working from most frequent to less frequent. This allows the coding system to be ready for deployment early as a release candidate and immediately work well within hospital environments. The engine can continue to assimilate and tune *in situ* while being used.

NLP allows for many different use cases once medical terminology and its slang are captured. With respect to coding, NLP can be used to generate suggested the principal and admit diagnosis, which can help provide hospital administration with much needed data for concurrent coding capability including real time reimbursement case analysis, as well as Present on Admission (POA) status, and Hospital Acquired Condition (HAC) events.

Beyond coding, AMI NLP technology can be deployed as a single platform for a variety of purposes including Clinical Documentation Improvement (CDI), Quality indicator abstractions, NLP data abstraction, NLP for clinical research applicability, and upstream predictive speech & text applications.

While there are many CDI solutions and organizations looking to capture a burgeoning market, no company to date can apply NLP rules written that address CDI hospital requirements. Roughly 2000 or more custom rules with Pivot Grammars and Medical Terminology are designed to address desired documentation flags. Examples of the types of NLP CDI querying include tracking the number of patients admitted each day with congestive heart failure, identifying where they are admitted within the facility, or what their length of stay is depending on nursing unit or attending physician, etc. and whether documentation was provided to justify the services provided. Another example could be the tracking of reimbursement patterns based on antibiotics used on patients admitted with community acquired pneumonia and whether the physician documentation provided justification for the antibiotic usage.

Quality indicator abstractions can be approached through data storage, specifically the synergy between NLP and a medical database already in place with CAC, allowing the NLP engine to comb through the database to extract query results. Examples of this involve the tracking length of stay for ventilator dependent respiratory failure patients and the actual ICU location, surgical vs. medical, while monitoring decubitus and urinary tract infection rates, or monitoring beta blocker usage in known chronic congestive heart failure patients upon admission vs. upon discharge.

Probably the most exciting usage of NLP for direct improvement potential of patient care is in the application of NLP for clinical research. A recent research abstract example is given in Figure 6, was developed using clinical hospital data at Robert Wood Johnson University Hospital over a five year period. The relevance of this type of NLP research ability can't be understated as healthcare institutions become sensitized to the impact of particular and admission diagnosis states on disease outcomes allowing for the adjustment of the treatment or the creation of more aggressive management protocols to more effectively deal with these disease states. In this case, the result could be that there is more immediate focus on proactive hypertension control to mitigate increased length of stay for disease.

Impact of a Hypertension Diagnosis on the Clinical Outcome of Patients Admitted with Coronary Artery Disease, Heart Failure or Diabetes Mellitus to an Academic Medical Center: 5 Year Retrospective Analysis Using Natural Language Processing Techniques. AB Covit[1,2,3], J Bershad[1,2], ME Familant[3], [1]Robert Wood Johnson University Hospital, [2]UMDNJ-Robert Wood Johnson Medical School, New Brunswick, NJ, [3]Artificial Medical Intelligence Inc, Eatontown, NJ
Hypertension (HTN), affecting 60 million people in the United States, creates a significant burden on the nation's health care system requiring often multi-drug therapy and increasing risk for stroke and heart attack. New, innovative, cost effective methods of defining the impact of therapy or illness on already overburdened healthcare facilities are needed to better deploy resources and optimally manage disease process. Using the EMscribe® computer assisted coding system database employed at an academic healthcare center from January 2005 through December 2010, we investigated the effect of a hypertension diagnosis on disease outcomes and length of stay (LOS) in patients with a diagnosis of coronary artery disease (CAD), diabetes mellitus (DM) or heart failure (HF). The total adult inpatient population over age 18 in the EMscribe® database was 284,831 (152,489 male, 132,342 female). Orthogonal groups of CAD (19,637), DM (15,582) and HF (10,107) were identified. The presence of a diagnosis of HTN was determined in the CAD (14,877, 76%), DM (11,988, 77%) and HF (7,186, 71%) groups. HTN was associated with increased mortality for CAD (1.3% vs. 0.5%), DM (1.6% vs. 0.4%), and HF (2.7% vs. 1.1%), p<.001. The presence of HTN carried an odds ratio requiring a skilled nursing facility at discharge of 1.35 in CAD, 1.45, DM and 1.42, HF. A HTN diagnosis was associated with increased LOS compared to those without in DM (6.0 vs. 5.3 days, p<.008), CAD (5.4 vs. 4.6 days, p <.0001) and HF (8.0 vs. 7.3 days, p <.005). Total time required for database and statistical analyses was under 4 hours.
In conclusion, the use of EMscribe® database analysis can provide a rapid means of generating disease data and a platform to evaluate prospective hospital resource utilization to optimize patient care. In this study, HTN is associated with increased hospital mortality and resource utilization manifested as increased LOS and discharge disposition to a skilled nursing facility in CAD, DM, and HF. Finding that a HTN diagnosis on admission is associated with increased clinical care burden supports data driven decisions to provide more aggressive management to patients with HTN admitted with CAD, DM and HF.

Fig.6 Clinical abstract example

Types of clinical research example ideas that can be applied by the user with our NLP are complications arising from antibiotic usage or the incidence of urinary tract infection depending on baseline training of the inserting provider, transfusion rates in newly diagnosed lymphoma patients depending on induction chemotherapy protocol, or factors in incidence of hyponatremia in patients presenting with chronic heart failure. These are just a few examples of the limitless NLP capability on the research front.

NLP data abstraction goes beyond established criteria using a backend database for customized correlational business intelligence or for clinical research academic purposes. Instead of typical queries, custom Pivot Grammars with Medical Terminology are applied to comb data elements. Some examples of this are an analysis of hospital length of stay for patients begun on warfarin vs. darbigatran for pulmonary emboli or with early mortality rate, antibiotic usage and long term outcome for patients diagnosed with septic shock depending on location of onset comparing surgical vs. medical services.

The NLP potential for upstream predictive speech & text requires zero training and is designed to help mitigate high output error rates. Examples include the ability to allow transient medical student usage for EMR documentation, or providing for increased speed and accuracy for real time speech recognition.