Appendix 4: Development of the Concept-Matching Algorithm in the Test Phase

Use of a Stop-Word List

The stop-word list was used for two purposes. First, it controlled the behavior of the phrase recognizer. Second, along with an implementation of the Porter stemmer, it was used to preprocess the MRCON table of the UMLS (which stores all terms for concepts). To this table, we added two extra columns that stored the *stemmed form* of each term, and the number of words in the stemmed form respectively. By stemmed form, we mean that the term is broken up into its individual words and stop-words are removed. The remaining words are converted to lower case, stemmed, and sorted alphabetically. Thus, high blood pressure gets transformed to blood high pressur , with three words. We soon discovered that the standard stop-word list needed considerable modification to serve its intended purpose.

1. Common words like first , second , third , low , high , left , right , early , open , great/er and less/er should not be stop-words. Otherwise, one cannot distinguish the greater trochanter of the femur from the lesser trochanter (in hip surgery), or between first- and second-degree heart block. Similarly, while single letters are placed on most stop-word lists, their complete elimination results in failure to recognize X ray , or the P, Q, R, S and T waves of the electrocardiogram.
2. We needed to add the acronyms NOS (not otherwise specified) and NEC (not elsewhere classified), as well as the words un/specified and un/classified , to the list. This is because while such qualifiers are important for ICD-9 based

disease classification (NEC is subtly different from NOS) they would almost never be encountered in medical narrative, and their presence in the stemmed form would interfere with concept matching, as discussed shortly.

3. Words like surgery and operation are stop words for the present domain.
4. It might be expected that certain other words are common enough in medicine to be considered stop words- e.g., disease/s and disorder/s . However, in the typically terse medical summaries that we encountered, such words were used judiciously and with relatively low frequency (as in post-traumatic stress disorder). Therefore such words should not be placed on stop word lists for processing of medical narrative.

After pre-processing the MRCON table, several terms yielded the same stemmed form., e.g., both Skin and Skin, NEC yielded skin . We wrote a batch query that extracted all such terms (and their stemmed forms) from MRCON. The first statement in the query identified entries with duplicated stemmed forms: the second statement extracted all information—concept ID, description, stemmed form— on the duplicated entries from MRCON to a separate table. The data in this table was used to reduce the number of concepts in our thesaurus subset by a semi-manual process, which we now describe.

Creation of a UMLS subset

To increase the effectiveness of matching medical narrative, we created a UMLS subset, eliminating several types of terms/concepts for one or more of the following reasons:

1. They were, from the indexing viewpoint, redundant equivalents of other terms. Thus, if an identical stemmed form matched multiple concepts, concepts that contained forms such as NOS , NEC , other , etc., were removed.
2. They were highly unlikely to be encountered in medical narrative, or unimportant with respect to clinical medicine. Thus, we eliminated semantic categories such as alga and Governmental or Regulatory Activity , and systematic chemical substance names e.g., 1,2-Dipalmitoylphosphatidyl-choline .
3. They were specific forms of more useful general concepts, and were unlikely to be matched successfully, e.g., pharmaceutical preparations that had the volume and/or dosage strength as part of the concepts name.
4. They were highly compound concepts based on lab tests. For example, UMLS entries for many concepts derived from LOINC (Logical Observation Identifiers, Names and Codes) (31) combine the substance being assayed with several qualifiers, e.g., TRYPSIN: CATALYTIC CONCENTRATION: POINT IN TIME: PLASMA: QUANTITATIVE . It is neither easy nor desirable to try to infer lab test codes and results from medical narrative; tables of laboratory test results are more appropriate for this task. LOINC-derived terms with embedded colons were therefore eliminated.
5. They were highly compound concepts that were considered unlikely to match phrases in text. An example is Supine and quarter turn to right - foot bed raised 14 inches (Read codes). Thus, after creating stemmed forms of terms, as described in the previous section, terms containing eight or more non-stop words in the stemmed form were eliminated.

6. They contained acronyms or abbreviations that would interfere with matching. Many concepts derived from the Read codes are preceded with an acronym, e.g., O/E Dorsalis Pedis — L and MPNST - Malignant peripheral nerve sheath tumour . The former is also a compound concept, and the simplest equivalent concept, e.g., Dorsalis Pedis artery , is more useful. Read Code-derived terms beginning with a capitalized word were therefore removed.
7. They were suppressable synonyms (abbreviated forms of other terms), e.g., sigmoid colon is a suppressable synonym for Malignant neoplasm of sigmoid colon . When the phrase sigmoid colon is encountered in narrative, malignancy is not necessarily implied, so such synonyms are misleading. (We thank Dr. Betsy Humphreys of the National Library of Medicine for valuable advice in this matter.)

Phrase Recognition Program

We used IBM s Feature Extraction Tool (FET) for phrase recognition. FET is part of Intelligent Miner for Text, a suite that includes text analysis tools, a relational database engine, and a Web crawler. While not a true part-of-speech tagger, FET s proprietary algorithm uses NLP heuristics along with an online database and a user-specified stop-word list to tag potentially interesting phrases. It identifies single and multiword phrases, names of places, people, and organizations, abbreviations; dates, money, and numbers. It can also distinguish between many homonyms based on context, e.g., can (verb) versus can (noun).

Because we have not performed comparative evaluations with rival products, we cannot comment on FET s features and performance versus other packages. One important reason for our using FET was its provision to us at no cost. (Commercial packages are not cheap: LinguistX, for example, costs $25,000 with academic pricing.) FET s major advantage, from our viewpoint, was that it saved us the considerable trouble of writing our own recognizer, and allowed us to focus on the task of concept matching.

FET runs like a UNIX filter , i.e., it takes its input from either a text file or the redirected text output of another program, and writes its output to a readily parsed text file. Its behavior is controlled with command-line parameters. FET processes more than 100 KB of text per second on a 233 MHz Pentium II with 128 MB RAM running Windows NT. While this is impressive, LinguistX is advertised as running at similar or greater speeds.

One drawback of FET is that its database s structure is undocumented, and therefore inaccessible to external programs. One cannot, for example, augment its contents by bulk-uploading subsets of the UMLS. Also, FET does not know about the medical domain: it appears to use simple rules, such as the presence of a sequence of words more than once in the document, to identify a phrase of potential interest. Thus, for example, if the phrase blood pressure occurs only once in a document, blood and pressure are flagged only as individually interesting words. FET is also vulnerable to variations of case and grammatical errors in the document, e.g., the phrase on the Morphine caused FET to categorize Morphine as possibly being a place name.

We post-processed FET s output, eliminating terms identified as places, people, and organizations, dates, money, and numbers. (Numeric data and patient names, while important, are better accessed from the structured part of the medical record, e.g., patient demographics, lab test and pharmacy tables.) Further, sequences of words identified by

FET that were not separated by punctuation or by stop words were concatenated into phrases, and these phrases were then used for concept identification.

To reduce the workload of manual checking, we shrunk the output of FET by eliminating multiple instances of the same concept match within the same note. Thus, if the concept post-traumatic stress disorder was matched three times in the note (which was quite likely if this was a primary diagnosis), we recorded only the first instance. In production concept-indexing, of course, one would want to record every instance of a match, because the frequency of occurrence of a concept in a note is important for relevance-ranking of that note with respect to a user query.

Database and Front-End

The database that houses a relational version of UMLS is an expanded version of a system previously built for Concept Locator (32), a UMLS Concept Searcher that uses complex Boolean query. In addition to functionality specific to the present study (described shortly), the MS-Access front end allows browsing of the UMLS in various ways. For example, we can inspect all details of a particular concept, e.g., its definition, associated terms, related concepts and semantic types. We can also locate concepts containing one or more words (optionally using wildcard symbols and Boolean operators); the authors used this feature to map phrases in the documents to UMLS concepts manually in cases where the automatic match had problems. Certain tables that are not part of the UMLS distribution have been added, e.g., a list of unique words occurring in the UMLS, and the number of concepts containing each word.

One of the forms, illustrated in fig. 1, provides a workbench , with the color-coded note in the top half, and the concepts recognized in it shown in the bottom half. Various buttons let one locate one or more instances of a phrase in text, search for UMLS concepts using a phrase that has been selected with the mouse, and move between a row in the output and the location of the corresponding phrase in the text.

As reproduced, fig. 1 is gray-scale, and therefore it is more important to appreciate the rationale for color-coding than to be concerned with the colors themselves. As stated before, we use FET to help identify phrases (noun phrases or otherwise) that can be mapped to one or more UMLS concepts. Because FET is not totally accurate, we must be able to easily ascertain visually where it works correctly and where it does not. Therefore we color-code the text. *Multi-word phrases* identified as such by FET are marked in red, *single-word* terms in orange, single-word terms that have been *concatenated into phrases* (by our code) in green, and stop words or uninteresting text in the default black. The coding scheme is now explained.

As stated earlier, FET performs limited recognition of multi-word phrases. In the text of figure 1, the phrases chronic deep venous thrombosis , dry gangrene , vital signs and cerebrovascular accident are correctly flagged, but the sequence of words lower extremity wrapped (with ACE wrap) has been incorrectly flagged as a single phrase.

In most cases, rather than recognizing a sequence of words as a phrase, FET will return each word in the sequence separately. In such a case, if the words are not separated by intervening punctuation or stop words, they can be programmatically concatenated into a single phrase. Examples of such word sequences visible in the text are Potassium chloride , pitting edema , partial thromboplastin time and right shoulder surgery ; however, old black male transferred represents an error with this approach.

In cases where interesting single-word phrases have been identified by FET, and these are separated from adjacent phrases by punctuation or stop words, such phrases are flagged in orange. Examples are, Embolus , Vancomycin , urokinase and afebrile . Note that both examples of erroneous phrase detection are due to FET s failure to explicitly tag verbs (wrapped , transferred), whose appearance in the text should signal the end of the preceding noun phrase.

Concept Matching against the UMLS

Each candidate phrase was used to search the UMLS. The steps of the matching algorithm, summarized in the flowchart of figure 2, were as follows:

1. From the phrase, we removed the occasional stop-words that had not been eliminated by FET, and words that were absent in the UMLS; the latter would never match. (To readily determine which words were absent in the UMLS, we preprocessed UMLS s MRXNW.ENG table to create a smaller table, MRXW_CON, with only unique combinations of concept ID and word columns being preserved. We then processed MRXW_CON to create a new table, MRWORDS, containing one record for each unique word in the UMLS. Checking for the existence of a particular word in this table was then reduced to a rapid table lookup operation.)

   If the remainder of the phrase contained more than five words, it was not matched, and was flagged as algorithm failure (phrase too long). We discuss the consequences of choosing the cut-off of five non-stop words later.
2. We first tried to find concepts whose terms contained all words in the phrase, using the MRXW_CON concordance table of the UMLS, which cross-references words with the concepts containing them. We did this by performing set intersections as

described in the Computational Complexity section. This step is insensitive to word order: thus non insulin dependent diabetes mellitus matches diabetes mellitus, non- insulin dependent .

3. If the entire phrase was successfully and uniquely matched, we moved to the next phrase. If matching failed, and the phrase contained more than one word, we then attempted to find complete matches to subsets of the phrase, as follows.
    1. If the phrase contained N words, we generated all combinations of N-1 words, and looked for complete matches. If no match was found, we then generated all combinations of N-2 words, looking for matches again, and so on. In the worst-case scenario (no matches), we had to match individual words to concepts. (These would always match, since we had eliminated words not in the UMLS in step 1.)
    2. If at any point a subset of the words matched, then we selected the words that were not yet matched, and returned to step 2 using these words.

Thus, for example, the five-word phrase post-traumatic stress disorder symptoms eventually matched stress disorder, post-traumatic and symptoms , while deep venous thrombosis ultrasound matches deep , venous thrombosis NEC and ultrasonography **.**

A potential flaw in this method is that the order in which the combinations are generated can influence matching. For example, the noun phrase spleen rupture with normal stomach may yield the false match stomach rupture . We did not see such a problem in the output of our program, possibly because FET almost always splits phrases on stop-words (e.g., with), supplying two separate phrases to our program, e.g., stomach rupture and normal spleen . The vast majority of the phrases matched by our code did not exceed three words.

4. If in steps 2 and 3, the algorithm needed to match concepts for a phrase comprising a single word, we checked if this word was on UMLS s ambiguous strings list. If so, it was not processed further, and was flagged as such. Note that such words should not be rejected right from the beginning. Thus, while the word anesthesia is ambiguous, the phrase endotracheal anesthesia is not.

5. If in steps 2 and 3, multiple concepts matched a phrase or a sub-phrase, we attempted disambiguation. We did this by generating the stemmed form of the (sub) phrase, and comparing the stemmed form with the stemmed forms for terms corresponding to each candidate concept. (As previously stated, stemmed forms of all terms in the MRCON table were pre-computed and stored in an additional column.) If an exact match of stems was uniquely found, we considered the corresponding concept to be uniquely matched. For example, using the MRXW_CON table alone, the phrase occupational therapy in the text matches the four concepts occupational therapy , occupational social therapy, NOS , occupational therapy evaluation and encounter for occupational therapy . However, only the stemmed form of the first concept would be an exact match to the stemmed form of the phrase.

   There were two failure conditions that could arise with this approach.

   a. Sometimes, even after generating the stem, more than one concept matched the stemmed phrase. We refer to this as the *non-unique stem problem*. It was the discovery of this problem that led us to explore systematic ways to eliminate concepts with NEC , NOS , unspecified and the like, if these are redundant with the equivalent unqualified concepts. Thus, an example of failure is paranoid schizophrenia , where the matched concepts schizophrenia, paranoid and paranoid schizophrenia, unspecified have the identical stemmed form, paran schizophreni . (The stop word unspecified is removed during creation of the stemmed form.).

If the phrase in the text consisted of a single word, it could often be disambiguated and matched uniquely to a concept by doing a case-insensitive comparison of the phrase to the original (un-stemmed) terms in the MRCON table. This approach, however, was not reliable for multi-word phrases, because of vulnerability to word order within the phrase.

b. Sometimes, no stemmed phrase matched exactly. This was typically due to the missing general concept situation described in the section composite vs. general concepts .

Steps 2 through 4 do not involve string comparisons of the entire phrase with terms for candidate concepts. Instead, we look up concept IDs from the MRXW_CON and Ambiguous_Strings tables, using individual words in the phrase. Only step 5 (disambiguation using stems or entire phrases) involves actual string comparisons.