An Iterative Method for the De-identification of Structured Medical Text

György Szarvas[1], Richárd Farkas[2], Szilárd Iván[2], András Kocsor[2], Róbert Busa-Fekete[2]

[1] University of Szeged, Department of Informatics, 6720 Szeged, Árpád tér 2., Hungary

[2] Research Group on Artificial Intelligence of the Hungarian Academy of Sciences and University of Szeged
H-6720 Szeged, Aradi vértanúk tere 1., Hungary

Abstract

The process of removing personal health information (PHI) from clinical records is called de-identification. There are many methodologies in use for de-identification, and most of them are based on a named entity recognition (NER) technique. We introduce here a novel, iterative NER approach intended for use on semi-structured documents like discharge records and it can successfully identify PHI in several steps. First, our method looks for semantic information, labelling all entities whose tags can be inferred from the structure of the text and then it utilises this information to find further PHI phrases in the document.

INTRODUCTION

The identification and classification of named entities in a plain text is of key importance in numerous natural language processing applications like the de-identification of clinical records. This task is crucial in the human life sciences because a de-identified text can be made publicly available for non-hospital researchers as well, to facilitate research on human diseases. However, the records about the patients include explicit personal health information, and this fact hinders the release of many useful data sets because their release would jeopardise individual patient rights. According to the guidelines of Health Information Portability and Accountability Act (HIPAA) the medical discharge summaries released must be free of the following seventeen categories of textual PHI: first and last names of patients, their health proxies, and family members; doctors' first and last names; identification numbers; telephone, fax, and pager numbers; hospital names; geographic locations; and dates. Removing these kinds of PHI is the main goal of the de-identification process.

In the literature many de-identification approaches have been introduced, and most of them are based either on a pattern-matching algorithm that uses a thesaurus [1] or on a statistical model [2]. In this paper we adapt some Named Entity Recognition (NER) techniques for the de-identification of clinical records.

The NER task was introduced during the nineties as a part of the shared tasks in the MUC conferences [3]. Later, research oriented to multilinguality and specific domains, like bioinformatics (CoNLL conferences [4], JNLPBA-2004 [5]). Machine learning methods have been applied to the NER problem with remarkable success. The most frequently applied techniques were the Maximum Entropy Model, Hidden Markov Models and Support Vector Machines. In our former work we used AdaBoostM1 and C4.5 learning techniques, and found the combination of boosting and C4.5 competitive to other models in NER of newswire articles [6].

We extended our NER model – which was designed for learning on business domains and achieved slightly better results on CoNLL shared task than any other currently published method – by adding the following: we applied regular expressions and we distinguished between structured (information given in unambiguously identifiable fields) parts of the document and parts containing flow text. Our iterative learning method described below utilise the information given in the structured parts of the texts to improve the accuracy of PHI recognition in flow text.

In this paper we present results on the I2B2 de-identification shared task, consisting of 671 medical reports with tagged entities described earlier (14314 pieces of PHIs). This dataset differs in its characteristics from the ones of previous shared tasks (e.g. MUC, CoNLL); its domain is very special and these medical reports are semi-structured documents. This means that the headings of the

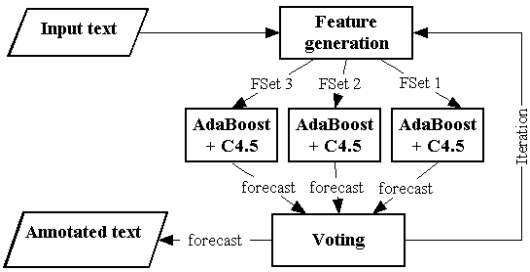

Figure 1: A schematic overview of our complex model

records can contain much useful information that can be easily extracted using our approach.

Our paper is organised as follows. In the next section we will discuss the feature sets used and we will introduce our new iterative method. Then we provide a brief overview of the Machine Learning models we employed in experiments. Next we will give a summary of the performance of our NER system on the I2B2 dataset, and lastly, we summarise our results and conclusions drawn from the study.

OUR APPROACH

We regard the de-identification problem essentially as the classification of separate tokens. We believe that this approach is competitive with the – theoretically more suitable – sequence tracking algorithms (like Hidden Markov Models, Maximum Entropy approaches or Conditional Random Fields), hence we applied a decision tree learning algorithm. Requiring less computational time, the application of C4.5 allowed us to use a feature set of enormous size. Of course our model is capable of taking into account the relationship between consecutive words using a window of appropriate size.

Figure 1 sketches the structure of our complex model, the details of its building blocks are described in this section.

Feature set

We employed a very rich feature set for our word-level classification model, describing the characteristics of the word itself along with its actual context (a moving window of size four). We did not use deep knowledge information (like POS, chunk codes or ontologies) or any complex domain specific resources (like MeSH IDs in [2]).

Our features fell into the following main categories:

- *Orthographical features*: capitalisation, word length, common bit information about the word form (contains a digit or not, has uppercase characters inside the word, and so on) and several regular expressions that describe the common surface characteristics of AGE, PHONE and ID classes,
- *Frequency information*: We gathered the frequencies of tokens from a huge corpus consisting texts collected from the Internet. We used the frequency of the token, the ratio of the token's capitalised and lowercase occurrences, the ratio of capitalised and sentence beginning frequencies of the token,
- *Phrasal information*: a forecasted class of several preceding words (we used an online evaluation) and common phrase suffixes (e.g. "Hospital") seen in the train set,
- *Dictionaries*: first names, cities in the US and so on: we collected five lists from the Internet,
- *Contextual information*: sentence position, the closest section heading, trigger words from the train text that often precede or follow PHI (see below), whether the word fell between quotes and so on.

We applied the feature set used for a common domain (as in our previous studies) and introduced only two new features: (i) regular expressions that try to cover the well formulated classes (they did not occur in previous shared tasks) and (ii) our model can infer knowledge from the structure of the document using the common headings observed in typical discharge records (we extracted the most frequent subject headings from the training set).

The use of trigger words is not straightforward, however, so we used them in three different way in our experiments: we collected the three preceding and three subsequent tokens of all *tagged tokens* in the train set (we refer to this feature set as the *token trigger* later on); similarly, we collected subsequent tokens of *tagged phrases* and used a wider window for this feature (*phrase trigger*); and third we collected the uni-, bi- and trigramms around the phrases of the train texts (*trigramm trigger*). The collected lists for each of the three cases were filtered according to their frequency and information gain on the class labels.

A significant difference in the predictions was noticed in experiments where only the use of triggers was changed, hence we decided to combine their forecasts to exploit the advantages of all of them.

Iterative learning

The structured parts of the text can be processed more easily than the flow text and the named entities in the record fields can occur in other parts of the text in the same or similar form. To utilise this latter fact we tagged only trusted named entities (appearing in document sections belonging to certain headings) in a first training phase. We considered a heading unambiguous if its cross class Shannon entropy was less then 0.1 on the train set. The named entities found in this first phase and their acronym became trusted phrases and their lists are added to the feature set of a second training phase.

We made the hypothesis that there are trigger words (like "dr.") which indicate trusted phrases as well. But the experiments with this kind of trusted named entities achieved worse results than ones without them, so we abandoned this hypothesis. This was probably caused by the artificially added ambiguity to PHI phrases in the data set (for example if we found a phrase "Dr. He" and accepted "He" as a trusted phrase, the model tended to treat all occurrences of the word "He" as the name of a doctor while it's a non-taggable common word in the majority of cases).

Fortunately, the structured parts of data usually contain full formed phrases and thus incorporating PHI found there proved to be beneficial to the model.

In the last phase of the iteration we standardize the tagging of the same phrases, because our token based classification approach can fail with tagging whole phrases. We collect all predicted phrases from the previous iteration and overwrite every occurrence of them with the predicted class of the longest matching phrase.

Classifiers

Boosting and C4.5 are well known algorithms for those who are acquainted with pattern recognition. Boosting has been applied successfully to improve the performance of decision trees in several NLP tasks. A system that made use of AdaBoost and fixed depth decision trees [7] came first on the CoNLL-2002 conference shared task, but gave somewhat worse results in 2003 (it was ranked fifth with an F measure of 85.0%). We have not found any other competitive results for NER using decision tree classifiers and AdaBoost.

Boosting was introduced by Shapire as a way of improving the performance of a weak learning algorithm. The algorithm generates a set of classifiers (of the same type) by applying bootstrapping on the original training data set and it makes a decision based on their votes. The final decision is made using a weighted voting schema for each classifier that is many times more accurate than the original model. In our investigation 30 iterations of Boosting were performed on each model as further iterations gave only slight improvement.

C4.5 is based on the well-known ID3 tree learning algorithm, which is able to learn pre-defined discrete classes from labelled examples. Classification is done by axis-parallel hyperplanes, and hence learning is very fast. This makes C4.5 a good subject for boosting. We built decision trees that had at least 5 instances per leaf, and used pruning with subtree raising and a confidence factor of 0.35.

Combination of the classifiers

The decision function we used to integrate the three hypotheses (learnt with different usage of triggers) was the following: *if any two of the three learners' outputs coincided we accepted it as a joint prediction, and forecasted 'O' label referring to a non-PHI entity class otherwise.* This cautious voting scheme is beneficial to system performance as a high rate of disagreement often means a poor prediction accuracy.

EXPERIMENTS

We extracted first the features – introduced above – for every token from the train set. 138 numerically encodable attributes describe each token (included features from a window around the token itself). Our previous experiments on NER problems showed that a feature space of this size can be handled by our learning algorithms for datasets smaller than 1 million tokens, hence we ignored any feature selection procedure. In our experiments we used an implementation based on the WEKA library [8], an open-source data mining software written in Java.

We split the train data into ten pieces (it was cut on the document boundaries), and made ten-fold cross validation on these subsets. We used two baseline methods to get a morec lear picture about our results:

C4.5 We used a single C4.5 learner instead of AdaBoostM1 with C4.5 on the token trigger feature set.

Trusted features We used here AdaBoostM1 and C4.5 but on an extremely decreased feature set. We kept only the features which are

thought to be the most significant (triggers, initial letter, predicted class of previous token and other four features).

	P	R	$F_{\beta=1}$
token trigger	97.48	95.74	96.60
phrase trigger	97.17	95.78	96.47
trigramm trigger	97.56	95.89	96.72
voting	98.02	95.82	96.91
C4.5	95.73	94.38	95.05
Trusted features	81.03	79.42	80.15

Table 1: Results of the first iteration and the baseline methods

Table 1 contains the accuracies[1] of the models and baseline methods after the first training phase. Each value in this table is the – size weighted – average of the ten train-test folds. All of the three models learnt on the different trigger features significantly outperformed the baseline methods, which shows the real value of our enriched feature set (against trusted features) and what boosting can achieve.
The results of the three trigger methods are somewhat similar to each other but their predictions are far from identical; consequently they perform well where the other two fail. The accuracy increased in their combination (voting), which confirms this point as well.

	phrase level	token level
First train	98.0/95.8/96.9	99.5/98.5/99.0
Trusted PHI	97.8/96.4/97.1	99.5/98.7/99.1
Standard.	98.1/96.7/97.4	99.5/99.0/99.3

Table 2: Precision/Recall/$F_{\beta=1}$ of the three iterations by two evaluation metrics

All of the results of Table 2 are the final results of the corresponding phase, after the voting of the three models trained with the different trigger methods. We made use of the prediction of the first iteration (the voting row in Table 1) by gathering the word forms of trusted PHIs (see previous section) and adding these lists to the next training phase as new features. In the last phase we standardised the recognised phrases based on the forecast of the previous phase.

[1]In this article we use everywhere the phrase level evaluation metrics introduced at the CoNLL conferences for a NER shared task. The script can be downloaded from the CoNLL website.

Because of the trusted phrases the second trained model tagged more instances to PHI than the first model (resulting in a higher recall) but there were several mistakes among these tagged phrases (the precision decreased). Because the phrase level evaluation metric penalised the partially tagged phrases twice (it effects both precision and recall), the standardisation of the predicted phrases increased both precision and recall.

	prec	rec	$F_{\beta=1}$	pred#	etal#
ID	99.5	99.2	99.33	3670	3678
AGE	100.0	91.7	95.00	12	13
DATE	99.3	99.3	99.25	5191	5193
PHON	100.0	97.3	98.61	170	175
DOC	96.9	94.9	95.88	2635	2690
PAT	96.6	95.9	96.21	683	685
HOSP	95.5	90.1	92.69	1634	1736
LOC	75.8	57.1	63.79	108	144
all	98.1	96.7	97.41	14104	14314

Table 3: The per class accuracies and frequencies of our final, best model

Table 3 gives an overview of the accuracies achieved (after the three iterations) on all PHI classes separately. The most accurate ones are the well-formed classes (ID, AGE, DATE, PHONE), with an $F_{\beta=1}$ measure above 98.5%. This is mainly due to the fact that they can be processed by simple regular expressions and they occur in the same form in the unstructured texts, as seen in the fields of the records (iterative learning utilises this fact).
We made bad predictions on the class LOCATION but considering the complexity of its recognisability and the amount of available training examples (we had less than 200 examples available for this class) it seemed to be really an intractable problem. The performance of the classes DOCTOR, PATIENT, HOSPITAL were similar to the ones we published previously – and described in the related NER works – for Named Entity classes on newswire articles. The better results achieved here on the de-identification task were probably due to the semi-structured characteristics of the documents (iterative learning).
Every learnt model in our experiments was significantly better on precision than recall. It may be because they learn just the more certain patterns (this is strengthened by our voting schema as well). Recall can probably be increased (in the worst case a tradeoff between recall and precision is attainable) by tuning the parameters of C4.5

and AdaBoostM1. We did not perform any parameter fine tuning due to the lack of time.

We consider the above results fairly promising, as they are probably quite near the inconsistency level of the labelling of data we used. We have no information on the agreement rate of the annotators though, which could explain the precision of training data and give a theoretical upper bound for the accuracy of classification.

We should also mention here that we used an evaluation script that implemented a phrase-level evaluation of the labelling using $F_{\beta=1}$ measure. Probably this is not the best fitting evaluation method for the de-identification of medical records, as the removal of *all* PHI is extremely important, so perhaps recall should be given a higher priority. Also, the failure of the removal of one PHI or another is many times not equally serious (consider the failure of the removal of a patient family name and the left of a small part of a hospital name, like "of" in the document – the former seriously conflicts with the HIPAA guidelines, while the latter not). Thus, it is not straightforward to give an ideal evaluation metric for the de-identification task, but we think the phrase-level evaluation we used is still a good characterisation of the quality of our results.

CONCLUSIONS

We introduced a machine learning model which was designed to recognise and classify Named Entities in newswire articles, and was transplantable for the de-identification task with a few extensions: we used two new features (regular expressions for the well defined classes and subject heading information) and we introduced a novel iterative learning approach which was inspired mainly by the characteristic of the discharge records (they are semi-structured).

Our model achieved 97.4% $F_{\beta=1}$ accuracy by tenfold validation which shows the success of the transplantation. We would like to emphasize here again that we reached this competitive result without any deep parsing information (even POS codes) and without any domain specific resources. Our success is probably due to the very rich token level feature set we collected, hence we think that our system can be used (or easily adapted) to other domains as well.

Similarly, the iterative learning seems to be a promising approach for every document type that consists of parts with different characteristics (like the discharge records having structured and unstructured parts).

As the system we constructed was trained and tested on a data set that contained re-identified PHIs, it's quite certain that our model would perform even better in a real-life application. We state this based on two facts: First, some features that would undoubtedly help the recognition of real PHI (like a list of possible first names for example) fail on the re-identified PHI in this dataset. Second, the artificially increased ambiguity of re-identified PHIs made this task particularly challenging and the results on such data is probably somewhat poorer.

Address for Correspondence

György Szarvas, University of Szeged, Department of Informatics, 6720 Szeged, Árpád tér 2., Hungary,
E-mail: szarvas@inf.u-szeged.hu

References

[1] Latanya Sweeney. Replacing personally-identifying information in medical records, the scrub system. In *Cimino, J., ed. Proceedings, Journal of the American Medical Informatics Assoc*, pages 333–337. Hanley & Belfus, 1996.

[2] Tawanda Sibanda, Ozlem Uzuner, and Ozlem Uzuner. Role of local context in automatic deidentification of ungrammatical, fragmented text. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 65–73, New York City, USA, June 2006. Association for Computational Linguistics.

[3] Nancy Chinchor. Muc-7 named entity task definition. In *Proceedings of the Seventh Message Understanding Conference (MUC-7)*, 1998.

[4] Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In Walter Daelemans and Miles Osborne, editors, *Proceedings of CoNLL-2003*, pages 142–147. Edmonton, Canada, 2003.

[5] Y. Tsuruoka J-D. Kim, T. Ohta and Y. Tateisi. Introduction to the bio-entity recognition task at jnlpba. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA), Geneva, Switzerland*, 2004.

[6] Gy. Szarvas, R. Farkas, and A. Kocsor. A multilingual named entity recognition system using boosting and c4.5 decision tree learning algorithms. *DS2006, LNAI*, 4265:267–278, 2006.

[7] Xavier Carreras Luís Màrquez, Adrià de Gispert and Lluís Padro. Low-cost named entity classification for catalan: Exploiting multilingual resources and unlabeled data. *Proceedings of Workshop on Multilingual and Mixed-language Named Entity Recognition, ACL 2003*, pages 25–32, 2003.

[8] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann, second edition, June 2005.