A System for Automated Lexical Mapping

by
Jennifer Y. Sun

B.S. Electrical Engineering
Massachusetts Institute of Technology, 1994

M.D.
Dartmouth Medical School, 1998

Submitted to the Harvard-MIT Division of Health Sciences and Technology
in partial fulfillment of the requirements for the degree of

Master of Science in Medical Informatics
at the
Massachusetts Institute of Technology

May 2005 [June 2005]

Signature of Author:____
Division of Health Sciences and Technology
May 16, 2005

Certified by:____
Isaac S. Kohane, MD, PhD
Lawrence J. Henderson Associate Professor of Pediatrics and
Health Sciences and Technology, Harvard Medical School
Thesis Supervisor

Accepted by:____
Martha L. Gray, PhD
Edward Hood Taplin Professorship of Medical Engineering
Director, Medical Engineering and Medical Physics Program
Director, Harvard-MIT Division of Health Sciences and Technology

A System for Automated Lexical Mapping

by
Jennifer Y. Sun, MD

Submitted to the Harvard-MIT Division of Health Sciences and Technology
on May 16, 2005 in Partial Fulfillment of the
Requirements for the Degree of Master of Science in
Medical Informatics

ABSTRACT

Merging of clinical systems and medical databases, or aggregation of information from disparate databases, frequently requires a process where vocabularies are compared and similar concepts are mapped. Using a normalization phase followed by a novel alignment stage inspired by DNA sequence alignment methods, automated lexical mapping can map terms from various databases to standard vocabularies such as UMLS (Unified Medical Language System) and SNOMED (the Systematized Nomenclature of Medicine). This automated lexical mapping was evaluated using a real-world database of consultation letters from Children's Hospital Boston. The first phase involved extracting the reason for referral from the consultation letters. The reasons for referral were then mapped to SNOMED. The alignment algorithm was able to map 72% of equivalent concepts through lexical mapping alone. Lexical mapping can facilitate the integration of data from diverse sources and decrease the time and cost required for manual mapping and integration of clinical systems and medical databases.

Thesis Supervisor: Isaac Kohane MD, PhD
Title: Lawrence J. Henderson Associate Professor of Pediatrics and
Health Sciences and Technology, Harvard Medical School

I. Introduction

Access to medical information is hindered by the variation that is inherent in the lexicon of medical terminology. Electronic medical records are bringing about a major revolution in health care. The significant impact of electronic medical records will be at the cost of automated applications to manage the vast clinical information resources. Implementation of health-related data in electronic form is becoming more widespread, especially in the free-text format. This format requires extensive computational support to track content and for organization. As the medical field moves towards electronic health records, portability of patient information, and sharing of information across institutions, the need for a method to computationally normalize and map non-standard terms and concepts to a standard vocabulary becomes more important.

Information retrieval from free text, specifically consultation letters, is important for access for providers – a quick way for providers to determine the reason for consultation at a glance. Another important reason is for billing – allowing for extraction of a reason for visit can be matched to an ICD-9 code to allow for charge capture. Lastly, for research purposes, queries to a database can allow searches for distributions of patients being referred for a disease of interest.

Several algorithms have previously been proposed to automate translation between medical vocabularies including the use of frames, semantic definitions, digrams, and a combination of lexical, logical and morphological methods [1-7]. The UMLS, a product of the National Library of Medicine, exists to help in the development of systems to "understand" the language of biomedicine and health [8-9]. Tools developed include lexical variant generation, tools for customizing the Metathesaurus (MetamorphoSys), and extracting UMLS concepts from text (MetaMap) [10-11]. The LOINC database (Logical Observation Identifiers Names and Codes) also has a mapping program called RELMA, the Regenstrief LOINC Mapping Assistant developed by the Regenstrief Institute [12]. None of these systems, however, have been evaluated as a fully automated mapping system on production databases from multiple healthcare institutions.

The Lexical INtegrator of Concepts (LINC) system performs completely automated lexical mapping of medical vocabularies. The following sections will illustrate the stages involved in implementation of the LINC system and will discuss key issues and trade-offs in performance.

II. Background

IIA. Natural Language Processing in Medicine

There are many issues with natural language processing (NLP) in medicine [13]. First, clinicians have very diverse ways of describing data and the meaning of different terms may vary depending on the context. An example being a description of pneumonia – which can be expressed as "pneumonia cannot be excluded", "rule out pneumonia", "evidence of pneumonia" or "pneumonia in 1985". Second, the same concept may be expressed in different ways. An example is the term "myocardial infarction" which could also be "heart attack" or "MI". Or the same word may have different meanings in different contexts, such as "discharge from hospital" versus "discharge from wound". Lastly, there also may be ambiguous relations among words – such as "no acute infiltrate" – which could mean that there is no infiltrate or that there is an infiltrate, but it is not acute. To enable accurate access to electronic information, natural language processing systems must encode the information using a standard terminology from a well-defined vocabulary, be able to represent relationships between concepts in an unambiguous fashion, represent the context of these concepts, and have a way of representing vague concepts.

Natural language processing in medicine does not only involve extraction and encoding of medical records, but also voice recognition, computerized translation, question-answering systems, knowledge representation and acquisition, literature searching and indexing, and medical vocabularies.

IIB. Medical Terminology

Understanding any textual language involves three important components: syntax, semantics, and domain-knowledge. The syntax is the structure of sentences – subjects, verbs, objects, etc. The semantics are the meanings of the words and how they are combined to form the meaning of a sentence. Domain-knowledge is information about the subject matter – in this case medical terminology.

It is important to have a well-defined vocabulary that organizes the terminology into a hierarchy and is able to delineate well-defined relationships among the concepts. These relationships will help systems to make inferences and thus "understand" what certain terms mean. For example, if *pneumonia* were defined as a lung disease, a system would be able to infer that a patient has lung disease when it encountered the term *pneumonia*. The domain of medical terminology is considered a sub-language and thus has less variety, ambiguity, and complexity than the general language domain. A sub-language is a technical language, which has general words that have "normal" meanings, but also other general words that take a more restricted meaning in the context of the sub-language [14]. In addition, for the sub-language, there exists a large amount of specific vocabulary that is exclusive to its domain. Thus, it is possible to define specific categories suitable only to medicine and then find patterns amongst these categories and hopefully interpret them unambiguously. For example, a category in medicine is *body location* (like chest). There are also categories of *symptom* (like pain) and *severity* (like mild). So finding a pattern of severity + body location + symptom (like mild chest pain) allows the interpretation of a *symptom* "pain" associated with the *body location* "chest" and that the *symptom* "pain" is associated with *severity* "mild".

Obviously a standardized lexicon is at the foundation of all NLP systems. The National Library of Medicine has developed the Unified Medical Language System (UMLS) [8]. The purpose of the UMLS is to aid the development of systems that help health professionals and researchers retrieve and integrate electronic biomedical information from a variety of sources and to make it easy for users to link disparate information systems, including computer-based patient records, bibliographic databases, factual databases, and expert systems. In the UMLS Metathesaurus, each concept is given a unique identifier, and all synonymous concepts have the same identifier. This feature helps NLP systems to link words in text to a controlled vocabulary. The UMLS also contains a semantic network that assigns a category to all concepts. In addition, the SPECIALIST lexicon assigns syntactic categories to words and phrases in biomedical text. These modules are important for identifying relevant units of information, extraction, and indexing. As of January 2004, the SNOMED vocabulary is available through the UMLS Metathesaurus. SNOMED CT, the Systematized Nomenclature of Medicine – Clinical Terms, produced by the College of American Pathologists is a comprehensive clinical terminology of over 345,000 terms. It is the most comprehensive clinical vocabulary available in English (or any language), covering most aspects of clinical medicine.

IIC. Natural Language Processing Extraction Systems

Information extraction is the process of scanning text for information relevant to some interest, including extracting entities, relations, and most challenging, events [15]. Structured or coded data in electronic medical records is not widely available. Instead, the main source of clinical information is in unstructured-text reports and unconstrained data. Many natural language processing systems have been developed for the purpose of extraction of clinical medical information. These systems all include text parsers, some type of normalization method, and mapping of the individual words/terms found. Again, one of the core issues with these systems is the mapping of the normalized words to a standard vocabulary. Another issue is the development of specific lexicons for each individual system that are not necessarily generalizable.

A major issue in natural language processing is a lack of a truly comprehensive clinical vocabulary. The Large Scale Vocabulary Test [16-17] showed that a combination of existing terminologies can represent a majority of concepts needed to describe patient conditions in a range of health care settings and information systems. This finding is significant in that it can serve as a strategy for improving current vocabularies and hopefully in establishing a standard national vocabulary.

Sager et al developed the Linguistic String Project which is one of the first comprehensive natural language processing systems for general English and was later adapted to medical text [18]. It has been applied to discharge summaries, progress notes, and radiology reports. The system aims to allow providers to extract and summarize sign/symptom information, drug dosage and response data, to identify possible side effects of medications, and to highlight or flag data items. The processing chain is composed of levels that parse and tag different semantic parts of the sentence, mapping the words of the sentences into appropriate fields, and normalization, which recovers implicit knowledge and maps the parsed sentences into a relational database structure [19].

The MedLEE system (Medical Language Extraction and Encoding system) was developed as part of the clinical information system at Columbia by Friedman et al [20] for use in actual patient care and was shown to improve care. The system consists of processing phases, the first of which parses the text into a preliminary structured output using a semantic grammar parser and a lexicon. The next phases regularize and encode the forms into unique concepts by mapping to a controlled vocabulary.

The MENLAS system aimed to provide better access to information in patient discharge summaries using natural language processing and knowledge representation techniques [14]. Two applications were developed – a document-indexing system and a consulting application to provide users with access to the information in the documents via the indexing system.

The MedsynDikate system developed at Freiburg University is used for extracting information from pathology findings reports [21]. Their novel approach involved researching interdependencies between sentences while creating a conceptual graph of sentences parsed. The system aimed for "deep" text understanding, and better knowledge sources for grammar and domain knowledge. This system tries to overcome the shortcomings of sentence-centered NLP analysis by looking for reference relations between sentences [22]. The researchers also focused on alternative ways to support knowledge acquisition in order to foster scalability of the system. One method was to use automatic concept learning embedded in the text understanding process. Another method was to re-engineer the medical knowledge assembled in various other terminological repositories, such as the UMLS, and transforming their data structures into a description logics framework that could be used with the system.

IID. Natural Language Processing Mapping Systems

The development of tools to manage variation among the many medical vocabularies throughout clinical and research domains is a daunting but necessary task. As the medical field develops increasingly complex information systems to store and exchange electronic information, the task of reconciling variations in terminology becomes increasingly urgent. Many different approaches have been created to help solve this problem.

The University of California San Francisco UMLS group began the undertaking of intervocabulary mapping in 1988 using lexical matching. Sherertz et al [23] used filters and rules to transform input for mapping. They attempted to map disease names and disease attributes to MeSH terms with mapping in 47-48% of phrases in both cases.

The SPECIALIST lexicon, a UMLS knowledge source, is an English language lexicon that contains biomedical terms. Lexical records contain base form, spelling variants, acronyms and abbreviations. In addition, there is a database of neoclassical compound forms – single words that consist of several Greek or Latin morphemes, which are very common in medical terminology [24]. The lexicon entry for each word or term records the syntactic, morphological, and orthographic information. The Specialist system is composed of many modules. The lexical component encodes the information specific to the words in the language. The morphological component is concerned with the structure of words and the rules of word formation. The syntactic component treats the constituent structure of the phrases ad sentences. The semantic component is concerned with the meaning of words and sentences.
Language processing tools have been developed to mediate between existing vocabularies and the lexicon. These programs consist of modules for normalization and lexical variant generation, including lowercasing, removing punctuation, removing stop words, sorting words in a term alphabetically, generating inflectional variants, reducing words to their base forms, and generating derivational variants for words [25]. However, these tools do not automate the task of mapping existing vocabularies to the UMLS Metathesaurus.

The MetaMap Program, by Aronson et al [26-27] is a configurable program that maps biomedical text to concepts in the UMLS Metathesaurus. The algorithm uses variant generation using knowledge from the SPECIALIST lexicon then maps these variants to the Metathesaurus. Candidate mappings are evaluated based on a weighted scoring method consisting of variation and how much of the candidate mappings the text and in how many pieces.

Lau et al [28] propose a method for automated mapping of laboratory results to LOINC codes. Their method uses a series of rules to parse and map the laboratory tests to the LOINC descriptors and then to the LOINC codes. This method is very specific in its mapping process and would be difficult to generalize to other medical vocabulary domains.

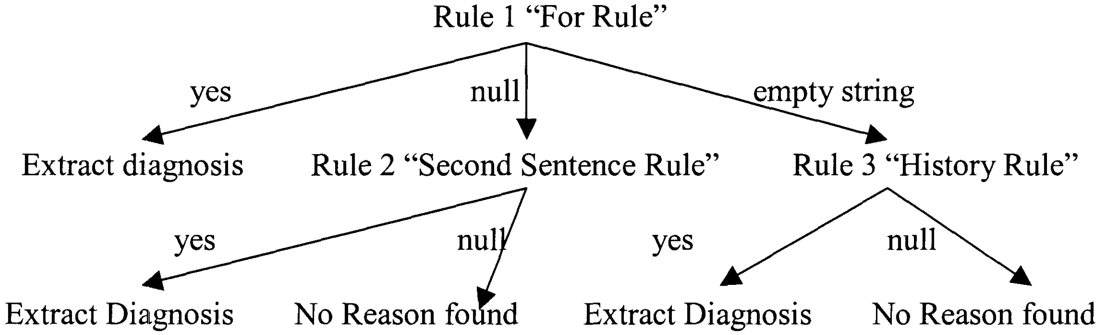
III. Methods

IIIA. Text Extraction

The Children's Workstation Database contains 1128 consultation letters in the doc_store SQL database. Each letter was extracted into a separate text file. Ten percent of the letters were examined – 120 files, to extract patterns with which to make rules to extract the "reason for referral" from these letters. On examination, three major patterns were found and rules were formulated from these patterns.

- Rule 1: In paragraph one (after the string "dear"), the reason for referral exists after the string "for".
- Rule 2: In paragraph one, if the string "for" is not found, the reason for referral exists after the string "he/she has".
- Rule 3: If neither rule one nor two match, in paragraph two (after the string "history"), the reason for referral exists after the string "for" or "with".

All extraction and rules were coded using Java 1.4.2_06. The rules were implemented in Java and results were written to a file for evaluation. The specific decision tree followed in the extraction process was as follows:



IIIB Lexical Mapping

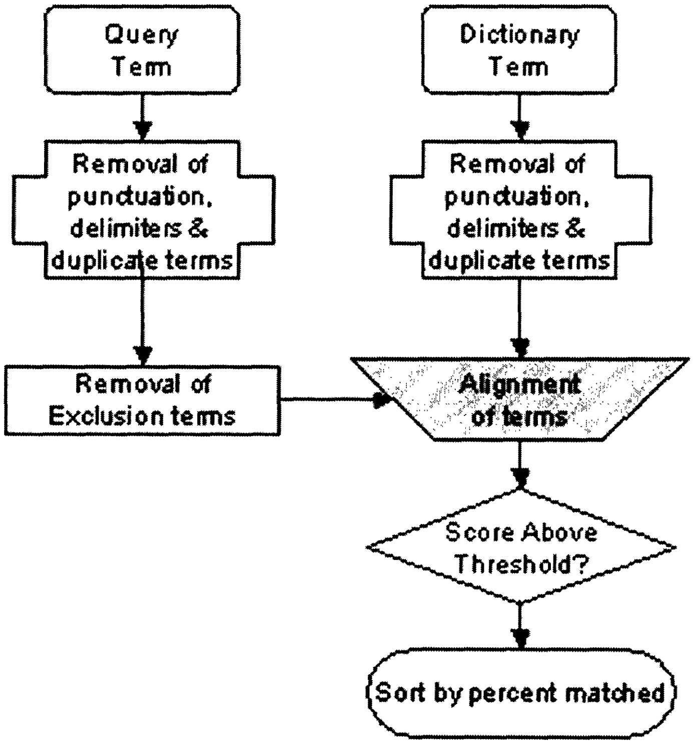
(1) Databases used

A database was created from the extracted reasons for referral from the consultation letters. This database, representing the *query terms*, was obtained from CWSSCRUBBED database from the Children's Hospital in Boston, Massachusetts. The dictionary source SNOMED (the Systematized Nomenclature of Medicine) has 863,277 unique terms. The SNOMED terms were subsetted from the UMLS Metathesaurus 2004AA using MetamorphoSys [30].

(2) Overview

A graphical flowchart of the processes used in LINC is summarized in Figure 1. The algorithm was modified from the original to suit this particular mapping problem. Details of the entire original algorthim are described in a paper pending publication [30]. The following sections present the individual steps within the overall method – pre-processing the query and dictionary terms, the alignment process, and post-alignment sorting of the candidate mappings.

Figure 1 – An overall flowchart detailing the steps of the alignment process.


Query Term
Dictionary Term
Removal of punctuation, delimiters & duplicate terms
Removal of punctuation, delimiters & duplicate terms
Removal of Exclusion terms
Alignment of terms
Score Above Threshold?
Sort by percent matched

(3) Pre-processing for normalization

To account for variations in the query vocabulary compared to the dictionary vocabulary, normalization of each term was necessary. The normalization process converting the term to lower case, splitting the term into tokens (its constituent words), removing involves: punctuation, and removing duplicate tokens. For the query vocabulary, the 1% most frequent terms are removed for the initial mapping (see details in Exclusion Terms section below). The normalized tokens are concatenated back into a string and passed to the alignment method and scored as a whole.

Two other methods for normalization were also tested, but not used in this particular study. The first method was sorting the tokens alphabetically, the second to sort tokens by frequency of the word (token) within its vocabulary. Both methods were employed to try to deal with variation in ordering between the different database vocabularies.

(4) Alignment

The inspiration for this algorithm comes from DNA sequence alignment algorithms such as BLAST (Basic Local Alignment Search Tool) [31-35]. LINC uses a matrix structure to find the best alignment between two strings – the query term and the dictionary term (see Figure 2). In Figure 2, the query term is on the y-axis, the dictionary term on the x-axis. The matrix is initialized with zeros, and then for every cell where a character of the query term matches a character of the dictionary term (not including spaces), the numeral "1" is placed as a marker in the cell. To fully explore all possible alignments, the algorithm iteratively performs depth first searches to find the "best" alignment. Using principles of dynamic programming, the larger problem of finding the optimal alignment is solved by finding the solution to a series of smaller problems, namely, finding the optimal alignment for substrings. The smaller problems do not require recalculation because their results are saved, thus improving computational efficiency.

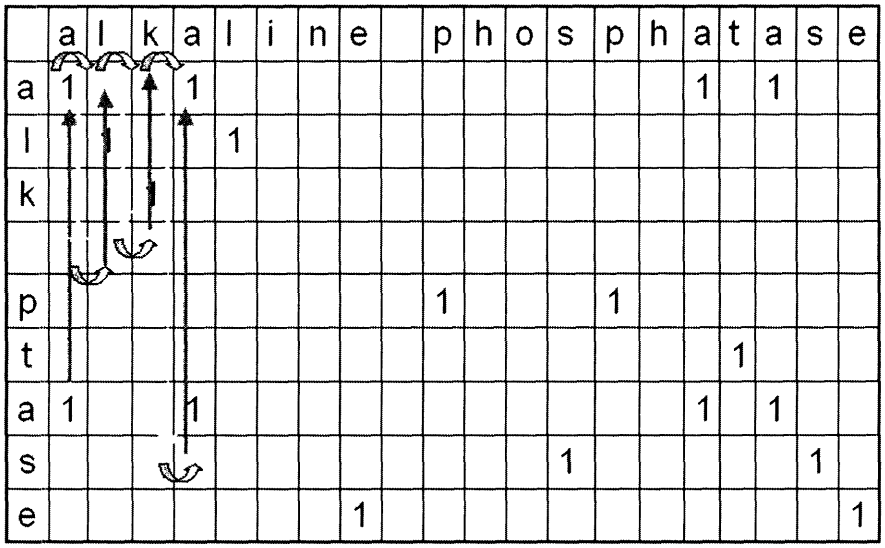
Figure 2 – Matrix representation of alignment. Query term is alk ptase (on the vertical) and dictionary term is alkaline phosphatase (on the horizontal). Highlighted cells represent positions from the highest scoring alignment where a character of the query term matches a character from the dictionary term within this alignment.

		0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*	*19*
		a	***l***	***k***	***a***	***l***	***i***	***n***	***e***		***p***	***h***	***o***	***s***	***p***	***h***	***a***	***t***	***a***	***s***	***e***
0	***a***	1			1												1		1		
1	***l***		1			1															
2	***k***			1																	
3																					
4	***p***										1				1						
5	***t***																	1			
6	***a***	1			1												1		1		
7	***s***													1						1	
8	***e***								1												1

The alignment proceeds as follows. Each matrix cell containing a "1" (a character match) is represented by a node. The nodes are then linked together into a chain starting at the lower left corner of the matrix, and proceeding to the right (see Figure 3 and 4). Each node has an optimal score associated with it, representing the highest possible score that can be attained from that node forward (see Scoring Algorithm section below). The algorithm tracks the scores for nodes that have already been traversed. Finding the node with the maximum score and following the path from that node through the matrix retrieves the best overall alignment. Each term in the query database is aligned to every term in the dictionary and only the top twenty highest-scoring mappings are kept for each query term.

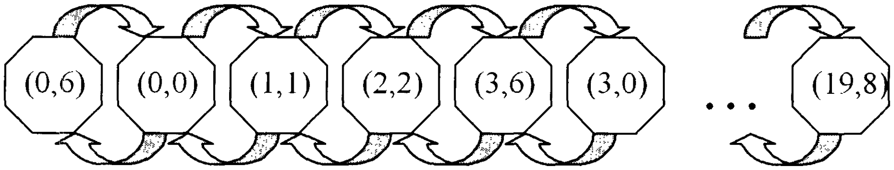
Initially, the alignment was done as a recursive "path-finding" through the matrix to truly do a depth first search. In successively trying each possible path of alignment, the algorithm would tally up the score each time and track the highest score. Because the system did not track computations that it had previously done, the system was quite inefficient, taking two minutes to map one query term to 11,033 terms in the UMLS. Solving the efficiency problem using dynamic programming and tracking scores that had previously been computed allowed the system to decrease the alignment time to two seconds to map one query term to the UMLS.

Figure 3 – A graphical representation of how the chain is linked together. The dark, solid arrows show the reading of the matrix cells from lower left and up the column. Each column is read from bottom to top, proceeding from the leftmost column to the right.



	a	l	k	a	l	i	n	e		p	h	o	s	p	h	a	t	a	s	e
a	1			1												1		1		
l		1			1															
k			1																	
p										1				1						
t																	1			
a	1			1												1		1		
s													1						1	
e								1												1

Figure 4 – Chain representation of the matrix. Coordinates represent the position of the cell in the matrix. The cells marked "1" are read from the matrix starting from the lower left corner, up the first column (column 0 in figure 2), then proceeding up the second column (column 1) and continuing through the table from left to right.



(5) Scoring algorithm

The scoring method is a combination of multiple factors that contribute to an optimal alignment. Each node (matched character) is given an initial score of 1. To penalize gaps between matching characters, the initial score is divided by the squared distance between the current node and the next mapped node (i.e. the gap).

Continuity between mapped characters is also tracked in order to benefit longer continuously matched node chains. A *proximity score* is calculated based on the Euclidian distance between any two nodes. If two matched characters are continuous within the original query term and dictionary term, then the proximity score equals 1. The *continuity score* is the sum of the proximity scores from all the nodes within the chain. But when two or more nodes have proximity scores equal to 1, the continuity score for that portion of the chain is squared prior to adding it to the overall continuity score. For example, a chain of 4 continuous nodes would have proximity scores of 1+1+1=3, and this would then be squared to add a continuity score of 9 to the overall score.

Overall, the scoring scheme is skewed towards greatly rewarding longer chains (greater continuity) of matched nodes.

(i) Formulas: Point1 (x_1,y_1) and Point2 (x_2,y_2)

Gap = $x_2 - x_1$

$$\text{Node Score} = \frac{1}{(x_2 - x_1)^2}$$

$$\text{Proximity Score} = \frac{\sqrt{2}}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}}$$

(ii) Specific examples of scoring

Example 1:

query term = **AST**
dictionary term = **AST**
Node Score (A) = 1
Node Score (S) = 1
Node Score (T) = 1
Proximity Scores = 2 *(A-S is continuous, and S-T is continuous)*
Continuity Score = 4 (2 squared)
Overall score = 1 + 1 + 1 + 2^2 = 7

Example 2:

query term = **AST**
dictionary term = **ALT**
Node Score (A) = 1
Node Score (S) = 0
Node Score (T) = 1
Proximity Score = 0 *(A-T is not continuous)*
Continuity Score = 0
Overall score = 1 + 0 + 1 + 0 = 2

(6) Exclusion Terms

In the pre-processing of the query vocabulary, the 1% most frequent terms in the vocabulary are removed. Because these terms are so prevalent within the vocabulary, they are often non-specific and contribute an excessive amount to the alignment mapping between terms. Examples of these terms are shown in Table 1. This exclusion method was employed due to the fact that the scoring is skewed towards giving a higher score to longer chains of matching characters, and also penalizing terms without this type of continuity.

A method not employed in this particular study is a second alignment, which adds back the excluded terms. The query term with the previously excluded words is then aligned to the candidate mappings from the initial alignment. In initial testing of the algorithm with laboratory databases, these excluded words were important clinically, but the word may be present in many different contexts and so does not allow differentiation of the context, i.e. vocabulary phrase, in which it is used. But with the second alignment, a higher score can then be assigned for the mappings, which are more specific because they contain these excluded terms.

Table 1 – Example of words excluded from the alignment

of	to	for	hers	on	with	rule	an
up	evaluation	and	his	years	ago	age	associated
follow	secondary	the	in	at	out	been	

(7) Match threshold

As a part of the scoring, a threshold was set to determine whether two terms were appropriate mappings based on the scoring algorithm described previously. The threshold is a percentage of a "perfect match", a perfect match being the case where the query term and the dictionary term are identical (normalized) character strings. Query terms that had no candidate mappings scored above the threshold are defined as "unmapped". Previous testing on other vocabularies [30] led to a setting of the threshold at 85% to optimize sensitivity and specificity of the mapping.

Another reason for setting a threshold was for evaluation purposes. If a match was deemed to be below threshold, it would then be possible to determine whether that term truly did not have a match (a true negative) or if the term did have a match but it was not found (false negative).

(8) Post-processing

Several different techniques were used to reorder the candidate mappings so that the "best" mapping would be sorted to the top of the list. Various sorting methods were tested in the original with the original algorithm [30]. These methods included:

- Sort by dictionary term length
- Sort by position – summing the x-coordinates of the matched characters within the dictionary term
- Sort by position 2 – summing the first and last x-coordinates of the matched characters of the dictionary term
- Sort by score and position – a new score which is a combination of the score and position of mapped terms –
    - score – the score of the current query term
    - max score – the highest scoring mapping for the current query term
    - position sum – the sum of all the x-coordinates for the mapped characters of the dictionary term
    - max position sum – the maximum position sum of all the dictionary terms that map to the current query term.
    - new score = $\left(\frac{score}{\max score}\right) * \left(\frac{\max positionsum}{positionsum}\right)$
- Sort by percent mapped – a new score which is a combination of the score and the percentage of the dictionary term mapped
    - new score = score * $\left(\frac{\# charactersmatched}{\# charactersofdictionaryterm}\right)$

In this mapping project, the results were sorted by percent mapped, since in previous tests this proved to be the most effective.

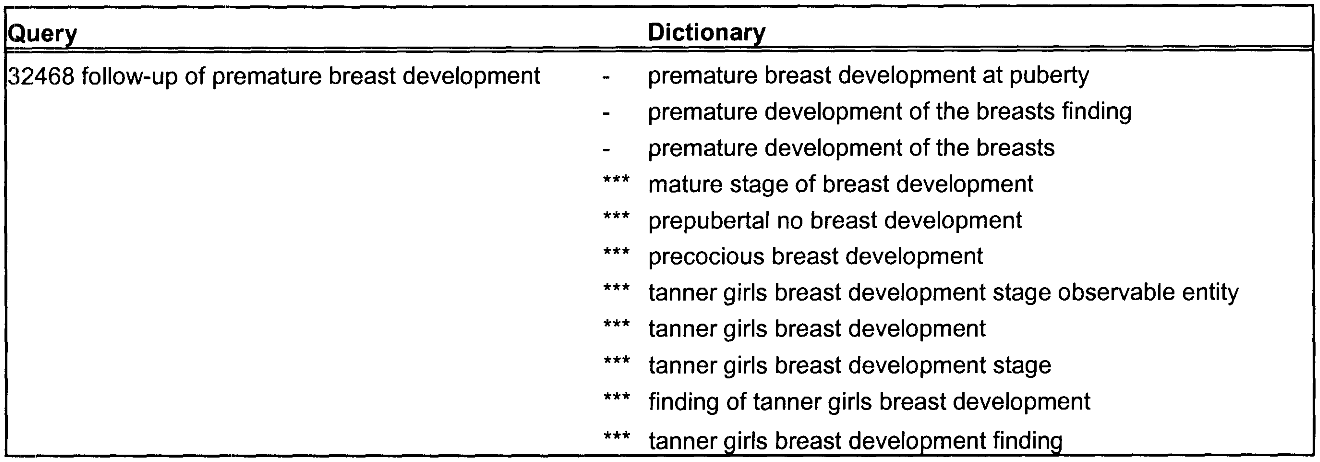
(9) Evaluation

The reasons for referral were evaluated against the actual letters from which they were extracted. A total is tallied of the true positives (the correct reason was extracted from the letter), false positives (extracted text is not the actual reason for referral or text is extracted when there is no reason for referral documented in the letter), false negatives (no reason is extracted when a reason for referral does exist in the letter), and true negatives (correctly does not extract any reason for referral).

For each LINC mapping run, a 2 by 2 truth table was then constructed to tally true positives (the algorithm found the correct mapping in the dictionary), false positives (the mapping found was incorrect, but scored above the threshold), false negative (the mapping scored below the threshold, but the term was present in the dictionary), and true negatives (the mapping was below the threshold and the term was not in the dictionary). The "gold standard" for the truth table was based upon manual evaluation of the mappings by the investigator.

A mapping was labeled true positive as long as an appropriate term from the dictionary scored above the threshold. In some cases, there were multiple mappings above the threshold, of which some would be correct mappings and some would be incorrect mappings – these were still labeled as true positives in that at least one correct mapping was identified. If the query term was ambiguous or undecipherable (as was the case with many abbreviated terms), those mappings were considered true negatives or false positives depending upon the mapping results, since there was no means by which to confirm a mapping. Additionally, for all terms where the system did not generate a mapping scoring above the threshold, the dictionary vocabulary was manually searched for the query term to determine if the mapping should be labeled as a false negative.

Figure 5 – An example of a true positive mapping produced by LINC. The column marked -/*** denotes whether the mapping was above threshold (-) , or below the threshold (***). The number before the query term is the document ID so the reason for referral could be checked against the true document. The ID number was not used in the alignment



Query		Dictionary
32468 follow-up of premature breast development	-	premature breast development at puberty
	-	premature development of the breasts finding
	-	premature development of the breasts
	***	mature stage of breast development
	***	prepubertal no breast development
	***	precocious breast development
	***	tanner girls breast development stage observable entity
	***	tanner girls breast development
	***	tanner girls breast development stage
	***	finding of tanner girls breast development
	***	tanner girls breast development finding

IV. Results

IVA. Text Extraction

As mentioned in the methodology, 10 percent of the documents were manually reviewed to identify patterns that could we used to develop rules to extract the information required. The actual reasons for referral were documented by manually reviewing all letters.

- 1075 letters had reason for referral explicitly mentioned
- 53 letters had no clear reason for referral mentioned

The extracted reasons for referral were then compared with the actual reasons. The following definitions were used for evaluation:

- TP – Extracted reason is the actual or real reason for referral
- FP – Extracts a reason which is not the actual reason or extracts a reason when there is no reason documented in the letter
- TN – Extracts no reason for referral when document does not contain reason for referral
- FN – Extracts no reason for referral when reason for referral is present in the document

TP = 1022 (90.6%)	FP = 46 (4.1%)
FN = 18 (1.6%)	TN = 42 (3.7%)

- **Percentage correctly extracted (TP+TN): 94.3%**
- Sensitivity = TP / (TP + FN) = 0.983
- Specificity = TN / (TN + FP) = 0.477
- Positive Predictive Value = TP / (TP + FP) = 0.96
- Negative Predictive Value = TN / (FN + TN) = 0.7

In addition, the incorrect mappings were examined to find that in 49 cases, none of the three rules fit to extract the reason for referral, but the reason could be found elsewhere in the letter. In only three cases was there a mistake in parsing due to punctuation.

IVB. Lexical Mapping

The mapping algorithm produced the top twenty matches that were the most lexically similar in mapping the extracted reason for referral to the SNOMED dictionary. The following definitions were used for evaluation:

- TP – an appropriate term from the dictionary and the mapping scored above the threshold
- FP – the mapping scored below the threshold, but the term was present in the dictionary
- TN – the mapping was below the threshold and the term was not in the dictionary
- FN – the mapping scored below the threshold, but the term was present in the dictionary.

TP = 657 (58.2%)	FP = 48 (4.3%)
FN = 273 (24.2%)	TN = 150 (13.3%)

- **Percentage correctly mapped (TP+TN): 71.5%**
- Sensitivity = TP / (TP + FN) = 0.706
- Specificity = TN / (TN + FP) = 0.756
- Positive Predictive Value = TP / (TP + FP) = 0.932
- Negative Predictive Value = TN / (FN + TN) = 0.355

There were 601 terms that included at least one excluded word. In 86 cases, the exact diagnosis/reason for referral was not found in SNOMED, but in 34 cases a synonym was found for the same query term. There were 25 terms that were misspelled, of which 22 were false negatives due to the misspelling. In addition, there were 27 terms that were abbreviated or contained an abbreviation, the lexical mapping still mapped 19 terms because the SNOMED dictionary also contains abbreviations. The lexical mapping was not designed to deal with multiple diagnoses, therefore in 174 cases, a false negative occurred because the reason for referral extracted contained more than one diagnosis term. Each term separately would have been found in SNOMED. One of the limitations of the lexical mapping algorithm is that every word within a term is given equal weight – whether it is a relevant word or extraneous information. In 96 cases, there were extraneous words within the terms, and in 70 of these cases, the extraneous information caused the algorithm to generate a false negative – because the score was not above the threshold due to these extraneous words.

V. Discussion

A comparison of the LINC system against the previously mentioned undertakings is highlighted by three key features. First, the system is flexible enough to allow mapping of different vocabularies – whether it is mapping a query vocabulary from a legacy system to a standardized vocabulary or a mapping between two disparate legacy vocabularies – which is not the case with any of the above-mentioned algorithms. The system is not dependent on the creation of specialized lexicons in order to perform the mappings. In addition, the mapping system is fully automated, requiring no pre-formatting, manual input or construction of rules or filters. Finally, the LINC system has been evaluated with multiple real-world vocabularies against both the UMLS, LOINC [30] and now with SNOMED.

In mapping query terms to dictionary terms, it was noted that the order of words within the terms differed from vocabulary to vocabulary. For example, the query term “type i glycogen storage disease” does not exist in that exact form in the SNOMED. Instead, SNOMED has the term “glycogen storage disease type i”.

To deal with this situation, two methods were employed to deal with these types of variation. The first method was alphabetizing the words within each term. Through the string processing method, the query term would then become "disease glycogen i storage type" and the SNOMED term would become "disease glycogen i storage type" – so then there would be a perfect match.

The second method was to break each term into tokens (words) and perform a separate alignment on each token, from which the scores were joined to obtain an overall score. In this method, the order of words would then not matter because as long as the token from the query term was found within the dictionary term, it would be mapped.

With the alphabetization method, although words will be ordered in a standard manner across vocabularies, the ordering may create gaps in the alignment due to words in the dictionary term that do not occur in the query term. An example – if the query term was "follow up of type i glycogen storage disease", then it would be reordered to "disease follow glycogen i of storage type up" – thus creating gaps in the perfect alignment.

Another issue that hindered lexical mapping was the existence of less "discriminating" or irrelevant words within the query term – such as "his", "hers", "follow up", and "rule out". These terms are clinically relevant, but not the critical tokens that differentiate a query term. The way LINC addressed this issue was to remove the most frequent terms in the query vocabulary (top 1%) from the query term for the initial alignment/mapping

LINC tackles the obstacle of abbreviation in both the query terms and the dictionary terms by running several phases of abbreviation expansion. Because of the limitations of the available abbreviation dictionaries, however, many appropriate expansions may be unavailable. Because SNOMED does contain abbreviations within its terms, this phase was not run as part of the alignment.

Another major consideration is how to choose the "optimal mapping". A single query term may map to multiple terms in the dictionary vocabulary, but in an automated system, we only want the "best" mappings. Two methods were used to try to extract the best mappings from the list of candidate mappings generated. The first method was to use a mapping threshold. Using this threshold, the mapping specificity was controlled – an exact mapping would have a threshold of 100%, meaning all words in the query term must appear in the dictionary term as exact string matches.

The second method to extract the optimal mapping was to sort the list of candidate mappings according to various scoring metrics. As described previously, several sorting methods were evaluated in prior testing. Initial sorting methods were based on the length of the dictionary term and the sum of positions of the character mappings of the dictionary terms – under the assumption that a shorter dictionary term without other extraneous tokens would be more relevant. The two algorithms to sort by position by summing the x-coordinates of the matched characters also addressed the issue of the length of the dictionary term. In this way, the dictionary term with the matching characters spread over a smaller "distance" in the dictionary term would be counted as the best match. The issue with these techniques was that the actual score of the alignment was not taken into account.

The sorting by score and position is a variation of scoring by position allowing a weighting of the score. The higher scoring terms would automatically have an advantage in the sorting, but the mappings would also be sorted based on the "shorter dictionary term" theory.

In the end, the best sorting method scored mappings by percent of characters in the dictionary term that matched. This method took into account both the score and the length of the dictionary term. In comparing two dictionary terms, if they have the same score, the shorter one would be sorted to the top of the mapping list.

VI. Limitations

A major limiting factor to the quality of mappings is the lack of a truly comprehensive, standardized medical vocabulary. Neither the UMLS Metathesaurus nor SNOMED covers the entire domain of clinical diagnoses that is available at this time.

While the system is automated in its generation of potential mappings for a query term, an expert/clinician still needs to needs to confirm the correct mapping from the list of generated candidates. Thus, there still exists a trade-off between the speed of automation and manual accuracy. Although manual confirmation of the lexical mapping is more time intensive than allowing the system to function in a totally automated fashion, this still represents an improvement in efficiency compared to systems that require manual search or collation of vocabularies.

VII. Conclusion

In the drive to expedite and improve the efficiency of information exchange, the mapping of local clinical terminology to standardized vocabularies will always be necessary to accommodate the legacy systems of individual institutions. LINC utilizes novel methods to automate the lexical mapping of terms between medical terminologies. The evaluation of LINC on a real world database and a "standardized" medical vocabulary illustrates the continuing obstacles that confront data mapping efforts and the performance of the system demonstrates promise to facilitate data exchange using automated lexical mapping.

References

[1] Rocha RA, Rocha BHSC, Huff SM. Automated translation between medical vocabularies using a frame-based interligua. Proc Annu Symp Compt Appl Med Care. 1993:690-4.

[2] Rocha RA, Huff SM. Using digrams to map controlled medical vocabularies. Proc Annu Symp Compt Appl Med Care. 1994:172-6.

[3] Dolin RH, Huff SM, Rocha RA, Spackman KA, Campbell KE. Evaluation of a "lexically assign, logically refine" strategy for semi-automated integration of overlapping terminologies. J Am Med Inform Assoc. Mar-Apr 1998; 5(2): 203-13.

[4] Elkin PL, Cimino JJ, Lowe HJ, Aronow DB, Payne TH, Pincetl PS, Barnett GO. Mapping to MeSH: the art of trapping MeSH equivalence from within narrative text. Proc 12th SCAMC. 1988: 185-90.

[5] Barrows RC, Cimino JJ, Claton PD. Mapping clinically useful terminology to a controlled medical vocabulary. Proc Annu Symp Comput Appl Med Care. 1994: 211-5.

[6] Hahn U, Honeck M, Piotrowski M, Schulz S. Subword segmentation – leveling out morphological variations for medical document retrieval. Proc AMIA Symp. 2001: 229-33.

[7] Cimino JJ, Barnett GO. Automated translation between medical terminologies using semantic definitions. MD Comput. 1990 Mar-Apr; 7(2): 104-9.

[8] McCray AT, Aronson AR, Browne AC, Rindflesch TC. UMLS knowledge for biomedical language processing. Bull Med Libr Assoc. 1993 Apr; 81(2): 184-94.

[9] Humphreys BL, Linfdberg DAB, Schoolman HM, Barnett GO. The Unified Medical Language System: An informatics research collaboration. J Am Med Inform Assoc. 1998 Jan-Feb; 5(1): 1-11.

[10] Divita G, Browne AC, Rindflesch TC. Evaluating lexical variant generation to improve information retrieval. Proc AMIA Symp. 1998: 775-9.

[11] Bodenreider O. The Unified Medical Language System (UMLS): integrating biomedical terminology. Nucleic Acids Res. 2004 Jan; 32 Database issue: D267-70.

[12] Mcdonald CJ, Huff SM, Suico JG et al. LOINC, a universal standard for identifying laboratory observations: a 5-year update. Clin Chem. 2003 Apr; 49(4): 624-33.

[13] Friedman CP, Hripcsak G. Natural language processing and its future in medicine. Academic Medicine,1999 August; 74:890-895.

[14] Spyns P. Natural language processing in medicine: An overview. Methods of Information in Medicine, 1996; 35: 285-301.

[15] Hobbs, JR. Information extraction from biomedical text. Journal of Biomedical Informatics, 2002; 35: 260-264.

[16] Humphreys BL, McCray AT, Cheh ML. Evaluating the Coverage of Controlled Health Data Terminologies: report on the Results of the NLM/AHCPR Large Scale Vocabulary Test. J Am Med Inform Assoc 1997 Nov-Dec; 4(6): 484-500.

[17] McCray AT, Cheh ML, Bangalore AK, Rafei K, Razi AM et al. Conducting the NLM/AHCPR Large Scale Vocabulary Test: A distributed Internet-based experiment. Proc AMIA Symp. 1997: 560-4.

[18] Sager N. Lyman M, Nhan NT, Tick LJ. Medical language processing: Applications to patient data representation and automatic encoding. Methods of Information in Medicine, 1995; 34(1): 140-146.

[19] Sager N. Lyman M, Bucknall C, Nhan N, Tick LJ. Natural language processing and the representation of clinical data. Journal of the American Medical Informatics Association, Mar/Apr 1994; 1(2): 142-160.

[20] Friedman CP, Alderson PO, Austin J, Cimino JJ, Johnson SB. A general natural language text processor for clinical radiology. Journal of the American Medical Informatics Association, 1994;1:161-74.

[21] Hahn U, Romacker M, Schulz S. MedsynDikate – A natural language system for the extraction of medical information from findings reports. International Journal of Medical Informatics, 2002; 67: 63-74.

[22] Hahn U , Romacker M, Schulz S. Discourse structures in medical reports-Watch out! The generation of referentially coherent and valid text knowledge bases in the MedsynDikate system. International Journal of Medical Informatics, 1999; 53:1-28.

[23] Sherertz DD, Tuttle MS, Blois MS, Erlbaum MS. Intervocabulary mapping within the UMLS: the role of lexical mapping. Proc Annu Symp Comput Appl Med Care. 1988: 201-6.

[24] McCray AT, Srinivasan S, Browne AC. Lexical methods for managing variation in biomedical terminologies. Proc Annu Symp Comput Appl Med Care. 1994: 235-9.

[25] McCray AT. The nature of lexical knowledge. Methods Inform Med 1998 Nov; 37: 353-60.

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

[27] Background information on MetaMap available at http://mmtx.nlm.nih.gov/ and http://skr.nlm.nih.gov.

[28] Lau LM, Johnson K, Monson K, Lam SH, Huff SM. A method for automated mapping of laboratory results to LOINC. Proc AMIA Symp. 2000: 472-6.

[29] More information about UMLS is available at http://umlsks.nlm.nih.gov

[30] Sun J, Sun Y. A System for Automated Lexical Mapping. In press. Please contact author for further information. jennifer.sun@alum.mit.edu

[31] Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. J Mol Biol. 1990 Oct 5; 215(3): 403-10.

[32] Altschul SF, Madden TL, Schaffer AA et al. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. nucleic Acids Res. 1997 Sep 1; 25(7): 3389-402.

[33] Thompson JD, Plewniak F, Poch O. A comprehensive comparison of multiple sequence alignment programs. Nucleic Acids Res. 1999 Jul 1; 27(13): 2682-90.

[34] Pretsemlidis A, Fondon JW 3rd. Having a BLAST with bioinformatics (and avoiding BLASTphemy). Genome Biol. 2001; 2(10): Reviews 2002. Epub 2001 Sep 27.

[35] Altschul Sf, Boguski MS, Gish W, Wootton JC. Issues in searching molecular sequence databases. Nat Genet. 1994 Feb; 6(2): 119-29. Review.