A term extraction tool for expanding content in the domain of functioning, disability, and health: proof of concept

Marcelline R. Harris, a,b,* Guergana K. Savova, a Thomas M. Johnson, a and Christopher G. Chute a

a Department of Health Services Research, Division of Medical Informatics Research, Mayo Clinic, Harwick 8-31, 200 1st St SW, Rochester MN, 55905 USA
b Department of Nursing, Division of Nursing Research, Mayo Clinic, Rochester MN, 55902 USA

Received 22 August 2003

Abstract

Among the challenges in developing terminology systems is providing complete content coverage of specialized subject fields. This paper reports on a term extraction tool designed for the development and expansion of terminology systems concerned with functioning, disability, and health. Content relevant to this domain is the emphasis of the foci and targets of many nursing terminologies. We extend previously published term extraction algorithms by applying two filters. The first filter is based on the raw frequency of the content words in the lexical string under consideration. The second filter applies the notion of a complete syntactic node to discover relevant noun or verb phrases. While we report on a limited corpus (30,607 words comprising 4103 terms from 60 dismissal note summaries), the recall, precision, and F-measures we observed are encouraging and suggest continued development and testing of the tool is merited.

© 2003 Elsevier Inc. All rights reserved.

Keywords: Term extraction; Terminology tools; Functioning, disability, and health; Functional status; Nursing terminologies, ICF

1. Introduction

Many nursing terminology initiatives currently emphasize techniques and processes that will support an evolution from classifications to formal systems of concepts. The stages of such an evolution have been described as “generational” [1]. First-generation systems are characterized by pre-coordinated phrases, and consequently there is limited re-use of data, and limited flexibility and extension of the systems. Semantic-based machine processing of first-generation systems is not possible. Second-generation systems include semantic categories that organize sets of concepts and a knowledge base of dissections, or structural patterns of the terminology. Third-generation systems feature formalisms or inference engines that enable concepts to be manipulated in order to compose phrases from primitives [2].

A document proposing reference models for nursing terminology systems was prepared by Technical Committee ISO/TC215 (Health Informatics) of the International Organization for Standardization (ISO) [3]. The models describe structures for the dissection of nursing diagnoses and nursing interventions that will enable the integration of nursing terminologies into reference terminologies, i.e., second- and third-generation systems. Descriptors specific to focus and judgment are required for diagnoses while descriptors specific to target and actions are required for interventions. As such, these models can be considered a “top-down” approach to nursing terminologies because the starting point is the implicit structure of the major nursing terminologies, not actual instances of clinical data within patient records.

The ISO model for nursing diagnoses has been shown to accommodate the dissection of two of the major nursing terminologies (NANDA and Omaha) and their integration into SNOMED CT, suggesting the ISO model offers an adequate structure to direct the
expression of those two terminologies in second-generation systems [4]. There is further evidence demonstrating the formal expression of one first-generation nursing terminology system (NANDA) as a third-generation system [5]. Together, these two studies lend substantial support to the feasibility of modeling first-generation nursing terminology systems as second-generation systems, and of applying the formalisms associated with third-generation systems in order to post-coordinate the expressions within first-generation nursing terminology systems.

Among the remaining challenges is the expansion of the content within nursing terminology systems, which appears to have been constrained by several factors. First, the manual identification and validation of relevant entry terms is labor intensive and time consuming. For many of the nursing terminology systems the funding for this effort is rather modest. Second, many nursing terminology systems have been developed using consensus-based approaches to agree on the terms to be included in a terminology. The need to balance clinical specificity with generalizability across settings and patient groups has undoubtedly required that nursing terminology developers exclude some potentially useful terms and phrases in the process of obtaining consensus. Finally, as in all fields there is some resistance to the notion of terminology standardization. These factors have, both independently and jointly, affected the extent of content coverage provided by any specific nursing terminology system.

The automated and semi-automated processing of text in order to identify single or multiple word terms associated with concepts relevant to a subject field is referred to as term extraction. Term extraction with no preconceived overall terminology structure is considered a "bottom-up" approach to terminology development in that it produces the raw material for terminology databases, i.e., the raw material that has to be examined, tested and validated for inclusion in a terminology database [6]. Ideally, the development of terminology systems requires a close coordination between bottom-up and top-down approaches as bottom-up approaches allow the data to drive the structure and top-down approaches provide a necessary general flow of the structure.

The purposes of this paper are to (1) provide background on the goals, methods, and evaluation metrics associated with automated term extraction tools and (2) report on an initial evaluation of a term extraction tool we have developed related to the domain of functioning, disability, and health. The naming of the concepts within this domain has long presented significant challenges not just for nursing, but for many other disciplines as well. For example, the World Health Organization's International Classification of Functioning, Disability, and Health (ICF) is a notable example of an international effort to describe conceptual classes that begin to organize the conceptual space and definitions of terms for this domain. Coded terms are suggested within each ICF class although, as with the nursing terminology systems, the number of terms is quite limited and they are not yet organized as second- or third-generation terminology systems [7,8]. While not a focus of this study, we think it is likely that a structure such as the ICF will be helpful to nursing terminology development, as this domain is such an important emphasis of the foci and targets of nursing terminologies, and therefore of the ISO models.

2. Term extraction

Term extraction (TE) is just one of the processes in terminology development in which automation is likely to produce significant benefits. However, the availability of tools to assist in TE is somewhat variable, in part because corpus analysis is a relatively new field [9]. Among the important considerations in the development and/or selection and use of TE tools are the motivation for the tool, methodological approaches, and evaluation metrics. Each is briefly discussed below.

2.1. Motivations for term extraction tools

Tools for term extraction are motivated by different uses including terminology development and/or enhancement; building glossaries, thesauri, terminological dictionaries, and knowledge bases; automatic indexing; machine translation; and corpus analysis [10]. Tools tailored for one use may be difficult to apply to another use.

Automatic indexing in order to optimize information retrieval (IR) and document retrieval (DR) serves as the motivation for many TE tools. From the standpoint of IR systems, the primary goal of automated term extraction is to identify an indexing terminology system of single word and/or multi-word items that are tailored to a particular domain. The indexing terminology thereby serves as an aid to information retrieval queries. Document retrieval answers questions about which documents or references include specific concepts, and in addition to indexing terminologies, DR tools often additionally emphasize the metadata that provides access to documents. The goal of TE for both IR and DR is to isolate terms that contain enough "semantic load" to support retrieval based on terms users supply when querying a set of documents [11].

Another motivation of automated TE tools is term discovery and the use of automated term acquisition methods to find concepts and associated terms in order to develop formal terminologies (or ontologies) within a specific domain of interest [12]. TE tools motivated by
ontology development do not completely overlap with IR and DR, and terminologists argue that the different intended uses need to be differentiated when evaluating TE tools [9,11]. A related motivation in the use of TE tools is to discover subject-specific terms and map them to existing terminology systems. In this case, the output of TE tools must be linked to existing semantic networks or systems such as the Semantic Network and Meta-Thesaurus of the National Library of Medicine (NLM)s Unified Medical Language System (UMLS) [13–15].

2.2. Methods and tools for term extraction

The general process of TE using natural language processing (NLP) techniques consists of term acquisition proper followed by term recognition. Acquisition proper identifies the candidate terms from a corpus of free-flowing text. Recognition verifies those candidate terms against an established list or an expert to identify known and unknown terms[16]. TE tools can be grouped into three methodological categories: linguistic, statistical, or hybrid [11]. Of note, regardless of the approach, all TE tools analyze text and produce terms that need to be confirmed for their relevance through collaborations between terminologists and domain experts.

Linguistic approaches rely on the detection of typical syntactic patterns that define the structure of the terms for the domain of interest. For example, if domain relevant terms have the syntactic make-up of a noun phrase consisting of an adjective and a noun (NP → Adj N), tools based on purely linguistic algorithms will detect all phrases with that syntactic pattern and flag them as term candidates. We refer to rules that form syntactic phrases as node-internal syntactic patterns, e.g., in the above example the noun phrase NP is the syntactic node or phrase and its node-internal syntactic pattern is an adjective followed by a noun (Adj N). Current term extraction systems focus mainly on noun phrases (NPs), but several studies show that for the healthcare domain, verb phrases (VPs) are as important as NPs [17,18]. As an example, analyzing verb and noun phrases, Grobe and associates were able to demonstrate an automated categorization algorithm for nursing interventions [17]. A limitation of purely linguistic approaches is that a rule base of all node internal syntactic patterns must be maintained; terms that do not “fit” a specific rule will not be extracted [11].

Tools based on purely statistical algorithms do not use any linguistic information, rather they rely on detecting lexical units based on metrics that reflect the importance of a given word or a given combination of words within a phrase. The primary metrics reported are frequency of occurrence and, less often, mutual information (the strength of association between two words) [11,20]. The most commonly cited limitation of statistically based approaches relates to the difficulties in identifying low-frequency terms consisting of more than two words as those frequencies decrease dramatically and proportionally to the length of the string [11].

Hybrid approaches to TE tool development offer a combination of the linguistic and statistical approaches, relying on linguistic filters, syntactic parsing (shallow or full) and statistical measures [11]. Hybrid methods appear to increase the correctly identified terms (recall) and reduce the noise, or increase precision [19].

2.3. Evaluation of term extraction tools

The standard measures used to evaluate TE tools are recall, precision and their combination, the F-measure [20]. Recall and precision statistics can be derived from $2 \times 2$ tables as illustrated in Fig. 1.

Since precision reflects noise in the tool, the failure analysis of a TE tool might include an estimate of false positives, or the number of terms identified by the tool that were not determined to be correct terms in the source text. Tools motivated by content coverage are designed to optimize recall, which causes more noise. In contrast, a tool built to demonstrate high precision (fewer false positives) will experience lower recall [20].

The F-measure (or F-score) is a statistic that allows tool developers to adjust for this relationship between recall and precision by applying a parameter $\beta$ to weight either recall or precision. The intended use of the tool will determine whether precision or recall is more important. $\beta$ Values less than one favor recall, while $\beta$ values greater than one favor precision. Recall and precision are equally weighted when $\beta$ is assigned a value of one [20].

3. A TE tool related to functioning, disability and health: a proof of concept

Our interest in a TE tool is motivated by two goals. The first is to enhance the content coverage of terminology systems specific to the domain of functioning, disability, and health. A second goal is to assure that the
terms within our local patient medical record information system can be mapped to reference terminologies for reuse in various applications such as the post coordination of nursing terminologies and also clinical and administrative decision support, alerts, outcomes analyses, and regulatory report generation.

Various limitations were noted in a review of published TE tools [10]. First, the majority of the published TE algorithms deal with noun phrases only; preliminary work by members of our group and published work by others requires the identification of verb phrases as well [17,18]. Second, the linguistic filters in the published algorithms and tools rely on strict node-internal syntactic patterns forming a term candidate. For example, to be considered a term, a noun phrase must consist of an adjective followed by a noun (NP → Adj N). Since automatic syntactic parsers are prone to errors, we were concerned that parts of the corpus would not be considered for term candidacy as the parsing error would preclude the exact match of the syntactic patterns. A final concern was that in many published algorithms, longer low frequency occurrences are not recognized as terms candidates because they do not pass the imposed statistical threshold. Our need was to go beyond noun phrases and include verb phrases tightly coupled with the application of statistical thresholds and a knowledge-lean approach that does not utilize a database of node internal syntactic patterns.

The study we report on below should be viewed as a proof of concept study because of the relatively small test corpus size. Our specific purpose at this point was to compare the recall, precision, and F-measures obtained by our algorithm to other linguistic and hybrid algorithms. Because our algorithm is a hybrid aimed at extracting NPs and VPs, we first compared it to a purely linguistic approach that extracts all NPs and VPs, we first compared it to a purely linguistic algorithm imposed a threshold of raw occurrence of 2; thus “frail seniors” passes it and is put on the list of term candidates. We simulated Justeson and Katz's algorithm as it was described in [21] but did not use their original implementation, as we did not find an available source.

Our experimental algorithm, also a hybrid approach, is described below.

3.2. Experimental algorithm

The algorithm underlying our TE tool relies on a fully parsed text, for which we used Charniak's parser [22]. The error rate of Charniak's syntactic parser on our data was calculated as 9%. Thus, a corpus processed through Charniak's parser is the expected input to our TE tool; our algorithm is implemented on top of the Charniak parser and does not introduce improvements of parsing techniques per se.

After parsing the dismissal summary texts, we next determined the ngrams within the corpus by keeping the links between the words. Ngrams are simply sequences of words as they occur in a corpus. 1-gram or a unigram represents one word, e.g., “seniors”; a bigram or 2-gram represents two sequential words, e.g., “frail seniors”; trigrams or 3-grams are sequences of three words, e.g., “of frail seniors”; 4-grams or tetragrams are sequences of four words, e.g., “the complaints of frail seniors”; 5-grams or pentagrams are chains of five words, e.g., “the complaints of frail seniors”; and so on. (Note, this phrase is presented only as an easily readable example of ngrams, the words “the” and “of” are function words that are part of our stopword list.)

Our term extraction algorithm applies two filters to the parsed text. Filter I is the frequency of a content word within given ngrams in the corpus; it is not the
ngram as a whole as used in the other algorithms, e.g., [21]. Filter 2 applies the notion of “syntactic node/syntactic phrase completeness.” We use “complete syntactic node” and “complete syntactic phrase” interchangeably to indicate a string of words that is syntactically well formed. For example, in the sentence “The patient requires maximum assistance with bathing,” “the patient,” “maximum assistance,” “with bathing,” “bathing,” and “requires maximum assistance with bathing” are complete syntactic nodes (noun phrase, noun phrase, prepositional phrase, noun phrase, and verb phrase, respectively). “Requires max” is not a complete syntactic node as it is only the beginning of a verb phrase.

We define a complete syntactic node as a NP or a VP at the uni-, bi-, and tri-gram levels and as an NP, VP, and sentence (S) at the tetra- and penta-gram level. If any given ngram is a complete syntactic node, it is extracted as a term candidate. Going through the ngrams, the algorithm extracts not only phrases consisting of one node (e.g., a NP consisting of an adjective and a noun like in “expressive aphasia” or “frail seniors”) but also embedded phrases (e.g., a NP consisting of a modified head noun followed by a prepositional phrase (PP) like in “a coordination and dexterity program for her right arm”). The search space is presently constrained by a maximum length of 5 content words (5-grams/pentagrams) because pilot studies we have conducted indicate domain experts tend to identify terms associated with specific concepts in this domain with 5 or fewer words. Stoplist words are not counted as content words. Presently, the stoplist consists of 183 items including prepositions, auxiliary verbs, pronouns, wh-words (e.g., “which,” “who,” “where,” “when,” etc.), pronoun/auxiliary abbreviations, quantitative adjectives (“many,” “any,” “none,” and “some”), and punctuation.

The following example illustrates the algorithmic steps for frequency cutoff of 5.

*Sentence in corpus:* The stroke patient requires maximum assistance with bathing. Stoplist words (not included in ngrams): the, with

**Unigrams and unigram frequency:**
- stroke: 1
- patient: 3
- requires: 4
- max: 2
- assistance: 10
- bathing: 5

**Bigrams:**
- stroke patient
- patient requires
- requires maximum
- maximum assistance
- assistance bathing

**Trigrams:**
- stroke patient requires
- patient requires maximum

**Fourgrams:**
- requires maximum assistance
- maximum assistance bathing

**Tetragrams:**
- stroke patient requires maximum
- patient requires maximum assistance
- requires maximum assistance bathing

**Pentagrams:**
- stroke patient requires maximum assistance with bathing
- patient requires maximum assistance bathing

**Complete syntactic nodes (NPs/VPs/S) (words on stopword list surrounded by brackets):**
- (the) stroke patient requires maximum assistance (with) bathing
- maximum assistance
- maximum assistance (with) bathing
- maximum assistance with bathing
- bathing

The unigrams that pass the frequency threshold of 5 are “assistance” and “bathing.” “Assistance” does not pass the 2nd filter (complete syntactic node) but “bathing” does. Therefore our algorithm will rule out the term “assistance” and leave the term “bathing.” At the bigram level, the terms “maximum assistance” and “assistance bathing” both pass the frequency filter as they contain at least one content word that passes the cutoff, but only the term “maximum assistance” passes the complete syntactic node filter and is added to the list of candidate terms.

The following list is the candidate term list generated from the sample sentence above, after processing all ngrams. (Words on the stoplist are in brackets):

**bathing**
- maximum assistance
- maximum assistance (with) bathing
- requires maximum assistance (with) bathing
- (the) stroke patient

This algorithm can be thought of as discovering frequent terms and their potential neighbors via a context of four content words. Our assumption is that frequent words are necessary descriptors of the domain. Extracting them as complete syntactic nodes/phrases encapsulates them in well-formed units able to stand by themselves as leaf nodes in a reference terminology, e.g., “bathing” and “maximum assistance” from the above example. Discovering their neighbors within a syntactic node gives us possible nested constructions and combinations of primitive concepts, e.g., “maximum assistance (with) bathing” and “requires maximum assistance (with) bathing” from the above example.

The algorithm dissects candidate terms into primitive concepts as long as the primitive concepts by themselves pass the two filters. The algorithm does not partition a complete syntactic node into suggested components if those components do not pass the requirements of the filters. For example, “dexterity program and coordination training” is on the candidate term list, as are its primitives “dexterity program” and “coordination training.”

*Algorithm:* This algorithm is designed to produce a list of terms that are both frequent in the corpus and also likely to be collocated with other frequent terms. It does this by first identifying all possible unigrams, bigrams, trigrams, fourgrams, and pentagrams in the corpus. For each of these ngrams, the algorithm checks to see if it meets the frequency threshold (if any). If it does, it is added to the candidate term list. Then, the algorithm applies two filters: the complete syntactic node filter and the complete syntactic phrase filter. The complete syntactic node filter requires that the term be a complete syntactic node, which means it must be a noun phrase, verb phrase, or sentence. The complete syntactic phrase filter requires that the term be a complete syntactic phrase, which means it must be a string of words that is syntactically well formed. If a term passes both filters, it is added to the final candidate term list. The algorithm then returns the list of candidate terms.
training” because one of their constituent content words passes the frequency cutoff and they form complete syntactic nodes (NPs in this case). The algorithm will not make further suggestions to include “dexterity,” “program,” “coordination,” and “training” unless those words occur elsewhere in the corpus by themselves and by themselves pass the two filters.

A goal in the development of this algorithm was to neutralize parsing errors from the automated parser we used and minimize database maintenance; the algorithm therefore does not rely on a database of node-internal syntactic patterns for the candidate terms as the linguistic filter, rather it checks for node completeness. An example of parse error tolerance is the term “requires max assist”:

```xml
<VP>
  <VB> requires </VB>
  <S>
    <NP> max </NP>
    <VP> assist </VP>
  </VP>
</S>
</VP>
```

Although the parse within the external VP is incorrect (the incorrect parse implies the meaning of “requires [that] Max assist”), the string is extracted as it forms a complete syntactic node (VP). The parse error is that the string “max assist” is a NP, not a sentence <S>, “max” is an adjective, not a noun <NNP>, and “assist” is not a verb <VB> but a noun. A database approach would rely on finding the syntactic pattern “VP V NP” (a verb phrase consisting of a verb followed by a noun phrase) and would not be extracted.

### 3.3. Implementation

The algorithm is implemented as a Java application on top of a relational database for persistence of extracted terms, inter-relationships, and statistical information. This is done with two primary goals in mind. First, it isolates applications from the complexities of storing and retrieving the syntactically parsed text, frequencies, and relationships. Second, it isolates the application from the storage medium (i.e., substituting a different relational database or another storage mechanism). Fig. 2 presents a high level view of information storage and retrieval through the Java application programming interface (API). The corpus is processed through the syntactic parser and transformed to extensible markup language (XML), then imported into a relational database (RDB). The information in the database includes the words, respective part-of-speech (POS) tag, e.g., noun, verb, adverb, etc., parent syntactic nodes, frequency of occurrences, ngram links up to five words, and the context (sentence) for each word and ngram.

Fig. 3 presents the high level architecture of the tool. Corpus information is externalized in object form; examples of objects include words, ngrams, syntactic nodes, references, and source documents. Each object then encapsulates relevant information such as frequency, syntactic tag, and text. New objects are easily generated as Java beans and inserted to persistent storage (the database) via the persistence application programming interface (API). Responsibility for handling new corpus information is delegated to a specialized application, or import utility. The import utility is responsible for interpreting the syntactically parsed XML, creating respective objects, assigning statistical and relational properties, and committing the objects to storage for later retrieval and analysis.

Once inserted, objects can also be retrieved, with statistical information and relationships intact, by applications performing specific analysis (e.g., our automated term extraction algorithm). Words, ngrams, and other objects can be queried based on a number of criteria. For example, an application might query for all ngrams with a minimum of 2 words and maximum of 4 words that occur at least 10 times and include both a noun and verb.

Support is provided to further filter retrieved objects. For example, a list of retrieved objects may be further narrowed by applying a regular expression to the object text (‘bob ran’), syntactic tag pattern (‘<nn>’/nn>
or both ("<nn>bob</nn><vb>ran</vb>"). Objects satisfying the regular expression are then processed by the application.

In addition, in order to produce normalized strings that are also persisted, the tool can optionally be linked to other applications. For example, in the present study we have plugged in the Lexical Variant Generator (LVG) [14] developed by the National Library of Medicine (NLM) so that queries could be performed on non-normalized and normalized content.

3.4. Tool evaluation

Our algorithm was tested to normalized and non-normalized words. Normalization (or lemmatization) is a process that strips inflectional endings and identifies one base form for its variants, e.g., “run” will be the normalized word for “run,” “running,” “runs,” “ran.” Testing against normalized and non-normalized frequencies shows to what extent terms vary in their forms. The frequency thresholds we experimented with are 2, 3, 5, and 7.

As an initial quality check to determine whether the computer-derived terms and the expert identified terms were derived from the same sentences, we randomly checked 25 terms (five at each ngram level). In all cases, the source of the computer-derived term coincided with the source sentence of the expert derived term.

We then compared the terms extracted by each of the three algorithms (linguistic, hybrid and experimental) against the test set at every term length. The results of our experimental algorithm are reported as “Experimental algorithm: non-normalized terms” and “Experimental algorithms: normalized terms.” To evaluate tool performance, we estimated recall and precision using the following formulas as discussed above. The F-measure (F-score) was estimated with a β value of one

Recall = \( \frac{\text{valid computer derived terms}}{\text{expert derived terms}} \),

Precision = \( \frac{\text{valid computer derived terms}}{\text{all computer derived terms}} \).

To estimate the actual noise, we estimated two additional measures:

Noise before expert validation

\[ \text{noise before expert validation} = \frac{\text{fp}}{\text{all computer derived terms}} \]

Noise after expert validation

\[ \text{noise after expert validation} = \frac{\text{fp after expert validation}}{\text{all computer derived most specific terms}} \]

3.5. Results

Table 1 presents detailed recall results by algorithm and ngram or term length (i.e., the number of content words in a term candidate). The results for the experimental algorithm are reported for both non-normalized and normalized terms. As was pointed out in Section 3.4., normalization (or lemmatization) shows the extent of term form variation. In addition to the frequency thresholds of 2, 3, 5, and 7 that we include in the algorithm, we also report on a frequency threshold of 0 as a more controlled comparison to the purely linguistic approach. The recall observed in the hybrid algorithm is particularly remarkable in that while comparable at the unigram level, a steep decline is observed as the term length increased beyond the unigram level. At the

<table>
<thead>
<tr>
<th>Term length</th>
<th>Linguistic approach</th>
<th>Hybrid approach</th>
<th>Experimental algorithm: non-normalized terms</th>
<th>Experimental algorithm: normalized terms</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Freq = 0</td>
<td>Freq = 2</td>
<td>Freq = 3</td>
<td>Freq = 5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>94%</td>
<td>85%</td>
<td>94%</td>
<td>85%</td>
</tr>
<tr>
<td>2</td>
<td>87%</td>
<td>40%</td>
<td>86%</td>
<td>84%</td>
</tr>
<tr>
<td>3</td>
<td>80%</td>
<td>17%</td>
<td>85%</td>
<td>83%</td>
</tr>
<tr>
<td>4</td>
<td>72%</td>
<td>10%</td>
<td>71%</td>
<td>68%</td>
</tr>
<tr>
<td>5</td>
<td>58%</td>
<td>0%</td>
<td>59%</td>
<td>59%</td>
</tr>
<tr>
<td>&gt;5</td>
<td>63%</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
</tbody>
</table>

*A constraint on term lengths >5 were imposed on the hybrid and experimental algorithms.*
pentagram level, recall for that algorithm was 0%. Across frequencies reported for the experimental algorithm, a decline in recall is observed but it is not steep as more nodes have a chance of being flagged as term candidates because their constituent content words by themselves pass the frequency threshold.

Table 2 first presents the summary recall, $F$-measure, precision, and percentage of valid terms in false positives for each of the three algorithms as well as the noise before and after expert validation. Recall is lower in the experimental algorithm than in the linguistic approach mainly due to the exclusion of words longer than 5 content words as our tool considers combinations up to 5 content words (pentagrams). The hybrid approach demonstrated dramatically lower recall than either the linguistic or experimental algorithm.

Precision is higher for our experimental approach than was observed in either the linguistic or hybrid approaches. Both the non-normalized and the normalized experimental algorithms have better $F$-measures overall (range is 0.18–0.23) than the linguistic or hybrid methods, although they are still low due to the low precision as $F$-measures combine the contributions of both recall and precision.

Because precision was lower than we expected for our experimental approaches, we conducted a failure analysis of the false positives, i.e., the terms that were identified by the algorithm but not by our expert reviewer. In our study, we asked the expert to mark "the most specific term" relevant to the domain of functioning, disability and health. We did not ask her to mark primitive terms that may have been nested within more specific terms. Our algorithm was constructed to identify nested primitives within complex phrases if they passed the two filters. We did not ask our expert to similarly identify nested terms. For example, multiword terms such as "ability to walk," "ability to bathe," and "ability to bathe max assist" were identified by both our clinical expert and the algorithm, but the expert did not additionally mark "ability" alone as a term. Our algorithm would identify both the single word term "ability" and the multiword terms such as "ability to walk" as terms. This contributed to the number of false positives we observed. Therefore, we presented the false positives to the expert for additional validation to determine whether they are true errors. The valid terms in false positives is 37% for the linguistic approach, 15% for the hybrid approach, and 30–34% for our experimental approaches.

The noise scores improved substantially after expert validation of false positives. Again, the difference between the linguistic and experimental approaches is within 4% points. The hybrid approach shows a considerably lower result than either the linguistic or experimental methods although we expect with a larger corpus, noise will be further reduced in the experimental algorithm.

Finally, we expected that normalized terms would perform better than the non-normalized terms in regard to both recall and precision. That is not supported by the results obtained in the current study. The most likely explanation is that candidate terms tend to appear in one preferred form rather than its variations within the source text, e.g., "upper extremities" is the preferred form occurring in the corpus rather than its normalized form "upper extremity" and changing to the normalized form would not affect frequencies.

### 4. Conclusions

A critical component of terminology development is term acquisition. As models related to the structure of nursing terminologies become stable (e.g., the ISO models), an expanded emphasis on the content of nursing terminologies will be needed. The results of this study suggest that the algorithm we present for term extraction contributes to this goal by providing a data-driven approach. Such an approach has great potential to enhance the scope of the nursing terminologies.

Importantly, the linguistic filter we introduce extends previous research [21] by going beyond nominal expressions and node-internal syntactic patterns and applies the notion of a "complete syntactic node" in order

### Table 2
Summary results for recall, precision, and $F$-score (detailed results are in Table 1)

<table>
<thead>
<tr>
<th></th>
<th>Linguistic approach</th>
<th>Hybrid approach</th>
<th>Experimental algorithm: non-normalized terms</th>
<th>Experimental algorithm: normalized terms</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Freq = 0</td>
<td>Freq = 2</td>
<td>Freq = 3</td>
<td>Freq = 5</td>
</tr>
<tr>
<td>Recall</td>
<td>77%</td>
<td>24%</td>
<td>71%</td>
<td>69%</td>
</tr>
<tr>
<td>Precision</td>
<td>8%</td>
<td>12%</td>
<td>10%</td>
<td>11%</td>
</tr>
<tr>
<td>$F$-Measure</td>
<td>0.14</td>
<td>0.16</td>
<td>0.18</td>
<td>0.18</td>
</tr>
<tr>
<td>Valid terms in FP</td>
<td>37%</td>
<td>15%</td>
<td>30%</td>
<td>32%</td>
</tr>
<tr>
<td>Noise before expert validation</td>
<td>92%</td>
<td>88%</td>
<td>90%</td>
<td>89%</td>
</tr>
<tr>
<td>Noise after expert validation</td>
<td>41%</td>
<td>65%</td>
<td>41%</td>
<td>40%</td>
</tr>
</tbody>
</table>
to accommodate verb phrases as well as noun phrases. The frequency-based filter we introduce extends previous research [21] by applying the frequency filter to the constituents of the lexical string under consideration. The algorithm appears to adequately take care of low frequency multi-word combinations as it relies on the frequency of any content word constituent within a given n-gram to pass the frequency cutoff. In that, our algorithm offers a piece of the solution to the limitations of frequency-based approaches. However, the algorithm does not extract low frequency n-grams in which there is no single constituent that passes the frequency threshold.

While we are pleased that the recall and precision statistics did not decline with these algorithm expansions, we were hopeful they would have improved more. It is likely that the precision results we report may be artificially low as a consequence of the way we instructed the expert to markup the “most specific terms” and not their primitive constituents. Our algorithm was constructed to accommodate the nesting of primitives within complex phrases. Single word terms tend to represent more general concepts, while multi-word terms are required for more specific concepts. As we continue to evaluate this tool on a larger corpus with more heterogeneous text sources, it will be essential that experts mark up the text in at least two ways, one looking for most specific terms, and the other looking for nested terms within complex phrases. We are also investigating the use of term ranking [23] and semantic clustering [25] as ways of decreasing noise.

Limitations of the evaluation we have reported are recognized. First, the test set is relatively small and is not representative of the entire domain of functioning, disability, and health. The small size of the test set has effects on determining the frequency cutoff. With a larger corpus, it is expected that the threshold would be increased. We also expect that the difference between the linguistic approach and our experimental approach would widen with a larger test set, as the linguistic approach would extract every noun and verb phrase indiscriminately of their frequencies thus producing more noise.

A larger number of clinical domain experts will be required to confidently evaluate the reliability of terms acquired using this tool, or we will need to investigate alternative methods to evaluating the algorithm. Creating large, expert-tagged test corpora is a time-consuming and cognitively intense task. Thus, a multi-site collaborative initiative for creating test beds would provide researchers with a robust gold standard.

A final limitation is the lack of comparison against a purely statistical approach such as the use of mutual information or log likelihood. Such a comparison is one of our future goals. Our intent here was to conduct an initial evaluation of the algorithm, and to determine whether more ambitious efforts in this direction are merited.

The broad goal of this research is to develop automated approaches to expanding the content within the domain of functioning, disability, and health. We believe a classification system such as the ICF can be integrated with the nursing terminologies through the focus and target axes of the ISO structure. To that end, we are collaborating with a colleague who is developing a “top-down” frames-based representation of the ICF [24]. We are hopeful that we will be able to export candidate terms to that application, and exploiting the nesting feature of our algorithm, auto-classify terms. Similarly, we plan to investigate the placement of terms in systems such as the SNOMED CT and the UMLS.

In addition to increasing the size and representativeness of the text corpus, future work will include the investigation of other metrics besides frequency as thresholds, e.g., log likelihood and mutual information [20], in order to make the algorithm robust to processing larger corpora and relatively independent of empirically determined thresholds. Term clustering methods will be examined as way to handle synonyms, and near-synonyms [25].

Finally, we have as a goal the development of a web-based interface that will make this tool available to nursing terminology developers in order to bring a data-driven approach to the development of those systems.

Acknowledgments

This research for this paper was supported in part by Grant LM07453 from the National Library of Medicine (Harris PI). Our thanks to Lori Rhudy, a true clinical expert who spent many hours tagging text. Special thanks to Jim Buntrock, Sergey Pakhomov, Harold Solbrig, and members of the Division of Medical Informatics Research who provide valuable critique at Thursday seminars.

References
