A repository of semantic types in the MIMIC II database clinical notes

Richard M. Osborne
Computational Bioscience
University of Colorado
School of Medicine
richard.osborne@ucdenver.edu

Alan R. Aronson
National Library of Medicine
Bethesda, MD
alan@nlm.nih.gov

K. Bretonnel Cohen
Computational Bioscience
University of Colorado
School of Medicine
kevin.cohen@gmail.com

Abstract

The MIMIC II database contains 1,237,686 clinical documents of various kinds. A common task for researchers working with this database is to run MetaMap, which uses the UMLS Metathesaurus, on those documents to identify specific semantic types of entities mentioned in them. However, this task is computationally expensive and time-consuming. Research in many groups could be accelerated if there were a community-accessible set of outputs from running MetaMap on this document collection, cached and available on the MIMIC-II website. This paper describes a repository of all MetaMap output from the MIMIC II database, publicly available, assuming compliance with usage agreements required by UMLS and MIMIC-II. Additionally, software for manipulating MetaMap output, available on SourceForge with a liberal Open Source license, is described.

1 Introduction

1.1 The MIMIC II database and its textual contents

The Multiparameter Intelligent Monitoring in Intensive Care II (MIMIC-II) database is a public-access intensive care unit database that contains a broad array of information for over 33,000 patients. The data were collected over a 7 year period, beginning in 2001 from Boston's Beth Israel Deaconess Medical Center (Saeed et al, 2011; Goldberger et al., 2000).

Of particular interest are the 1,237,686 clinical documents, which are broadly classified into the following four groups: MD notes, discharge summaries, radiology reports and nursing/other. Each free-text note contains information describing such things as a given patient's health, illnesses, treatments and medications, among others.

1.2 Motivation for the resource: MetaMap runtimes

Part of the motivation for making this resource publicly available is that considerable resources must be expended to process it; if multiple groups can share the output of one processing run, the savings across the community as a whole could be quite large. To illustrate why this would be valuable from a resources perspective, we provide here some statistics on the performance of MetaMap.

Random samples of each category (10% each) were chosen and Monte Carlo simulation was performed (1,000 iterations per note) to obtain the running times presented below. The clinical notes ranged from a minimum of 0 words to a maximum of 6,684 (some of the notes were 0 bytes because the note for a particular patient and day contained no text). The mean, median and mode per document processed by MetaMap were 17, 5 and 2 seconds, respectively, with a minimum of 1 and a maximum of 216 seconds.

Figure 1 below plots the number of words against processing times in seconds for each of the of notes, sampled as mentioned above.

The majority of the processing was done on a Sun Fire X4600M2 server with 16 (4 x Quad-Core AMD Opteron(tm) Processor 8356 cores, 2.3GHz), 128GB memory and 12 TB of disk storage, currently running Fedora Core 17 Linux. (An Apple MacBook Pro and a Windows desktop server were also used to speed processing. The analysis of the random sample of notes was performed in its entirety on the Sun machine, thereby providing consistent results for the data in Figure 1.)

Proceedings of the 2014 Workshop on Biomedical Natural Language Processing (BioNLP 2014), pages 93–97,
Baltimore, Maryland USA, June 26-27 2014.

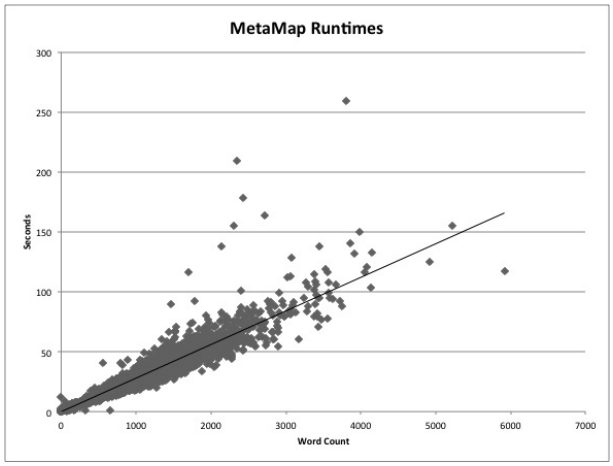

Figure 1: MetaMap Runtimes

1.3 Motivation for the resource: reproducibility

Any large-scale run of MetaMap over a huge document collection will have occasional failures, etc. The odds of any two runs having the same output are therefore slim. Moreover, there is potential variability in how documents are preprocessed for use with MetaMap. Using this repository of MetaMap outputs will ensure reproducibility of experiments and also preclude the necessity of performing the same preparatory work and MetaMap processing on the same data.

1.4 Motivation for the resource: semantic types

The creation of the MIMIC-II repository is an intermediate step in our research. We are extracting the semantic types found in each clinical note in an attempt to determine if there exists evidence of subdomains across the categories used by MIMIC-II to group the notes.

2 Materials and Methods

2.1 Materials

MetaMap is a program developed at the National Library of Medicine (NLM) that maps biomedical concepts to the UMLS Metathesaurus and reports on the corresponding semantic types.[1] The program is used extensively by researchers in the field of biomedical text mining. See Aronson, 2001; Aronson and Lang, 2010.

[1]Users of MetaMap must comply with the UMLS Metathesaurus license agreement (`https://uts.nlm.nih.gov/license.html`).

Although our focus is on the clinical notes contained in a single table, **noteevents**, MIMIC-II is both a relational database (PostgreSQL 9.1.9) containing 39 tables of clinical data and bedside monitor waveforms and the associated derived parameters and events stored in flat binary files (with ASCII header descriptors). For each Intensive Care Unit (ICU) patient, Saeed et al. (2011) collected a wide range of data including *inter alia* laboratory data, therapeutic intervention profiles, MD and nursing progress notes, discharge summaries, radiology reports, International Classification of Diseases, 9th Revision codes, and, for a subset of patients, high-resolution vital sign trends and waveforms. All data were scrubbed for personal information to ensure compliance with the Health Insurance Portability and Accountability Act (HIPAA). These data were then uploaded to a relational database thereby allowing for easy access to extensive information for each patient's stay in the ICU (Saeed et al.). A more detailed description of the use of the MIMIC-II database may be found in(Clifford et al. , 2012).

The abbreviated schema in Table 1 below shows that each ID uniquely identifies a note along with a SubjectID, a Category and Text.[2] We added the ID attribute to **noteevents** as a primary key because SubjectID, Category and Text are not keys. Thus, a particular patient might have many notes and many categories but each note is uniquely identified.

Attribute	Type	Cardinality	Sample Values
ID	integer	unique	1, 2, 3, 4...
SubjectID	integer	many to one ID	95, 100, 100, 99,
Category	character varying(26)	many to one ID	radiology
Text	text	many to one ID	interval placement of ICD

Table 1: Schema MIMIC-II noteevents table.

As mentioned above, the notes in the MIMIC-II database are categorized as MD reports, radiology reports, discharge summaries and nursing/other reports. The contents of these notes varied greatly. The MD and nursing notes tended to be short and unstructured with a number of abbreviations and misspellings, whereas the radiology reports were longer, more structured and showed fewer errors.

The MIMIC-II version used for this research is 2.6 (April 2011; 32,536 subjects).

The distribution of reports with summary statistics is below in Table 2.

[2]The full **noteevents** table has ten other attributes such as admission date, various timestamps and patient information but these were not relevant to our research.

	MD	Discharge	Radiology	Nursing	Totals
min words	0	0	0	0	0
max words	632	6684	2760	632	6684
median words	108	963.5	174	108	135
average words	131.6	1009.2	265.5	131.6	194.7
total notes	23,270	31,877	383,701	798,838	1,237,686

Table 2: MIMIC-II Clinical Note Summary Stats.

2.2 Methods

We used MetaMap to process the clinical notes in order to find semantic concepts, the latter of which are being used in our current research. For the work in this paper, we used MetaMap 2013 with the 2013AB database.

Before processing the notes with MetaMap, a number of preparatory steps were taken. As mentioned above, a primary key was added to the **noteevents** table to provide a unique id for each note. A Python script then queried the database extracting each note and storing it in a file named according to the following convention: uniqueID_subjectID_category.txt where uniqueID is the primary key value from the **noteevents** table, subjectID is the unique number assigned to each patient and category is one of the four categories mentioned above.

Each of the notes was then processed by a Bash shell script to remove blank lines and control characters. (This important step was added after a significant amount of processing had already taken place. If this is not done, a number of problems arise when running MetaMap). Finally, all files with 0 bytes were removed. These files were present because many tuples in the **noteevents** table contained clinical note entries with no data.

The number of options available when running MetaMap is considerable so we chose those that would provide a full and robust result set which would be useful to a wide range of researchers. In our first run, we limited the threshold for the Candidate Score to 1,000. However, for the repository, no threshold was set so that a full range of output is provided. [3]

The output is in XML in order to structure the data systematically and provide an easier and consistent way to parse the data. Although we chose XML initially, we intend to provide the same data in plain text and Prolog formats, again to provide utility to a broad range of researchers.

In order to process all files, a Bash shell script was created that called MetaMap on each note and created a corresponding XML file, named according to the same convention as that for notes but with the txt extension replaced by xml.

3 Results

3.1 The repository of MetaMap output

The repository for the MetaMap output contains an XML file for each note that originally contained text in the MIMIC-II database. Each XML file contains a wealth of information about each note and a discussion of this is beyond the scope of this paper (see `http://metamap.nlm.nih.gov/Docs/MM12_XML_Info.shtml`).

For our research, we are interested in the semantic types associated with phrases identified by MetaMap. Below is a section from output file 768591_19458_discharge.xml. This is a discharge summary for subject 19458 with a unique note id of 768591. The note contained the phrase "Admission Date" which MetaMap matched with a candidate score of 1000 and indicated that it is a temporal concept (tmco).

Ultimately, the MetaMap output files will be uploaded to the PhysioNet website and made available to the public.[4] The files will be organized in a fashion similar to the original data files on the site. Namely, data are grouped by subject ids and compressed in archives with approximately 1000 files each.

```
<Candidate>
<CandidateScore>-1000</CandidateScore>
<CandidateCUI>C1302393</CandidateCUI>
<CandidateMatched>Admission date
     </CandidateMatched>
<CandidatePreferred>Date of admission
      </CandidatePreferred>
<MatchedWords Count="2">
<MatchedWord>admission</MatchedWord>
<MatchedWord>date</MatchedWord>
</MatchedWords>
<SemTypes Count="1">
<SemType>tmco</SemType>
</SemTypes>
```

The original note contained 975 lines, whereas the MetaMap xml file contained 248,198. Thus it is obvious that there is a very large amount of MetaMap output that we don't consider but which may be of interest to other researchers.

[3] The exact MetaMap command we used was metamap13 –XMLf –silent –blanklines 3 filename.txt

[4] Subject again to the data usage agreement.

3.2 A Python module for manipulating MetaMap output

In order to make information in the XML files accessible to others, we developed a Python module (parseMM_xml.py) containing a number of methods or functions that allow one to parse the XML tree and extract relevant information.

Although we will add more functionality as needed and requested, at this point the following methods are implemented:

- parseXMLtree(filename) – parses the contents of filename and returns a node representing the top of the document tree.
- getXMLsummary(XMLtree) – summarizes the data contained in the parsed XML tree. The summary contains top-level elements and their corresponding text. The output is much like that contained in typical MetaMap text output.
- getCUIs(XMLtree) – returns the MetaMap CUIs found in the XML tree along with the matching concepts.
- getNegatedConcepts(XMLtree) – returns negated concepts and their corresponding CUIs.
- getSemanticTypes(XMLtree) – returns matched concepts, their CUIs, the candidate scores and the semantic types associated with the concept.
- findAttribute(attribute) – searches the document tree for an attribute of the user's choosing. Returns the attributes with their corresponding text values.

We chose Python to create our module because of its ease of use and its multi-platform capabilities. Once Python is installed and the parseMM_xml.py is placed in a directory along with the MetaMap xml file which is to be analyzed, retrieving relevant information is relatively straightforward. [5]

[5]Under most circumstances, Python is already installed on the Mac OS X and Linux operating systems.

A stylized version of our code is presented below.

```
# Parse XML tree and return semantic
    types.

import parseMM_xml
xml_tree = \
parseXMLtree("noteid_subid_category.xml")
semTypes = getSemanticTypes(xml_tree)

print(semTypes)
```

A truncated listing of the output:

CandidateCUI – C0011008
CandidateMatched – Date
1 – SemType – Temporal Concept
CandidateCUI – C2348077
CandidateMatched – Date
2 – SemType – Food

In order to fully test the robustness of our module, we will do further unit and regression testing, in addition to providing more exception handling. Ultimately, the code will be available on SourceForge, an Open Source web source code repository available at `www.sourceforge.net`.

Acknowledgments

We would like to thank George Moody of MIT for his help with questions concerning the MIMIC-II database.

References

Alan R. Aronson 2001. *Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program,* Proc AMIA Symp. 2001; 17:21.

Alan R. Aronson, Franois-Michel Lang 2010. *An overview of MetaMap: historical perspective and recent advances,* J Am Med Inform Assoc 2010; 17:3 229-236

Gari D. Clifford, Daniel J. Scott, Mauricio Villarroel 2012. *User Guide and Documentation for the MIMIC II Database,* `http://mimic.physionet.org/UserGuide/UserGuide.pdf`

Ary L. Goldberger; Luis A. N. Amaral; Leon Glass; Jeffrey M. Hausdorff; Plamen Ch. Ivanov; Roger G. Mark; Joseph E. Mietus; George B. Moody; Chung-Kang Peng; H. Eugene Stanley, 2000 *PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals,* Circulation 101(23): e215-e220.

Mohammed Saeed, Mauricio Villarroel, Andrew T. Reisner, Gari Clifford, Li-Wei Lehman, George Moody, Thomas Heldt, Tin H. Kyaw, Benjamin Moody, Roger G. Mark. 2011 *The Multiparameter intelligent monitoring in intensive care II (MIMIC-II): A public-access ICU database,* Critical Care Medicine; 39(5):952-960

MetaMap Release Notes Website 2013. *MetaMap 2013 Release Notes* `http://metamap.nlm.nih.gov/Docs/MM12_XML_Info.shtml`

MetaMap 2012 Output Website 2014. *MetaMap 2012 XML Output Explained* `http://metamap.nlm.nih.gov/Docs/MM12_XML_Info.shtml`

PhysioNet Website 2014. *PhysioNet MIMIC-II Website* `http://physionet.org/mimic2/`