

## **Author's response to reviews**

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

### **Authors:**

Ishna Neamatullah ([ishna@mit.edu](mailto:ishna@mit.edu))  
Margaret M Douglass ([douglass@mit.edu](mailto:douglass@mit.edu))  
Li-wei H Lehman ([lilehman@mit.edu](mailto:lilehman@mit.edu))  
Andrew Reisner ([areisner@partners.org](mailto:areisner@partners.org))  
Villarroel Villarroel ([maurov@mit.edu](mailto:maurov@mit.edu))  
William J Long ([wjl@mit.edu](mailto:wjl@mit.edu))  
Peter Szolovits ([psz@mit.edu](mailto:psz@mit.edu))  
George B Moody ([george@mit.edu](mailto:george@mit.edu))  
Roger G Mark ([rgmark@mit.edu](mailto:rgmark@mit.edu))  
Gari D Clifford ([gari@mit.edu](mailto:gari@mit.edu))

**Version:** 2 **Date:** 26 March 2008

**Author's response to reviews:** see over

Reviewer 1:

#### Minor Essential Revisions

1. Examples of regular expressions would be welcome. They would help more NLP-oriented readers to get a more precise idea of the kind of strings which can be matched.

> As requested, we have added a set of examples of regular expression in appendix B.

2. p. 27 refers to "The initial version of the de-identification software". Earlier in the paper I thought I had understood that the software described in the paper was an improvement over an initial version. But this sentence seems to mean that the older version is the one which was evaluated on the test corpus.

> The algorithm was essentially developed in three stages. In the first stage we built a gold standard corpus and evaluated the early algorithm on this, (See Douglass 2004, 2005. The algorithm presented in the paper is an improvement over that algorithm and was subsequently evaluated on the test corpus. However, since this test (performed about 1 year ago), significant improvements were made to the algorithm. However, since the test evaluation was performed by hand (on paper) for speed and ease of review, the test corpus could not be used to re-evaluate the software after the improvements. The 'gold standard' training corpus was amenable to re-evaluation (since the data was coded digitally). These are the final results quoted on the gold standard. We have updated the description on page 27 to read: "An early prototype of the de-identification software described here [4] was applied to the test data to generate a preliminary set of scrubbed nursing notes. Each of 14 reviewers was then assigned approximately 130 of these scrubbed nursing notes and was charged to identify any PHI remaining in the scrubbed text."

#### Discretionary Revisions

3. The UMLS is described as a "collection of standard medical dictionaries": "vocabularies" would seem more appropriate. Can all vocabularies included in the UMLS be called "standards"?

> "collection of standard medical dictionaries" has been replaced by "collection of medical vocabularies, some of which are considered standards in particular applications"

4. p. 20, Locations:

How was the list of locations compiled? How large is it?

> Appendix C now lists all the dictionaries and their respective sizes. There are several location dictionaries, compiled from several sources. The content from each file was extracted from public, state and national records.

5. p. 26, why not measure precision on the test corpus too? The effort needed to check for false positives seems much lower than that needed to check for false negatives, given the estimate of about 474 instances of PHI per 100,000 words, which predicts about 1,400 PHIs in the whole test corpus (about 100 per reviewer).

> The test evaluation was performed by hand (on paper) for speed and ease of review, and so the test corpus could not be used to re-evaluate after the improvements.

6. p. 33, "It should be note\*d\*" is the only typo I could spot in the whole paper.

> Thank you - we have corrected that.

7. p. 40, the URL "<http://www.physionet.org/physiotools/deid/>" is correct but the link in the PDF is incorrect (it includes the next words).

> Apologies - that must be an artifact of the BMC word-pdf converter. The URL is correct in the word document. We have pointed this out to the journal and will look for this in the final proof copy.

Reviewer 2:

#### 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.

> Thank you for pointing this issue out. We have tried to make this distinction clear in the article. As we understand it, the Common Rule defines research on PHI-bearing clinical data as human subjects research and states the responsibility of researchers to get IRB approval before doing this, as well as the responsibilities of IRB's to make sure the data are used safely. IRB's are allowed, in the case of "minimal risk" and significant benefit, to grant permission to use clinical data in research without explicit patient consent, if such consent would be impractical to get. For example, we have received IRB approval to use raw patient data to help develop de-id algorithms, with various safeguards to make sure that the data do not "leak". Our algorithm is not meant to subvert these rules, and we are not suggesting that our algorithm is completely sufficient to adhere to HIPAA, only that it goes much of the way towards addressing the issue.

> We have now altered the manuscript to ensure that it acknowledges the duty of researchers to obey the common rule, and that there is a significant risk of disclosing PHI when only automated scrubbing is done. Also we have emphasized that the bulk of our data (that is only automatically de-identified) will only be available under a limited data use agreement.

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.

> See response to point #1. We have also addressed this explicitly in the discussion and conclusion, to make sure that it is clear that we do not intend this to be used as generic auto scrubber for public release of data.

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.

> The reviewer is correct that our algorithm would be extremely slow to process all the text data produced by a hospital if run on a simple PC, and if the hospital were producing a massive 1Tb a week. However, this is not the aim of our algorithm. It has been designed to de-identify research databases offline. On the other hand, we could parallelize the de-identification algorithm and run it on 40 8-core processors to be able to process the 1TB/week in real time. Also, our main database is ~1Gb of uncompressed text data collected over 5 years, and we estimate that our local hospital produces at most 10 times this in text-based data, so we would expect to have to de-identify less than 1MB a day.

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 Common 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.

> We do agree, and have adjusted the paper to emphasize this point. We do not consider our algorithm acceptable to subvert IRB approval for scrubbing data. In fact, all of our gold standard corpus was meticulously scrubbed by a team of experts and an algorithm. We would expect no less for any publicly available research data.

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.

> Thank you for pointing out this update. We have corrected and updated our description of your algorithm as described on your website: "As Berman points out, 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, Berman has published a new scrubber algorithm based upon doublet (word pair) matching [26]. Berman's new approach parses through a text, matching every possible doublet (word-pair) in the text against the list of a list of approved identifier-free doublets (about 200,000). The doublet scrubber 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. Although a significant improvement, much useful text is still blocked."

> We have also added the following in the discussion: "An alternative approach to virtually ensure full HIPAA-compliant de-identification, is that of concept-matching [26] where the output is devoid of phrases that do not map to a reference terminology and is stripped of nonmedical and extraneous information. Although some relevant information may be removed, concept matching provides the terminology code for each medical term included in the sentence, making it possible to index and relate the terms to each other and standard biomedical ontologies."

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.

> Comparison between scrubbers is almost impossible given the lack of publicly available data, and it certainly wasn't our intention to criticize other scrubbers for this reason. We were just trying to describe the applicability of the other scrubbing methods to our data type. We are sure that the other scrubbers are entirely appropriate and well-constructed for their applicable data types, and that our scrubber is likely to have a lower performance because of the different nature of the data. Our framework is general and open-source, and so can be adapted to other data sets, but this may not be the best approach for certain types of data. We have adjusted the style of this review section to make this clear and avoid offending other authors who should certainly be credited for their excellent systems.

#### 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.

> We have no objections to the revisions and find them most useful.

#### 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?

> We felt that a discussion of these papers was warranted in the context of other works.

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?

> We felt that since the HIPAA rules may evolve over time, it was important to provide an explicit list of what definitions/rules we were attempting to address. We have therefore left them as a table, which may be ignored for those that are familiar with the definitions. Furthermore, our algorithm addresses each HIPAA PHI category separately, and so we felt it was important to list the manner in which we do this for each PHI category.