Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Biomedical Informatics 36 (2003) 189–201
Journal of Biomedical Informatics
www.elsevier.com/locate/yjbin

A vocabulary development and visualization tool based on natural language processing and the mining of textual patient reports

Carol Friedman,[a,*] Hongfang Liu,[a] and Lyudmila Shagina[a]

[a] *Department of Medical Informatics, Columbia University, 622 West 168 Street, VC-5 Bldg, New York, NY 10032, USA*

Received 16 May 2003

Abstract

Medical terminologies are critical for automated healthcare systems. Some terminologies, such as the UMLS and SNOMED are comprehensive, whereas others specialize in limited domains (i.e., BIRADS) or are developed for specific applications. An important feature of a terminology is comprehensive coverage of relevant clinical terms and ease of use by users, which include computerized applications. We have developed a method for facilitating vocabulary development and maintenance that is based on utilization of natural language processing to mine large collections of clinical reports in order to obtain information on terminology as expressed by physicians. Once the reports are processed and the terms structured and collected into an XML representational schema, it is possible to determine information about terms, such as frequency of occurrence, compositionality, relations to other terms (such as modifiers), and correspondence to a controlled vocabulary. This paper describes the method and discusses how it can be used as a tool to help vocabulary builders navigate through the terms physicians use, visualize their relations to other terms via a flexible viewer, and determine their correspondence to a controlled vocabulary.

Keywords: Natural language processing; Controlled vocabulary; XML-based graphical user interface; Text mining; Medical terminology

1. Introduction

Computerized healthcare systems can revolutionize medicine because they enable implementation of guidelines [1–6], decision support systems [7–10], quality assurance applications [1], improved access to the literature [11–13], as well as facilitate research through availability of large amounts of online patient data [14,15]. However, in order for widespread use and interoperability, automated systems must be able to communicate through a common terminology or be capable of being mapped between different terminologies [16–21].

Cimino [22] proposed that completeness is essential for a controlled vocabulary. Starren and Johnson [23] examined the completeness of the BIRADS coding system, which was developed by committee to represent relevant findings in mammography reports. By examining a sample of mammography reports, they found that critical concepts were missing from BIRADS. Elkin and colleagues [24,25] argued that compositionality is also an essential feature for a vocabulary. Zhang [26] suggested that external or controlled vocabularies (e.g., SNOMED [27], the UMLS [28]) are artifacts created to categorize the medical domain in a systematic way, which differ from the internal representations of the medical concepts as they exist in the minds of users, and believed that a critical issue is the relation between the external and internal models. We propose that a vocabulary system that is developed without regard to physician usage may not be complete and may not be intuitive for physicians to use, and that a system that can readily be linked to terminology physicians use will be more helpful for furthering the functionality of automated systems because it will be based on terms naturally expressed by physicians.

In this paper we present a method that helps clinical system builders capture and view terms physicians use. A description of an early version of the method was presented by Liu and Friedman [29], but the functionality has been expanded significantly since then. The method

* Corresponding author. Fax: 1-212-305-3302.
E-mail address: friedman@dbmi.columbia.edu (C. Friedman).

1532-0464/$ - see front matter © 2003 Elsevier Inc. All rights reserved.
doi:10.1016/j.jbi.2003.08.005

determines clinically relevant terms from a collection of medical reports, as well as their frequency, the frequency of related modifiers and other clinical terms. It also identifies the compositional nature of the terms, and determines mappings to a controlled vocabulary. This method is achieved by the use of a corpus of textual medical reports from a particular domain, an existing NLP system MedLEE [30] that generates XML structured output of the clinical information in the reports, and two additional components that manipulate and integrate the XML form generated by MedLEE. The method also includes a graphical user interface, called DynTreeViewer that provides navigation functionality by allowing users to browse the vocabulary easily in order to find out relevant information associated with the terms. An important new feature is that users can view the information organized by a variety of views, which they specify dynamically. The use of XML in the method provides flexibility because it allows users to dynamically view, navigate, manipulate, and edit the XML form. XML is also convenient to use because of the availability of tools for XML, such as XPATH [31], XSL [32], and XML parsers [33]. Another advantage of using XML is that it intrinsically is a tree and therefore can be manipulated through the use of various tree transformation languages such as XSLT [34] and DOM [35]. This method is not dependent solely on MedLEE or XML, and it may be used in conjunction with another NLP system. The most significant features are that a corpus must be used to capture the domain terminology, and an NLP system must be used to generate structured output by processing the corpus. In addition, the output must represent the compositionality of the terms.

2. Related work and background

Some work has been reported concerning vocabulary development based on natural language processing and large corpora of patient records. One method discussed by Kreis and Gorman [36] used term frequency analysis as a tool for designing a structured data entry system. Hersh et al. [37] used NLP methods to identify clinical findings in a large corpus of patient reports, which involved identification of noun phrases. He compared his findings with the UMLS Metathesaurus and determined that modifiers expressed by physicians were not included in the Metathesaurus. Elkin et al. [25] studied the compositional nature of clinical vocabularies and developed tools to assist users in composing complex clinical terms. Chute et al. [38] discussed desiderata for a clinical terminology server and proposed that the server should be capable of proposing coordinated standard terms. Cimino et al. [22] discussed desiderata for a clinical vocabulary and proposed that completeness was one of the requirements. Bodenreider et al. [39] proposed an unsupervised corpus-based method for extending the UMLS by finding new candidate terms through shallow syntactic analysis of MEDLINE phrases and UMLS terms as well as use of the UMLS semantic categories. That study focused on adjectival modification within the domain of disorders and procedures.

There are several interface systems for accessing and viewing terminologies in the biomedical domain. One is the UMLS knowledge source server [40], which contains a set of Web-based interaction tools, and an interface for computer programs that allows users and developers access to the biomedical terminologies found within the UMLS. Another system is called Metaphrase [41], which is a practical terminology server in healthcare enterprises. At New York Presbyterian Hospital, the Medical Entities Dictionary (MED) [42] has been used. Users can access the MED through either a browser called MEDviewer or an editing interface called MEDitor [43]. However, most of these interfaces are Web-based retrieving interfaces: they depend on the user to input a specific term. They then retrieve local information specific to that term and present the result through the Web.

Our method differs from the above types of vocabulary servers as well as their graphical user interfaces because it is not focused on a controlled vocabulary or ontological relations, such as a hierarchy. Instead, it shows usage statistics and the compositional structure associated with all candidate terms that are obtained from patient records. These candidate terms can then be included or excluded from the controlled vocabulary, based on expert review. Thus, it is not a controlled vocabulary system, but rather a controlled vocabulary development tool. It is also different from methods that propose lists of candidate terms because instead of proposing terms, it allows users to visualize the compositional structure of the terms in the corpus, determine their frequency and relation to other terms and, if desired, to establish links between the terms and a vocabulary system. Additionally, the user interface, DynTreeViewer, is not static, but is a flexible and dynamic navigation interface based on a tree structure. It provides different views of the data, which are not predefined but are dynamically specified by the user. DynTreeViewer is based on the use of XML to represent terms and related information. In the particular application we discuss in this paper, the XML tree was obtained by using a natural language processing system, called MedLEE.

3. MedLEE background

MedLEE processes text reports and generates output in the form of XML. A more detailed description of the MedLEE XML output form and a description of the

corresponding DTD is discussed by Friedman et al. [44]. Fig. 1 provides an example of the output form for the sentence *He has lower extremity edema*. The output for a sentence contains two components: one is called **structured**, which encloses the structured findings, and has an attribute **form** whose value is the output format, which in this case is XML. The second component contains XML tagged text (**tt**) of the original text, which is used to link the structured information to the original text. The structured component in this example consists of one clinical finding, which is an XML tag **problem** that has an attribute **v** (i.e., value) with value "edema." The term also corresponds to two UMLS concepts "edema" and "edematous," which are represented as values of an attribute called **umls**. The method used to determine the UMLS codes are not discussed in this paper, but will be described in detail in a subsequent paper. A clinical finding can have modifiers, and each modifier can also have modifiers. In Fig. 1, "edema" has a body location modifier **bodyloc** with a value "extremity," and in turn, "extremity" has a modifier **region** with a value "lower." References to the original text are represented using the **idref** attribute. These refer to portions of the tagged text component that have a tag **phr**, which have an **id** attribute that is equal to the value of the **idref** attribute. For example, in the structured component, the **idref** associated with "edema" is "p6," and in the **tt** component, the text enclosed in the **phr** tag that has an **id** attribute equal to "p6" is *edema*. In the structured component, "edema" has another modifier **certainty** whose value is "high certainty." If we follow the link to the original text, we can see that this was generated from *has* in the sentence. "Edema" has other modifiers (e.g., **parse mode**, **sectname**, and **sid**), which are contextual and are not in the actual sentence. They represent values for the parse

```
<sentence>
  <structured form = "xml">
    <problem v = "edema" idref = "p10">
      <bodyloc v = "extremity" idref = "p8">
        <region v = "lower" idref = "p6"/>
      </bodyloc>
      <certainty v = "high certainty" idref = "p4"/>
      <sid idref = "s1.1.1"/>
    </problem >
  </structured>
  <tt>
<sent id = "s1.1.1"> He <phr id = "p4"> has </phr> <phr id = "p6">
lower </phr> <phr id = "p8"> extremity </phr> <phr id = "p10"> edema
</sent> </tt>
</sentence>
```

Fig. 1. Example of XML output generated by MedLEE as a result of processing the sentence *He has lower extremity edema.* The XML output for the sentence consists of two components: a component **structured** consisting of the structured findings and modifiers, and a tagged text component **tt**, which consists of the original sentence with **phr** tags. The tags have identifiers (e.g., **id** attributes), which are assigned so that the tags in the structured component can be linked to the original text.

method used to obtain the output, the section that the sentence occurred in, and the identifier for the sentence. An additional modifier is the UMLS code, which was derived from the structured output. The UMLS encoder attempts to find the most specific UMLS concept that matches the problem along with the modifiers. In this example, MedLEE determined that the UMLS concept corresponding to "edema of lower extremity" was the most specific code. The links "p6 p10 p12" referring to the original text are also shown. Notice that the **umls** attribute refers only to the corresponding value "edema" without modifiers, whereas the **umls** modifier tag is more specific because it refers to the term with modifiers.

Data in an XML document, by nature, forms a tree where data associated with each tag is a node in a tree. XML data can be viewed using various XML navigation tools such as Internet Explorer. Additionally, a tree is among the most effective navigation structures, as seen in applications such as Windows Explorer. In a tree, the information is displayed in a hierarchical order where the more general topic is displayed at the top level while related or more specific items are stored as descendents.

4. Methods

4.1. Generating a XML vocabulary tree

An overview of the overall system is shown in Fig. 2. Initially, a large set of clinical reports is collected. These can be reports from a specialized domain, such as echocardiography or pathology reports, or from a broad variety of clinical reports, depending on the intended use. For the figures shown in this paper, 1000 complete discharge summaries were used as the collection.

The first step in the overall process consists of processing the text sentences using MedLEE to generate XML output consisting of primary clinical events and modifiers. The original XML output is then simplified by removing contextual tags (i.e., **parse mode**) and by incorporating the original text phrases that the concepts were derived from into the structured component. For example, the simplified XML output for the example is shown in Fig. 3A. The **idref** attributes were removed and the original phrases were included as nodes called **source**. The value for **source** was obtained by concatenating the corresponding original phrases in the **tt** component where concatenation is performed according to order in the original text. For brevity, function words in the text (e.g., *in*, and *of*) that do not change the underlying meaning, are not shown. For example, the structured finding "edema" was obtained from textual phrase *edema*, and has a **source** node as its child. Similarly, "edema" with a body location modifier "extremity" is associated with a **source** node with the value

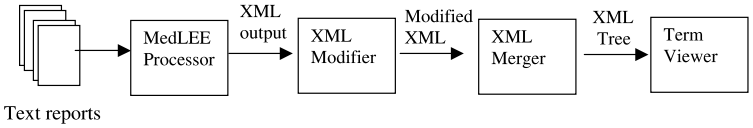

Fig. 2. Overview of vocabulary development method. There are four processing steps. The first step consists of using MedLEE to process the text reports, to generate output that represents the compositional structure of the clinical information. The XML output is then modified by the XML Modifier in preparation for the next step. After modification, the individual clinical events are merged by XML Merger to form a single XML tree. The XML tree can then be viewed by the graphical user interface, DynaTreeViewer.

"extremity edema." Note that in order to simplify the output for illustrative purposes, **umls** attributes and **umls** modifiers are not shown.

The second step of the tree building process consists of modifying the XML output generated by MedLEE using XML Modifier. This component is needed in order to separate the original tags that represent the type of information from the **v**(alue) attributes so that the user can be provided with a view where different values of the same type of information can be grouped together, which will be explained in the next step. Another function of this component is to compute the number of occurrences of the different clinical events. Fig. 3B shows the XML tree after modification. Note for brevity, we do not show the tag **source** in Fig. 3B. In this example, the attribute value "edema" was removed from the **problem** tag and an **item** tag with that attribute was inserted as its child; a similar transformation was performed for the **region** tag. In addition, a frequency attribute **fv**, which has a value of **40**, was added to all the tags. This signifies that there were 40 occurrences in the corpus of the components of the compositional concept *edema in lower extremities*.

The third step of the process consists of merging the XML trees representing each of the individual clinical events so that similar types of information occur

```
<problem v = "edema"> <source v = "edema"/>
    <bodyloc v = "extremity"> <source v = "extremity edema"/>
        <region v = "lower"> <source v = "lower extremity edema"/>
        </region>
    </bodyloc>
</problem>
```

A

```
<problem fv = "40">
    <item fv = "40" v = "edema">
        <bodyloc fv = "40">
            <item fv = "40" v = "extremity">
                <region fv = "40">
                    <item fv = "40" v="lower"/>
                </region>
            </item>
        </bodyloc>
    </item>
</problem>
```

B

```
<problem fv = "75">
  <item fv = "75" v = "edema">
    <bodyloc fv = "75">
      <item fv = "41" v = "extremity">
        <region fv = "41">
          <item fv = "40" v="lower"/>
          <item fv = "1" v="upper"/>
        </region>
      </item>
      <item fv = "34" v = "lung"/>
    </bodyloc>
  </item>
</problem>
```

C

Fig. 3. (A) Original XML output generated by MedLEE for *edema of lower extremity*. The primary event is a tag called **problem**, which has an attribute **v** with value "edema." The problem tag has nested body location modifier with value "extremity." Similarly, the **bodyloc** tag has a nested tag **region** with the value "lower." (B) Illustrates the tree after XML modification has been performed. The **v** attributes were replaced with nested tags called **item**. Additionally, a frequency attribute **fv** with the value "40" was added to each tag. The figure illustrated by (C) depicts the XML tree after the merging of three trees associated with *edema of lower extremity*, *edema of upper extremity*, and *pulmonary edema*, which occurred 40, 1, and 34 times in the corpus, respectively. The frequency value for "edema" is "75"; the frequency value of the body location modifier whose value is "extremity" is "41," and the frequency value of the modifier whose value is "lung" is "34."

together in the tree. Frequency information is also updated during this process. Thus, if several terms have the same XML tree structure, the frequency attribute will be the summation of the frequency values of each of the XML trees. Additionally, if different XML trees have a common ancestor, the merge operation will merge them into one tree, and the frequency values for the common ancestor will be the summation of the frequency values of each tree. Fig. 3C illustrates the XML tree that was generated by merging XML trees associated with tree structures generated for *edema of lower extremity*, *edema of upper extremity*, and *pulmonary edema*, which occurred 40, 1, and 34 times in the corpus (note we do not show the **source** tag in Fig. 3C). Therefore, the item "edema" has an **fv** value of "75," the body location item "extremity" has an **fv** value of "41" because it is comprised of 40 occurrences containing *lower* and one of *upper*. Additionally, the body location item **lung** has an **fv** value, which is "34." The merged tree also contains the original source terms so that the user can see the actual terms that were processed to obtain the associated trees.

The steps of modifying and merging have been achieved through PERL Scripts and a PERL module called XML Parser. We also developed an alternative method that performed the same functions but used the XML transformation language XSLT and JAXP, a JAVA package for XML. However, we found that the latter implementation was much slower than the implementation based on PERL, and was too inefficient for large trees.

4.2. DynTreeViewer: a tool for visualizing and manipulating the tree

Once generated, the XML tree can be viewed using DynTreeViewer. DynTreeViewer is a graphical user interface programmed in JAVA, which enables visualization of trees as well as providing other functions. Since an XML document forms a tree, visualization and dynamic navigation were straightforward to implement using JAVA. Fig. 4 shows a snapshot of the interface after loading the generated vocabulary tree for 1000 discharge summaries and expanding the medication (i.e., **med**) sub-tree. Notice that the node with the most frequent value (i.e., **fv** with a value of 1330) under **med** is called "FILTER9." In order to dynamically reduce the size of the tree and make it more efficient and manageable to manipulate, a frequency filter is dynamically computed when reading in the tree. In this case, a frequency filter of 9 was computed. The filter replaced individual nodes that had a frequency value of less than nine by merging them into one common node called "FILTER9." In Fig. 4, the value of the first term under **med** is "FILTER9," signifying the number of medication structures that occurred less than nine times, and that were filtered out in this view. Two of the most frequent medications are generic terms (e.g., *medication* and *antibiotics*), whereas most of the remaining ones are more specific (e.g., *lasix*, *coumadin*, and *prednisone*). The most frequent MedLEE semantic types associated with terms from in this corpus are **problem** with a frequency of 39,319, and **procedure** with a frequency of 14,379.

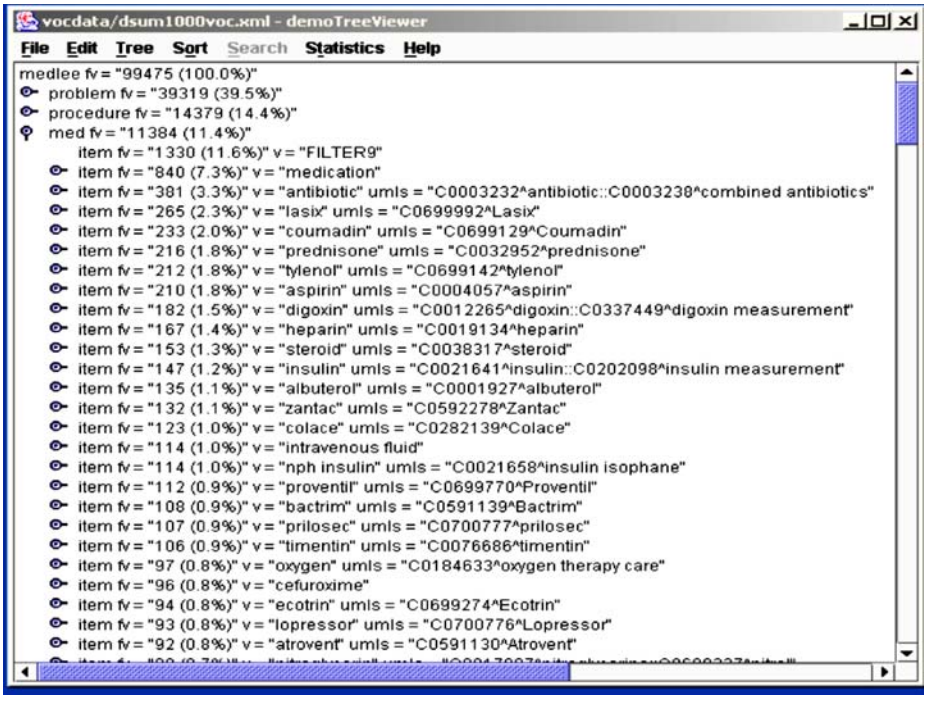

Fig. 4. A screen snapshot showing the user interface for DynTreeViewer that focuses on medications.

Fig. 5 shows a snapshot of the portion of the XML tree focusing on *edema*. The term (including variants) occurred 546 times in the corpus. It occurred with **certainty** values (i.e., values such as ''possible,'' ''no,'' and ''rule out'') 441 times, and with **body location** modifiers 359 times. Percentages are also computed relative to the frequency values of the parent node. Thus, the body location modifier occurred 65.75% (359/546) of the time in relation to ''edema.'' ''Extremity'' was the most frequent of the body locations (252/359) associated with ''edema'' whereas ''hip'' and ''thigh'' rarely occurred with ''edema.'' Another interesting aspect of the tree is that codes are shown when applicable. The coding is an additional option of MedLEE, and is table driven so that different coding systems or controlled vocabularies can be interchanged. For this paper, UMLS codes were obtained by MedLEE as part of the structured output. In the XML vocabulary tree, a code is represented as an attribute **umls** with a value that is the UMLS CUI (unique concept identifier) along with the associated preferred term. Thus, the item whose value is ''extremity'' has an attribute **umls** with a value ''C0085649^edema peripheral.'' Notice that in Fig. 5, not all of the terms are associated with codes. For example, no UMLS codes were found for ''edema of hip,'' ''edema of calf,'' or ''edema of shin.'' Sometimes more than one code is associated with a term. Thus, ''edema'' is associated with two UMLS concepts ''C0013604^edema'' and ''C0333239^edematous.''

The tag **source** shows the variety of original phrases in the text that correspond to the parent structure. Fig. 6 illustrates this feature for ''edema'' when it occurs with a body location modifier ''extremity.'' The frequencies of the original phrases are also shown. In this example, there are only several variations, some of which involve different combinations of upper and lower case. We can also see that there was only one occurrence of a phrase containing the source term *edematous*.

Fig. 7 provides another snapshot of the term tree using DynTreeViewer. It shows the different values of **problem** in order of descending frequency. This view was achieved by choosing an option in the **sort menu**, which is described below, to sort the children of each parent node according to frequency. This allows the user to focus on the most frequent clinical information first. The frequency value for the information type **problem** signifies that that type of clinical information occurred 39.5% of the time. Some other primary types of clinical events that occur but are not shown in this view of the tree are **procedure**, **medication**, **labtest**, and **body measurements**. According to Fig. 7, the most frequent child of **problem** was ''pain,'' which occurred in 3.5% of the problems. Other frequent problems were ''edema,'' ''hypertension,'' and ''fever.''

In addition to basic interface menus and operations, DynTreeViewer also contains additional functions in the form of menus that are useful for file handling, editing, vocabulary browsing, and development. The **File**, **Edit**, and **Help** menu perform the standard functions. The other menus are summarized below:

- **Tree** allows the user to expand and collapse the tree by one, two, or all levels. It also allows the user to

Fig. 5. A screen snapshot of the XML tree focusing on ''edema'' and some of the children, which are **bodyloc** modifiers. If a compositional term is associated with a umls code, it is shown as an attribute called **umls**, whose value is the UMLS concept unique identifier (CUI) and the associated preferred term.

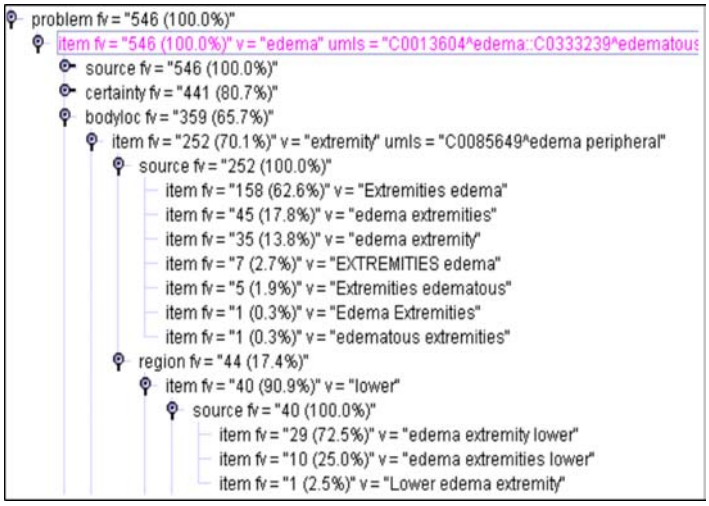

Fig. 6. This is a snapshot of the XML tree showing the values of a node called source, which is a child of ''extremity.'' The children of **source** show the original phrases from the corpus and their frequencies.

Fig. 7. This is a snapshot of the XML tree focusing on clinical conditions and their frequencies. The value of the first term is ''FILTER9,'' signifying that 7.8% of the terms occurred less than nine times, and were filtered out in order to reduce the size of the tree. Some other primary types of clinical events that are not shown in this tree are **procedure**, **medication**, **labtest**, and **body measurements**.

transform the tree to dynamically change the orientation. For example, the children of the root are typically primary events, such as **problem**, **procedure**, **medication**, **labtest**, etc. However, a first level node may also be a **bodyloc** node to obtain a body location oriented view of the terms or a **status** node to obtain a view oriented to temporal information. This is demonstrated in Fig. 8, which was obtained when viewing the XML tree by selecting the node ''chest'' and requesting a **lifting operation**, which is available in this menu. As a result, the body location node was lifted to become a top level node (i.e., to become a child of the root) of the tree and the resultant tree was organized by body locations. To obtain this view, the tree was transformed and, in addition, the frequencies were recomputed accordingly. In the snapshot shown in Fig. 8, a body location item with the value ''chest'' was expanded so that the user could see the various problems associated with it. The most frequent condition occurring with ''chest'' was ''pain'' (367 out of 507 occurrences of problems), and less frequent problems were ''thrush'' and ''rash.'' In this figure, a body location item with the value ''breast'' was also expanded to show the various problems associated with it, which are different than the problems associated with ''chest.'' Body location nodes between ''chest'' and ''breast'' were omitted in this view to save space. An attribute can also become a first level node by

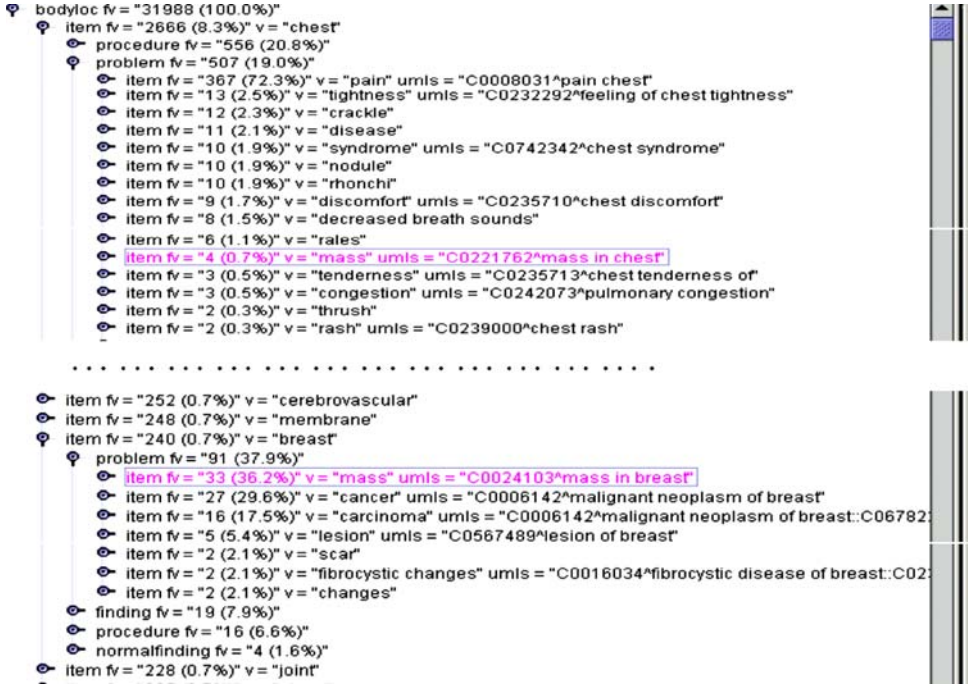

Fig. 8. This is a snapshot of the XML vocabulary tree, which focuses on a body location view. To obtain this view, the user requested that the **bodyloc** modifier be lifted up to become a top level node. The tree was transformed, and the frequencies were recomputed accordingly. In this view, some body location values were omitted, which is represented by the dotted line, in order to reduce the size of the figure.

selecting the attribute and requesting a lifting operation. In that case, the tree will be organized according to the attribute. This is demonstrated in Fig. 9, which was obtained by requesting that the attribute **umls** be lifted to be a top level node. In this view, users can see the frequencies according to controlled vocabulary terms.

- **Sort** allows the user to view the tree sorted in regular or reverse order according to the alphabetical order of the terms or of attributes, or numerically according to frequency. In the snapshots obtained for this paper, most of the views are in reverse numerical order so that the most frequent terms appear first.
- **Statistics** provides a report concerning the number of reports in the corpus if available, the number of nodes in the tree, the number of nodes with associated codes, and the number of nodes greater than a specific frequency value.

4.3. Testing the XML tree generation and DynTreeViewer

We tested the ability to generate an XML tree as well as the functionality of DynTreeViewer using two collections of documents. The first collection consisted of 1000 discharge summaries of inpatients at New York Presbyterian Hospital, and the second consisted of 104,149 radiological reports of the chest performed on patients at NYPH in the year of 1996. The discharge summary collection was used to test the methods on a set consisting of a broad variety of clinical information. The second set comprised a much larger collection of text than the discharge summary collection, and was used to test the methods on a large tree. In both cases, complete reports were used. The performance of MedLEE in processing the reports was not tested in this study, but previous studies have found that performance was satisfactory [14,4].

5. Results

The methods worked appropriately for each collection of output. For each collection, MedLEE generated XML output from which a single XML tree was successfully generated by the tree modification and merging methods. Using DynTreeViewer, it was possible to load and manipulated each tree properly. The node and attribute lifting functions worked properly, as did the sorting functions, and functions that computed the frequencies. Table 1 summarizes statistical information concerning the two corpora and the trees that were generated from each collection. There were 101,631 unique terms obtained from discharge summaries. These terms represent the clinically relevant source terms that were captured and structured by MedLEE. Of these, 60,145 different structures were generated when creating the tree because a common structure often represents phrasal variants. In addition, 47,905 (79.6%) of the structures contained UMLS codes either completely or

Fig. 9. This is a snapshot of an XML tree organized according to the attribute **umls**. To obtain this view, the user requested that the XML tree being viewed should be lifted so that the attribute **umls** becomes a top level node. The frequencies were recomputed accordingly. Note that several codes may be mapped to the same structure by MedLEE, as demonstrated by the codes corresponding to ''termination of pregnancy,'' ''pregnancy loss,'' and ''aborted pregnancy.''

Table 1
Statistics associated with the two corpora

	DSUM	CXR
No. of reports	1000	104,149
No. of words in corpus	545,000	8.9million
No. of unique cr terms	101,631	322,920
No. of unique cr structures	60,145	181,529
No. of structures with umls codes	47,905	97,108

DSUM denotes discharge summaries, CXR denotes radiological reports of the chest, and cr denotes clinically relevant.

partially. From the collection of radiology reports, there were 322,920 different clinically relevant phrases that were combined into 181,529 structures. Of these, 97,108 (53.5%) structures contained UMLS codes either completely or partially.

6. Discussion

There are some limitations to using a method based on MedLEE for vocabulary development. Because MedLEE is used to obtain parses of the sentences in the corpus, some clinical terms will be lost when a parse cannot be obtained. This occurs most frequently when a term is not in the MedLEE lexicon. This can become more evident when processing text in a domain that is new to MedLEE. In that case, MedLEE must first be refined so that the new terms are added to the lexicon prior to the vocabulary development effort. The vocabulary tree can still be useful for this purpose because MedLEE encloses unknown words with an **undef** tag in the **tt** portion of the output. The input to DynTreeViewer could be modified to incorporate undefined words into the tree, and to show their frequencies. This feature could be used to aid the knowledge engineer in identifying relevant clinical terms to add when adapting the NLP system to a new domain.

Another limitation concerns accuracy. Not all parses are completely accurate, which means that some semantic relationships in the XML tree will not be correct. We have found that for vocabulary development purposes this is not a problem. Since MedLEE has high accuracy, it is correct most of the time, and because errors are infrequent, they occur as noise. Since the majority of the vocabulary development work is associated with the more frequent terms, this is not a serious problem.

Another limitation may be scalability. If the size of the resultant tree becomes very large, certain operations,

such as loading the tree and dynamically lifting nodes or attributes may take an unacceptably long time. So far, we have tested a corpus of over 100,000 reports, and plan on testing a larger corpus. We found that filtering the tree based on frequency was necessary for efficiency and manageability to reduce the size of the tree, but then infrequent terms were lost. For example, it took 125 s on a Pentium 4 PC with 1 GB RAM to initially load the tree obtained from discharge summaries that consisted of 60,145 nodes, and 20 s to perform the lifting operation, but it took 34 s to load a tree (consisting of 7786 nodes) which had a frequency filter of 5, and 20 s to load a tree, which had a frequency filter of 9 that consisted of 4394 nodes. This could be a problem if rare terms were desired, but for vocabulary development purposes the focus is typically on capturing terms characteristically used by physicians.

In this paper, we have presented a vocabulary development tool for controlled clinical terminology based on terms found in actual reports. This tool has the advantages of providing (a) flexible views of the clinical information, (b) a display of the compositional nature of the terms, (c) a correspondence between the textual terms and controlled vocabulary concepts, (d) frequency of occurrence of the structured information, and (e) frequency of occurrence of text that corresponds to the structured form. Another significant advantage is that the tool is not based on simple string matching, but on the matching of semantic structures, where related terms are displayed even when there is a big string distance between them. For example, if the sentence *her extremities were nontender with moderate edema* were processed, an XML structure for **edema in extremity** will be generated that will be the same as for the sentences *edema in extremities*, and *edema in upper and lower extremities*. The structures for the corresponding sentences will not be identical, because some will have additional modifiers (e.g., *moderate*, *upper*, and *lower*) or additional main findings (e.g., *extremities were nontender*), but the portions of the XML tree associated with **edema in extremity** will be the same for all three.

Other advantages are that we were able to find additional information related to vocabulary from the composition of the tree. The tree was useful for identifying terms that are potentially compositional based on the frequency information provided. For example, there were 37 occurrences of descriptor modifiers associated with ''pneumonia,'' where the modifier values included ''reticular,'' ''focal,'' ''methicillin resistant,'' ''hospital acquired,'' ''subclinical,'' ''atypical,'' ''community acquired,'' ''diffuse,'' ''pan sensitive,'' and ''common,'' as shown in Fig. 10. Among these values, only ''atypical'' and ''community acquired'' occurred relatively frequently, signifying that it may be useful to include the compositional concepts **atypical pneumonia** and **community acquired pneumonia** in the controlled terminology, while the others may not be as useful because they are infrequent and could cause the vocabulary to increase substantially. Another advantage is that the tree is useful for identifying modifier values that are general and values that are specific to certain concepts, signifying that the specific ones are potentially compositional. This can be accomplished by lifting a particular modifier, in order to obtain a view specific to that modifier. By exploring the resultant tree, we can identify values that are associated with a very large number of different clinical findings versus modifiers that occur with only a few. Modifiers occurring with many different findings are likely to be true modifiers, whereas ones occurring with only a few

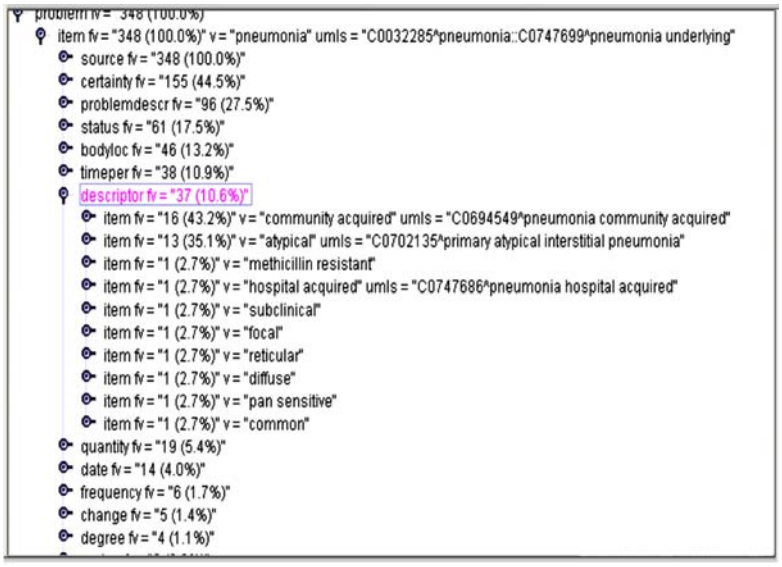

Fig. 10. Descriptor values associated with ''pneumonia'' organized by frequency.

findings are likely to be true parts of a term. For example, the certainty modifier value "rule out" occurred with more than 50 different findings, while the value "attempt" occurred with less than 10 different findings. The most frequent finding was "suicide," and therefore it is possible that the concept **suicide attempt** should be considered a single term.

Another characteristic that was interesting to explore consisted of comparing terms that were associated with negation and ones that were not. Ones that are infrequently negated may be diseases whereas the ones that are frequently negated may be symptoms or conditions. For example, edema occurred around 50% (276 out of 546) of the time with negation, while pneumonia occurred less than 3% (10 out of 348) of the time with negation. Another use would be to identify structures referring to the same concepts based on knowledge in the controlled vocabulary. Additionally, structural relatedness of terms in the controlled vocabulary can also be shown. In Fig. 11, the structural relatedness of UMLS concepts "C0000731^abdominal distension," "C0003899^swelling," and "C0232570^epigastric fullness," etc., is shown, and the semantic relation of these concepts may be easily detected.

Another significant feature of DynTreeViewer is that it is general and therefore is applicable to any well-formed XML structure. Therefore DynTreeViewer could be used as a flexible interface that provides dynamically oriented views to users. For example, it could be used to view a controlled vocabulary along with hierarchical information, as long as the vocabulary could be represented in XML form. DynTreeViewer could also be used for many different types of applications, such as for viewing different types of information in a patient registry or a patient problem list, and for organizing the views according to different orientations. Basically, any relational table can be exported to an XML form, and then viewed by DynTreeViewer.

DynTreeViewer could also be used to view an existing standard terminology that provides modifiers and has some compositionality. One such candidate is SNOMED CT. DynTreeViewer should be capable of using a large collection of medical reports to present the SNOMED CT tree with frequencies where frequencies can be computed either using string matching methods or using medical reports parsed and encoded into SNOMED by MedLEE.

The work presented in this paper is ongoing research. We initially focused on methods that organize and manipulate the XML structures to obtain a single tree, to compute frequencies, dynamic views, and perform sorting. The methods provided the desired functionality, and further work is needed in development of the interface by addressing interface design and usability issues.

7. Conclusion

We have presented a corpus-based method that displays term frequency, relations of terms to other terms, the compositional components of terms, and correspondences to an existing controlled vocabulary, such as the UMLS, via a flexible graphical XML-based user interface, which is intended to be used for facilitating vocabulary development. The method utilizes a natural language system and processes a large collection of patient reports in order to obtain clinical information in

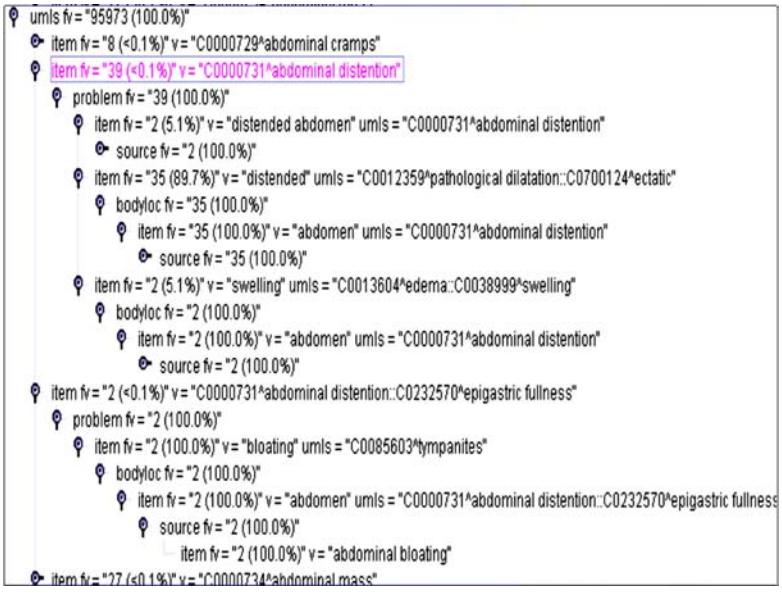

Fig. 11. Structural relatedness among different UMLS concepts.

structured XML form, which is then modified and merged so that it becomes one large XML tree. The tree can then be viewed, dynamically manipulated, and edited using a graphical interface called DynTreeViewer. This method was tested on two different corpora to ensure that it functioned appropriately. We believe the method provides substantial help for creating and enhancing vocabularies because it is based on text generated by physicians. Future studies will be aimed at evaluating utility and effectiveness.

Acknowledgments

This work was supported in part by Grant LM06274 from the NLM and from the Center for Advanced Technology at Columbia University.

References

[1] Bushko RG, Havlicek PL, Deppert E, Epner S. Impact of voice- and knowledge-enabled clinical reporting—US example. Stud Health Technol Inform 2002:80265–74.
[2] Ertle AR, Campbell EM, Hersh WR. Automated application of clinical practice guidelines for asthma management. In: Cimino JJ, editor. Proceedings of the 1996 AMIA fall annual symposium. 1996. p. 552–6.
[3] Fiszman M, Haug PJ. Using medical language processing to support real-time evaluation of guidelines. In: Proceedings of the AMIA symposium 2000, 2000. p. 235–239.
[4] Friedman C, Knirsch CA, Shagina L, Hripcsak G. Automating a severity score guideline for community-acquired pneumonia employing medical language processing of discharge summaries. In: Proc AMIA symp. 1999. p. 256–60.
[5] 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(2):94–100.
[6] Lenert LA, Tovar M. Automated linkage of free-text descriptions of patients with a practice guideline. In: Ozbold JG, editor. Proceedings of the 18th annual SCAMC, 1993. p. 274–278.
[7] Aronsky D, Fiszman M, Chapman WW, Haug PJ. Combining decision support methodologies to diagnose pneumonia. Proc AMIA Symp 2001:12–6.
[8] Chapman WW, Fizman M, Chapman BE, Haug PJ. A comparison of classification algorithms to automatically identify chest X-ray reports that support pneumonia. J Biomed Inform 2001;34(1):4–14.
[9] Hripcsak G, Cimino JJ, Johnson SB, Clayton PD. The Columbia-Presbyterian Medical Center decision-support system as a model for implementing the Arden Syntax. In: Clayton PD, editor. Proceedings of the 15th annual symposium on medical care. 1992. p. 248–52.
[10] Nadkarni PM. Clinical patient record systems architecture: an overview. J Postgrad Med 2000;46(3):199–204.
[11] Campbell D, Johnson, SB. Comparing syntactic complexity in medical and non-medical corpora. Proc AMIA Symp 2001;90–94.
[12] Sneiderman CA, Rindflesch TC, Aronson AR. Finding the Findings: Identification of Findings in Medical Literature Using Restricted Natural Language Processing. In: Cimino JJ, editor. Proceedings of the 1996 AMIA fall symposium, 1996. p. 239–243.
[13] Weeber M, Klein H, Aronson AR, Mork JG, de Jong-van den Berg, Vos R. Text-based discovery in biomedicine: the architecture of the DAD-system. J Am Med Inform Assoc 2000:903–7.
[14] Hripcsak G, Friedman C, Alderson PO, DuMouchel W, Johnson SB, Clayton PD. Unlocking clinical data from narrative reports. Ann Int Med 1995;122(9):681–8.
[15] Hripcsak G, Austin JHM, Alderson PO, Friedman C. Use of natural language processing to translate clinical information from a database of 889,921 chest radiographic reports. Radiology 2002;224(1):157–63.
[16] Delamarre D, Burgun A, Seka LP, Le Beux P. Automated coding of patient discharge summaries using conceptual graphs. Methods Inf Med 1995:34345–51.
[17] Heinze DT, Morsch ML, Sheffer RE, Jimmink MA, Jennings MA, Morris WC, Morch AEW. LifeCode—A deployed application for automated medical coding. Ai Magazine 2001;22(2):76–88.
[18] Jonassen K, Saboe R. The use of text encoding in the development of a terminology and knowledge system associated with the Norwegian version of the ICD-10. In: Greenes RA, editor. Proceedings of MEDINFO 95, 1995. p. 51–55.
[19] Lussier Y, Shagina L, Friedman C. Automated ICD-9 encoding using medical language processing: a feasibility study. In: Overhage M, editor. Proceedings of AMIA Symposium 2000, 2000. p. 1072.
[20] Lussier Y, Shagina L, Friedman C. Automating SNOMED Coding using medical language understanding: a feasibility study. In: Baaken S, editor. Proceedings of 2001 AMIA, 2001. p. 418–422.
[21] Sager N, Lyman M, Nhan NT, TIck LJ. Medical language processing: applications to patient data representation and automatic encoding. Methods Inf Med 1995:34140–6.
[22] Cimino JJ. Desiderata for controlled medical vocabularies in the twenty-first century. Methods Inf Med 1998:37394–403.
[23] Starren J, Johnson SB. Expressiveness of the Breast Imaging Reporting and Database System (BIRADS). Proc AMIA 1997:655–9.
[24] Elkin PL, Bailey KR, Chute CG. A randomized controlled trial of automated term composition. In: Chute CG, editor. Proceedings of the fall AMIA 1998 annual symposium, 1998. p. 765–774.
[25] Elkin PL, Tuttle MS, Keck K, Campbell K, Atkin G, Chute CG. The role of compositionality in standardized problem list generation. In: Cesnik B et al., editors. Proceedings of MEDINFO 98, 1998. p. 660–664.
[26] Zhang JJ. Representation of health concepts: a cognitive perspective. J Biomed Inform 2002:3517–24.
[27] Bidgood WD, editor. SNOMED DICOM Microglossary. Northfield, IL, Coll. Am. Path. 1997.
[28] Lindberg DAB, Humphreys B, McCray AT. The unified medical language system. Methods Inf Med 1993:32281–91.
[29] Liu H, Friedman CA. Method for vocabulary development and visualization based on medical language processing and XML. AMIA 2000:502–6.
[30] Friedman C, Alderson PO, Austin J, Cimino JJ, Johnson SB. A general natural language text processor for clinical radiology. JAMIA 1994;1(2):161–74.
[31] XPATH website; Available from: http://www.w3.org/TR/xpath20/. Last visited 8/5/2003.
[32] XSL webite; Available from: http://www.w3.org/XSL/. Last visited 8/5/2003.
[33] XML parser websites; Available from: http://www.alphaworks.ibm.com/tech/xml4j; http://xml.apache.org/. Last visited 8/05/03.
[34] XSLT website; Available from: http://www.w3.org/XSLT/. Last visited 8/5/2003.
[35] Domain object model website; http://www.w3.org/DOM/. Last visited 8/5/2003.
[36] Kreis C, Gorman P, Word frequency analysis of dictated clinical data: a user-centered approach to the design of a

structured data entry interface. In: Masys DR, editor. Proceedings of the 1997 AMIA fall annual symposium, 1997. p. 724–728.

[37] Hersh, WR., Campbell, EM., Evans, DA., and Brownlow, ND. Empirical, automated vocabulary discovery using large text corpora and advanced natural language processing tools. In: Cimino JJ, editor. Proceedings of the 1996 AMIA annual fall symposium, 1996. p. 159–163.

[38] Chute CG, Elkin PL, Sherertz DD, Tuttle MS. Desiderata for a Clinical Terminology Server. Proc AMIA Symp 1999:42–6.

[39] Bodenreider O, Rindflesch TC, Burgun A. Unsupervised, corpus-based method for extending a biomedical terminology. In: Johnson SB, editor. Proceedings of the workshop: natural language processing in the biomedical domain, 2002; p. 53–60.

[40] UMLS Knowledge Server. Available from: http://umlsks.nlm.nih.gov. Last visited 08/05/03.

[41] 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. Methods Inf Med 1998:37373–83.

[42] Cimino JJ, Clayton PD, Hripcsak G, Johnson SB. Knowledge based approaches to the maintenance of a large controlled medical terminology. J Am Med Inf Assoc 1994:135–40.

[43] Medical Entities Dictionary. Available from: http://med.dmi.columbia.edu. Last visited 08/05/03.

[44] Friedman C, Hripcsak G, Shagina L, Liu Hongfang. Representing information in patient reports using natural language processing and the extensible markup language. J Am Med Inf Assoc 1999:676–87.