JAMIA Original Investigations

Research Paper ■

Automated Encoding of Clinical Documents Based on Natural Language Processing

CAROL FRIEDMAN, PHD, LYUDMILA SHAGINA, MS, YVES LUSSIER, MD, GEORGE HRIPCSAK, MD, MS

Abstract **Objective:** The aim of this study was to develop a method based on natural language processing (NLP) that automatically maps an entire clinical document to codes with modifiers and to quantitatively evaluate the method.

Methods: An existing NLP system, MedLEE, was adapted to automatically generate codes. The method involves matching of structured output generated by MedLEE consisting of findings and modifiers to obtain the most specific code. Recall and precision applied to Unified Medical Language System (UMLS) coding were evaluated in two separate studies. Recall was measured using a test set of 150 randomly selected sentences, which were processed using MedLEE. Results were compared with a reference standard determined manually by seven experts. Precision was measured using a second test set of 150 randomly selected sentences from which UMLS codes were automatically generated by the method and then validated by experts.

Results: Recall of the system for UMLS coding of all terms was .77 (95% CI .72–.81), and for coding terms that had corresponding UMLS codes recall was .83 (.79–.87). Recall of the system for extracting all terms was .84 (.81–.88). Recall of the experts ranged from .69 to .91 for extracting terms. The precision of the system was .89 (.87–.91), and precision of the experts ranged from .61 to .91.

Conclusion: Extraction of relevant clinical information and UMLS coding were accomplished using a method based on NLP. The method appeared to be comparable to or better than six experts. The advantage of the method is that it maps text to codes along with other related information, rendering the coded output suitable for effective retrieval.

■ **J Am Med Inform Assoc.** 2004;11:392–402. DOI 10.1197/jamia.M1552.

The electronic patient record contains a rich source of valuable clinical information, which could be used for a wide range of automated applications aimed at improving the health care process, such as alerting for potential medical errors, generating a patient problem list, and assessing the severity of a condition. However, generally, such applications are not yet feasible because much of the information is in textual form,[1] which is not suitable for use by other automated applications. In addition, for the information to be effectively utilized by a broad range of clinical applications, it must be interoperable among different clinical applications, different clinical domains, and different institutions. The Clinical Document Architecture (CDA) effort,[2] which is a key effort that addresses interoperability, has established the importance of a common well-delineated representational model for each document in a document class (e.g., radiology reports) and also for the concepts in the documents. An effective method that automatically maps relevant text in clinical documents into standardized concepts would help further the CDA effort.

Many reports have been published[3–10] on methods that automatically map clinical text to concepts within a standardized coding system, such as the Unified Medical Language System (UMLS). These approaches, which primarily have been applied to indexing applications, use methods based on string matching, statistical processing, or linguistic processing, utilizing part of speech tagging (i.e., classifying words as nouns, verbs, etc.) followed by identification of noun phrases. However, for accurate retrieval of clinical information, coding concepts in the documents in isolation of other information, such as modifiers, is not enough because the underlying meaning of the concepts is significantly affected by other co-occurring concepts. For example, in a patient document, concepts may be modified by negation (i.e., *without significant tremors*), by temporal information (i.e., *previously admitted for pneumonia*), by family history (i.e., *family history of heart*

Affiliation of the authors: Department of Biomedical Informatics, College of Physicians and Surgeons, Columbia University, New York, NY.

Supported by grants LM06274 and LM7659 from the National Library of Medicine.

Correspondence and reprints: Carol Friedman, PhD, Department of Biomedical Informatics, Columbia University, 622 West 168 Street, VC-5, New York, NY 10032; e-mail: <friedman@dbmi.columbia.edu>.

Received for publication: 02/04/04; accepted for publication: 04/13/04.

disease), or by modifiers denoting that the event may not have actually occurred (i.e., *exposed to tuberculosis, admitted for syncope workup*). For many automated applications, such as alerting applications, to perform effectively, modifiers of the concepts also have to be accounted for. Some work concerning identification of negated concepts has been reported,[11,12] which is critical for many automated applications. However, other types of modifiers were not detected.

In this report we describe a coding method that uses advanced NLP techniques to generate structured encoded output consisting of findings and corresponding modifiers. The method attempts to find the most granular corresponding code by matching the structured output, which was generated as a result of parsing the sentences, with structures in a coding table in which the structures have been associated with codes. A code is obtained by successfully matching a finding along with modifiers, based on an assumption that a match consisting of a finding with the most modifiers is preferable to a match consisting of the same finding with fewer modifiers because it is the most specific. For example, for the sentence, *Status post myocardial infarction in 1995*, the UMLS code **C0856742** (corresponding to the UMLS concept **post mi**) would be obtained along with a **date** modifier with the value **1995**. Note that the code **C0027051** (corresponding to the UMLS concept **myocardial infarction**) would also be appropriate, but is not as specific as the one corresponding to **post mi,** and therefore is not chosen. In addition, we also describe an evaluation of the method and the results that were obtained. The Background section discusses related work, the Methods section describes the techniques and the evaluation design, the Results section presents performance measures, the Discussion section discusses the strengths and weaknesses of our method and issues associated with encoding and with the evaluation.

Background

Related Work

Cooper and Miller[5] developed and evaluated three different methods for mapping terms in clinical text to MeSH terms. These methods were not intended for use with completely automated systems but were meant to suggest a list of relevant MeSH terms to physicians to help them improve access to the literature. The first method, PostDoc, achieved a recall of 40% and a precision of 44% and was based on lexical indexing and string matching. The second method, Pindex, which achieved a recall of 45% and a precision of 17%, was based on statistical indexing in which associations were made between phrases in text and MeSH terms based on frequency. A third method that was reported was a hybrid of the two and achieved a recall of 66% and a precision of 20%.

More recent work associated with indexing concepts in clinical documents to a controlled vocabulary has been reported,[3,7,9,13,14] and Nadkarni et al.[7] provide a detailed description of the basic techniques. Many of the methods were based on string matching or a more advanced linguistic method that requires part-of-speech tagging of the text (i.e., identifying words as nouns, verbs, etc.) and identification of simple noun phrases. The noun phrases were then used for mapping to the UMLS. Lowe et al.[13] reported using the Sapphire indexing engine[15] to index radiology reports to support retrieval of these images from the patient record. Good recall but poor precision were reported, and a subsequent effort to improve precision explored context-based mapping.[9] This method restricted the UMLS terms to be used for the mapping according to the section of the report and the source vocabulary. This resulted in improvement in precision without affecting recall. Elkin et al.[16] developed a method called *Concept-Based Indexing* that uses techniques based on noun phrase identification and term compositionality, utilizing the ontology of clinical concepts in SNOMED-RT.[17,18] Concepts that are indexed are mapped to SNOMED-RT codes instead of to UMLS codes. Nadkarni et al.[7] report on an application that indexed clinical information in discharge summaries and surgical notes to UMLS codes, also using phrase-based methods. Another indexing system, called *IndexFinder*, was presented by Zou et al.,[10] which also is based on a variation of phrase identification.

Other work on automated UMLS indexing was associated with efforts to improve information retrieval of the biomedical literature.[6,19–21] Berrios et al.[19] used a method that involved tokenization, syntactic parsing, and noun phrase identification to identify concept values in the UMLS occurring in a chapter of a textbook associated with infectious diseases. They reported that 66% of nouns and noun-modifiers and 81% of noun phrases were correctly matched to UMLS concepts. In a subsequent report, Berrios[22] also reported a different indexing technique that used a statistical-based method and a vector-space model for the same task.

MetaMap[6,20] can also be used to index text or to map terms from one vocabulary (including natural language) to another. It discovers a set of UMLS concepts in a body of text and ranks the concepts in the set. This is achieved in a five-step process that involves (1) using the SPECIALIST minimal commitment parser[23] to identify simple noun phrases in the text (i.e., noun phrases without right modifiers), (2) generating variants of each phrase, (3) retrieving candidate phrases consisting of all strings with at least one variant, (4) using linguistic principles to compute a score for each candidate by comparing it with the input phrase, and (5) using linguistic principles to compose mappings and calculate scores by combining candidates in disjoint parts of the phrase. MetaMap has many options and can be used for clinical text as well as biomedical literature. An interesting application of MetaMap was presented by Brennan and Aronson[8] that involved mapping electronic messages of patients to UMLS codes to see if the codes could be used to search knowledge resources and to bring helpful health information to the patients.

Srinivasan et al.[24] reported on a method that mapped the entire collection (as of 2001) of MEDLINE citations, consisting of text in the titles and abstracts, into UMLS concepts. The method first uses a part of speech tagger and the SPECIALIST mimimal commitment parser and then obtains three types of noun phrases associated with different structural complexities. String matching is then performed with different degrees of strictness and flexibility. The number of phrases obtained and the number of matches to the UMLS using the different matching methods were reported as well as comparison between the 2001 and 2002 versions of the UMLS Metathesaurus. Nonmatches were also analyzed, but precision was not measured because it was not the focus in this effort.

Our Method

Our method focuses on mapping clinical information to coded forms. Our work differs from related work because we base our method on an NLP technique that parses the entire document (not just noun phrases) to obtain clinical information along with the modifiers and other values that are associated with the information. The most specific UMLS codes are then obtained based on structural matching in which the modifiers are an inherent part of the matching process. When codes are obtained, they are associated with modifiers as well. Some types of modifiers are not coded by themselves, such as **certainty, quantitative, degree,** and **temporal types**, because the UMLS semantic categorization of those types is not granular enough for clinical purposes. For example, the UMLS concept **C0332149 (possible)** is categorized as a **Functional Concept,** the concept **C0332125 (no evidence of)** is classified as a Qualitative Concept, and **C0241889 (family history of)** is classified as a **Finding**. An additional problem is that there are no UMLS concepts associated with various modifiers found in clinical reports, such as *unlikely* and *cannot evaluate*.

MedLEE

A number of previous reports discussed the architecture of MedLEE[25,26] and the different modules, including the encoding component. Figure 1 shows the different components; knowledge-based components are shown as ovals and programming components as rectangles. A brief summary of the components is given below. This summary describes an earlier version of the system, upon which the thrust of the current coding method is based:

- **Pre-processor**: segments the text into sections, paragraphs, sentences, and words and performs lexical lookup to identify and classify words and multiword phrases and to determine their canonical forms using a lexicon. Thus, a sentence *Myocardial infarction in 1995* would be considered a sequence of three terms, *myocardial infarction*, *in*, and *1995* because the first term is considered a multiword phrase in the lexicon. *Myocardial infarction* would be classified as a **finding** type, *in* as a general English preposition, and *1995* as a number. This component also handles abbreviations using an abbreviation table and performs some word sense disambiguation based on contextual rules.

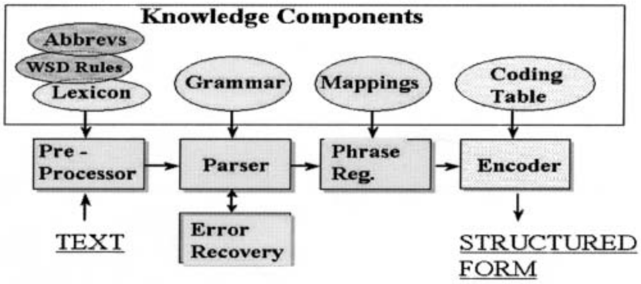


Figure 1. Overview of components of MedLEE. The knowledge-based components are shown as ovals; the processing engines are shown as rectangles. The new work discussed in this report involves the final stages of processing, the encoding process, which occurs after structured output is obtained.

- **Parser**: determines the initial structured form for each sentence using a grammar that includes syntactic and semantic rules that specify the structure of the language components and their interpretation. The structure is frame based in the form of a list in which the first element corresponds to the type of information, the second the value, and the remaining elements modifiers of the value. For example, *Status postmyocardial infarction in 1995* would be structured **[problem,'myocardial infarction', [date,'19950000'], [status,post]]**. In this example, the primary finding is **myocardial infarction;** it has a temporal modifier **status** with the value **post** and a date modifier **19950000,** which is a normalized form YYYYMMDD for date. Zeros are used in the normalized form when the year, month, or day fields are unknown.
- **Error recovery**: attempts to obtain a parse if the initial effort failed; this may involve skipping words and segmenting the text into chunks.
- **Phrase regularization**: composes a multiword phrase after the parsing stage if the sentence contains a noncontiguous phrase (i.e., the individual words of the phrase are separated). For example, *enlarged spleen* is defined in the lexicon as a phrase, and the target form is specified as **enlarged spleen**. When that phrase is contiguous and occurs in a sentence, such as *enlarged spleen noted*, the output would be: **[problem,enlarged spleen, [certainty,'high certainty']]**. However, if the words in the phrase are separated, as in *spleen was enlarged*, the output would be different because it would be formed from the individual words and not the phrase *(i.e.,* **[problem,enlarged, [bodyloc,spleen], [certainty,'high certainty']]***)*. The regularization component uses a compositional table containing the phrases and their corresponding compositional structures to compose terms that are not contiguous so that after regularization, the output would be the same as for sentences with contiguous phrases. For example, after the regularization phase the output for *spleen was enlarged* would be: **[problem,enlarged spleen, [certainty,'high certainty']]**. The compositional table is generated using MedLEE also, but it is created only once when the lexicon for MedLEE is compiled. Another aspect of phrase regularization involves using domain knowledge to add information to the output that is implicit in the domain. For example, **infarct** denotes **myocardial infarction** in the domain of cardiology reports, but could refer to another body location in other types of reports. The domain knowledge is specified by entries in a table that is manually created using domain expertise. In the above example, when the domain is cardiology and a sentence contains *infarct*, the term will be changed to **myocardial infarction**.
- **Encoding**: uses an encoding table to add codes, such as UMLS codes, to the regularized output form. Encoding in this version is straightforward and involves mapping the primary finding to a coded form. When processing reports at New York Presbyterian Hospital, the encoding phase maps a target term in the MedLEE system to a Medical Entities Dictionary (MED) code[27] so that the output can be stored in the clinical repository. For example, **acute myocardial infarction** would be mapped to an MED code **2618** that corresponds to that term. Modifiers would also be mapped separately to MED codes, but findings would not be combined with modifiers to determine

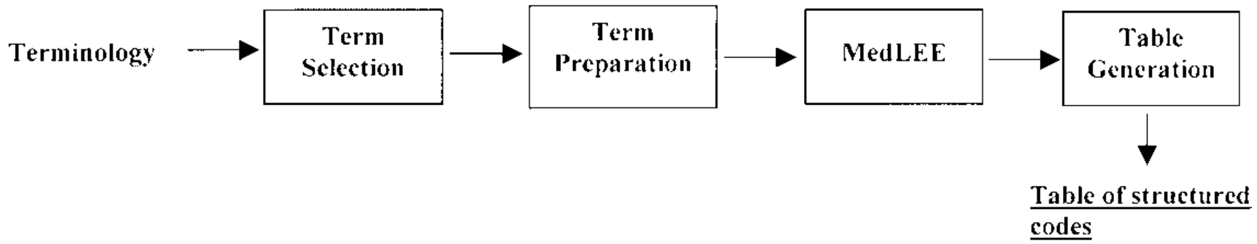


Figure 2. Process for creating a structured coding table of clinical information that is required for encoding of documents.

a more specific code. Creation of the coding table is a time-consuming manual process because many of the entries have to be entered manually. Each entry in the table consists of a target term generated by MedLEE and the corresponding MED term and code. The bottleneck consists of the task of associating a target MedLEE term with a MED term. Since many of the MED terms are not similar to terms found in patient documents, only a partially automated process can be used to facilitate the mapping. For example, the target form **trifascicular block** generated by MedLEE corresponds to the MED term **complete (third degree) atrioventricular block**. Note that the coding method discussed above describes the previous version; the current version is described in the Methods section.

Methods

Encoding Method

The revised method used by MedLEE to encode clinical documents is similar to the method shown in Figure 1 in that a coding table is still used to generate encoded output. However, in the revised method, the process of creating the table, the structure of the coding table, and the technique for encoding have been changed from the earlier version of MedLEE, which was described in the Background section. In the revised method, table creation is a more complex but completely automated process that consists of four steps: term selection, term preparation, parsing using MedLEE, and table generation, as shown in Figure 2. Once the table is created, it is used by MedLEE, as shown in Figure 1, and the coding technique is based on a process involving matching of structures and not just strings.

Creating the Encoding Table

The first step in the table creation process involves **term selection**, because some terms in controlled vocabularies are inappropriate for clinical documents, some terms cause redundancies, and some terms are highly ambiguous. Therefore, this task may differ for different terminologies. For UMLS encoding that is reported here, this process consists of (1) selecting UMLS semantic classes that are appropriate for the clinical domain and then selecting terms associated with those classes for the table; (2) removing terms that include the word *other* (e.g., **other otosclerosis**); (3) removing terms containing *nos*, *nec*, *unspecified*, and *classified elsewhere* if corresponding terms exist in the terminology without them (e.g., **anemia** occurs in the UMLS; therefore, **anemia nec** was removed); (4) removing veterinary terms only based on knowledge in SNOMED; and (5) removing abbreviations (details describing the method used for identifying abbreviations in the UMLS were described previously[28]) because they are highly ambiguous and were found previously to be a frequent cause of coding errors.

Term preparation is used to add variant forms of a term to the coding table if the variant is not already in the table to capture variant structures. An example of a variant form concerns terms with commas (e.g., **infarction, myocardial**). Terms with commas are normalized so that the comma is removed (e.g., **infarction myocardial** is generated), and the part of the phrase following the comma is moved to the left of the part preceding the comma (e.g., **myocardial infarction** is generated). When the term preparation step is completed, a list is formed in which each term is associated with a unique concept identifier (CUI) that is a unique identifier for that concept. Thus, the coding table at this stage will consist of two fields. The first field is the unique identifier for the concept (i.e., the CUI in this study), followed by a string associated with the concept, which is not necessary but is useful for debugging purposes. Currently, the string that is chosen is based on the preferred form specified by the UMLS or a variation of the form. The second field is the UMLS term or variant of the UMLS term. Therefore, terms that are synonyms according to the UMLS because they correspond to the same UMLS concept will correspond to the same code. Below is an example of some rows containing synonyms (according to the UMLS) associated with the concept **myocardial infarction**:

C0027051^myocardial infarction | myocardial infarction
C0027051^myocardial infarction | heart attack
C0027051^myocardial infarction | myocardial infarction syndrome
C0027051^myocardial infarction | myocardial necrosis
C0027051^myocardial infarction | attack coronary
C0027051^myocardial infarction | necrosis myocardium
C0027051^myocardial infarction | myocardial necrosis syndrome
C0027051^myocardial infarction | coronary thrombosis
C0027051^myocardial infarction | cardiopathy necrotic
C0027051^myocardial infarction | infarction of heart
C0027051^myocardial infarction | infarction, myocardial
C0027051^myocardial infarction | infarction myocardial

Parsing is performed for each term in the initial table. This is accomplished by treating each term as a complete sentence and parsing it using MedLEE to obtain a structured output form for that term. In addition, if a term has multiple words, it is parsed in the standard mode (in which multiword phrases are treated as atomic units) and in the decompositional mode in which each word of a multiword phrase is treated individually. This is done because sometimes there is no single compositional phrase in the UMLS that corresponds to the term, but there may be codes for components. For example, there is no UMLS code for *rash on big toe*, but there is a code for *rash and big toe*. In the examples above associated with the concept **C0027051**, the **terms myocardial infarction, heart attack, myocardial infarction syndrome,** and so forth would each be parsed. Parsing is achieved using the strict mode, in which case each word of the term must be known to the system, and the parse must include all the words of the term. The output that is generated is the internal MedLEE output form, which is a list form as shown in the Background section. The structured output forms generated by parsing the first three terms above would consist of the following six structures, which would each be associated with the code **C0027051:**

```
[problem,'myocardial infarction']
[problem,infarction,[bodyloc,myocardium]]
[problem,'heart attack']
[problem,attack,[bodyloc,heart]]
[problem,'myocardial infarction',[problemdescr,syndrome]]
[problem,infarction,[bodyloc,myocardium],[problemdescr,
  syndrome]]
```

In the examples, the first output structure corresponds to a parse of **myocardial infarction** and consists of a primary finding of type **problem** with a value **myocardial infarction,** which does not have any modifiers. The second structure corresponds to a parse in which **myocardial** and **infarction** were parsed individually. The third and fourth structures correspond to parses of **heart attack**. The fifth and sixth output structures were generated by parsing **myocardial infarction syndrome.** In these structures, **myocardial infarction** has a modifier called **problemdescr** with the value **syndrome.** After each parse is obtained, it is combined with the CUI. In addition, for ease of readability, the CUI is combined with a string representing the concept. Thus, the six entries to the UMLS encoding table from the example shown above would have the code **C0027051^myocardial infarction** added to the first field of each structure to associate the code with a structure.

Table generation is the final step in the process. In this step, entries with parsing problems that can be detected automatically are removed, but no entries are removed manually. Although this is possible, it would be too costly. One type of problem is detected if a UMLS term is classified in the UMLS as one type of semantic category, but the primary finding obtained by MedLEE is an incompatible type. An example of this would be if the type according to the UMLS is **medical device,** but the primary finding determined by MedLEE is a different type of finding, such as **problem.** This may occur due to ambiguity. For example, **patch** is ambiguous; it may correspond to **C0994894** (e.g., *drug form patch*), **C0445403** (e.g., *surgical material*), **C0332461** (e.g., plaque), or to a descriptive term associated with the skin (e.g., *skin examination revealed a black macular patch)*, which does not have a corresponding UMLS code. During parsing the term *patch,* the ambiguity associated with the term led to an incorrect interpretation. More specifically, when the term **patch** associated with **C0994894** was parsed, the output generated by MedLEE was the sense that denotes skin patch, which is an error because the semantic class of **C0994894** is medical device.

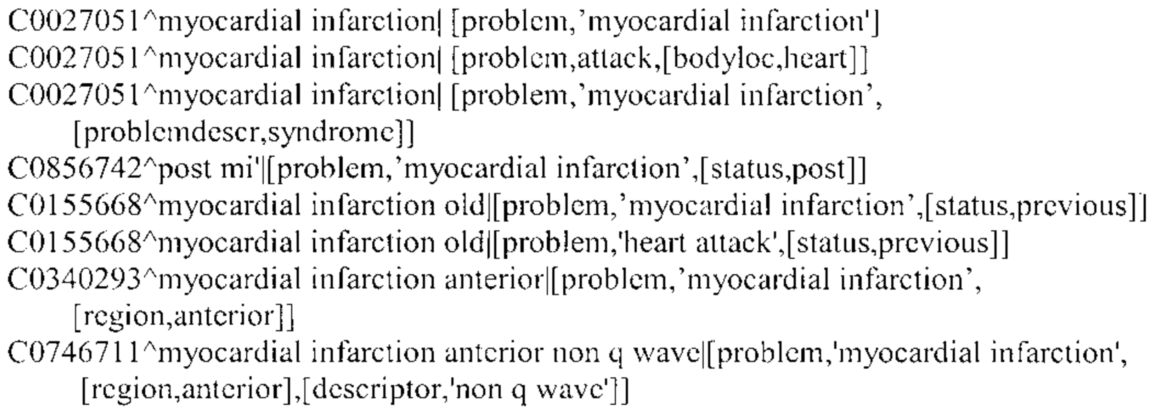
Ambiguity is a challenging problem for NLP systems and is often a source of error. Contextual information, such as neighboring words, is sometimes used by MedLEE to help resolve ambiguity. The case of table creation is even more problematic because individual terms are parsed instead of complete sentences; therefore, the usual contextual clues are generally missing. Ambiguity is currently handled during lexical lookup by using contextual rules that are currently written manually. We have experimented with various machine-learning methods that automatically create classifiers to resolve ambiguities[29,30]; however, these have not been integrated with the system.

Once the known types of errors are automatically removed during table generation, all the remaining entries are combined into one coding table that is subsequently used when parsing and encoding sentences. An example of a simplified form of the UMLS coding table is shown in Figure 3. Note that the same code corresponds to different structures, each of which represents a parse of a synonymous term.

Mapping Text to Coded Form

This step consists of using MedLEE to parse sentences in clinical documents, as described above. All the steps required for processing are followed, except in this version the new encoding method and the new coding table schema are used. At this point in sentence processing, the structured output form of the sentence contains the primary findings and modifiers that are associated with the findings. The encoding step consists of matching the structured output obtained by parsing the sentence with the structured outputs that are in the encoding table to find the entry or entries in the table that contain the closest match(es). The closest match is determined by successfully matching the primary finding and as many modifiers as possible to a table entry. The closeness criterion is established on a structural basis, in which the match containing the greatest number of modifiers is considered the most specific match. For example, the matching method would find that the most specific match for the structure **[problem,'myocardial infarction',[date,19950000], [status,post]],** which was obtained by parsing *Status post myocardial infarction in 1995*, would be the structure **[problem,'myocardial infarction',[-status,post]]** in the coding table, which is associated with the code **C0856742** corresponding to the preferred form **post mi**. The most specific match in this case includes the primary finding and the status modifier. A different code would be obtained for the sentence *She was admitted to rule out myocardial infarction*. The structure prior to encoding would be: **[problem, myocardial infarction,[certainty,rule out],[timeper,admission]]**. Because there is no UMLS code that contains any of the above modifiers, the code **C0027051,** which corresponds to **myocardial infarction,** would be obtained as the code.

In some cases, more than one code may be obtained. An example in which this would occur would be the sentence ***He is status post an anterolateral myocardial infarct*** where the parsed structure would be **[problem,'myocardial infarction', [status,post],[region,anterolateral]]**. In this case there are two structures in which the primary finding and one modifier match a table entry; one corresponds to *anterolateral myocardial*



```
C0027051^myocardial infarction| [problem,'myocardial infarction']
C0027051^myocardial infarction| [problem,attack,[bodyloc,heart]]
C0027051^myocardial infarction| [problem,'myocardial infarction',
    [problemdescr,syndrome]]
C0856742^post mi'|[problem,'myocardial infarction',[status,post]]
C0155668^myocardial infarction old|[problem,'myocardial infarction',[status,previous]]
C0155668^myocardial infarction old|[problem,'heart attack',[status,previous]]
C0340293^myocardial infarction anterior|[problem,'myocardial infarction',
    [region,anterior]]
C0746711^myocardial infarction anterior non q wave|[problem,'myocardial infarction',
    [region,anterior],[descriptor,'non q wave']]
```

Figure 3. Simplified form of the coding table. The actual table has additional fields to take advantage of indexing for improved efficiency when searching for matches. Several sample entries in the coding table are shown for some UMLS concepts associated with myocardial infarction. The first field consists of the CUI followed by the preferred former variant, which is provided for ease of readability. The second field consists of the structure obtained by MedLEE as a result of parsing the input terms in the table creation phase.

infarction (e.g., **C0262564**) and the other to *post mi* (e.g., **C0856742**).

When a code is obtained, it is added to the primary finding as a modifier, which is named *umls*. Thus, for the example above, the output structure would become **[problem, 'myocardial infarction', [status,post],[region,anterolateral], [umls, C0262564^anterolateral myocardial infarct],[umls, C0856742^post mi]]**.

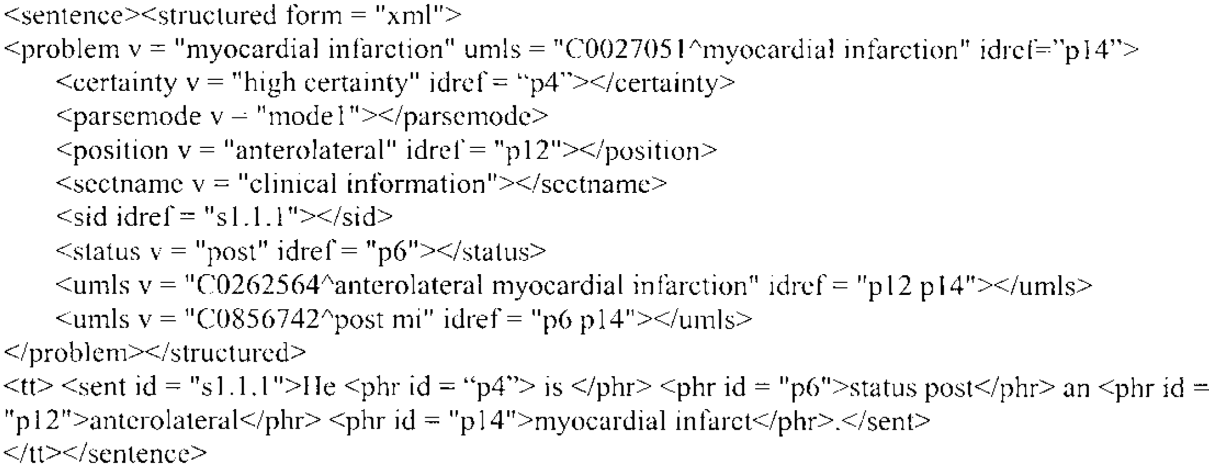
In the examples above, we showed a simplified version of the intermediate output form. The final output form is XML and is more complex because it contains links to phrases in the original text, which are associated with the corresponding UMLS code. The actual version also contains contextual attributes that are not shown in the example above. A detailed explanation of the XML schema and the process that generates it is described by Friedman et al.[31] The actual output form for the sentence *He is status post an anterolateral* is shown in Figure 4. The output consists of a structured component (**structured**) and a textual component (**tt**). The textual component consists of the original text with sentence identifiers and phrasal identifiers added. Only phrases that are referenced by the structured component have identifiers.

The structured output consists of tags corresponding to the type of information and attributes associated with it. The **v** attribute corresponds to the value of the information (e.g., the target output form), the **umls** attribute corresponds to the umls code associated with the term (e.g., the **v** attribute only). This is different conceptually from the component constituting the **umls** tag because the **umls** attribute corresponds to the primary term only, whereas the **umls** tag corresponds to the primary term along with modifiers, if applicable. This is explained in more detail below. A tag may also have an **idref** attribute, which may be a single value or a string of values referring the identifiers of the phrases in the original text that are associated with the tag. Thus, in Figure 4, the tag **problem** has a **v** attribute *myocardial infarction*, a **umls** attribute *C0027051^myocardial infarction*, and an **idref** attribute *p14* corresponding to the phrase in the sentence *myocardial infarct*. The **problem** tag has components **certainty, parsemode, position, sectname, sid, status,** and **umls**, which are modifiers in which each different type of modifier represents a different type of information.

The modifiers **parsemode, sectname,** and **sid** are different from the other modifiers because they do not correspond to phrases in the sentence but to contextual information and have different attributes than the other tags. The **parsemode** is associated with the method used to generate the parse, the **sectname** corresponds to the section of the report the information was obtained from, and the **sid** refers to the sentence identifier, which consists of three components: a section identifier, a paragraph identifier, and a sentence identifier. The **certainty** tag has a value *high certainty*, which corresponds to the phrase *is* in the sentence. The **status** tag has the value *post*, and the **position** tag has the value *anterolateral*. There are two **umls** tags. One tag has the value **C0262564^anterolateral myocardial infarction**, which was obtained from the phrases **p12** (*anterolateral*) and **p14** (*myocardial infarction*). The other **umls** tag has the value *C0856742^post mi*, which was obtained from the phrases **p6** (*status post*) and **p14** (*myocardial infarction*). Notice that in this case, *status post* is separated from *myocardial infarction* by a few words. Since the coding method is based on structural matching and not string matching, the match is considered good even when the parts of the text that correspond to the code are not consecutive. For example, in *He is status post an anterolateral and posterior myocardial infarction*, the code **C0856742** (e.g., *post mi*) would be obtained as well as the codes **C0262564** and **C0340319** (e.g., *anterolateral myocardial infarction and posterolateral myocardial infarction*, respectively). Notice that these codes are more specific than **C0027051** (e.g., *myocardial infarction*).

Evaluation of UMLS Coding

Recall and precision were evaluated in two separate studies to reduce the effort of the medical experts who served as subjects for the two studies. The studies used a collection of discharge summaries of de-identified patients admitted to New York Presbyterian Hospital during 2000 to obtain the two test sets. This collection consisted of 818,000 sentences. In the study measuring precision, the test set consisted of 150 sentences, which were randomly selected from the collection.



```
<sentence><structured form = "xml">
<problem v = "myocardial infarction" umls = "C0027051^myocardial infarction" idref="p14">
     <certainty v = "high certainty" idref = "p4"></certainty>
     <parsemode v = "mode1"></parsemode>
     <position v = "anterolateral" idref = "p12"></position>
     <sectname v = "clinical information"></sectname>
     <sid idref = "s1.1.1"></sid>
     <status v = "post" idref = "p6"></status>
     <umls v = "C0262564^anterolateral myocardial infarction" idref = "p12 p14"></umls>
     <umls v = "C0856742^post mi" idref = "p6 p14"></umls>
</problem></structured>
<tt> <sent id = "s1.1.1">He <phr id = "p4"> is </phr> <phr id = "p6">status post</phr> an <phr id =
"p12">anterolateral</phr> <phr id = "p14">myocardial infarct</phr>.</sent>
</tt></sentence>
```

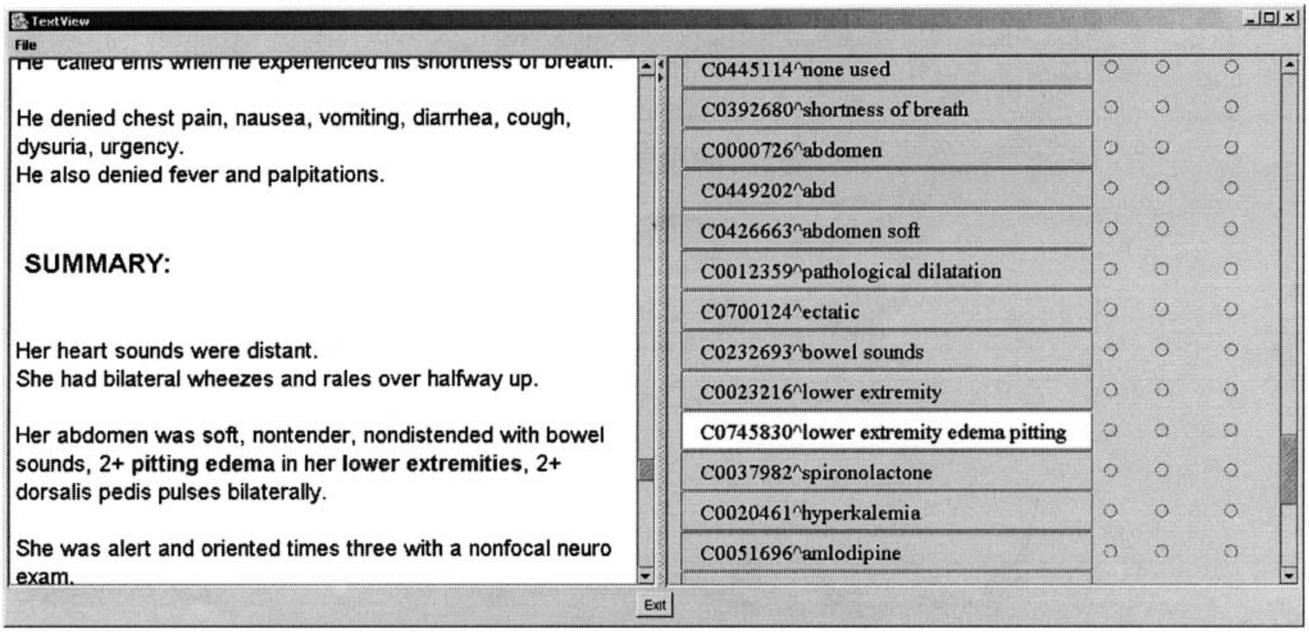
Figure 4. XML output generated by MedLEE for *He was status post an anterolateral myocardial infarction*. Modifiers are indented in this figure for readability, but in the actual XML output form, they are not indented. The structured component is enclosed within the **structured** tag, and the original text is enclosed within the *tt* tag. In addition, the sentence and terms are also tagged.

MedLEE was used to process the sentences, extract relevant clinical information consisting of findings and modifiers, and perform UMLS coding. Six experts participated in this part of the study. They were asked to judge the correctness of the UMLS codes that MedLEE obtained as a result of processing the test set of sentences. Each expert was given a set of codes obtained by MedLEE that were associated with 75 of the sentences in the test set, and each sentence was reviewed by at least two experts. To reduce the effort, the experts were given a program to use that included an interface in which they could easily view the codes, view the corresponding phrases and sentences in the test set, and register their judgments regarding the codes. They were also given written instructions and were asked to participate in a trial run before doing the actual evaluation. The interface, which is shown in Figure 5, consisted of three windows. The window on the left contained the original paragraphs with the applicable sentences highlighted. The complete paragraph was shown to provide contextual information. The middle window contained the UMLS codes that were generated by MedLEE, and the window on the right contained radio buttons in which the subjects could enter their judgments. To evaluate the correctness of each UMLS code, they were asked to click on each code in the coding window. For example, in Figure 5, the user clicked on the code **C0745830** associated with **lower extremity edema pitting**. The corresponding text then was shown in the window on the left, and the relevant phrases in the text sentence for the UMLS code were highlighted in red. The expert was given three choices (**correct, incorrect,** or **maybe correct**), and was asked to register his or her choice by clicking the appropriate radio button on the far right window. Precision was calculated as the ratio of the number of correct UMLS codes over the total number of UMLS codes in which a *maybe* counted as half correct.

The second part of the evaluation measured recall and consisted of two parts. Another test set of 150 sentences was randomly selected for this evaluation. In the first part, the same six experts were asked to manually read the sentences in the test set to determine information that they thought was clinically important, but they were not asked to code the information because that would be too time consuming. Each expert was shown 75 sentences, and each sentence was read by three experts. The experts were asked to restrict extraction of the information in the sentences to information denoting diseases, signs and symptoms, medications, body locations, vital signs and other measurements, procedures, functionality, and medical devices and not to extract admission, discharge, follow-up or other types of information, such as social history. To help clarify the task, several examples were given in the instructions. The experts were shown each test sentence, which was highlighted along with the paragraph in which it occurred, to provide context. They were asked to note the clinical terms in the highlighted sentence and to list the terms in a special location provided for them. They were asked to include body locations separately from other findings, such as diseases, signs and symptoms, and body locations associated with findings. For example, for the sentence *rib fracture noted*, they were expected to list **rib** and **rib fractures**, and also to expand abbreviations. In addition, they were asked to enter **XXX** if the sentence did not contain any of the specified types of information. A list of terms associated with each sentence was then compiled by combining all the terms listed by each expert for that sentence. A reference standard of relevant terms was obtained using majority opinion (i.e., at least two of the three experts). Note that the reference standard did not consist of UMLS codes but just relevant clinical terms.

The same set of sentences was processed by MedLEE, which generated structured output consisting of extracted findings, modifiers, and UMLS codes. One of three outcomes was noted by comparing the output of MedLEE with the reference standard: (1) MedLEE found a code corresponding to the term in the reference standard, (2) MedLEE generated structured output corresponding to the term but no code, or (3) MedLEE did not generate any output at all for that term.

Since this part of the evaluation study measured recall, we recorded whether a code was generated by MedLEE but did not determine whether it was correct. However, if a code was not generated, it could be because there was no corresponding UMLS code. Therefore, we utilized a seventh expert, who is both a board-certified physician and an expert in clinical



He called ems when he experienced his shortness of breath.

He denied chest pain, nausea, vomiting, diarrhea, cough, dysuria, urgency.
He also denied fever and palpitations.

SUMMARY:

Her heart sounds were distant.
She had bilateral wheezes and rales over halfway up.

Her abdomen was soft, nontender, nondistended with bowel sounds, **2+ pitting edema in her lower extremities**, 2+ dorsalis pedis pulses bilaterally.

She was alert and oriented times three with a nonfocal neuro exam.

C0445114^none used ○ ○ ○
C0392680^shortness of breath ○ ○ ○
C0000726^abdomen ○ ○ ○
C0449202^abd ○ ○ ○
C0426663^abdomen soft ○ ○ ○
C0012359^pathological dilatation ○ ○ ○
C0700124^ectatic ○ ○ ○
C0232693^bowel sounds ○ ○ ○
C0023216^lower extremity ○ ○ ○
C0745830^lower extremity edema pitting ○ ○ ○
C0037982^spironolactone ○ ○ ○
C0020461^hyperkalemia ○ ○ ○
C0051696^amlodipine ○ ○ ○

Figure 5. Screen shot of the graphic user interface of the application used by experts to evaluate the precision of MedLEE's UMLS coding. This view was obtained by clicking on the code **C0745830** in the right window. As a result, the information in the report that was used to obtain the code is highlighted in the window on the left.

terminologies, to manually check whether there was a corresponding code in the UMLS for those terms in the reference standard that MedLEE failed to code. Four different measures were calculated for the system:

1. Recall for extracting all terms in the reference standard regardless of whether the terms had corresponding UMLS codes. This measured the recall performance of the system with respect to the entire reference standard.
2. Recall for extracting those terms that had corresponding UMLS codes. This measured the recall performance of the system with respect to the terms that have corresponding codes.
3. Recall for obtaining UMLS codes for all terms in the reference standard regardless of whether the terms had corresponding UMLS codes. This measured the portion of terms in the reference standard that was coded by the system.
4. Recall for obtaining UMLS codes considering only those terms that had corresponding UMLS codes. This measured the proportion of terms in the reference standard that were coded and that were possible to code.

Recall was computed as the ratio of the number of correct terms for the particular category being measured over the total number of terms in the particular category of reference standard. Recall for extracting terms was also computed for each subject. In this case, the reference standard for the subject was obtained by excluding that expert. Therefore, the reference standard was either a majority opinion (i.e., agreement with both of the two other subjects) or a coin toss if there was agreement with only one expert. This calculation was performed 1,000 times and then averaged.

Expert precision was also measured, based on judgment of the seventh expert, for terms that had UMLS codes. For this task, the seventh expert registered judgment of the other experts by answering **yes** or **no** for each depending on whether he agreed or disagreed with them. Note that this differs from the precision study of MedLEE in several ways. MedLEE had to code the terms, whereas the experts just had to extract the terms. Also, the experts judged the precision of MedLEE, whereas the seventh expert judged the experts' precision (they may have different set-points). Finally, the seventh expert only answered *yes* or *no*, whereas the other experts had three choices.

Results

For the precision part of the evaluation of MedLEE, 465 UMLS codes were generated automatically from the test set of sentences. Precision was .89 (95% CI .87–.91), where *maybe correct* received half credit, and it was .86 (.83–.88) where *maybe correct* received no credit. In each result presented, the figures in parentheses represent a 95% confidence interval, signifying that if the same experiment were performed 100 times, on average, the true value would be within the confidence interval 95% of the time. This is a standard way to report experimental results.[32] Confidence intervals are a factor of sample size and measure how precise results of experiments are. The interrater reliability was .75, which is sufficient to make a reasonable judgment of precision. The precisions of the six experts that rated MedLEE were .89 (.85–.93), .83 (.79–.87), .77 (.72–.83), .61 (.51–.71), .79 (.73–.83), and .85 (.80–.89).

Figure 6 shows the results for recall. Recall of the system for extracting terms in the reference standard regardless of whether a corresponding UMLS code existed was .84 (.81–.88), and recall for extracting terms that had a corresponding UMLS code was .88 (.84–.91). Recall for coding all terms in the reference standard was .77 (.72–.81). This code tells us how much of the relevant information in the record that should be coded was coded. Recall for coding all terms in the reference standard that had UMLS codes was .83 (.79–.86). This result tells us the recall of the system for coding terms that were possible to code. Recall of the six expert subjects for extracting terms (regardless of whether there was a corresponding UMLS code) ranged from .69 (.58–.74) to .91 (.88–.95). Figure 6 shows that MedLEE extracted terms just as well as the experts and also that the recall performance of MedLEE for obtaining codes was as good as the extraction recall of five of the six experts.

Discussion

An analysis of the errors in precision was performed. A number of errors in coding were due to ambiguous terms. For example, *serous drainage* was mapped correctly to **C0012621**, which corresponds to **drainage of a body substance**, but it was also mapped to **C0013103**, which corresponds to **drainage procedure**. Other errors caused by ambiguity occurred when *withdrawn* in the sentence *patient was withdrawn* was mapped to a generic procedure **withdraw** and *no evidence of abuse* was mapped to **not used**. In the last example, the ambiguity was indirect because in MedLEE *abuse* is synonymous with **use** in the context of **drug use** or **alcohol use**. Some of the errors in ambiguity can now be resolved with our improved method of automatically removing inappropriate parses during table creation. For example, in the revised method, a code of **C0013103** (e.g., **drainage procedure**) would automatically be removed from the table during table creation if the structured output did not represent it as a procedure. This would correct the coding error described above, because the parsed output of the sentence containing *serous drainage* would only map to the UMLS sense **drainage of body substance**, which would be correct. However, this means that the procedural sense of *drainage* would be missing from the UMLS table, which could cause another type of coding error. If another sentence containing *drainage* was parsed so that *drainage* was correctly interpreted to be a procedure, the code **C0013103** would not be found because it would be missing from the table. Other ambiguities would be harder to detect and are still likely to cause errors. To correct those errors, more work on disambiguation is needed. Other errors in precision usually were associated with very general terms, such as **not used** or **vessel positions**, and these terms by themselves are not clinically useful. They could be removed from the UMLS coding table and the errors would be eliminated; however, this would require knowledge engineering and manual review and would be costly.

Errors in recall were also analyzed. The majority of errors were due to a modifier that the experts coded in isolation from findings, whereas MedLEE primarily coded a modifier when it was related to and combined with a finding. For example, there were terms such as *po*, *right*, and *postoperative day* that caused errors when they occurred alone in a sentence without a finding. This occurred because the finding was in a previous sentence. Since the sentences were chosen at random, these types of sentences could not be avoided.

Although the performance of MedLEE in coding was similar to expert performance, there are a few issues in this study that should be noted. One issue is that this study evaluated precision and recall of MedLEE encoding in isolation of modifiers and other information in the sentence. This design was similar to other studies cited above in that terms were coded without regarding modifiers. The coding technique of MedLEE involves encoding of modifiers when combined with primary findings even if the modifiers are separated from the primary findings; it does not currently encode modifiers independently except for body locations. This is because body locations are generally well defined in the UMLS and other coding systems, but other types of modifiers containing degree, change, and temporal information are generally not categorized finely enough to be useful for retrieval. Even if uncoded, the modifiers are valuable for applications that retrieve the structured output, and, more importantly, coded information alone is not sufficient for clinical applications. Modifiers and other values that co-occur with the codes are also important to capture. For example, if a code of **C0027051** denoting **myocardial infarction** is generated, it could occur in the context of *at risk for myocardial infarction*, *family history of myocardial infarction*, *rule out myocardial infarction*, or *myocardial infarction in the past*, which do not denote a recent event of **myocardial infarction**. Similarly, the fact that a sentence contains a code of **C0886414** corresponding to **body temperature measurement** would not be useful unless a value (i.e., 98° F) is associated with the code. The association of related clinical information that corresponds to the UMLS codes, which is one of the strengths of MedLEE, is performed by MedLEE, but this aspect was not evaluated in the current study. It would have been too time consuming to have the experts manually capture and associate modifiers and values with the terms they listed. An improved evaluation would be needed to address compositionality and modification. However, MedLEE was evaluated numerous times previously[31,33,34] without coding, and one reason high precision was obtained was because the modifiers and values of measurements were reflected in the query along with the terms. As an example, if a clinical variable was needed comprising a current event of *myocardial infarction*, the different modifiers would be used to filter out sentences, such as those above, that contain *myocardial infarction* but do not assert that it recently occurred. Another example is that the application that assessed the risk status of a patient with community-acquired pneumonia[31] needed variables associated with quantitative values. For example, one variable was a fever that was greater than 104°F. It is possible to obtain this variable because the

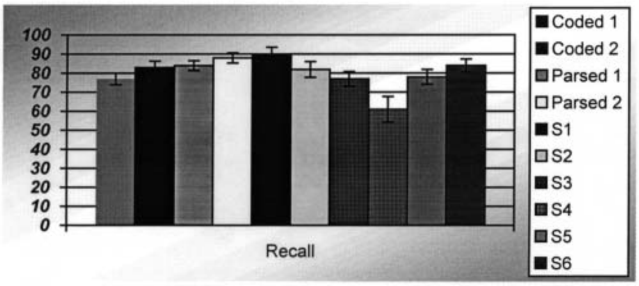


Figure 6. The first four recall measures are for the system, and the remaining six (S1-S6) for the six subjects. "Coded 1" represents the performance of MedLEE coding when considering all terms in the reference standard. "Coded 2" represents performance for coding terms in the reference standard that have corresponding UMLS codes. "Parsed 1" represents extraction performance (i.e., extracting the term from the text but not necessarily mapping it to a UMLS code) for all terms, "Parsed 2" represents the extraction performance of terms that have existing UMLS codes.

structured output associated with the finding *fever* or *temperature* would have a measurement modifier containing the value, which could be retrieved.

Another issue in this evaluation is that the experts were not required to find UMLS codes for information in the reports but only to extract the terms. This was designed to reduce effort because UMLS coding would be much more time consuming than listing terms. For example, in a previous report, it took an expert 60 hours to manually code one discharge room report[35] that contained approximately 540 words into SNOMED codes. The effort was large because of the density of SNOMED codes obtained for the report (431 SNOMED codes) and the problem of redundancy in coding because of compositionality. Thus, in the current study, we could not compare the recall of MedLEE in UMLS coding with that of the physicians' ability to perform UMLS coding. However, it is highly likely that recall of physicians for coding would be lower than recall in extraction because the former requires much more knowledge than extraction, is more time consuming, and is a more complex process.

A third issue involves the instructions given to the experts. The instructions excluded certain types of information from the extraction process, such as follow-up, admission, discharge, and social history and gave examples. However, if the experts felt that information that should have been excluded was clinically important they listed it anyway. For example, *follow-up*, *postoperative*, and *discharge* (e.g., as in *discharge from hospital*) were listed by the experts, but the instructions stated they were supposed to be excluded. Additionally, some of the experts also extracted modifiers in isolation from their findings. For example, the terms *po*, *left*, and *diffuse* were listed separately by some experts as terms, but they had no corresponding finding. This type of information was also supposed to be excluded, because it currently is not coded by MedLEE and is not well defined in isolation of other terms. A large number of *errors* in recall were due to experts listing terms that should have been excluded.

Another issue associated with our method of coding is the complexity of the method itself. Creation of the coding table must be performed first and involves parsing; however, this task is only necessary initially, and then it can be used to code a number of reports. Table creation is based on the lexicon and other components involved in parsing; therefore, update of any of the components requires that the table be updated also, but this is an automated process. In addition, this method is more complex than a method based on string matching, and therefore it is likely to require more execution time than a string-matching algorithm. The average processing time to generate structured encoded output on a Sun Blade 2000 (Sun Microsystems, Inc., Santa Clara, CA) with dual 2.1 GHz processors and 4G RAM was approximately .4 seconds for a radiologic report of the chest, and approximately 8 seconds for a complete discharge summary. The method is generalizable in that it can be adapted to different coding systems. For example, in the clinical domain, we have used it to map to ICD-9 codes[36] and to SNOMED codes.[35] In the biological domain, we are using it to map to Mammalian Phenotype codes and Gene Ontology (GO) codes[37] when processing the biomedical literature. The method can also be generalized to other systems that generate structured output.

The technique we used for coding was achieved completely through automated methods. For improved performance, knowledge engineering or machine learning may be needed. For example, when coding radiologic reports of the chest to assign ICD-9 codes, manual coders usually assign reports asserting *infiltrate* or a *possible infiltrate* an ICD-9 code that corresponds to **pneumonia**. This would require domain knowledge and inferencing to link findings to appropriate codes. The knowledge could be obtained using domain experts or machine learning methods to automatically associate findings with codes. The main thrust is that some other process would be needed subsequent to MedLEE encoding to obtain the codes. This can become quite complex. For example, to obtain the UMLS code **post mi** for *myocardial infarction in 1995*, a process would have to check for a date modifier and then check that its value signifies that the event occurred in the past.

Another limitation of the method is that if a parse is not obtained, a code is not generated. Thus, if the input closely matches the name of a term in the coding system, but if there is a spelling error, a code would not be obtained. However, it may be obtainable using a string-matching algorithm, which, in the future, could be combined with the structural matching method.

Conclusion

We have presented a method that maps clinical text to UMLS codes using NLP techniques that obtain codes along with modifiers of the codes. Once coded, the output is in a comparable form that can be accessed reliably by automated clinical applications, which can be performed at different institutions and used for a broad range of purposes. MedLEE had a recall of 0.83 (95% CI 0.79–0.87) for coding terms that had corresponding UMLS codes and a recall of 0.84 (95% CI 0.81–0.88) for extracting terms. Extraction recall of the experts ranged from 0.69 (95% CI 0.58–0.74) to 0.91 (0.95% CI 0.88–0.95). The precision of the system was 89% (95% CI 0.87–0.91), and precision of the experts ranged from 0.61 (95% CI 0.51–0.71) to 0.89 (0.85–0.93). Results in evaluation of the method showed that the system performed similarly to or better than the experts in both recall and precision. Future work will involve further refinement of the coding method and a finer-grained evaluation design.

References ■

1. Tange HJ, Schouten HC, Kester AD, Hasman A. The granularity of medical narratives and its effect on the speed and completeness of information retrieval. J Am Med Inform Assoc. 1998;5:571–82.
2. Dolin RH, Alschuler L, Beebe C, et al. The HL7 clnical document architecture. J Am Med Inform Assoc. 2001;8:552–69.
3. Elkins JS, Friedman C, Boden-Albala B, Sacco RL, Hripcsak G. Coding neuroradiology reports for the Northern Manhattan Stroke study: a comparison of natural language processing and manual review. Comput Biomed Res. 2000;33(1):1–10.
4. Tuttle MS, Olsen NE, Keck KD, Cole WG, et al. Metaphrase: an aid to the clinical conceptualization and formalization of patient problems in healthcare enterprises. Meth Inform Med. 1998;37:373–83.

5. Cooper GF, Miller RA. An experiment comparing lexical and statistical method for extracting MeSH terms from clinical free text. J Am Med Inform Assoc. 1998;5:62–75.
6. Aronson AR. Effective mapping of biomedical text to the UMLS metathesaurus: the MetaMap program. 2001. AMIA Symposium Hanley & Belfus, 2001:17–21.
7. Nadkarni P, Chen R, Brandt C. UMLS concept indexing for production databases: a feasibility study. J Am Med Inform Assoc. 2001;8:80–91.
8. Brennan PF, Aronson AR. Towards linking patients and clinical information: detecting UMLS concepts in e-mail. J Biomed Inform. 2003;36(4-5):334–41.
9. Huang Y, Lowe H, Hersh W. A pilot sutdy of contextual UMLS indexing to improve the precision of concept-based representation in XML-structured clinical radiology reports. J Am Med Inform Assoc. 2003;10:580–7.
10. Zou Q, Chu WW, Morioka C, Leazer GH, Kangarloo H. IndexFinder: a method of extracting key concepts from clinical texts for indexing. Proc AMIA Symp. 2003:763–7.
11. Mutalik PG, Deshpande A, Nadkarni PM. Use of general-purpose negation detection to augment concept indexing of medical documents: a quantitative study using the UMLS. J Am Med Inform Assoc. 2001;8:598–609.
12. Chapman WW, Bridewell W, Hanbury P, Cooper GF, Buchanan BG. A simple algorithm for identifying negated findings and diseases in discharge summaries. J Biomed Inform. 2001;34(5): 301–10.
13. Lowe H, Antipov I, Hersh W, Smith CA, Maillot M. Automated semantic indexing of imaging reports to support retrieval of medical images in the multimedia electronic medical record. Methods Inf Med. 1999;38:303–7.
14. Hersh W, Mailhot M, Arnott-Smith C, Lowe H. Selective automated indexing of findings and diagnoses in radiology reports. J Biomed Inform. 2001;34(4):262–73.
15. Hersh WR, Leone T. The SAPPHIRE Server: A New Algorithm and Implementation. SCAMC 95. Philadelphia: Hanley & Belfus Inc, 1995: 858–62.
16. Elkin PL, Tuttle MS, Keck K, Campbell K, Atkin G, Chute C. The role of compositionality in standardized problem list generation. Proceedings of MEDINFO 98. Amsterdam: IOS Press, 1998: 660–4.
17. The Systematised nomenclature of human and veterinary medicine: SNOMED International. In: Cote RA, Rothwell DJ, Palotay JL, Beckett RS, Brochu L, (eds). The Systematised nomenclature of Humans and Veterinary Medicine. Northfield: College of American Pathologists, 1993.
18. Spackman KA, Campbell KE. Compositional concept representation using SNOMED: towards further convergence of clinical terminologies. Proc AMIA Symp. 1998:740–4.
19. Berrios DC, Kehler A, Fagan LM. Knowledge requirements for automated inference of medical textbook markup. Proc AMIA Symp. 1999:676–80.
20. Aronson AR, Bodenreider O, Chang HF, Humphrey SM, Mork JG, Nelson SJ, et al. The NLM Indexing Initiative. Proc AMIA Symp. 2000:17–21.
21. Srinivasan P, Rindflesch T. Exploring text mining from MEDLINE. Proc AMIA Symp. 2002:722–6.
22. Berrios DC. Automated indexing for full text information retrieval. Proc AMIA Symp. 2000:71–5.
23. Rindflesch TC, Tanabe L, Weinstein JN, Hunter L. EDGAR: extraction of drugs, genes and relations from the biomedical literature. Pac Symp Biocomput. 2000:517–28.
24. Srinivasan S, Rindflesch TC, Hole WT, Aronson AR, Mork JG. Finding UMLS Metathesaurus concepts in MEDLINE. Proc AMIA Symp. 2002:727–31.
25. Friedman C, Alderson PO, Austin J, Cimino JJ, Johnson SB. A general natural language text processor for clinical radiology. J Am Med Inform Assoc. 1994;1:161–74.
26. Friedman C, Starren J, Johnson SB. Architectural requirements for a multipurpose natural language processor in the clinical environment. In: Gardner RM (ed). Proceedings of SCAMC 1995. Philadelphia: Hanley & Belfus, 1995:347–51.
27. Cimino JJ, Clayton PD, Hripcsak G, Johnson SB. Knowledge Based approaches to the maintenance of a large controlled medical terminology. J Am Med Inform Assoc. 1994;1:35–40.
28. Liu H, Lussier Y, Friedman C. Evaluating Abbreviations in the UMLS. Proc 2001 AMIA. Philadelphia: Hanley & Belfus, 2001: 393–7.
29. Liu H, Johnson SB, Friedman C. Automatic resolution of ambiguous terms based on machine learning and conceptual relations in the UMLS. J Am Med Inform Assoc. 2002;9:621–36.
30. Liu H, Lussier YA, Friedman C. Disambiguating ambiguous biomedical terms in biomedical narrative text: an unsupervised method. J Biomed Inform. 2001;34(4):249–61.
31. Friedman C, Hripcsak G, Shagina L, Liu H. Representing information in patient reports using natural language processing and the extensible markup language. J Am Med Inform Assoc. 1999;6:76–87.
32. Hogg RV, Tanis EA. Probability and Statistical Inference 6 (ed). Upper Saddle River, NJ: Prentice Hall, 2000.
33. Hripcsak G, Friedman C, Alderson PO, DuMouchel W, Johnson SB, Clayton PD. Unlocking clinical data from narrative reports. Ann Intern Med. 1995;122:681–8.
34. Knirsch CA, Jain NL, Pablos-Mendez A, Friedman C, Hripcsak G. Respiratory isolation of tuberculosis patients using clinical guidelines and an automated decision support system. Infect Control Hosp Epidemiol. 1998;19:94–100.
35. Lussier Y, Shagina L, Friedman C. Automating SNOMED Coding using medical language understanding: a feasibility study. Proc 2001 AMIA. Philadelphia, PA: Hanley & Belfus, 2001: 418–22.
36. Lussier Y, Shagina L, Friedman C. Automated ICD-9 encoding using medical language processing: a feasibility study. Proc AMIA Symp. 2000:1072.
37. Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, et al. Gene ontology: tool for the unification of biology. Nat Genet. 2000;25:25–9.