Reproducing MetaMap Evaluation: First 100 FDA Labels

June 16, 2016

François-Michel Lang

flang@mail.nih.gov

This note presents instructions for reproducing MetaMap's Linux processing of the first 100 FDA labels. In case FDA chooses to not download and run MetaMap locally, we have provided NLM's MetaMap output and evaluation results for all 100 drug labels, as described below in items 4 and 5 of section 2.1.

All our processing uses MedDRA 18.0, which we received by e-mail from Taxiarchis Botsis on June 18, 2015, and the 2016AA version of the UMLS.

Although this document contains considerable detailed information, a vast majority of it is explanatory only—very few steps are required to replicate NLM's results once the download and installation explained in Sections 1 and 2 are complete. Throughout this document, all steps to be followed are shown in **boldface red**; Linux commands to be run are also shown in `red Courier font`, and indicated with "*".

1 Download and Install MetaMap Locally

Reproducing NLM's results requires (a) obtaining a UMLS Metathesaurus license, and then (b) downloading and installing MetaMap locally on a Linux machine.

1.1 Obtain UMLS Metathesaurus License

To request a UMLS Metathesaurus license, **visit the page**

https://uts.nlm.nih.gov/license.html

and **follow the instructions on that page**. Once your request has been approved, you will be able to download and install MetaMap.

1.2 Download and Install MetaMap

To download MetaMap, first **review the MetaMap prerequisites** at

https://metamap.nlm.nih.gov/MetaMap.shtml

to ensure your Linux environment is adequately configured. In addition to the MetaMap prerequisites, **verify that the Linux host has Perl available at `/usr/bin/perl`**.

Then **visit**

https://metamap.nlm.nih.gov/MainDownload.shtml

and **select "MetaMap 2016 Linux Version"** to download the public MetaMap distribution.

Finally, **follow the installation instructions**, which are available at

https://metamap.nlm.nih.gov/Installation.shtml

When the installation is complete, MetaMap will be installed in a directory called `public_mm`. We will refer to that `public_mm` directory as `$METAMAP`.

*`cd` to the `$METAMAP` directory

*`setenv METAMAP $cwd`

2 Download and Unpack Repository

A Github repository containing necessary programs and data beyond the MetaMap installation is available at

https://github.com/lhncbc/fda-ars

2.1 Contents of Unpacked Directories

The `MetaMapFiles` entry in the repository contains seven directories whose contents are the following:

1. `Brat`: files with an `ann` extension, containing the human annotations of the `txt files` in brat standoff format.[1] The naming of the `ann` files follows the pattern `DRUGNAME_section`, e.g.,
   - `VORAPAXAR_adverse_reactions.ann`
   - `VORAPAXAR_boxed_warnings.ann`
   - `VORAPAXAR_warnings_and_precautions.ann`
2. `DATA`: Four data files that were created using MedDRA 18.0:[2]
   (a) `CUI.MedDRA.PT`: Mappings from UMLS CUIs to MedDRA PTs using only rows in the Metathesaurus `MRCONSO.RRF` file with a `PT` Term Type.
   (b) `CUI.MedDRA.PT.LLT`: Mappings from UMLS CUIs to MedDRA PTs using rows in the Metathesaurus `MRCONSO.RRF` file with a `PT` or `LLT` Term Type.
   (c) `CUI.MedDRA.HT.PT.LLT`: Mappings from UMLS CUIs to MedDRA PTs using rows in the Metathesaurus `MRCONSO.RRF` file with a `HT`, `PT` or `LLT` Term Type.

[1] See `http://brat.nlplab.org/standoff.html`.

[2] Instructions and programs for creating these files can be provided if desired.

   (d) `StopList`: A list of terms often found by MetaMap, but judged by the annotators to be too general to be reliably considered an adverse reaction. Some terms on the stop list are: `abnormalities`, `activity`, `adverse event`, `adverse reaction`, and ≈100 others.

   The use of the `CUI.MedDRA` files is covered in Section 4.2.

3. `Gold`: files with an `offsets` extension, containing the relevant annotations and text offsets extracted from the annotations files in the `Brat` directory. These files are our gold standard, against which MetaMap's results are compared. Lines in these files are of the form

   `Offsets|UMLS CUI|UMLS String|MedDRA PT`

   e.g.,

   `2407,2420|C0020456|hyperglycemia|10020635`.

   Note that the offsets can contain multiple `StartPos,EndPos` pairs, e.g.,

   `2422,2431;2456,2459|C0151904|increased AST|10003481`.

4. `MetaMap.NLM`: Files generated by NLM and provided for comparison purposes; they include (a) files with an `MMI` extension, containing the output generated by MetaMap[3] when run on the files in the `Txt` directory and (b) files with an `offsets` extension, containing the relevant UMLS concepts identified by MetaMap with their offsets. The offsets files contain lines of the form

   `Offsets|UMLS CUI|UMLS String|UMLS Semantic Type(s)`

   e.g.,

   `1004,1014;1030,1039|C0009691|congenital cataracts|cgab`

   The files in this directory are provided in case FDA chooses to not download and run MetaMap locally.

5. `Results.NLM`: Final results generated by NLM, again provided for comparison purposes. See Section 3.3 for more information.

6. `Txt`: files with a `txt` extension, containing the text extracted from the XML labels. The naming convention is the same as for the files in the `Brat` directory, e.g.,

   - `VORAPAXAR_adverse_reactions.txt`
   - `VORAPAXAR_boxed_warnings.txt`
   - `VORAPAXAR_warnings_and_precautions.txt`

7. `bin`: All the executable programs needed to reproduce NLM's results.

[3] The output format is Fielded MetaMap Indexing (MMI) Output, which is described in great detail at `https://metamap.nlm.nih.gov/Docs/MMI_Output_2016.pdf`.

2.2 Replace MetaMap Executable

This section is not necessary if you have downloaded MetaMap16V2; it *is* necessary for MetaMap16.

In order to best process the 200 FDA labels, we developed special MetaMap functionality that is not yet available in the production MetaMap binary; we are therefore providing a development version of MetaMap that includes this functionality, which will be included in the next public MetaMap release. The next two commands will replace the default MetaMap with the development version.

```
*cd $LABELS
```

```
*./bin/replace_binaries
```

3 Running Scripts

This section explains the steps necessary to reproduce NLM's MetaMap results.

3.1 Run MetaMap

If you have downloaded and installed MetaMap and are able to run it locally, follow the instructions in this section; otherwise, skip this section and simply use the MetaMap results generated by NLM, which are found in the `MetaMap.NLM` directory, described above in item 4 of section 2.1. To continue, run these two commands:

```
*cd $LABELS
```

```
*./bin/run_MetaMap
```

This script will

1. call MetaMap on all the text files in the `Txt` directory for which a non-empty annotation file exists in the `Brat` directory. This script will generate a large amount of output, which will be shown on the screen and preserved in the file `typescript.MetaMap`. If the host Linux machine has N>3 processors, the `run_MetaMap` script will run N-2 parallel MetaMap processes. Using six 2.13GHz processors, MetaMap took about fifteen minutes to run 236 files at NLM; thirty 2.13GHz processors required only three minutes.

2. remove any existing `$LABELS/MetaMap` directory, create a new `MetaMap` directory, and move all the `MMI` MetaMap output files described above in item 4 of Section 2.1 to the new `MetaMap` directory;

3. and finally, generate the `offsets` files also described above in item 4 in the new `MetaMap` directory.

Once MetaMap finishes, run

```
*diff MetaMap MetaMap.NLM
```

to compare the output just generated by MetaMap to the MetaMap output generated at NLM. If the call to `diff` generates no output, your run of MetaMap was consistent with NLM's. If `diff` does generate output, please let us know so we can investigate.

3.2 Evaluate MetaMap Results

This section explains how to evaluate the performance of MetaMap against the Gold Standard. Begin by running these two commands:

```
*cd $LABELS
```

```
*./bin/gen_results
```

This script will

1. for each drug file, compare the `offsets` file in the `Gold` and `MetaMap` directories and generate True Positives (TPs), False Positives (FPs), and False Negatives (FNs) for the drug label file in a newly created `Results` directory, and
2. compute the Precision, Recall, and F_1 for each drug file, and the overall micro average P/R/F_1 across all files.

Once `gen_results` finishes, run

```
*diff Results.0 Results.NLM
```

to compare the output just generated to the `gen_output` output generated at NLM. If the call to `diff` generates no output, your run of `gen_results` was consistent with NLM's. If `diff` does generate output, please let us know so we can investigate.

3.3 Files in the `Results` Directory

The `gen_results` script will create a directory called `Results.N` containing the results for each drug label file, as well as overall results. The value of `N` is 0, 1, 2, or 3, depending on the match strictness, which is explained in Section 4.1.

The `Results.N` directory will contain an overall results file `OVERALL.N` which is pipe-separated, and therefore suitable for loading into Excel; the fields in each line are

```
N|Drug file|GS|MM|CUIDiffs|MedDRADiffs|TPs|FPs|FNs|Precision|Recall|F1
```

where

- `N` is the line number,
- `GS` is the total number of Gold Standard annotations,
- `MM` is the total number of MetaMap annotations,
- `CUIDiffs` and `MedDRADiffs` are explained below in Sections 3.3.4 and 3.3.5.

The remaining fields are self-explanatory. A sample line from the `OVERALL.N` file is

```
232|BELIMUMAB_adverse_reactions|53|54|4|0|52|2|1|96.30|98.11|97.20
```

For each drug label file, the `gen_results` script will also create a file in the `Results.N` directory named `DRUGNAME_section.Results.N`, e.g.,

```
IVACAFTOR_warnings_and_precautions.Results.0
```

Each label-specific results file will contain lines identifying TPs, FPs, FNs, CUIDiffs, and MedDRADiffs, all of which are explained next in Sections 3.3.1–3.3.5.

3.3.1 TPs

True Positives are represented as lines such as

```
462,467|TP|edema|C0013604|10030095|462,467|edema|C0013604|10030095|fndg
```

whose schema is

```
Gold Offsets|TP|Gold Text|Gold CUI(s)|Gold MedDRA PT(s)|
    MetaMap Offsets|MetaMap Text|MetaMap CUI(s)|MetaMap MedDRA PT(s)|SemType(s)
```

3.3.2 FPs

False Positives are represented as lines such as

```
80,86|FP|asthma|C0004096|10003553|dsyn
```

whose schema is

```
MetaMap Offsets|FP|MetaMap Text|MetaMap CUI(s)|MetaMap MedDRA PT(s)|SemType(s)
```

3.3.3 FNs

False Negatives are represented as lines such as

```
10780,10788|FN|bleeding
```

whose schema is

```
Gold Offsets|FN|Gold Text
```

Finally, the `gen_results` script also records CUIDiffs, which are TPs for which the Gold and MetaMap results contain different CUIs, and MedDRADiffs, which are TPs for which the Gold and MetaMap results contain the same CUI, but different MedDRA PTs.

3.3.4 CUIDiffs

CUIDiffs are represented as lines like

```
71,76|CUIDiff|fatal|C1306577|10011906|71,76|fatal|C1705232|10060933|fndg
```

whose schema is

```
Gold Offsets|CUIDiff|Gold Text|Gold CUI(s)|Gold MedDRA PT(s)|
    MetaMap Offsets|MetaMap Text|MetaMap CUI(s)|MetaMap MedDRA PT(s)|SemType(s)
```

3.3.5 MedDRADiffs

MedDRAiffs are represented exactly like CUIDiffs, but with `MedDRADiff` instead of `CUIDiff` in the second field.

4 Different Configurations for `gen_results`

We provide two ways of modifying the processing of the `gen_results` script.

4.1 Match Strictness (-s)

The `gen_results` script counts as a True Positive an offset match between the Gold Standard and MetaMap output, regardless of CUI or MedDRA PT, subject to strictness criteria:

- A *strict* match requires that the entire GS and MM offsets be identical, this criterion is the default of the `gen_results` script.

- A *relaxed-1* match requires that matching GS and MM offsets contain at least an identical `StartPos,EndPos` pair, e.g., 14069,14077;14092,14102 and 14092,14102, representing *gingival ulceration* and *ulceration*.

- A *relaxed-2* match requires that matching GS and MM offsets contain at least an identical StartPos AND an identical EndPos (but not necessarily in the same pair), e.g., 24344,24367,and 24344,24353;24361,24367, representing *tightness of the throat* and *tightness throat*.

- A *relaxed-3* match requires that matching GS and MM offsets contain at least an identical StartPos OR an identical EndPos (but not necessarily in the same pair), e.g., 23121,23137 and 23121,23126, representing *edema peripheral* and *edema*.

To generate results with non-default strictness (i.e., N = 1, 2, or 3), simply call `gen_results` as described at the beginning of Section 3.2, as

*`./bin/gen_results -s N` where `N` is the desired value.

Not surprisingly, the higher the `N` value, the looser the match, and therefore the higher the F_1 score.

4.2 Different CUI → MedDRA Mappings (-c)

The gen_results script by default uses the CUI.MedDRA.PT file described in item 2 of Section 2.1. Our experiments showed that the CUI.MedDRA.PT file generated higher F_1 scores than the CUI.MedDRA.PT.LLT and CUI.MedDRA.HT.PT.LLT files, although these two did improve recall.

To run gen_results with a CUI→MedDRA mapping file other than the default CUI.MedDRA.PT, simply call e.g.,

```
*./bin/gen_results -c CUI.MedDRA.PT.LLT.
```

The -s flag described immediately above in Section 4.1 and the -c flag can be combined in one call to gen_results, e.g.,

```
*./bin/gen_results -s 2 -c CUI.MedDRA.PT.LLT.
```

5 Conclusion

As we mentioned in the introduction, this document contains considerable detailed information. We are available to discuss it, answer questions, and provide additional information.