Extracting Diagnoses from Discharge Summaries

William Long, PhD

CSAIL, Massachusetts Institute of Technology, Cambridge, MA, USA

Abstract

We have developed a program for extracting the diagnoses and procedures from the past medical history and discharge diagnoses sections in the discharge summary of a case and coding these using SNOMED-CT in the UMLS. The program uses a limited amount of natural language processing. Rather, it makes use of the relatively standard structure of the discharge summary, a small dictionary to divide the text into phrases, and the extensive collection of phrases for concepts in the UMLS to do the coding. With this approach the program finds 240 of 250 desired concepts with 19 false positives in 23 discharge summaries.

Keywords: diagnosis, UMLS, natural language processing, coding, text extraction

Introduction

Discharge summaries contain much useful information about the patient, useful not only for subsequent visits but for a variety of other tasks. To turn a collection of discharge summaries into an effective resource for research or quality assurance, it is necessary to index the cases. The most common need is to select a subset based on diagnoses or procedures done on the patients. The discharge summaries contain this information but the location, format, and description of diseases is highly variable. To make it usable, we need to extract the diseases and procedures and code them into a standard vocabulary.

There have been a number of efforts to extract useful data from discharge summaries. Friedman's group at Columbia have been prominent in using a detailed linguistic approach to natural language processing with their MedLEE parser to interpret discharge summaries and extract diseases and other information from them[1,2,3]. Another effort used triggering words to look for adverse events, with less success because of the variety of ways these events may be expressed[4].

Discharge summaries are organized into a number of sections. Usually included among these sections are the "past medical history" and possibly the "discharge diagnoses". These sections are lists of the diseases and associated procedures. The function of the program is to extract those diseases and procedures and code them. With the addition of SNOMED-CT to the UMLS, there is great incentive to use this resource to code the diseases into this widely recognized vocabulary[5]. The MetaMap program available from the NLM is a tool for coding phrases into the UMLS that provides a standard for comparison[6].

There are a number of problems that make disease extraction challenging. The diseases are found in sections with a variety of names, separated in several ways. The disease statements may have additional descriptive text. The diseases themselves may be more specific than is codable in a single code or there may be no code for the disease as described. We have developed a program that handles these problems sufficiently to extract and code almost all of the diseases in a training sample of 23 discharge summaries of patients in intensive care units of a teaching hospital.

Methods

Our strategy is to extract the diseases and procedures in the discharge summary by using the structure of the summary to locate the appropriate text, a very limited amount of natural language processing to find phrases that might contain the desired data, and the large number of phrases per concept in the UMLS to find the concepts. The natural language processing consists of recognizing punctuation, conjunctions, prepositions, a small number of common verbs, and a few other words unlikely to be part of a disease or procedure name, using these as boundaries to divide the text. We chose this approach rather than use an existing tagger and chunker to take advantage of the features of the discharge summaries and because we viewed the tagging as providing only limited benefit beyond this simple approach.

> HISTORY OF PRESENT ILLNESS: The patient was an 80-year-old…
>
> PAST MEDICAL HISTORY: Included chronic obstructive pulmonary disease, Wegener's vasculitis, steroid induced diabetes mellitus, no history of coronary artery disease.
>
> MEDICATIONS: On admission, …

Figure 1: Example Medical History

Figure 1 is text from a typical discharge summary showing a common format and the past medical history section with the list of diseases. This list along

with another from the discharge diagnosis section is what the program needs to code in SNOMED-CT.

To accomplish this task, the program first locates the sections with the desired information, determines the individual descriptions in those sections, divides the descriptions into phrases, looks for codes covering the maximum length phrases by normalizing the words and looking up the phrases in the UMLS, augments these with any modifiers, and provides this list as the result, as described below.

Finding the Diagnosis Sections

Most discharge summaries have both a list of diagnoses for the patient as they arrived and as they left. The arrival diagnoses are usually called the "past medical history" (as in figure 1) or "admitting diagnoses" and the exit diagnoses are called the "discharge diagnoses". There are variants: the "diagnoses" may be "diagnosis". More generally, anything labeled with "diagnosis" or "diagnoses" is probably relevant. On the other hand there are several possible sections labeled with "history" that are not relevant, including the "social history".

The discharge summary is divided into sections usually with a label in upper case and separated with a colon as in figure 1. Each section may have multiple paragraphs. Sometimes, the past medical history is a paragraph within the "history of present illness", identified just by the phrase in the text. Another possibility is encountering "history of" in the history of present illness. There are also instances where there is no past medical history. We encountered this in some trauma cases where any previous diseases are not relevant to the current problem. The case may not have discharge diagnoses either, certainly when the patient dies but also in other circumstances.

The program first tries to recognize how sections are labeled by looking for some standard section names at the beginning of a line and determines how these are formatted (e.g., upper case followed by a colon). Once the pattern of labeling is identified, the program looks for sections labeled as diagnoses or medical history. If there is no labeled admitting diagnosis or medical history section, the program looks for the present illness section and tries to find a paragraph starting with an appropriate phrase. Otherwise the program looks for "history of" followed by a disease or procedure in the present illness section.

Individual Diseases

The sections with diseases often consist of numbered lists, which helps identify what should be considered an item and how any modifiers should be associated. If the diseases are numbered ("1.", "#1", etc.), the program uses the numbering to separate the section into separate diseases. In other summaries, the diseases are just separated by punctuation as in figure 1 and that punctuation is used to separate the items. In figure 1 the program uses the commas to identify four phrases for potential coding into diseases.

Phrases

Once the program has identified a segment of text likely to have a disease, it still needs to divide it into phrases. A typical statement is: "Congestive heart failure with hospitalizations in June and July 2001 and an ejection fraction of 20-30%." The program uses a dictionary of about 200 common punctuation marks and words including prepositions, conjunctions, and other common words that are not usually part of the disease name to divide the statement into phrases. In this case, the phrases are "Congestive heart failure", "hospitalizations", "June", "July 2001", "ejection fraction", and "20-30%". This serves to focus the coding problem on phrases, which if they are of the desired types, are likely to be in the UMLS.

Normalization and Coding

Next we find the longest contiguous subphrase that can be coded. This is done using the UMLS normalized string index. In order to look up a phrase, it is necessary to normalize it. The normalization algorithm is not easily duplicated (sometimes normalization even divides a word into two as "cerebrovascular" is normalized to "cerebro vascular") so we use the "norm" program (part of the specialist lexical tools available from the NLM[7]) to normalize each word, sort the normalized words in the desired phrase to match the normalized phrase structure in the index, and check the UMLS. Looking for the maximal coding returns a code for "steroid induced diabetes mellitus" in figure 1. If the whole phrase is not found, the possible subphrases are tried in order of length.

Some words have more than one normalization (e.g., femara normalizes to femara, femaron, or femarum) so there is potential for combinatorial search. Fortunately, there are often more than one of the possible normalizations of a phrase indexing the same concept, so the program is likely to find the concept even if not all possible normalizations are tried. One heuristic that greatly reduces the search is to normalize a word to itself only, when that is one of the possible normalizations. This still leaves a search through some normalization options ("lower" normalizes to

"low" and "lour") but most words then have a single normalization.

Concept Types

The next problem is to determine which terms are diseases, procedures, medications, etc. The UMLS provides a mapping from concept to type, but there are several types that would be considered diseases and the other categories of interest. Table 1 shows UMLS types mapped to *disease* and an example concept.

UMLS type	Example concept
Disease or Syndrome	Acute myocardial infarction
Fungus	Candida parapsilosis
Injury or Poisoning	Fractured nasal bones
Anatomical Abnormality	Ventral hernia
Congenital Abnormality	Congenital hemangioma
Acquired Abnormality	Nodule
Mental or Behavioral Dysfunction	Dementia
Hazardous or Poisonous Substance	Cocaine
Neoplastic Process	Hemangioma
Pathologic Function	Hematoma

Table 1: Mapping of UMLS types to *disease* and examples

Many concepts have more than one UMLS type, so the program assumes that if there is a type that maps into disease that is the one to use. For example, *Cocaine* also has the type "Pharmacologic Substance", but listed in a diagnosis section, the program takes this as a disease. If the program had found the phrase "cocaine abuse", that would have been coded as a concept with type "Mental or Behavioral Dysfunction" and hence mapped to *disease*. Our mapping of types to disease is similar to that of McCray,[8] except that we included "Hazardous or Poisonous Substance" and "Fungus" as diseases because of the context.

In the congestive heart failure example above, *congestive heart failure* is a concept classified as a disease, while *hospitalization* and *ejection fraction* are concepts but not diseases or procedures and while *hospitalization* is close enough in the text to be a modifier it is not of an acceptable type, so these concepts are discarded.

Some concepts span a separator. That is, there may be a comma or preposition in the concept. For example, "lung cancer" may be described as "cancer of the lung". When the program finds a disease, and the separator for the next phrase is one of those that may link a concept, the program tries to find a more encompassing code. This strategy counts on the concept part bounded by separators also being a disease concept, but it is effective in our training set and keeps the search for maximal concepts tractable.

Not every disease or condition in a discharge summary has a single concept in the UMLS exactly corresponding to it. The most common problem is a condition that may apply to a variety of parts of the body, such as a fracture. While some specific concepts for fractures exist (*fractured nasal bones* in table 1), most are missing. The program handles this by allowing modifiers. Thus, "left front intracranial hemorrhage" is coded as *Intracranial haemorrhage NOS*, *left*, *front*. The program allows as modifiers, spatial (e.g., bilateral), body parts and locations, temporal (e.g., chronic), and qualitative concepts (e.g., severe). This results in a few unneeded modifiers such as *history of* as a temporal modifier, but no confusion. The search for modifiers also looks beyond some separators and attempts to associate them with the correct disease.

The most important modifiers are those indicating the absence of a disease rather than the presence. Thus, "no history of coronary artery disease" in figure 1 needs to be either removed or put in a category of negative findings. We do not have a large enough training set to determine the most likely ways these are represented but they certainly include "no", "ruled out", etc. This problem has been handled effectively for findings and diseases[9,10] so a published algorithm should be effective for a larger data set. The problem of negatives for diseases is simpler than that for findings since constructs like "denies" or "no sign of" are not used for diseases.

Sometimes in a phrase with a disease there are words for which there is no code in the UMLS or there may be a phrase coded as a disease connected by "and" or "or" for which there is no UMLS code. In such cases, it is likely that the uncoded words are part of a disease description and should be extracted from the text. In the example in figure 1, "Wegener's vasculitis" has no UMLS code but "vasculitis" is a disease and there is no code for "wegener". There is a concept for "Wegener's granulomatosis" but that is inaccessible from the vasculitis concept. The program would code this as *vasculitis*, "Wegener's".

The final step is simply to pull out the disease and procedure concepts identified with their modifiers for indexing.

Results

This program was developed on a training set of 23 discharge summaries. While we can not say how well it will perform on other cases until we have a test set, we can say what problems it encounters on the training data. The first issue is what should be considered a disease or procedure. An unbiased criterion is whatever the writer of the discharge summary listed in the diagnosis sections. From a reading of the discharge summaries this includes some symptoms and findings, such as "history of increased INR" or "falls". To include these items, we ran the program looking for diseases, procedures, symptoms, and findings in the diagnosis parts of the 23 summaries. By this criterion there were 250 such statements. The program missed 10 of these and got 19 false positives. 18 of the false positives were the result of not locating the end of the past medical history when it was part of the history of present illness and ended in the middle of a paragraph. This could be corrected by recognizing such phrases as "presented with", "on the DOA", or "on the day PTA" as starting the description of the present illness. The missed diagnoses represent a variety of problems.

Two of the missed diagnoses were abbreviations: "XRT to larynx" meaning X-ray therapy to larynx as a procedure, and MVR meaning mitral valve replacement in the phrase "CABG/MVR". Some abbreviations are in the UMLS, e.g., CABG is properly coded. Medications were missed as the cause of something else: "interstitial lung disease secondary to methotrexate". This could be corrected by adding medications to the extracted concepts but we would need to distinguish between phrases indicating causation versus treatment. The program left ambiguous phrases uncoded. "Breast reduction" has two different codes, one of which is a procedure and the other is a finding, so it was not coded.

Half of the missed diagnoses are phrases that many physicians would not consider diagnoses. These include "borderline high cholesterol", "loss of appetite", "anxiety" (mental process), "bronchoscopy" (diagnostic procedure), and "deconditioning".

Nonstandard phrasing in "polysubstance abuse (cocaine and alcohol)" caused a miss. The UMLS has codes for *polysubstance dependence*, for *cocaine abuse* and *alcohol abuse*, but the program was incapable of assembling a disease out of the phrase.

There are also two diagnoses that were added to the knowledge base manually. We added *depression* as a disease because it is ambiguous in general but in the context of a disease list it is a disease. We also added *tamponade* as cardiac tamponade because it is very important in our patient set and would only rarely mean anything else.

There were also a few modifiers that were missed or misattributed. For example, in "right 1.3cm infiltrating ductal carcinoma" the "right" was missed because size was not considered a modifier so it was isolated. Since many kinds of statements could separate modifiers from their object, this is a more general problem requiring more natural language processing.

We compared the results of the program to the UMLS codes returned by MetaMap applied to the same phrases extracted from the summaries. MetaMap missed 31 of the diagnoses and had 23 false positives. 8 of the missed diagnoses were "hypertension" which was coded as *hypertension induced by pregnancy,* which the program avoids by preferring concepts with names close to the phrase it is looking for.

Discussion

Spelling correction could be used to improve the performance. For example, the program missed the phrase "noninsulin dependent diabetes mellitus" because "noninsulin" is not in the UMLS. The program correctly finds the phrase if "non insulin" or "non-insulin" is substituted. A spell correcting program will make this substitution.

The multiple dictionaries in the UMLS have significant benefits for coding but also introduce problems. The problems in this set of summaries were manifest as phrases with more than one possible encoding. For example, "lower extremity" maps to two codes: *lower extremity* and the procedure *knee strapping*; "cardiac catheterization" maps to *cardiac catheterization procedure* and *cardiac catheterisation as the cause of abnormal reaction of patient, or of later complication, without mention of misadventure at the time of procedure* which is an injury and therefore one of our disease categories; and "down" which maps to *downward* and *Down's syndrome*. Since each undesired mapping was to a type the program was looking for, the simple heuristic of picking the disease or procedure code gets the wrong code. In each case the undesired mapping was contributed by a dictionary other than SNOMED-CT. On the other hand, if we were to only use mappings from strings to codes in SNOMED-CT, the program would miss 27 correct mappings that were found using the multiple dictionaries. These included several abbreviations such as CHF, CABG, TAH BSO, and UTI; more specific encodings including the "steroid induced diabetes mellitus" in the example, "sacral decubitus ulcer", and "kidney stone removal"; and commonly used names including "obesity", "diabetes", "type 2

diabetes", "lung cancer", and "chronic renal insufficiency". The appropriate strategy seems to be to restrict some dictionaries from consideration, but we will need more data to determine the appropriate set to include.

We considered restricting the concepts extracted to just diseases and procedures but that did not match closely the concepts listed in the discharge summaries in the disease sections. Patients who are hospitalized as the result of an accident have a finding rather than a disease. For example a patient with a fall complicated by a lumbar artery bleed has no diseases as classified by the UMLS. Another common problem is to use a finding or symptom to identify the disease. One patient was described as having a "history of angina" rather than coronary artery disease. Another had "bradycardia" without specifying the cause. One patient had "recurrent right pneumothorax". *Pneumothorax* is a disease but *right pneumothorax* is a finding and "recurrent" is a modifier. Since the program looks for the longest phrase to code, it selects *right pneumothorax*, *recurrent*, which is a finding.

It is clear that there are situations where better natural language processing would help. However, in the single example where the program was unable to code the disease, "polysubstance abuse (cocaine and alcohol)", it would take a deep knowledge of English to associate the drugs with "abuse" and unlikely that a parser designed for general English would know what to do with "polysubstance". Most of the need for natural language understanding is obviated by the structure of the discharge summaries.

Conclusion

We have developed a program for extracting the diseases and procedures from patient discharge summaries and coding them, complete with coded and unrecognized modifiers, using SNOMED-CT from the UMLS. The program uses a limited amount of natural language processing, only using a small (less than 200 word) dictionary to divide up disease statements into phrases for coding. With these short phrases, the program is able to quickly find the most specific codes available in SNOMED-CT for the statements in the summary.

The program has been developed and tested on 23 discharge summaries containing 250 phrases to be coded. The program does an effective job coding all but 10 of the phrases with 19 false positives.

We expect that the program will be an effective tool for providing a disease and procedure index for a large set of discharge summaries from the same source. From examination of discharge summaries from other hospitals, the technique should be transferable with possible changes to the code for finding the appropriate sections.

Acknowledgments

This work was supported by Grant Number R01 EB001659 from the National Institute of Biomedical Imaging and Bioengineering.

References

[1] Lussier YA, Shagina L, Friedman C. Automating SNOMED coding using medical language understanding: a feasibility study. Proc AMIA Symp. 2001;:418-22.

[2] Cao H, Chiang MF, Cimino JJ, Friedman C, Hripcsak G. Automatic summarization of patient discharge summaries to create problem lists using medical language processing. Medinfo. 2004;2004(CD):1540.

[3] Friedman C, Shagina L, Lussier Y, Hripcsak G. Automated encoding of clinical documents based on natural language processing. J Am Med Inform Assoc. 2004 Sep-Oct;11(5):392-402.

[4] Murff HJ, Forster AJ, Peterson JF, Fiskio JM, Heiman HL, Bates DW. Electronically screening discharge summaries for adverse medical events. J Am Med Inform Assoc. 2003 Jul-Aug;10(4):339-50.

[5] Lindberg, DAB, Humphreys BL, McCray AT. The Unified Medical Language System. *Meth Inf Med.* 1993;32: 281-91.

[6] Aronson AR. Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. Proc AMIA Symp. 2001;17-21.

[7] National Library of Medicine, Specialist lexical tools, http://specialist.nlm.nih.gov/LexTools.html

[8] McCray AT, Burgun A, Bodenreider O. Aggregating UMLS semantic types for reducing conceptual complexity. Medinfo. 2001;10(Pt 1):216-20.

[9] 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 Oct;34(5):301-10.

[10] 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 Nov-Dec;8(6):598-609.