Deidentifying Clinical Text: NLM Scrubber Error Analysis

Mohammad Arvan, PhD, Karl M. Kochendorfer, MD, Shane Borkowsky, MD, Aaron Chaise, MD, Bhrandon Harris, MD, Natalie Parde, PhD [1]

[1]University of Illinois Chicago, Chicago, Illinois

Introduction

This poster explores the NLM Scrubber's effectiveness in deidentifying patient notes, emphasizing its limitations, error patterns, and the role of human evaluation. NLM Scrubber is a deidentification tool developed at the National Library of Medicine. Clinical text deidentification is the process of removing Protected Health Information (PHI) and is vital for privacy compliance. While automatic tools can support the PHI removal process, they may misclassify data. By studying differences between scrubbed and original patient notes, the project aims to reveal tool error patterns, enhancing accuracy and compliance with privacy regulations in healthcare settings. This study was approved by the Institutional Review Board (IRB) at the University of Illinois Chicago (#2020-0418).

Methods

We followed a structured methodology to collect and preprocess 45 patient notes, then used the NLM Scrubber to deidentify them. Four expert evaluators (physicians) reviewed the original and scrubbed notes. The expert evaluators annotated errors and corresponding error types for both false negatives and false positives.

Results

The NLM Scrubber identified 1894 cases of PHI. We report precision and recall (sensitivity) of 0.84 and 0.98, respectively. Specifically, we found 18 false negatives, or cases of PHI that were left unscrubbed. Twelve of these were timestamps (e.g., "06.25," or "Jan 2018,"). The remaining six errors involved the omission of personal names and identifiers, with half of these errors related to middle name initials left unscrubbed. Additionally, there was a case where a non-English name and a name entirely in lowercase were not appropriately scrubbed.

The prevalence of false positive cases was higher (293 total). Specific medical terms and abbreviations (e.g. "NoDiff," "COVID," or "IONCA") consistently triggered flags. Moreover, certain vitals, dates and timestamps (e.g., "190/88") were labeled as addresses and the abbreviation "2/2 to," commonly understood as “secondary to,” was misinterpreted as "etiology [DATE] to." Timestamps ("02/17 02:21" or "06/05 14:15") were misread as dates or appended with alphanumeric identifiers, and terms such as "DAY," "PLAN," "Author," "Meeting-active," and "Disorder" were incorrectly identified as personal names.

Discussion and Conclusions

Overall, we observed that the NLM Scrubber tends to excessively label non-PHI as PHI, indicating an inclination toward over-scrubbing. While the NLM Scrubber offers an allowlist (whitelist) option, its source code is not public, limiting further investigation. Removing medical terms is particularly harmful as it leads to information loss. Hence, an effective tool should allow the integration of a custom dictionary. The tool struggles with name detection, often relying too much on capitalization. To address these challenges, employing Large Language Models (LLMs) could prove transformative. LLMs can understand and generate human-like text, making them adept at discerning contextual nuances in clinical documents. We are now considering two specific pathways forward. First, we are exploring the development of a new tool tailored to address these noted deficiencies effectively. Alternatively, we may contribute to enhancing an existing open-source project, Philter, which already shows promise in the field of clinical text deidentification. Both approaches aim to incorporate the versatility and precision of LLMs to improve PHI detection and minimize information loss.

Limited data availability is still one of the biggest challenges for healthcare-oriented machine learning, largely due to the presence of PHI in clinical notes. In this work, we shed light on automated deidentification quality, reporting common error patterns and providing recommendations on how to address them. Importantly, we believe there is a need for an *open-source* tool that is easily *customizable* and utilizes the latest advancement in text processing.