A Normalized Lexical Lookup Approach to Identifying UMLS Concepts in Free Text

Vijayaraghavan Bashyam\textsuperscript{a}, Guy Divita\textsuperscript{b}, David B Bennett\textsuperscript{a}, Allen C Browne\textsuperscript{b}, Ricky K Taira\textsuperscript{a}

\textsuperscript{a} Medical Imaging Informatics Group, University of California, Los Angeles, United States
\textsuperscript{b} National Library of Medicine, National Institutes of Health, United States

Abstract

The National Library of Medicine has developed a tool to identify medical concepts from the Unified Medical Language System in free text. This tool – MetaMap (and its java version MMTx) has been used extensively for biomedical text mining applications. We have developed a module for MetaMap which has a high performance in terms of processing speed. We evaluated our module independently against MetaMap for the task of identifying UMLS concepts in free text clinical reports. A set of 1000 sentences from neuro-radiology reports were collected and processed using our technique and the MMTx Program. An evaluation showed that our technique was able to identify 91% of the concepts found by MMTx in 14% of the time taken by MMTx. An error analysis showed that the missing concepts were largely those which were not direct lexical matches but inferential matches of multiple concepts. Our method also identified multi-phrase concepts which MMTx failed to identify. We suggest that this module be implemented as an option in MMTx for real-time text mining applications where single concepts found in the UMLS need to be identified.

Keywords:
MetaMap, natural language processing, text mining, UMLS

Introduction

The recent past has seen a phenomenal growth in the amount of textual information generated in various areas of biomedicine such as clinical reports associated with electronic medical records and scholarly publications [1],[2]. There have been repeated calls for the development of text mining and natural language processing tools to extract information from the biomedical literature [3]. To cater to such needs, the National Library of Medicine (NLM) has developed several open-source text processing tools for the biomedical domain. The MetaMap Program is a tool developed at the NLM to map free text to concepts in the Unified Medical Language System Metathesaurus (UMLS) [4]. MetaMap and its java version MMTx, henceforth referred synonymously as MM, have been extensively used for a wide variety of biomedical text mining applications. We have developed a technique for fast and accurate mining of whole UMLS concepts. This is one of the applications for which MM has been used extensively, but for which MM is considered to be slow [5]. We report the details of this system and suggest that it be implemented as an optional module in future releases of the MM toolkit.

Background

In order to get an understanding of the various scenarios in which MM is being applied, we conducted a literature review to identify all reports of MM usage in the past three years. We identified 8, 13 and 9 articles within the proceedings of MEDINFO 2004, AMIA 2005 and AMIA 2006 respectively, all of which directly used MM for their respective tasks. We ignored the articles which made secondary use of MM via other NLM applications like SemRep. We also did not analyze journal articles since conference publications often get extended to journal articles.

Recent uses of MetaMap

Recently, MM has been used for several different applications both within and outside the NLM. Whalen et al. used MM to identify UMLS concepts for use in medical textbook summarization [5]. Mary et al. report the usage of MM for identifying genes and proteins in scientific articles related to molecular biology [6]. Chapman et al. used MM to index emergency department reports by UMLS terms related to respiratory findings [7]. Hsieh et al. tried to identify UMLS terms used by patients in email messages using MM [8]. Kim et al. indexed descriptive clinical text from a web-based repository of dermatology cases to MeSH terms using MM[9]. Bernhardt et al. identified diagnoses and causes of death and mapped them to MeSH terms using MM [10]. Lacson et al. identified nouns and replaced them with the UMLS semantic labels through MM [11]. Wedgwood used MM to map medical terms to the UMLS as a module within a medical question answering system [12]. Gay et al. indexed the title and abstracts of biomedical articles using MM [13]. Niu et al. identified UMLS semantic categories for analyzing polarity information in medical text [14]. Kahn et al. annotated radiology images with patient demographics and UMLS concepts generated using MM for a digital searchable library [15]. Mirhaji et al. used MM to identify single UMLS concepts found in chief complaint data [16]. Silfen et al. processed biomedical publications using MM to identify UMLS con-
cepts [17]. Peace et al. used MM to index concepts in
nursing procedure manuals [18]. Meyestre et al. customized
the target dataset of MM adapted to 80 targeted medical
problems and used it to identify a predefined list of
problems [19].

Baud et al. used MM to analyze the variability of Medical
terms [20]. Leroy et al. tried to disambiguate word senses
using UMLS concepts identified by MM and their semantic
types[21]. Hagerty et al. processed clinical guidelines
using MM [22]. Tringali et al. used MM to process gas-
trointestinal endoscopy reports [23]. Meng et al. identified
anatomy-related concepts from surgical reports with MM
[24]. Coonan coded clinical narratives to UMLS concepts
using MM [25]. Yetisgen-Yildiz et al. used MM as a
phrase chunking tool [26]. Zhou et al. identified UMLS
concepts in documents related to five health domains using
MM [27]. Crowley et al. used MM as a coding module of
an intelligent tutoring system [28]. Lieberman et al. used
MM to extract concepts from chief complaints entered into
an ambulatory medical electronic medical record [29].
Wang et al. report the use of MM as an identification tool
for medical terms which are later manually labeled for an
information retrieval system [30]. Ogren et al. tried to
annotate corpora using UMLS semantic types using MM
[31]. Patel et al. attempted to mine cross terminology links
in the UMLS and used MM to process their dataset [32].
Hagerty et al. used MM to identify UMLS concepts and
associated concepts for encoding clinical text in a markup
language [33]. Ruiz used MM to first identify English con-
cepts and then translate them to French [34].

Need for a new module
Our examination indicated that 15 [5-19] of the 30 men-
tioned reports used MM primarily to identify whole
concepts in the UMLS. This task does not require some of
the sophisticated processing modules present in MM
which are computationally complex and hence time con-
suming. By making a reasonable assumption that this set
of articles accurately reflects the distribution of MM usage
in the medical informatics community, we safely conclude
that around 50% of MM applications only seek whole
UMLS concepts and not multiple concept mappings. A
method bypassing the time-consuming modules of MM
would be very suitable for these applications. Our exper-
iment was a step toward this goal.

Materials and methods
Our methodology consisted of four stages — 1) Identifying
the data to be processed our method and by MM, 2) Pro-
cessing the data by our method, 3) Processing the data with
MMTx and, 4) A comparative evaluation of the two meth-
ods. The details of the individual stages follow:

Data Collection
Collection of the sentences to be indexed to the UMLS
was the first stage in our experiment. We followed the
steps outlined below to collect sentences.

Step 1: A set of one thousand radiology reports related to
the neuro-radiology domain were collected from the exist-
ing hospital database at our institution.

Step 2: Section boundary detection was performed on the
reports to break up a report into individual sections such as
HEADER, HISTORY, FINDINGS, CONCLUSIONS etc.
Next, sentence boundary detection was performed on the
sections. Both of these modules have been previously
tested, with recall and precision accuracies of over 99%
within the domain of radiology [35].

Step 3: A lexical analyzer processed each sentence per-
forming tokenization, part-of-speech tagging, semantic
class tagging and word-sense disambiguation. Our lexicon
categorizes tokens into twenty syntactic categories and
over eight hundred semantic categories.

Step 4: We identified a list of the primary semantic classes
which occur in radiology reports, such as anatomy, find-
ings, procedures, equipment etc. Using a high-recall
sentence-level parser, all target sentences were conserva-
tively identified while sentences that obviously have no
instances of our target semantic types were rejected. To
illustrate, the presence of the semantic type phys-
obj-anatomy (e.g. brain) in a sentence would
automatically result in the sentence being accepted as con-
taining a possible UMLS concept. This step is necessary so
that we do not try to find UMLS concepts in sentences
which might not have any. Thus, we chose to process the
sentences that are highly likely to contain UMLS concepts.

Step 5: From the target set of sentences, we randomly sam-
ped 1000 sentences for processing by our technique and
by MM.

Identifying UMLS concepts
We previously prepared a normalized word index of the
entire UMLS concepts tagged as English concepts and
serialized the index in a database. This index is similar to
the index used by MM, GSpell and other such tools de-
veloped at the NLM. Each sentence was preprocessed using
the ‘LuiNorm’ stemming tool developed at the NLM to
generate a sentence normalized at the word level [36]. We
identified a list of and word types words types such as con-
junctions which, when normalized are likely to give false
positive errors. These words were intentionally not
normalized.

Each normalized sentence was processed by our module to
identify all UMLS concepts by referring to the index. The
method used to identify the concepts is the longest sub-
string matching algorithm developed by Aho and Corasick. Details of this algorithm can be found in the
original publication [37]. The algorithm finds a list of
longest substrings according to the parse order (left to right
in this case) given a lengthy string. Thus, this method will
only identify a list of lexically matching concepts but not
attempt to find any partial matches, combinatorial
matches, etc. The output of this module for an example
sentence is shown below.
Discussion

Error analysis

False positives occur for several reasons. First, the UMLS Metathesaurus contains several concepts which are orthographically identical but have different concept unique identifiers (CUIs). When such concepts are encountered, our system adopts an ad hoc method of choosing one of the multiple mappings. An example of such a false positive would be the verb am mapped to the concept am (ante-meridiem).

Other types of false positives occur due to the natural ambiguity introduced when words are stemmed. Multiple words when normalized to the same stem increases the ambiguity. Replacing the stemmed word index by a regular word index can result in an increase in precision but a possible drop in recall. To illustrate, plural forms of words would still map to the same concept when normalized but would not when indexed directly. These types of errors could be potentially reduced by using a regular word index in addition to a normalized word index.

Table 1 – Results of Evaluation

<table>
<thead>
<tr>
<th></th>
<th>MMTx</th>
<th>Normalized Lexical Lookup</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of Concepts Identified</td>
<td>6859</td>
<td>6513</td>
</tr>
<tr>
<td>Large Phrases</td>
<td></td>
<td>+173</td>
</tr>
<tr>
<td>False Positives</td>
<td></td>
<td>-457</td>
</tr>
<tr>
<td>Total</td>
<td>6859</td>
<td>6229</td>
</tr>
</tbody>
</table>

Multiple phrasal concepts

Since we do not use a shallow parser to chunk sentences into phrases, we are able to identify concepts spanning across multiple phrases. An example of this is the phrase liver and spleen which exists as a concept in the UMLS (C08545797) but is not recognized by MM as a single concept returning the individual concepts liver (C0023884) and spleen (C0037993). Our method is able to return the whole concept liver and spleen. While it can be argued that failure to separate the coordinated concepts is an undesirable feature, we would like to stress that our system is also not intended to correct errors present in the UMLS. A retrieval system is only as good as the content it is retrieving. Though our method is intended to be as close to MM output, it is independent of, and can be used with any lexicon other than the UMLS. The system only needs a list of concepts with corresponding unique identifiers and can perform efficient lookup. Thus, as concepts are removed from or added to each new release of the UMLS, we don’t have to make changes to our system.
Methodology differences
Our technique differs from the MM approach in several ways:

1. MM tokenizes a sentence into terms and phrases whereas our method only tokenizes sentences into words.

2. To identify phrases, MM requires the services of a part-of-speech (POS) tagger to first tag the words or terms, and a shallow parser to use the tags to mark phrase boundaries and phrase head assignments. Our system bypasses the POS tagging and shallow parsing component.

3. To identify terms, MM performs a lexical lookup within the sentence to find the longest spanning terms from the SPECIALIST Lexicon. Our method does a longest spanning match to a normalized word index of the UMLS, bypassing the term tokenization and lexical lookup into the SPECIALIST Lexicon. One of the benefits to this approach is the potential of retrieving multi-phrase concepts as demonstrated previously.

4. MM has several modules to find a) matches to the terms within phrases, b) generate spelling and inflectional variants, synonyms, derivations, acronyms, abbreviations and expansions of the words in the phrase, and a recursive combination of these, to make the match. Our system does not have any of these modules.

5. MM categorizes the retrieved matches as partial matches, over matches, and matches with gaps to the phrase. This categorization requires an alignment mapping technique that is at the heart of MM. This is a technique that is computationally expensive. By default, MM filers out over-matches and matches made with concept gaps, to reduce the noise caused by such matches. Our system does not have this alignment mapping technique.

6. MM includes a ranking mechanism to evaluate the quality of the matches. It ranks ‘exact’ and ‘near exact’ matches higher, and ‘partial matches’ with matches made on the basis of some form of lexical variation, lower. Our method does not include a ranking of the matches returned but returns a fixed list of concepts identified in a sentence.

Our method does not return matches made with gaps, but could return partial matches. Because of these differences, it is likely that our method will do a better job than MM at quickly finding exact and normalized exact matches within a sentence. However, since our method lacks the variant generation and ranking capabilities, it is likely to miss matches based on non-inflationary lexical variation or because of word order differences. Our method will do a better job retrieving multi-phrase concepts. It is an open question whether phrase spanning concepts occur more often in a target corpus than concepts that could only be matched via non-inflationary variation, and whether the added precision is worth the extra computational costs. These issues may be highly dependent on specific domains.

Scope for improvement and future work
Previous internal studies at the NLM have shown that over 95% of the time taken by MM in processing a sentence is due to the alignment mapping algorithm. Since we avoid the alignment mapping module, the shallow parser and the part-of-speech tagger, theoretically our method should perform much faster than its current performance. We took no particular effort in optimizing our implementation of the Aho-Corasick algorithm. Our aim was to show that for specific applications, it is possible to achieve results similar to that of MM using simpler modules and by avoiding the computationally complex modules. Thus, an efficient implementation has a possibility to be a lot faster than our current system. Additionally, there are methods other than the Aho-Corasick algorithm which could be used for finding the largest substrings. The performance of those methods for this particular application needs to be explored. Finally, a more extensive evaluation of MM versus our method needs to be conducted to arrive at conventional precision, recall and f-score measures.

Conclusion
A module for MetaMap is proposed with focus on high processing speed suitable for real-time applications. This module uses components already being used in MetaMap with minor changes. This module is likely to have multiple useful applications in medical informatics.

Acknowledgments
This research was supported in part by an appointment of the first author to the NLM Research Participation Program.

References


Address for Correspondence
Vijayaraghavan Bashyam
vbashyam @ ucla.edu