

Reviewer's report

Title: Automated De-Identification of Free-Text Medical Records

Version: 1 **Date:** 14 January 2008

Reviewer: Jules Berman

Reviewer's report:

General comments

The authors have added their auto-scrubber to the list of open source programs currently available. This paper describes their software and should be published. However, the paper contains inaccuracies that must be changed before the paper can be accepted.

Major Compulsory Revisions

1. The discussion is focused almost exclusively on HIPAA. The Common Rule precedes HIPAA, and there is language in HIPAA indicating that the Common Rule trumps HIPAA in areas where they both overlap. The Common Rule stipulates the protections that IRB must put in place for human subject research. The Common Rule holds that medical record research is human subject research. For most of the uses related to de-identification that were described in the paper, it is the Common Rule, not HIPAA, that must be satisfied. This paper should be re-written to include consideration for the Common Rule.

2. Beyond HIPAA and the Common Rule are issues related to tort and to privacy, both of which existed for researchers prior to either HIPAA or Common Rule. These issues apply equally to researchers outside of the U.S. The emphasis on U.S. Regulation is unnecessarily insular and does not cover all of the legal/sociologic issues that de-identification software must address.

Privatizing data involves removing information that is embarrassing, potentially incriminating, or just nobody's business.

If a nurse writes, "Dr. Dirty never washes his hands between patients," removing the name "Dirty" doesn't solve the problem incurred when disseminating the text. IRBs have several tools at their disposal: the waiver (under HIPAA or the Common Rule) or the limited data use agreement (under HIPAA) to share scrubbed but not perfectly de-identified data.

Privatization is sometimes a major obstacle for sharing de-identified medical records. The authors should address privatization as a potential legal limitation for auto-scrubbers.

3. The authors compare the speed of their software to that of humans doing the same job. They write that the corpus was 118 million words. That's about 1 Gigabyte. It took two days to parse a 1 Gigabyte text. That means that it would

take 2,000 days to parse 1 Terabyte (about the amount of data collected each week in large hospital systems). This means that their scrubber cannot keep up with output of a hospital.

The authors should comment that their scrubber is sufficiently fast for many research-sized fixed length files (on megabytes to gigabytes), but should comment that it may not suffice for large-scale real-time or data-streaming efforts using data collected from multiple medical systems.

4. The de-identifier reduces the number of identifiers in a corpus (without eliminating all of the identifiers).

They write:

"Medical records are said to be de-identified when the risk is *very small* that the information can be used alone or in combination with other reasonably available information to reidentify individuals associated with the records."

Well, if identifiers are left in the text, then the risk is very large that the information can be used to re-identify a patient. Since their de-identifier doesn't remove all of the identifiers, it does not produce an output that can be fully disseminated.

IRBs seldom consider program-scrubbed data to be truly de-identified. It would be unusual (and probably a terrible mistake) for an IRB application to claim exemption under HIPAA safe harbor or under the e4 paragraph of the Comon Rule. That's why nobody publishes their auto-scrubbed medical records in public places (let me know if you can find any). Waivers and limited data use agreements and their relevance to auto-scrubber projects should be discussed.

5. The authors describe my concept-match scrubber correctly but they draw a wrong conclusion about its functionality. Contrary to their statement in the paper, the concept-match scrubber will block "Mr. Parkinson" because "Mr. Parkinson" is not a term listed in UMLS. Also, the concept-match scrubber has no trouble handling misspellings: it just blocks them.

The limitation with the concept-match scrubber is that it blocks too much, so the output is full of asterisks (the blocking symbol) and the text is hard to read. Since publishing the concept-match scrubber, I've published a new scrubber. This one uses a list of identifier-free doublets (about 200,000). It parses through any text, matching every possible doublet in the text against the list of approved doublets. It preserves, in situ, those text doublets that match against one of the doublets in the "safe" list. Everything else in the text is blocked (with an asterisk). This produces an output that is much more readable than the concept-match output and which is also fully de-identified.

The doublet scrubber is fast, operating at 1 Megabyte per second on my 2.8 GHz computer with 512 MByte memory. This means that my scrubber would scrub a terabyte of text in about a week (compared to 2000 days for the authors' scrubber).

I devoted a chapter to this new scrubber in my recent book, Ruby Programming for Medicine and Biology, Jones & Bartlett Publishers, Sudbury, MA, 2008. (chapter 11, pp 157-163).

The description of my autoscrubber should be corrected and updated.

6. The authors compare their scrubber to other programs and unnecessarily criticize the other software applications. This only serves to make people angry. I think it would be much more useful (for the readers) if the authors simply stated how their scrubber might be used and how the other scrubbers might be used (and this is what the authors should do to revise the text). If you need a readable output on a relatively small corpus, the scrubber prepared by the authors might be ideal. If you need a scrubbed output on a very large corpus (terabytes) and you can tolerate an output that is less than optimally readable, the doublet scrubber would be better.

Minor Essential Revisions

I really think that all of the revisions I suggested are major issues and must be addressed. On the plus side (for the authors) is that they can all be handled by relatively small changes in the text. If the authors object to these revision requests, I'd be happy to re-consider.

Discretionary Revisions

The Latanya Sweeny papers have been discussed-to-death in earlier related papers. Is it really necessary to do it all again here?

The same goes for the safe harbor hipaa de-identifiers. Everyone familiar with HIPAA knows this stuff. Do you really need to list (again) the 18 types of identifiers?

What next?: Unable to decide on acceptance or rejection until the authors have responded to the major compulsory revisions

Level of interest: An article of importance in its field

Quality of written English: Acceptable

Statistical review: No, the manuscript does not need to be seen by a statistician.

Declaration of competing interests:

'I declare that I have no competing interests'