MetaMap 2014 Release Notes

August 10, 2014

1 Windows XP no Longer Supported

Microsoft ended support for Windows XP on April 8, 2014:

`http://www.microsoft.com/en-us/windows/enterprise/end-of-support.aspx`

The Windows version of MetaMap is now compiled and supported on Windows 7 only.

2 WSD Server Efficiency

Ron Katriel of Medidata Solutions pointed out an inefficiency in the WSD Server, which ran progressively slower when processing a long file containing mostly short terms.

The handling of requests to the WSD Server has been moved to a separate thread for each request. Moreover, the thread is discarded after handling the request is complete; consequently MetaMap with WSD runs over 50 times faster on lists of short terms.

3 No Display of Candidate Concepts by Default

MetaMap's default output has historically included both Candidate Concepts, which are intermediate results, and Final Mappings. In order to encourage users to use MetaMap's Final Mappings (and not the Candidate Concepts) for their research, beginning with MetaMap2014, Candidate Concepts will by default not appear in any MetaMap output (human-readable, Prolog Machine Output (MMO) or XML). Users still wanting to see the candidates should use the `show_candidates` (`-c`) option. A by-product of this change is the retirement of the `hide_candidates` option.

The following examples of MetaMap output will all be based on the input `heart attack`.

3.1 Human-Readable Output

MetaMap's default human-readable output will no longer include the candidates section appearing in red below:

```
Phrase: heart attack
Meta Candidates (Total=6; Excluded=0; Pruned=0; Remaining=6)
  1000   Heart attack (Myocardial Infarction) [Disease or Syndrome]
   861   Heart [Body Part, Organ, or Organ Component]
   861   Attack, NOS (Onset of illness) [Temporal Concept]
   861   attack (Attack behavior) [Social Behavior]
   861   Heart (Entire heart) [Body Part, Organ, or Organ Component]
   861   Attack (Observation of attack) [Finding]
Meta Mapping (1000):
  1000   Heart attack (Myocardial Infarction) [Disease or Syndrome]
```

3.2 Prolog Machine Output (MMO)

In MetaMap's default MMO, the candidates term will appear simply as

```
candidates(6,0,0,6,[]).
```

instead of (pretty-printed and condensed)

```
candidates(6,0,0,6,[
  ev(-1000,'C0027051','Heart attack','Myocardial Infarction',[heart,attack],
     [dsyn],[[[1,2],[1,2],0]],yes,no,['AOD','CHV','CSP','CST'], [0/12],0,0),
  ev(-861,'C0018787','Heart','Heart',[heart],
     [bpoc],[[[1,1],[1,1],0]],yes,no,['AOD','CHV','CSP','FMA'],[0/5],0,0),
  ev(-861,'C0277793','Attack, NOS','Onset of illness',[attack],
     [tmco],[[[2,2],[1,1],0]],yes,no,['AOD','CHV','CSP','MTH'],[6/6],0,0),
  ev(-861,'C1261512',attack,'Attack behavior',[attack],
     [socb],[[[2,2],[1,1],0]],yes,no,['AOD','CHV','MTH'],[6/6],0,0),
  ev(-861,'C1281570','Heart','Entire heart',[heart],
     [bpoc],[[[1,1],[1,1],0]],yes,no,['MTH','SNOMEDCT_US'],[0/5],0,0),
  ev(-861,'C1304680','Attack','Observation of attack',[attack],
     [fndg],[[[2,2],[1,1],0]],yes,no,['CHV','MTH','SNOMEDCT_US'],[6/6],0,0)]).
```

3.3 XML Output

In MetaMap's default XML output, the Candidates element will appear simply as

```
<Candidates Total="6" Excluded="0" Pruned="0" Remaining="6" />
```

rather than as

```
<Candidates Total="6" Excluded="0" Pruned="0" Remaining="6">
 <Candidate>
  . . .  XML for candidate  . . .
 </Candidate>
 <Candidate>
  . . .  XML for candidate  . . .
 </Candidate>
   . . .
</Candidates>
```

4 Minor Syntax Change in Machine Output (-q)

In MetaMap's Prolog Machine Output (MMO), the AAs term is now displayed as

`aas(["CF"*"cystic fibrosis"*(1,2,3,15)*['C0010674']])`

instead of

`aas(["CF"*"cystic fibrosis"*[1,2,3,15]*['C0010674']])`

(with square brackets instead of parentheses surrounding the token- and character counts of the short and long form of the acronynm/abbreviation).

5 Additional Data Available in Fielded MMI Output (-N)

MetaMap's Fielded MMI Output, which is produced by the `fielded_mmi_output` (`-N`) option, has historically looked like the following, for the input text `heart attack`:

```
00000000|MM|14.64|Myocardial Infarction|C0027051|[dsyn]|
   ["Heart attack"-tx-1-"heart attack"]|TX|0:12|
   C14.280.647.500;C14.907.585.500
```

2013 MMI (indented for readability)

Beginning with MetaMap14, each concept will be represented by a 6-tuple instead of the usual 4-tuple; the two additional fields shown below are the Lexical Category and Negated Status:

```
00000000|MMI|14.64|Myocardial Infarction|C0027051|[dsyn]|
   ["Heart attack"-tx-1-"heart attack"-noun-0]|TX|0:12|
   C14.280.647.500;C14.907.585.500
```

2014 MMI (indented for readability)

6 New MetaMap Source Vocabulary

The 2014 MetaMap data include an experimental synthetic vocabulary that is not in the UMLS and is designed to increase recall. The new vocabulary, called *NLMSubSyn* is intended to fill gaps in the UMLS' synonymy via subterm synonymy, or *subsynonymy*.

The kind of synonymy with which we are supplementing the UMLS' built-in synonymy can be understood from the following canonical examples:

Although *geriatric* and *elderly* are synonyms in C1999167 and *geriatric patients* is a string in C0870602, *elderly patients* does not occur in the UMLS, even though that string occurs over 40,000 times in PubMed. We therefore include *elderly patients* in the NLMSubSyn synthetic vocabulary as part of CUI C0870602 (preferred name *Geriatric patients*).

Similarly, although *evaluation* and *assessment* are synonyms in C1261322 and *clinical evaluation* is a string in C1516634, *clinical assessment* is not in the UMLS, even though that string occurs over 16,000 times in PubMed. We therefore include *clinical assessment* in NLMSubSyn as part of CUI C1516634 (preferred name *research clinical testing*).

Users wishing to exclude the NLMSubSyn vocabulary can easily do so via the command-line option `-e NLMSubSyn`.

7 Dynamic NegEx Triggers

MetaMap has for several years included an implementation of Wendy Chapman's NegEx negation-detection algorithm, which is documented at http://code.google.com/p/negex. NegEx is invoked from command-line MetaMap simply by using the `--negex` option.

Beginning with MetaMap2014, users will have the ability to custom-craft NegEx triggers. The `public_mm/resources` MetaMap download directory contains a file called `negex_triggers.pl`, which encodes the Prolog representation of the NegEx triggers. Simply copy that file to a working directory, e.g., `/home/mydir/MetaMap/my_negex_triggers.pl`, and edit it as desired.

Sample Prolog clauses are

```
nega_phrase_tokens(absence, [of]).
nega_phrase_tokens(cannot, []).
nega_phrase_tokens(cannot, [see]).
nega_phrase_tokens(checked, [for]).

negb_phrase_tokens(have, [been,ruled,out]).
negb_phrase_tokens(has, [been,ruled,out]).
```

No detailed knowledge of Prolog syntax is required. simply ensure that

- any clauses added mirror the syntax of their neighbors,
- all clauses defining a given trigger type, e.g., `nega_phrase_tokens`, are contiguous, and

- all tokens in the definitions are lowercase and contain no punctuation.

In order to comment out a clause, simply place a `%` in front of it, e.g.,

```
% negb_phrase_tokens(have, [been,ruled,out]).
% negb_phrase_tokens(has, [been,ruled,out]).
```

Finally, call MetaMap with

```
--negex --negex_trigger_file /home/mydir/MetaMap/my_negex_triggers.pl
```

A message will be printed to standard output such as

```
###############################################################################
###### Loading NegEx trigger file /home/mydir/MetaMap/my_negex_triggers.pl #####
###############################################################################
```

and the new trigger file will be used in that run of MetaMap.

Note: The modified NegEx triggers will not be compiled into the MetaMap binary; the file defining them must be specified every time MetaMap is called.