Journal of Biomedical Informatics xxx (2015) xxx–xxx

Contents lists available at ScienceDirect

Journal of Biomedical Informatics

journal homepage: www.elsevier.com/locate/yjbin

Combining knowledge- and data-driven methods for de-identification of clinical narratives

Azad Dehghan [a,b], Aleksandar Kovacevic [c], George Karystianis [a,b], John A. Keane [a,d], Goran Nenadic [a,d,e,*]

[a] School of Computer Science, University of Manchester, Manchester, UK
[b] The Christie NHS Foundation Trust, Manchester, UK
[c] Faculty of Technical Sciences, University of Novi Sad, Novi Sad, Serbia
[d] Manchester Institute of Biotechnology, University of Manchester, Manchester, UK
[e] Health eResearch Centre, The Farr Institute of Health Informatics Research, UK

ARTICLE INFO

Article history:
Received 15 February 2015
Revised 22 May 2015
Accepted 30 June 2015
Available online xxxx

Keywords:
De-identification
Named entity recognition
Information extraction
Clinical text mining
Electronic health record

ABSTRACT

A recent promise to access unstructured clinical data from electronic health records on large-scale has revitalized the interest in automated de-identification of clinical notes, which includes the identification of mentions of Protected Health Information (PHI). We describe the methods developed and evaluated as part of the i2b2/UTHealth 2014 challenge to identify PHI defined by 25 entity types in longitudinal clinical narratives. Our approach combines knowledge-driven (dictionaries and rules) and data-driven (machine learning) methods with a large range of features to address de-identification of specific named entities. In addition, we have devised a two-pass recognition approach that creates a patient-specific run-time dictionary from the PHI entities identified in the first step with high confidence, which is then used in the second pass to identify mentions that lack specific clues. The proposed method achieved the overall micro F_1-measures of 91% on strict and 95% on token-level evaluation on the test dataset (514 narratives). Whilst most PHI entities can be reliably identified, particularly challenging were mentions of *Organizations* and *Professions*. Still, the overall results suggest that automated text mining methods can be used to reliably process clinical notes to identify personal information and thus providing a crucial step in large-scale de-identification of unstructured data for further clinical and epidemiological studies.

1. Introduction

A recent promise and the potential of wider availability of data from Electronic Health Records (EHRs) to support clinical research are often hindered by personal health information that is present in EHRs, raising a number of ethical and legal issues. De-identification of such data is therefore one of the main pre-requisites for using EHRs in clinical research. As a result, there is a growing interest for automated de-identification methods to ultimately aid accessibility to data by removing Protected Health Information (PHI) from clinical records. De-identification of unstructured data in particular is challenging, as PHI can appear virtually anywhere in a clinical narrative or letter. This task is often considered as Named Entity Recognition (NER), where mentions of specific PHI data types (e.g. patient names, their age and address) need to be identified in the text of clinical narratives.

Automated de-identification of unstructured documents has been a research topic for more than twenty years. As early as 1996, Sweeney et al. proposed a rule-based approach to recognize twenty five overlapping entity types they identified as PHI in EHRs [1]. Since then, a large number of systems have been introduced, including knowledge-based [2–5] and data-driven [6–11], as well as hybrid [12–14] methods that combine various approaches. In terms of types of clinical narrative, previous de-identification research has explored varied clinical documents such as discharge summaries [11,15], pathology reports [9], nursing progress notes [2] and mental health records [4].

The 2006 i2b2 de-identification challenge [15] was the first effort to provide a common test-bed for eight PHI entity types (mentions of *Patients*, *Doctors*, *Hospitals*, *IDs*, *Dates*, *Locations*, *Phone numbers* and *Age*) in clinical discharge summaries. The submitted systems ranged from rule-based [5] and machine-learning (ML) methods (e.g. using Conditional Random Fields [13], Hidden Markov Models [13], and Decision Trees [8]) with a wide range of features, to hybrid approaches (e.g. combining rules and Support Vector Machines [12]). A notable observation across

* Corresponding author at: School of Computer Science, Oxford Road, Manchester M13 9PL, UK.

E-mail address: g.nenadic@manchester.ac.uk (G. Nenadic).

http://dx.doi.org/10.1016/j.jbi.2015.06.029
1532-0464/© 2015 Elsevier Inc. All rights reserved.

methods was the use of knowledge-driven techniques (in particular rules) both for the direct recognition of PHI and in support of data-driven and hybrid methods. For example, rules were used as features in ML models (e.g. indicating whether a particular rule was triggered) [12], as a post-processing correction module [13] or combined with data-driven results at the final step (e.g. integration of ML and rule-based annotations) [14]. This trend was often motivated by the presence of a number of categories that are characterized by regularized expressions (e.g., date, phone, zip/postcode, and identification numbers), which make rules an efficient modeling technique. In general, the 2006 shared task showed that data-driven methods with features generated by rules for regularized expressions performed best [8,13]. They were followed by hybrid methods [12], while the pure rule-based systems proved to perform less well [5].

The 2014 i2b2/UTHealth [16] Shared Task in de-identification [17] of longitudinal clinical narratives focused on 25 entity types, inclusive of twelve types as defined by the Health Insurance Portability and Accountability Act (HIPAA). The entity types were grouped into seven main categories: *Names* (e.g., patient and doctor names), *Profession, Locations* (e.g., street, city, zip code, organizations), *Contacts* (e.g., phone, fax, email), *IDs* (e.g., medical record, identification number), *Age* and *Dates*. The organizers provided a fully annotated mention-level training dataset, as well as a test dataset for the evaluation. This paper describes a hybrid method that integrates the results of knowledge- (dictionary- and rule-based components) and data-driven methods. We present the results and further discuss the challenges in the de-identification task.

2. Methodology

The training data (790 narratives, 460,164 tokens) was released in two batches by the organizers. We have used the first batch (521 narratives, 316,357 tokens) for the initial design of the methods, whereas the second batch (269 narratives, 143,807 tokens) was used as a development set for validation and tuning. The initial analysis of the training data confirmed that some of the entity types are more lexically closed (e.g. country and city names) or regularized (e.g. zip codes, phones, etc.) than the others (e.g. patient and doctor names). The methods developed have largely followed that observation, devising a hybrid approach aiming to combine different methods where appropriate. Fig. 1 shows an overview of the system, and the steps are detailed below.

2.1. Pre-processing

The narratives were pre-processed with *cTAKES* [18] and *GATE* [19] to provide basic lexical and terminological features, including tokenization, sentence splitting, part-of-speech tagging and chunking.

2.2. Dictionary- and rule-based taggers

The dictionary-based taggers were used for the *Hospital*, *City*, *Country, State, Profession* and *Organization* entity types. The dictionaries (see Supplementary material for the full list) were collected from open sources such as Wikipedia, GATE and deid [2,20]. We have merged the entity-specific term lists from these sources and then manually filtered the resulting dictionaries to exclude ambiguous terms.

The rule-based tagger included a set of rules that exploited several types of features including the output of the dictionary-based taggers to recognize entities. Five feature types were used in the rule engineering:

1. *Orthographic* features, which include word characteristics such as *allCapitals*, *upperInitial*, *mixedCapitals*, or *lowerCase*; as well as token/word length.
2. *Pattern* features, which include common lexical patterns of specific entity types as derived from the training data set e.g., date (e.g., DD-MM-YYYY), zip (XXXX), telephone number (XXX-XXX-XXXX) and so forth.
3. *Semantic/lexical* cues or entity types. For example, *Street* names often include lexical cues such as ‘street’, ‘drive’, ‘lane’, *State* (e.g., “DC”, “CA”, etc.), and so forth.
4. *Contextual* cues that indicate the presence of a particular entity type. They include specific lexical expressions (e.g., person and doctor titles, months, weekdays, seasons, holidays, common medical abbreviations, etc.), symbols (e.g., bracket and colon, e.g. used for *Username* and *Medical record* respectively), and other special characters such as white space and newline.
5. *Negative* contextual cues (e.g., lexical and orthographic) are used for disambiguation (e.g., for entity types that are similar e.g., phone and fax number, patient and doctor names).

Using the combination of these features enabled us to craft a relatively small rule set of 5 rules on average per entity type (the minimum of 1 for zip, fax and email, and the maximum of 11 for age). The rules were developed using Java Annotation Patterns Engine (JAPE) [19] and Java regular expressions. An example rule is given in Table 1.

2.3. ML-based tagger

As target entities comprise spans of text, we approached the task as a token tagging problem and trained separate Conditional Random Fields (CRF) [21] models for each entity type. We used a token-level CRF with the Inside–Outside (I–O) schema [22], for each of the entity types separately. In this schema, a token is labeled with *I* if it is inside the entity span and with *O* if it is outside of it. For example: in sentence “*Saw Dr. Oakley 4/5/67*”, token “*Oakley*” will be tagged as *I_Doctor* (inside a doctor’s name), whereas all other tokens will be annotated as *O_Doctor* (outside doctor’s name). This schema provides more examples of “inside” tokens to learn from than the other schemas (e.g. the Beginning–Inside–Outside, B–I–O), and in our case, it also provided satisfactory results during training.

The feature vector consisted of 279 features for each token (see Supplementary material for the full list of features), representing the token’s own properties (e.g. lexical, orthographic and semantic) and context features of the neighboring tokens. Experiments on the development set with various context window sizes showed that two tokens on each side provide the best performance. The following features were engineered for each token:

1. *Lexical features* included the token itself, its lemma and POS tag, as well as lemmas and POS tags of the surrounding tokens. Each token was also assigned its location within the chunk (beginning or inside). All chunk types returned by *cTAKES* (see Supplementary material for the full list) were considered for this feature.
2. *Orthographic features* captured the orthographic patterns associated with gold-standard entity mentions. For example, a large percentage of hospital mentions are acronyms (e.g., *DHN*, *EHMS*), doctor and patient names are usually capitalized (e.g., *Xavier Rush*, *Yosef Villegas*), dates contain digits and special characters (e.g., “2069-04-07”, “04/07/69”), etc. We engineered two groups of orthographic features. The features in the first group captured standard orthographic characteristics (e.g., is the token capitalized, does it consist of only capital letters, does it contain digits, etc.). The second group aimed to further model