

Original Investigations

JAMIA

Research Paper ■

Use of General-purpose Negation Detection to Augment Concept Indexing of Medical Documents:

A Quantitative Study Using the UMLS

PRADEEP G. MUTALIK, MD, ANIRUDDHA DESHPANDE, MD,
PRAKASH M. NADKARNI, MD

Abstract Objectives: To test the hypothesis that most instances of negated concepts in dictated medical documents can be detected by a strategy that relies on tools developed for the parsing of formal (computer) languages—specifically, a lexical scanner (“lexer”) that uses regular expressions to generate a finite state machine, and a parser that relies on a restricted subset of context-free grammars, known as LALR(1) grammars.

Methods: A diverse training set of 40 medical documents from a variety of specialties was manually inspected and used to develop a program (Negfinder) that contained rules to recognize a large set of negated patterns occurring in the text. Negfinder’s lexer and parser were developed using tools normally used to generate programming language compilers. The input to Negfinder consisted of medical narrative that was preprocessed to recognize UMLS concepts: the text of a recognized concept had been replaced with a coded representation that included its UMLS concept ID. The program generated an index with one entry per instance of a concept in the document, where the presence or absence of negation of that concept was recorded. This information was used to mark up the text of each document by color-coding it to make it easier to inspect. The parser was then evaluated in two ways: 1) a test set of 60 documents (30 discharge summaries, 30 surgical notes) marked-up by Negfinder was inspected visually to quantify false-positive and false-negative results; and 2) a different test set of 10 documents was independently examined for negatives by a human observer and by Negfinder, and the results were compared.

Results: In the first evaluation using marked-up documents, 8,358 instances of UMLS concepts were detected in the 60 documents, of which 544 were negations detected by the program and verified by human observation (true-positive results, or TPs). Thirteen instances were wrongly flagged as negated (false-positive results, or FPs), and the program missed 27 instances of negation (false-negative results, or FNs), yielding a sensitivity of 95.3 percent and a specificity of 97.7 percent. In the second evaluation using independent negation detection, 1,869 concepts were detected in 10 documents, with 135 TPs, 12 FPs, and 6 FNs, yielding a sensitivity of 95.7 percent and a specificity of 91.8 percent. One of the words “no,” “denies/denied,” “not,” or “without” was present in 92.5 percent of all negations.

Conclusions: Negation of most concepts in medical narrative can be reliably detected by a simple strategy. The reliability of detection depends on several factors, the most important being the accuracy of concept matching.

Medical narrative, consisting of dictated free-text documents such as discharge summaries or surgery notes, is an integral part of the clinical patient record. Databases containing free-text medical narrative often need to be searched to find relevant information for clinical and research purposes. Researchers in the field of information retrieval have devised general methods for processing free-text documents so that relevant documents can be returned in response to queries.^{1,2} One aspect of processing is the indexing of text to improve the effectiveness and speed of its subsequent search. Most commercial, general-purpose information retrieval engines typically index the words in documents; few of them make use of synonym information that inter-relates words or phrases.

To eliminate the need for users to specify synonymous forms of keywords manually when searching a database of documents, and to thereby increase the sensitivity of the search, documents pertaining to a specific domain may also be concept indexed. Here, phrases in the document are identified and matched to concepts in a domain-specific thesaurus. Automated concept indexing based on the National Library of Medicine's Unified Medical Language System (UMLS) Metathesaurus³ or its MeSH (Medical Subject Headings) subset has been explored by several researchers.⁴⁻⁸

An important aspect of information-retrieval-based search is the ranking of matching documents by relevance,^{9,10} giving more weight to documents containing the specified keywords many times and to documents that contain keywords that are rarer in the collection as a whole. For a medical document, however, the presence of a concept does not necessarily make the document relevant for that concept. The concept may refer to a finding that was looked for but found to be absent or that occurred in the remote past.

Negation is a fundamental operation that can invert the "sense" of a sentence or document. In query of large medical free-text databases, the presence of negations can yield numerous false-positive matches,

because medical personnel are trained to include pertinent negatives in their reports. Thus, in a search for "fracture" in a radiology reports database, 95 to 99 percent of the returned reports would state "no evidence of fracture," or words to that effect. Therefore, to increase the utility of concept indexing of medical documents, it is necessary to record whether the concepts have been negated or not.

Negation in natural language can be extremely subtle, and insertion or omission of a single word can completely change the scope and force of a negation and even reverse its polarity.^{11,12} In medical narrative, however, we expect negations to be much more direct and straightforward, since clinicians are trained to convey the salient features of a case concisely and unambiguously, as the cost of miscommunication can be very high. We therefore hypothesized that negations in dictated medical narrative are unlikely to cross sentence boundaries and are also likely to be simple in structure. Simple syntactic methods to identify negations might therefore be reasonably successful, as they are in computer languages. In particular, most negations in medical documents might be identified, providing the documents are suitably pre-processed with the aid of a lexical scanner that carries out finite state processing and a parser that uses a context-free grammar, using tools of the type used for constructing computer language compilers.

We describe the design of a program for negation recognition (Negfinder) based on existing technology for implementing parsers based on context-free grammars. Such technology has been used successfully for processing both natural language¹³ and formal (computer programming) languages.¹⁴ Negfinder can be used in production mode as part of a pipeline of programs used to create a concept index for a collection of documents. Its output consists of information necessary to index the occurrence of a concept in the document, as described later.

Background

Syntactic Parsing of Natural and Formal Languages

Syntactic parsing mechanisms have been designed extensively for two kinds of grammars—context-free grammars (CFGs) and the more constrained regular or finite-state grammars (FSGs). Finite-state grammars can handle regular expressions, but they cannot handle nested structures of arbitrary depth, whereas CFGs can. This is important, because nested structures are common both in natural languages ("This is the cat that

Affiliation of the authors: Yale University School of Medicine, New Haven, Connecticut.

This work was supported by National Institutes of Health grant R01 LM06843-01 from the National Library of Medicine.

Correspondence and reprints: Pradeep G. Mutalik, MD, Department of Diagnostic Radiology, Yale University School of Medicine, 333 Cedar Street, New Haven, CT 06510; e-mail: <Pradeep.Mutalik@yale.edu>.

Received for publication: 12/22/00; accepted for publication: 5/15/01.

killed the rat that ate the malt that...”) and in formal (computer) languages (“if A then if B then if C then...”). On the other hand, since FSGs are more constrained than CFGs, it is computationally easier to implement efficient parsers for them.

Context-free grammars have been the grammars of choice in both natural language and formal languages, but often the need for computational tractability and efficiency have led investigators to use a restricted subset of CFGs¹⁵ or FSGs¹⁶ in constrained domains. For instance, the compiling and parsing of computer languages has been streamlined by the use of a restricted subset of CFGs called LR grammars, for which powerful software tools have been developed. In particular, the well-known utilities *lex* and *yacc*,¹⁷ originally developed for UNIX at AT&T Bell Laboratories, generate highly efficient parsers for a subclass of LR grammars called LALR(1) grammars, and their widespread use is one reason that almost every computer programming language in use today conforms to an LALR(1) grammar. LALR(1)—for Look-Ahead, Left-Recursive (1)—refers to a grammar that uses a look-ahead of one token to match input to a parsing rule, and where recursive constructs are specified left-most in a rule. (A token may map to a single word [lexeme] or to a lengthy phrase.)

Lex generates lexical scanners or analyzers (“lexers”) that work cooperatively with parsers that are generated by *yacc*. The programmer provides input to *lex* through a specification based on regular expressions,¹⁸ while *yacc* input is specified through a set of parsing rules written using a modification of Backus-Naur form, which is commonly used to specify a CFG. *Lex* generates a deterministic finite state machine or automaton (DFA) similar to that generated by parsers of FSGs. The DFA is a program that classifies strings in the input data into various lexical classes in a very time-efficient (though not necessarily space-efficient) manner. During parsing, the lexer sends the parser a token at a time based on patterns that it recognizes: several contiguous lexemes can be combined into a single token. The parser uses information about the token just received to activate an appropriate parsing rule, and the entire input is eventually transformed into a unique parse tree after several parsing rules have been applied in succession.

Negation and Semantic Concerns in Natural Language Processing

A number of complexities in the nature of natural language, such as ambiguous syntax, challenge any treatment of negation. Negation itself, in natural lan-

guages, is far more subtle and complex in force and scope than it is in formal logic and computer languages, where **not p** is simply the polar opposite of **p**. This has been extensively documented with numerous examples by Horn¹¹ and by Horn and Kato.¹² Consider the following example based on Horn¹⁹:

1. I’m not tired.
2. I’m not a bit tired. (= I am not at all tired.)
3. I’m not a little tired. (= I am quite tired.)

In sentence 1, the negative operator “not” scopes over “tired.” In sentence 2, the scope of “not” is still over “tired,” but the intercalated phrase “a bit” has increased the force of the negative. In sentence 3, the intercalated “a little,” a synonym of “a bit” in an affirmative sentence, redirects the scope of “not” to itself and away from “tired,” which is affirmed instead of negated, with intermediate force.

If the subtle negations and ambiguities that Horn and others have documented were commonplace in medical narrative, full-blown natural language understanding would be needed to detect them. However, as stated earlier, we expect that most negations in medical narrative would be straightforward in type, scoping over words only in their immediate vicinity, and therefore identifiable without extensive syntactic analysis of sentence structure.

Related Work in Medical Information Retrieval

A general-purpose approach to handling negations in queries of information retrieval databases was described by McQuire and Eastman.²⁰ (This problem is simpler than negations in free text, because queries are limited to relatively simple sentence structures that serve the purpose of interrogation.) These authors created a query front-end where, if ambiguity was detected, the user was asked to clarify which constituents of the query they intended to be negated. Their work indicates that it is not possible for a system to automatically disambiguate all uses of negation in queries.

A limited amount of previous work has addressed negation in medical narrative in specific contexts.^{21,22} NegExpander, by Aronow et al.,²¹ identifies a small number of negation variants to classify radiology (mammography) reports; in our pilot studies, these were seen to have the simplest negations. NegExpander does not do any syntactic parsing beyond detection of a small set of negative words and conjunctions, but it does expand conjunctive phrases and makes explicit the negation of each component concept. This is a critical function for a general-purpose negation detector, as is shown later.

MedLEE,²² by Friedman et al., does sophisticated concept extraction in the radiology domain and has been recently extended to handle discharge summaries. MedLEE has been ported to the Web²³ and has recently been augmented by representing the extracted information in XML (Extended Markup Language).²⁴ The latest version of MedLEE combines a syntactic parser with a semantic model of the domain. MedLEE recognizes instances where negatives are followed by words or phrases that represent specific semantic classes, such as degree of certainty, temporal change, or a clinical finding. It also identifies patterns where only the following verb is negated and not a semantic class (e.g., "X is not increased"). For a general-purpose negation detector, the coverage of both these types of negations needs to be exhaustive enough to have a high rate of success in diverse medical narrative.

To develop an open-ended semantic model for all medical narrative is an immense task. In its absence, negation of medical concepts in text can be determined if the concepts themselves are identified in a pre-processing step. Previous work by Nadkarni et al.²⁵ explored the feasibility of using the UMLS for this purpose; the concept-recognition algorithm described by them is used in the present work, with some augmentations. The algorithm correctly identified 76 percent of concepts (true-positive results) in a test set of documents; the error rate of 24 percent, however, was considered too high for concept indexing to be the only production-mode means of pre-processing medical narrative.

Among the causes of problems in matching were redundant concepts in the UMLS, homonyms, acronyms, abbreviations and elisions, concepts that were missing from the UMLS, proper names, and spelling errors. Some of these problems are discussed in more detail below, as they can significantly affect the detection of negations.

Composite Concepts and Homonyms and Their Effects on Negation

The complexities of natural language semantics are such that even if negation signals in medical narrative do turn out to be simple as we have hypothesized, problems with the accurate recognition of concepts may make it difficult to determine what exactly is being negated. Two specific problems in concept matching that can confound negation detection are *composite concepts* and *homonyms*.

As medical knowledge has evolved, simpler concepts have been combined to form higher-level concepts to

refer to them concisely. An example is elevation of blood pressure due to kidney disease, which is called nephrogenic hypertension. Higher-level concepts in common usage may enter the UMLS eventually. However, given N individual concepts, these can potentially be combined into N^2 pairs and N^3 triplets. Many of these combinations may be meaningless, whereas others may be perfectly valid composite concepts created on the fly (e.g., "the distal articular surface of the second left metatarsal"). One would expect that the majority of such meaningful combinations will not be recorded as distinct entities in the UMLS.

We define a composite concept as a phrase that does not match completely to a single UMLS concept but whose parts do. An example is *digitalis-induced atrial fibrillation*, which does not exist in the UMLS; parts of the phrase, however, separately match the UMLS concepts *digitalis* and *atrial fibrillation*. A practical problem with composite concepts is that if they are negated, it does not necessarily mean that the individual concepts that the composite comprises have been negated. For example, if digitalis-induced atrial fibrillation was ruled out, it might be that atrial fibrillation itself was present, but that digitalis was ruled out as a cause.

Homonyms are terms that may have multiple meanings. For example, "anesthesia" may refer to a procedure ancillary to surgery or to a clinical finding of loss of sensation. The UMLS records 13,000 ambiguous terms, but this list is by no means complete. For example, "supine," which is not in the list, could, if encountered in text, imply either of the UMLS concepts "supine position" or "supine decubitus." (The latter is a subset of the former, because "decubitus" implies position of the torso.)

If a phrase that is an unrecorded homonym is encountered in text, the concept matching process fails. All we discover is that "supine" is a word that is part of several terms for concepts; there is no simple way to differentiate it from words of no medical importance. This creates a problem when a homonym is negated. Although a negation detection program may correctly identify that the homonym lies within the scope of the negation, its index of negated concepts will not record this instance, because no unique UMLS concept ID can be assigned to the homonym.

Methods

Our program, Negfinder, consists of several components that operate in pipeline fashion. That is, each step uses the output of the preceding step as its input. The steps are as follows:

*DIAGNOSES: Pneumonia. The patient has a history of peptic ulcer disease with a Billroth I operation as well as vomiting... He denies shortness of breath, chest pain, fever, chills, nausea, vomiting, diarrhea, or abdominal pain. The patient had no other complaints... *PHYSICAL EXAMINATION... JVP estimated at 6cm. 2+ carotid pulsations bilaterally. No bruits... *CARDIAC: Irregularly irregular, with normal S1, split S2, with no murmurs, rubs, or gallops... He was without complaint and had no evidence of cough or sputum production. ...Sputum and blood cultures were negative for evidence of infection.

*DIAGNOSES: ~32285:17:9. The patient has a ~48604:340:7 of ~30920:351:20 with a ~192440:379:20 as well as ~42963:411:19... He denies shortness of ~225386:703:6, ~8031:711:10, ~15967:723:5, ~85593:730:6, ~27497:738:6, ~42963:746:8, ~11991:756:8, or ~737:769:14. The patient had no other ~231216:846:10... *PHYSICAL EXAMINATION... ~332254:1683:6. ~750572:1692:13 at 6cm. 2+ ~741982:1718:18 bilaterally. No ~6318:1651:6. ...*~205041:1812:7 Irregularly ~205271:1833:9, with normal S1, split S2, with no murmurs, rubs, or gallops... He was without ~231216:2489:9 and had no evidence of ~10200:2522:5 or ~242104:2531:17... ~38056:2869:6 and ~T213890:2880:14 were negative for evidence of ~21311:2925:9.

*DIAGNOSES: **Pneumonia**. The patient has a history of **peptic ulcer disease** with a **Billroth I operation** as well as **vomiting**... He *denies shortness of breath, chest pain, fever, chills, nausea, vomiting, diarrhea, or abdominal pain*. The patient had *no other complaints*. *PHYSICAL EXAMINATION... **JVP estimated** at 6cm. 2+ **carotid pulsations** bilaterally. *No bruits*. *CARDIAC: Irregularly **irregular**, with normal S1, split S2, with *no murmurs, rubs, or gallops*... He was *without complaint* and had *no evidence of cough or sputum production*... **Sputum and blood cultures** were *negative for evidence of infection*.

Figure 1 Excerpts from a discharge summary at various stages of the Negfinder pipeline. *Top*, Original document. *Middle*, Document transformed by coding of recognized concepts from UMLS 2000. Concepts are indicated by ~#:#:~ where the three numbers indicate the UMLS concept ID, the byte offset in the text, and the length of the phrase. Thus, "pneumonia" is replaced by ~32285:17:9. The only words that remain are stop words and phrases or standard headings; unrecorded homonyms (see discussion in text) such as "rubs," "S1," and "S2"; and unrecorded variants of standard terms such as "gallop," which is a variant of the UMLS preferred form "gallop rhythm." *Bottom*, Negfinder mark-up simulated in monochrome. Negating phrases are marked in italics, identified concepts in bold; of these, negated concepts are also italicized.

- A *concept-finding step*, described in a previous paper,²⁵ identifies UMLS concepts in a document. Each row of the output of this step consists of the phrase in the document that was matched, its offset in the original document and the length of the phrase that it replaced, and the concept that was matched. (The phrase is recorded for the purpose of human verification.) For composite concepts, there will be multiple entries for that phrase, with each concept recorded in a separate row.
 - An *input transformation step* combines the text of the document and the output of the concept-finding routine to replace every instance of a concept or compound concept in the original document with a coded representation that includes its UMLS concept ID (shown in the middle panel of Figure 1).
 - A *lexing/parsing step*, carried out by a lexical scanner that uses regular expressions and a parser that uses an LALR(1) grammar, processes the transformed document and identifies negations. The lexer identifies a very large number of negation signals and classifies them on the basis of properties such as whether they generally precede or succeed the concept they negate and whether they can negate multiple concepts. Each class generates a single token that is passed to the parser. The parser then applies its grammar rules to associate the negation signal with a single concept or with multiple concepts preceding or succeeding it. The output of this step is a modified version of the output of the original concept finder, with the negation information added.
 - A *verification step* marks up the original document by color-coding the text to assist human validation of the program's output. Negated simple concepts are highlighted in red, negated composite concepts in orange, negated non-concepts in magenta, and the negating phrases themselves in blue. (A monochrome equivalent is reproduced in the bottom panel of Figure 1.) This step will not be described further.
- Negfinder was designed and trained with a set of 40 medical documents from diverse specialties (radiology reports, surgery notes, discharge summaries). It was then evaluated with a test set of 60 documents (30 discharge summaries, 30 surgical notes) in which the marked-up text was inspected visually to quantify false-positive and false-negative results. To estimate the effect of priming bias caused by examining previously marked negations, we did a second evaluation with 10 documents (5 discharge summaries, 5 surgical notes) where the original (unmarked) text was inspected. Radiology notes were not used for

testing because, in our sample data, they turned out to be the easiest with respect to negation identification, yielding almost 100 percent results with just a few simple rules. As discussed later, discharge summaries, especially in psychiatry, proved to be the most challenging. The length of the notes in the test set varied between 600 to 2,000 words, with a sentence length varying between 3 and 26 words. Discharge summaries tended to be longer than surgery notes and also tended to have longer sentences.

UMLS Concept Recognition

The approach used to recognize UMLS concepts in the narrative has been described in detail earlier²⁵ and is summarized here (along with a brief description of recent enhancements and changes necessary to handle negation). We use a phrase-recognizing utility that is bundled with a commercial package (IBM Intelligent Miner for Text²⁶) to identify phrases of potential interest, which we then attempt to match to concepts in a subset of the UMLS Metathesaurus, 2000 Edition, that resides in a Microsoft SQL Server relational database management system. Negfinder also writes its output to the same database. The phrase recognizer uses a customizable stop-word list. (Stop words are very common words in English natural language that are not useful for indexing by themselves, such as common pronouns and conjunctions.) Some words that flag negations, e.g., “denies” and “absent,” also occur in the UMLS and must be filtered out initially, so that the phrases to be recognized do not contain any stop word or negation word.

The concept recognition algorithm seeks to match an entire phrase (up to five non-stop-words long) uniquely to a UMLS concept. If a match to the entire phrase is not obtained, it then attempts to match subsets of the words in the phrase. Ambiguous-term and ambiguous-string lists of the UMLS are used to ensure that the phrase being matched is not homonymous. (If it is, we treat it as a “pseudo-concept” and code it slightly differently from identified concepts.) The current algorithm has been modestly refined over the original to improve specificity and speed.

Input Transformation

The algorithmic details of input transformation are straightforward and will not be described. A segment of the transformed input is illustrated along with the original text in Figure 1. The figure shows that, while many non-stop-words and -phrases are matched to UMLS concepts, some are not. This can happen for several reasons—the word does not occur in UMLS

(*Sarajevo*), an exact concept match to a phrase cannot be found (*EEG Activity*), or the word is an “undiscovered homonym,” where multiple matches occur that cannot be disambiguated using the current versions of the UMLS homonym tables because they have not been recorded as such. Examples of undiscovered homonyms are *rub*, *S1*, and *S2*. *Rub* can refer to a friction sound (e.g., pleural or pericardial rub), or a physical action, while *S1* and *S2* can refer to the sacral vertebrae, the corresponding spinal nerves, or to the first and second heart sounds.

Identification of Negation

In our pilot work, we asked medically trained observers to exhaustively mark all instances of negation on randomly selected discharge summaries, surgical notes, and radiology reports. The purpose of this was twofold:

- To define exactly what we should call a negation
- To discover the types and complexity of the structures used to express negation, in order to decide what kind of syntactic approach was necessary

On the basis of this study, we decided to include only independent words implying the total absence of a concept or thing in the current situation. Words signifying temporal transitions such as stopping or discontinuing a drug were not treated as instances of negation, nor were instances in which a concept itself had a negative connotation, such as “akinesia.”

Most of the negations did turn out to be straightforward, and the words or phrases indicating negation (NegPs) were usually in close proximity to the concepts they negated, as we had expected. Nevertheless, several complexities had to be dealt with as shown below (NegPs are shown in italics and negated elements in bold):

- The negation signals were quite heterogeneous, from single words (“no,” “without,” “negative”) to simple phrases (“no evidence of”) and complex verb phrases (“could not be currently identified”).
- There is a large set of verbs that, when preceded by the word “not,” negate their subject or object concept (“X is not seen,” “does not show X”); but there are also a large number of verbs that do not do so (“X did not **decrease**,” “does not **affect** X”). These need to be correctly distinguished.
- The negation signals may precede or succeed the concepts they have scope over, and there may be several words between the two (“there was *absence* of this type of X,” “X, in this instance, is *absent*”).

- A single negation signal may serve to negate a whole list of concepts either preceding or following it (“**A, B, C, and D** are *absent*” “*without evidence of A, B, C, or D*”); or it may scope over some but not all of them (“there is *no A, B and C*, and D seemed normal”).

These observations made it apparent that employing the lexical scanning and LR parsing tools used for parsing computer languages was feasible but that a great deal of customization would be required to handle all the complexities.

We built Negfinder’s lexer and parser using a software package called *Visual Parse++* (SandStone Technology, La Jolla, California). *Visual Parse++*, which runs under the Windows operating system, incorporates the functionality of the Unix utilities *lex* and *yacc*. *Visual Parse++* is language independent and incorporates a visual debugging environment, allowing interactive debugging of a grammar specification. On the negative side, differences in the specification language (compared with *lex/yacc*) make porting of existing *lex/yacc* specifications a non-trivial chore; in this context, a *Visual Parse++* lexer specification tends to be considerably larger than the equivalent *lex* specification.

The Lexical Scanner

The main task of the lexer in our application is to identify words or phrases that signify negation and pass them as special tokens to the parser. This has to be followed by the more difficult task of deciding which particular concepts were negated.

On the basis of an initial training set of medical documents including discharge summaries, surgical notes, and radiology reports, we have set up our lexer to recognize more than 60 distinct words or patterns that express negation, such as “absent” or “not visualized.” Combinations and inflections of these forms are also recognized. Some of the patterns (such as words ending in “n’t”) match multiple negative words, and the lexer handles interspersed adverbial forms such as “not currently visualized.” As a result, the number of distinct NegPs recognized by the lexer easily runs into the thousands. A specific token is passed to the parser to represent the exact way in which the NegP is used for negation. Unique tokens are generated for combinations of all the following characteristics of NegPs:

- Does the NegP precede or follow the concepts it negates? (The word “no” precedes the concepts it negates, whereas the negated concept would precede the phrase “not present.” We term these prefix- and postfix-negatives, respectively.)

- Can the NegP negate multiple concepts? (A single “no” can negate a whole list of concepts, whereas “non” negates only the very next word.)
- If the NegP can negate a list of concepts, is the terminal conjunction an “or” or an “and”? (Prefix negatives generally take an “or,” as in “**no murmurs, rubs or gallops,**” whereas postfix negatives take an “and,” as in “**murmurs, rubs, and gallops** are *absent*”).

When it encounters words that usually end the scope of negations, the lexer also outputs a “negation-termination” token. This prevents the parser from mislabeling subsequent concepts as being negated. Common negation terminators are:

- Most prepositions, such as “at,” “in,” “after,” and “during.” These generally begin a prepositional phrase that may contain concepts that are not part of the preceding negation (e.g., the preposition “during” in the fragment “**no complications during** surgery,” where “**complications**” is negated but “surgery” is not). An exception is the preposition “of,” which is used in many noun phrases expressing composite concepts.
- Most conjunctions, such as “and,” “but,” and “or.” These generally begin coordinate clauses that do not participate in the preceding negation (e.g., “there was **no bleeding** but the blood pressure was low”). This is complicated by the fact that “and” and “or” are frequently used to coordinate noun phrases, gathering several concepts into a concept list. The two cases are distinguished by looking for the presence of an inflected verb that signals a coordinate clause (e.g., in the sentence “there was *absence of murmurs, S3, and S4* and pulse was normal,” the first “and” is part of a negated concept list, whereas the second “and” is followed by the verb “was,” indicating that it begins a coordinate clause and is therefore a negation terminator).
- Personal pronouns, such as “I,” “he,” “she,” and “it.” Sometimes in dictated reports conjunctions are dropped, and a pronoun begins an independent clause that terminates the negation (e.g., “the patient is *not delirious*, he appears to be aware of his surroundings”).
- Relative pronouns, such as “which” and “that.” These usually begin a non-negated subordinate clause modifying the negated concept (e.g., “There was *absence of temperature sense* which supports spinothalamic involvement”).

Since it detects large numbers of NegPs and negation terminators, Negfinder’s lexer does much more work

than lexers used for traditional programming languages, which typically return tokens corresponding to one or two words at most. The Negfinder lexer often returns a single token that corresponds to several possible combinations of words. Thus, combinations of *is/was/were/are/been* followed by an optional adverb followed by *denied/refused/omitted/lacking/excluded* will generate a single token that is passed to the parser. For this purpose, the lexer makes extensive use of regular expressions in addition to using limited part-of-speech information as described above.

The Parser

Negfinder's parser uses a grammar far simpler than those traditionally used for NLP, because it makes no attempt to parse the sentence structure in detail. Rather, it focuses on the occurrence of concepts matched to the UMLS, negation signals, negation terminators, and sentence terminators and treats most other words merely as fillers. The main task of the parser is to assemble contiguous concepts into a list, if required, to associate a concept or a list of concepts with a negative phrase that either precedes or follows it to form a negation, and to accurately determine where the negation starts and ends. A flavor for the kinds of grammar rules used and the types of difficulties that the parser encounters are given below.*

The first of the tasks listed above—that is, to assemble contiguous concepts into a list,—is accomplished by using the following type of grammar rule:

conceptlist : conceptlist ',' concept.

This rule recursively allows a concept list to incorporate two or more contiguous concepts separated by commas. As discussed above, these rules take into account punctuation (such as commas) and conjunctions (such as "or" and "and") that may constitute part of a concept list. The situation is made more challenging by the fact that the parser has to cater to those cases in which some concepts are not recognized and coded but are still parts of negated lists. Thus, in a sequence of the type "no filler1, filler2 or concept1," the parser still has to recognize that concept1 is negated, even though no formal concept list, as defined above, has been started. To account for such cases, a total of 16 such rules are required for this function alone.

Determining when a negation begins and ends can be a difficult problem for a simple parser such as Negfinder's that does not analyze complex sentence

structure. A perfect solution would require accurate sentence segmentation and perhaps a rich semantic model to identify cases in which the concept being negated is remote from the negation signal or separated from it by a long phrase or clause. Our parser relies on the list of "negation terminators" as described above. When these do not occur, the parser arbitrarily relies on a window of three intervening filler words (not concepts or NegPs) to terminate the current instance of negation or concept-list formation. (The choice of three words was found to provide a suitable tradeoff of sensitivity vs. specificity.) Thus, two concepts are not considered to be part of a list if they are separated by more than three filler words, and a concept is considered to be negated only if the NegP does not precede or follow it by more than three intervening filler words. As the results show, this simplistic strategy does fairly well in practice in medical documents.

The final output of Negfinder consists of an entry for each concept—its concept ID, its starting and ending byte offsets within the document, the presence of negation, and whether the negation represents a compound concept. This output is similar to the data that are stored in proximity indexes generated by information retrieval engines when individual words are indexed.

Evaluation

Two separate evaluations were carried out. In the first, a set of 60 test documents was passed through Negfinder and the marked-up documents were distributed between three observers,[†] who were assigned the following tasks:

- To look at the marked-up negated concepts and decide whether they were, in fact, negated
- To ignore the mark-up and look through the document to find any instances of missed negations
- To point out and comment on instances where Negfinder did not accurately mark the beginning or end of a negation

The first task yielded the number of true-positive results (TP) and false-positive results (FP), and the

[†] The observers were the three authors. The first and third authors (PGM and PMN) were the developers of Negfinder, while the second author (AD) was the primary reviewer. The final tally of the results was done only after all three observers conferred and discussed their findings, which were then re-examined by the primary reviewer, taking into account types of errors found or missed by all the observers. Although the reliability of the observers was not measured, this iterative process was found to considerably increase the reliability of the error detection.

*More details are available online, at <http://www.jamia.org>, as appendixes to this article.

second one identified the false-negative results (FN). The true-negative results (TN) were calculated as the number of concepts identified in the documents that were not negated.

From these numbers, the specificity ($TP \times 100 / (TP + FP)$) and sensitivity ($TP \times 100 / (TP + FN)$) of Negfinder could be determined.

Since the identification of negations on already marked-up documents could introduce a bias, a second evaluation was carried out using a different set of ten documents that were independently checked for negations by the primary reviewer and Negfinder, and the results were compared. (A smaller number of documents were used in this evaluation because of the significantly increased cost of evaluation.)

Results

Figure 2 shows the results of the quantitative evaluation using 60 test reports. The specificity of the negation finder for the test set was 97.7 percent, and the sensitivity was 95.3 percent.

Figure 3 shows the result of the additional, independent evaluation using ten separate reports. The specificity dropped to 91.8 percent, whereas the sensitivity was about the same, 95.7 percent.

The difference between the results of the two evaluations, however, barely missed statistical significance ($P = 0.0517$ by chi-square test), indicating that the differences might have been due to chance. If we assume, however, that the observed difference in specificity is real, it may be accounted for partly by differences in the content of the two document collections and partly by priming bias during the first experiment. The relative contribution of each factor, however, is hard to quantify.

	Negation found by observer	Negation not found by observer
Negation found by program	True positives: 544	False positives: 13
Negation not found by program	False negatives: 27	True negatives: 7,774

Figure 2 Results of Evaluation 1, showing performance of Negfinder on a test set of 60 documents (30 discharge summaries, 30 surgical notes) using human evaluation of color-coded text previously marked up by Negfinder. This test has the possibility of priming bias.

Discussion

In this paper we have described Negfinder, a program that uses tools designed for parsing computer languages to identify negations in medical documents. There is no doubt that these tools are not powerful enough to handle the general problem of natural language understanding, which is extremely difficult. Nevertheless, the current work shows that, as we had hypothesized, the detection of negation in medical narrative is a constrained problem that is amenable to lexical scanning using a finite state machine and to parsing using the restricted LALR(1) type of CFG.

Negfinder finesses an important limitation of LALR(1) grammars—the problem of single token look-ahead—by identifying and passing complex negating phrases, such as “could not be specifically visualized,” as single tokens and also by the use of states that change the behavior of the lexer on the basis of the occurrence of a negation or concept. Negfinder’s lexer generates a finite state machine that is several orders of magnitude larger than its parser, and this is where most of Negfinder’s complexity resides. Despite the simplicity of the grammar and the lack of extensive analysis of sentence structure, Negfinder seems to do a fairly adequate job of identifying negations in medical narrative.

The results show that Negfinder has a sensitivity and specificity between 91 and 96 percent in detecting negations in medical documents. The slightly worse specificity values on our second evaluation suggests some evidence of priming bias when the negations are marked up by Negfinder as opposed to being independently discovered by a human observer. Nevertheless, the results are not substantially worse (sensitivity unchanged, specificity lower by 6 per-

	Negation found by observer	Negation not found by observer
Negation found by program	True positives: 135	False positives: 12
Negation not found by program	False negatives: 6	True negatives: 1,716

Figure 3 Results of Evaluation 2, showing performance of Negfinder on a test set of 10 documents (5 discharge summaries, 5 surgical notes), using an unbiased design of independent evaluation by a human observer and Negfinder.

cent, but still about 92 percent) in the independent mark-up case; as previously stated, the differences between the two evaluations did not reach statistical significance.

An interesting side observation of the second evaluation was that, even though the human observer took great care in proofreading the ten notes, Negfinder still flagged a few obvious negations that had been overlooked by him, and these were comparable in number to the more difficult ones that it missed. This means that Negfinder's sensitivity was as good as that of the human observer on this very tedious task. This observation indicates that it might be useful to use a program like Negfinder to carry out an automatic color mark-up of dictated medical documents before they are approved by the dictating physician, to highlight negations and make sure that they have been accurately transcribed (especially by medical dictation systems that use automated speech recognition technology).

Analysis of Errors

The errors made by Negfinder can be classified as follows. (In the examples, concept strings are shown in bold, negated concept strings in bold italics, and negating phrases in italics):

- *Errors caused by omissions in the lexer's list of negatives or negation terminators.* Thus, in the fragment "*no seizure activity* throughout his **detoxification**," the program correctly identified "*seizure activity*" as being negated but also marked "**detoxification**" as being negated, because the word "throughout" was not on its list of negation terminators.
- *Errors caused by non-standard language usage.* Negfinder is flexible enough to accommodate non-standard usage when it is unambiguous, but this is not always possible. The following sentence fragment is from our test set: "There is *absent vibration sense* distal to..." Here the program could not identify "*vibration sense*" as being negated because in standard usage, "*absent*" succeeds the words it negates.
- *Errors caused by Negfinder's use of a three-word window to terminate a negation.* As mentioned, Negfinder terminates a concept list or negation if there are more than three intervening words between concepts or between a negating phrase and a concept. Often, this gives the desired result of avoiding FPs. However, the presence of interposed parenthetical phrases between the concept and its negating phrase can give FNs because the

window is now too narrow. In the fragment "several **blood cultures**, six in all, had been *negative*," Negfinder could not identify the "**blood cultures**" as the concept that was being negated by the word "*negative*," because it was too far away.

- *Errors caused by some double negatives.* The program does correctly parse some double negatives, such as "**X-rays** were *negative* except for..." but fails on others such as "The patient was *unable* to walk for long periods *without* **dyspnea**," where it identified **dyspnea** as being negated. It would be hard to correct these errors using a syntactic parser alone. (In the first example and the one in the previous paragraph, the word "*negative*" actually means "negative for significant findings" where the omitted phrase "significant findings" is the actual concept being negated and not the test itself. Negfinder does not currently distinguish this from the case where no test was done. The word "*negative*" is also somewhat problematic in that it may precede or follow the concept it negates: "**HIV** *negative*" or "*negative* **HIV test**." Negfinder can usually identify the correct negated concept if there is no other concept in the vicinity but may flag the wrong concept in a construction such as "**Neck**: *Negative* **JVD**.")
- *Errors caused by improper recognition of noun phrases representing temporary composite concepts that are the actual entities being negated.* In the fragments "*no signs* of **recurrence** of **diverticulitis**" and "*no complications* of the **procedure**," the negation involves the entire composite concept (noun phrase) following the word *no*, but not necessarily every component of it. In both these cases, Negfinder flags all the component concepts as being negated. This is correct in the first case but not in the second, where "**complications**" is negated but "**procedure**" is not.
- *Miscellaneous errors.* Negfinder does not currently recognize single words with contained negatives such as "*Seidel-negative*" or "*non-distended*," as instances of negations, since its lexer only passes word-level tokens. The same negations are recognized if they are presented as separate words—"Seidel *negative*" or "*non distended*."

In most cases, errors made by Negfinder are easily correctable by syntactic methods and involve minor modifications of the lexer or parser. However, in some cases semantic methods may be required, such as better characterization of temporary composite concepts using noun phrase detection combined with a rich semantic model of the domain.

Distribution of Negating Words and Phrases

Analysis of the negating phrases shows that a small number of them are very common and a relatively large number appear only occasionally. In our test material, four words appear in 92.5 percent of the negations. These are “no” (49.3 percent), “denies/denied” (20.5 percent), “not” (12.8 percent) and “without” (9.8 percent).

At first glance, this would seem to indicate that a relatively simple parser that recognized these four NegPs alone could be quite accurate. However, further examination shows that this is not the case. First, “no” includes several variations, such as “no evidence of,” “no sign of,” and “no history of.” Second, quite often a single word negates a large number of concepts as a list, as in “Patient *denies* **shortness of breath, chest pain, fever, chills, nausea, vomiting, diarrhea, and abdominal pain.**” Thus, a parser must be able to detect concept lists negated by a single NegP. Without this capability, the simple parser mentioned above would catch only 67.8 percent of negated concepts in our test set.

Third, more than 50 percent of the time that the word “not” occurs, it does not negate a concept at all but only a succeeding verb. Thus, “**dyspnea** is *not* seen” is quite different from “dyspnea did *not* **improve,**” as in the first instance “dyspnea” is negated, whereas in the second it is not. Thus, Negfinder incorporates large lists of verbs like “see” or “show” that are used to negate concepts and also notes whether they precede or follow the negated concept (“**fracture** is *not* seen on X-ray” vs “X-ray does *not* **show** **fracture**”). Such detailed knowledge is the basis of the high sensitivity and specificity shown by Negfinder.

The sophistication of negation in our material was much greater in the discharge summaries than in the surgical notes or radiology reports. (In our first test set of 60 documents, which had equal numbers of discharge summaries and surgical notes, the discharge summaries accounted for the vast majority of errors—34 of 40 errors, or 85 percent.) Hence it is expected that Negfinder should do better on surgical notes and radiology reports.

Necessity of Recognition of Compound Concepts

One weakness of our current concept recognition system in recognizing noun phrases affects detection of the beginning and end of negations. Thus, when a phrase such as “shortness of **breath**” was negated, Negfinder marked the words “shortness of” and the concept “breath” as negated. However, this is not

quite right, since “breath” is not being negated at all, but rather the compound concept “shortness of breath.” If “shortness of breath” is passed to Negfinder as a single concept, then it is correctly flagged as negated. This emphasizes the need for better recognition of compound concepts, perhaps by using a true noun-phrase detection program.

Future Directions

Several enhancements to both negation detection as well as concept recognition have suggested themselves in the course of the evaluations described in this work, and we plan to address these in the near future. As described previously,²⁵ certain other enhancements are required in the phrase recognizer, which is not smart enough to split phrases on verbs, so that it sometimes yields curious pseudo-phrases such as “*lower extremity wrapped with ACE bandage*” (phrase italicized).

In the next version of the recognizer, we plan to incorporate lookup of the SPECIALIST lexicon,²⁷ which is distributed along with the UMLS and contains part-of-speech information about most of the common words in English and their inflected forms. In the above example, this will allow the word “wrapped” to be recognized as a past-participial form that is unambiguously a verb (as opposed to the finite form, “wrap,” which can also be a noun), and this information can be used to split the pseudo-phrase into smaller parts. It remains to be seen to what extent this approach will improve concept matching. We may need to consider the use of a stochastic tagger to resolve the noun-vs.-verb ambiguity, in addition to the existing IBM text analysis tool. (In its defense, the latter taps into a large database of place, person, and organization names and identifies most of them correctly as such, so the developer can easily eliminate many words from further consideration.) Combining the input from both taggers is an interesting challenge.

A final issue is that many UMLS concepts themselves represent antonymous forms of other concepts, e.g., words beginning with “anti-,” “an-,” “un-,” and “non-.” Such forms are not necessarily negations. (Thus, an anti-epileptic drug is used when epilepsy is present; “non-smoker,” however, is a true negation.) Antonym information is not currently recorded in the MRREL table of UMLS. It is possible that antonym information is not useful for query-broadening strategies that use negation, since clinicians who query a system would probably specify an antonym prefix explicitly in the query. This issue, however, requires further experimentation and feedback from users.

The lexer/parser specification can be downloaded from <http://ycmi.med.yale.edu/mutalik/negation.zip>. Evaluation of the lexer/parser requires *Visual Parse++*, a free time-limited evaluation copy of which can be downloaded from <http://www.sandstone.com>. The Visual Basic code uses a SQL server database, the schema of which can be requested from the third author (PMN).

The authors thank Prof. Carol Friedman PhD, of Columbia University and the City University of New York, for making a demo version of MedLEE publicly accessible at <http://cat.cpmc.columbia.edu/medleexml/demo/>. They also thank the reviewers of the earlier version of this paper, as their comments helped to greatly improve its quality.

References ■

- Hersh WR. Information Retrieval: A Health Care Perspective. New York: Springer-Verlag, 1996.
- Salton G. Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer. Reading, Mass.: Addison-Wesley, 1989.
- Lindberg DA, Humphreys BL, McCray AT. The Unified Medical Language System. *Methods Inf Med*. 1993;32:281-91.
- Aronson A, Rindfleisch T, Browne A. Exploiting a large thesaurus for information retrieval. *Proc RIAO '94 Conf*; New York; October 1994. 1994:197-216.
- Rindfleisch TC, Aronson AR. Ambiguity resolution while mapping free text to the UMLS Metathesaurus. *Proc Annu Symp Comput Appl Med Care*. 1994:240-44.
- Elkin PL, Cimino JJ, Lowe HJ, et al. Mapping to MESH: the art of trapping MESH equivalence from within narrative text. *Proc Annu Symp Comput Appl Med Care*. 1988:185-90.
- Wagner MM. An automatic indexing method for medical documents. *Proc Annu Symp Comput Appl Med Care*. 1991:1011-7.
- Goldberg H, Hsu C, Law V, Safran C. Validation of clinical problems using a UMLS-based semantic parser. *Proc AMIA Fall Symp*. 1998:805-9.
- Salton G, Wu H, Yu CT. Measurement of term importance in automatic indexing. *J Am Soc Inf Sci*. 1981;32(3):175-86.
- Wilbur WJ, Yang Y. An analysis of statistical term strength and its use in the indexing and retrieval of molecular biology texts. *Comput Biol Med*. 1996;26(3):209-22.
- Horn LR. A natural history of negation. Chicago, Ill.: University of Chicago Press, 1989.
- Horn L, Kato Y. Negation and polarity: syntactic and semantic perspectives. New York: Oxford University Press, 1999.
- Tomita M. Efficient parsing for natural language: a fast algorithm for practical systems. Boston, Mass.: Kluwer Academic Publishers, 1986.
- Aho AV, Sethi R, Ullman JD. Syntax-directed translation. In: *Compilers: Principles, Techniques, Tools*. Reading, Mass.: Addison-Wesley, 1988:33-40.
- Tomita M. An efficient augmented-context-free-parsing algorithm. *Comput Linguistics*. 1987;13:31-46.
- Roche E, Schabes Y. Finite-state language processing. Cambridge, Mass.: MIT Press, 1997.
- Levine JR, Mason T, Brown D. *lex & yacc*. 2nd ed. Sebastopol, Calif.: O'Reilly, 1992.
- Friedl JEF. *Mastering Regular Expressions*. Sebastopol, Calif.: O'Reilly, 1997.
- Horn LP. Pick a theory (not just any theory): indiscriminatives and the free-choice indefinite. In: Horn L, Kato Y (eds). *Negation and Polarity*. New York: Oxford, 1999.
- McQuire AR, Eastman CM. Ambiguity of negation in natural language queries to information retrieval systems. *J Am Soc Inf Sci*. 1988;49(8):686-92.
- Aronow D, Feng F, Croft WB. Ad hoc classification of radiology reports. *J Am Med Inform Assoc*. 1999;6(5):393-411.
- Friedman C, Alderson P, Austin J, Cimino J, Johnson S. A general natural-language text processor for clinical radiology. *J Am Med Inform Assoc*. 1994;1(2):161-74.
- Friedman C, Shagina L, Socratous SA, Zeng X. A WEB-based version of MedLEE: a medical language extraction and encoding system. *Proc AMIA Fall Symp*. 1996:938.
- Friedman C, Hripcsak G, Shagina L, Liu H. Representing information in patient reports using natural language processing and the extensible markup language. *J Am Med Inform Assoc*. 1999;6:76-87.
- Nadkarni PM, Chen RS, Brandt CA. UMLS concept indexing for production databases: a feasibility study. *J Am Med Inform Assoc*. 2001;8(1):80-91.
- Intelligent Miner for Text 2.3. Armonk, NY: IBM Corp, 1999.
- McCray A, Aronson A, Browne A, Rindfleisch T, Razi A, Srinivasan S. UMLS knowledge for biomedical language processing. *Bull Med Libr Assoc*. 1993;81(2):184-94.