On-time clinical phenotype prediction based on narrative reports

Cosmin A. Bejan, PhD[1], Lucy Vanderwende, PhD[2,1], Heather L. Evans, MD, MS[3], Mark M. Wurfel, MD, PhD[4], Meliha Yetisgen-Yildiz, PhD[1,5]

[1]Biomedical and Health Informatics, School of Medicine, University of Washington, Seattle, WA; [2]Microsoft Research, Redmond, WA; [3]Department of Surgery, School of Medicine, University of Washington, Seattle, WA; [4]Pulmonary and Critical Care Medicine, School of Medicine, University of Washington, Seattle, WA; [5]Department of Linguistics, University of Washington, Seattle, WA

Abstract

In this paper we describe a natural language processing system which is able to predict whether or not a patient exhibits a specific phenotype using the information extracted from the narrative reports associated with the patient. Furthermore, the phenotypic annotations from our report dataset were performed at the report level which allows us to perform the prediction of the clinical phenotype at any point in time during the patient hospitalization period. Our experiments indicate that an important factor in achieving better results for this problem is to determine how much information to extract from the patient reports in the time interval between the patient admission time and the current prediction time.

Introduction

The narrative clinical notes in electronic medical records (EMRs) are very important sources of information that capture the progress of the overall clinical state of patients. The clinical information encoded in these narrative reports, however, is as yet unstructured and therefore cannot be accessed by software applications in a manner similar to accessing the structured clinical data. In general, to facilitate the automatic extraction of such information from clinical notes, natural language processing (NLP) technologies are employed.[1][2]

To address the need of identifying large cohorts of critically ill patients for clinical and translational studies, the focus of automatic extraction clinical notes has been on identifying complex illness phenotypes. In this paper, we describe a clinical information extraction system that is able to automatically discover clinical phenotypes of intensive care unit (ICU) patients. More specifically, the system employs natural language processing and machine learning technologies to process and analyze the clinical reports associated with each ICU patient from our database in order to predict whether or not the patient exhibits a specific phenotype. Although in this study we focus on pneumonia identification, the methodologies we propose do not require phenotype specific clinical expertise and therefore they can be easily adapted to detect other phenotypes as well with sufficient annotated training data. Furthermore, the system is able to predict whether a patient is positive or negative for a specific phenotype at any point in time during the patient hospitalization period. Consequently, in this scenario, the only reports available to the system are the ones with a timestamp on or before the timepoint for which the prediction is performed. Therefore, one of the main challenges for this application is to make an accurate prediction when only a limited set of reports is available.

Phenotype prediction can be used in hospital surveillance applications to automatically assess the clinical condition of patients in real-time. Phenotype prediction, and therefore identification can also be employed in clinical and translational research studies where large cohorts of patients exhibiting a particular phenotype need to be identified in EMRs. Examples of potential applications using such large cohorts of patients are studies of disease-disease and disease-drug interactions as well as studies on identifying genetic variations associated with diseases over time.[3-5]

Our contributions in this paper are as follows. First, we annotated a new dataset of clinical reports corresponding to a cohort of 100 ICU patients. For this dataset, a decision regarding pneumonia was made for each of the reports associated with the patient, i.e., the phenotype was annotated at the report level, in contrast to our previous work[6-8] where annotation was performed at the patient level. Second, based on the dataset described above, we designed a novel learning architecture to compute the prediction for a patient being positive or negative for pneumonia at a given timestamp. The fact that the annotations were performed at the report level allows us to explore whether the information more recent to the prediction time is more relevant to the task of on-time phenotype identification than the information from reports with an older timestamp. And third, we performed detailed experiments in order to

determine the best configuration setup of the prediction system. These experiments validate the robustness of assertion classification and statistical feature selection for on-time phenotype identification – two methodologies we previously developed for phenotype identification.[6][8]

Related Work

Fiszman et al. developed one of the first systems to use clinical notes for the task of pneumonia identification.[9] The main purpose of their system (called SymText) is to automatically identify pneumonia related concepts from chest x-ray reports. The results achieved by SymText were similar to those of a physician and superior to other methods. Furthermore, the same group of researchers used the output generated by SymText in order to identify chest x-ray reports that support acute bacterial pneumonia.[10] In another research project, Mendonça, Haas, and colleagues showed how a general NLP system and clinical rules can be employed to improve the automated surveillance for the presence of healthcare associated pneumonia in neonates.[11][12]

Recently, our research group reported results on pneumonia identification using various types of clinical reports (e.g., admit notes, ICU daily progress notes, cardiology progress notes, discharge summaries) corresponding to a cohort of 426 patients.[7] Based on the fact that the dataset corresponding to the 426 patients was annotated at the patient level, the main research question we addressed is how accurately a patient a positive for pneumonia can be identified at a given point in time when only the set of reports timestamped on or before the specific timepoint are available. To solve this problem, we built a sequence of supervised classifiers, where each classifier corresponds to a timepoint of interest (for example, "pneumonia on day 6") and each data instance consists of a restricted set of time-ordered reports.

In our current work, however, the research problem is to identify whether an ICU patient is positive or negative for pneumonia at any given timepoint. With annotations at the report level, it now becomes possible to tackle this problem. One of the main characteristics when building the classification framework is the fact that we can encode a data instance with information corresponding to both patients and timepoints of interest. In the section describing the system architecture, we present additional details on how a data instance is represented.

Dataset

As mentioned briefly in the introduction, the dataset considered in this study consists of various types of clinical reports corresponding to a cohort of 100 ICU patients. The retrospective review of the reports was approved by the University of Washington Human Subjects Committee of Institutional Review Board.

To create this dataset, we first projected the reports associated with each ICU patient on a special timeline of reports. Each element of this timeline is associated with a set of reports having their corresponding timestamp in a specific time interval as follows. The first timeline element, denoted as ICU Day 0, is associated with all the reports timestamped on the day when the patient was admitted into the ICU. This first timeline element also contains the admit note(s) of the patient. In a similar manner, we assigned reports to the timeline elements from ICU Day 1 to ICU Day 6 based on the day when these reports were created. Finally, we assigned all the reports timestamped after the 7th ICU day as well as the discharge summary of the patient to the final timeline element.

Based on this assignment, the reports associated with an ICU patient were then analyzed by a clinical expert to determine whether they contain relevant information which indicates the presence of pneumonia. If, from the set of reports associated with a specific timeline element, the clinical expert found at least one report indicating the presence of pneumonia, then the corresponding timeline element of the patient was labeled as *positive*. Otherwise, it was labeled as *negative*. Due to the fact that the clinical expert considered the reports for one timeline element in isolation, it is possible for a timeline element to have been annotated as positive for pneumonia while a subsequent timeline element was annotated as negative. For instance, the annotation for patient A in our dataset is *Annotation*(patient A) = {ICU Day 0 = *negative*, ICU Day 1 = *negative*, ICU Day 2 = *positive*, ICU Day 3 = *positive*, ICU Day 4 = *negative*, ICU Day 5 = *negative*, ICU Day 6 = *negative*, ICU Day 7 = *positive*}. To account for this discrepancy, we performed a relabeling of the timeline elements such that, once a patient was labeled as positive for a specific timeline element, he or she will remain positive for the rest of the elements. After relabeling the annotations for patient A presented above, the labels corresponding to the timeline elements ICU Day 4, ICU Day 5, and ICU Day 6 become positive.
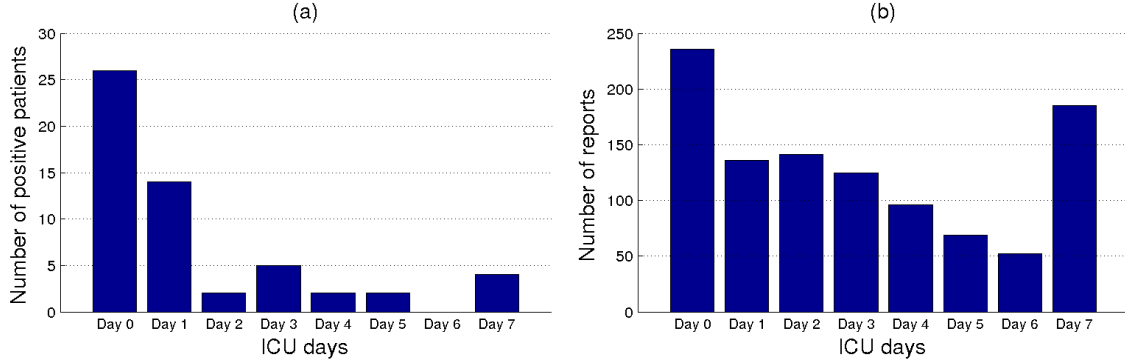
Figure 1. (a) The distribution of patients with respect to the ICU day when they were first identified as positive patients for pneumonia. (b) The distribution of reports with timestamps in a specific time interval.

Using the methodology described above, 55 of the patients were annotated as having at least one positive timeline element (i.e., 55 positive patients) whereas the remaining 45 patients were annotated as patients negative for pneumonia. The plot in Figure 1(a) shows the distribution of the 55 patients according to their earliest timeline element annotated as positive. During the annotation process, the clinical expert also annotated the type of pneumonia for each positive case, although, we grouped all the pneumonia cases into a single class. However, the majority of positive patients were identified as having community acquired pneumonia. This is also in concordance with the distribution shown in Figure 1(a), where most of the patients were identified as having pneumonia in the first 48 hours since ICU admission.

Overall, the dataset consists of 1040 reports. Most of these reports are ICU input records (474), admit notes (208), and discharge summaries (103). Of note, not all the patients from the dataset have a specific report type. For instance, only 95% of patients had admit notes, 99% had ICU input records, and 89% had discharge summaries. The reason for this is that some of the patients may have been transferred to ICU from other medical units or they stayed in ICU for a short period of time. The plot in Figure 1(b) shows the report distribution for each of the timeline elements considered. From the plot, we observe that more reports were written in the first days of ICU stay which may be explained by more detailed clinical examinations being performed for each patient during this period. Another explanation for the shape of this distribution is the fact that many patients that do not have pneumonia are gradually discharged from ICU after several days of stay. It is important to recall that ICU Day 7 covers reports timestamped in a larger time interval (i.e., all reports timestamped after the 7th ICU day) as well as all the discharge summary reports.

Approach

Using the information encoded in clinical reports, we developed a supervised learning framework in order to predict whether or not a patient is positive for pneumonia at any given timepoint. However, unlike our previous work[7] where each data instance was associated with a patient, a data instance in this framework is uniquely identified by a patient and the timepoint for which we want to make the prediction. More exactly, each of the possible prediction timepoints is associated with one element on the report timeline and the reports used for performing a specific prediction are selected from the reports corresponding to a restricted set of timeline elements.

Figure 2 illustrates how data instances are represented for the patient A who is first identified as positive for pneumonia on ICU Day 2. By definition, all the elements after the first time element where the patient is positive, i.e. ICU Day 2 in Figure 2, are required to have a positive label. To represent the data instance for ICU Day 3 as indicated in this figure, the only available reports are the reports corresponding to this current timeline element, ICU Day 3, as well as the reports corresponding to all elements before ICU Day 3. It is important to mention that we cannot represent a data instance using reports timestamped after the prediction time (after ICU Day 3 in this example) since a surveillance application using this framework in real-time for any patient in the intensive care unit does not have access to his/her future reports. We denote the instance from this example as *Instance*(patient A, ICU Day 3) and assign the label it as positive, which is the same as the label of its corresponding timeline element.

Additionally, in order to control the size of the report set associated with each data instance, we introduce a new parameter called LOOKBACK PERIOD, which can take values from 0 to 7. If LOOKBACK PERIOD = 0, each data instance is comprised of only the reports of their corresponding timeline element. In general, if LOOKBACK PERIOD = lp, each
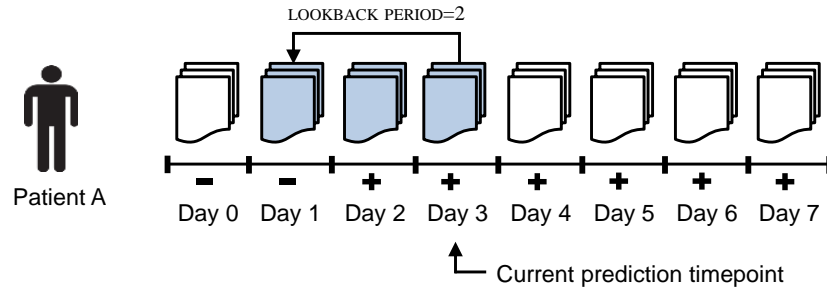
Figure 2 Graphical representation of a data instance for the problem of on-time clinical phenotype prediction.

data instance of the form *Instance*(*P*, ICU Day *X*) will be represented by the set of reports associated with the elements from the interval [ICU Day *X* − *lp*, ICU Day *X*]. Here, *lp* and *X* can take any value between 0 and 7, and *P* can be any patient from our dataset. By convention, for the cases when *lp* > X, we consider all the reports associated with the elements from the interval [ICU Day 0, ICU Day *X*]. As illustrated in Figure 2, the reports corresponding to *Instance*(patient A, ICU Day 3) are depicted with a shaded background. It is also important to mention that the LOOKBACK PERIOD parameter allows us to perform experiments which determine whether the reports with a timestamp closer to the prediction time contain clinical information that is more relevant for predicting pneumonia than the information from the reports with an older timestamp.

Once the data instances are defined, we can now build their corresponding feature vectors from the clinical reports they represent. For this purpose, we used the SPLAT toolkit[13] to split the content of each report into sentences as well as to tokenize each sentence. In this data preprocessing phase, we also extracted the Unified Medical Language System (UMLS) concepts from the clinical reports using MetaMap[14], a tool developed at the National Library of Medicine. In this extraction process, we set the MetaMap parameters such that only the UMLS concepts with the highest mapping score for each match are considered. To perform the classification, we used LIBLINEAR[15], an implementation of the support vector machine algorithm.

The most intuitive method to decide whether a patient is positive for pneumonia or not is (1) to identify in the patient reports the textual expressions that refer to the pneumonia concept and (2) to compute the assertion value associated with these expressions. In this way, we can capture those cases when clinical experts explicitly mention in the patient reports whether the patient is positive or negative for pneumonia. To solve the first desideratum, we selected from the output generated by MetaMap those expressions that were mapped to the pneumonia concept (CUI:C0032285) in the UMLS Metathesaurus. Furthermore, to increase the coverage of these expressions, we run regular expressions to identify *pna* − an abbreviation colloquially used by clinicians in reports and which is not tagged as a pneumonia concept in the UMLS Metathesaurus. For the second desideratum, we employed our state-of-the-art assertion classifier[8], which is able to determine the assertion value corresponding to any medical concept expressed in free text. The task of assertion classification was introduced in 2010 as a shared task[16] within the Integrating Biology and the Bedside (i2b2)/Veteran's Affairs (VA) challenge. This task was formulated such that each medical concept mentioned in a clinical report to be associated with one of the six assertion categories: *present*, *absent*, *conditional*, *hypothetical*, *possible*, and *not associated with the patient*. Examples with text expressions extracted from our dataset that express some of the assertion categories associated with pneumonia are listed in Table 1.

After we assigned an assertion value to each of the pneumonia concepts found in the clinical reports of a data instance, the algorithm counted how many times each of the six values were identified for this specific instance and mapped the most frequent value to one of the two categories for pneumonia identification (i.e., *positive* or *negative*). In our previous research[8] we found that an appropriate mapping for pneumonia identification is {*present*} → positive pneumonia and {*absent*, *possible*, *conditional*, *hypothetical*, *not associated with the patient*} → negative pneumonia. We computed the category labels using the methodology described above and used this information to build a binary feature (called *assert* or *assertion feature*) for each data instance from our dataset. By convention, we assigned a category of negative pneumonia to all data instances for which we could not find at least one pneumonia expression mentioned in their corresponding report set.

Although assertion classification is appealing and intuitive to use, the identification of complex phenotypes such as pneumonia often requires a deeper understanding of the clinical information encoded into narrative reports. For

Table 1 Examples of assertion categories associated with the pneumonia concept.

assertion value	example
present	*Patient remains critically ill with pneumonia and bacteremia in the setting of novel H1N1 influenza.*
	We put him on broad spectrum antibiotics for pneumonia and then drained his pleural effusions.
absent	*Patient was improving without antibiotic treatment and no evidence on chest x-ray for pneumonia.*
possible	*This likely represents atelectasis; however, pneumonia cannot be excluded.*

instance, in many clinical reports from our dataset there are no explicit expressions that mention whether their corresponding patient has pneumonia or not.

To better predict pneumonia from clinical notes, we enhance the feature space by extracting word and UMLS concept n-grams from the clinical reports associated with our data instances. Of note, before extracting the word n-grams, we filtered out the punctuation tokens. Furthermore, in order to use the most informative n-gram features for pneumonia prediction, we employed statistical feature selection as described in our previous work on phenotype identification.[6] Specifically, in this study, we used the *t* statistical test in order to compute the association strength between each n-gram feature from the training set and the two categories for pneumonia identification (i.e., positive and negative pneumonia). This will generate an ordering of the features such that the most informative ones will be ranked on the top of the list whereas the noisy and redundant features for our task will be placed at the bottom of the list. In order to select the most relevant features that improve the results on pneumonia identification, we experimented with various threshold values for each feature type such that only the feature above those threshold values will be used for classification.

Experimental Results

Due to the fact that a data instance in our framework is uniquely identified by a patient and timeline element, the total number of data instances in our dataset is equal to 800 (100 patients × 8 timeline elements). In these 800 instances, 361 are identified as positive and the remaining 439 as negative. Since our main interest is to determine how well we can identify a positive patient for pneumonia at a specific prediction timepoint, we considered F-measure as the primary measure in our experiments; however, we also report results based on negative predictive value and specificity. To evaluate our system, we considered a 5-fold cross validation scheme.

Figure 3 shows the performance results achieved by our system under various feature type combinations and parameter configurations. As baseline systems, we considered configurations of our framework that use various combinations of word and UMLS concept n-grams. Additionally, the F1-measure results achieved by these systems are reported for each possible value of the LOOKBACK PERIOD parameter along the horizontal axis. These baseline results are shown in the top plots of Figure 3. The notations w1, w2, and w3 are used to denote a specific experiment name in these plots corresponding to word unigrams, bigrams, and trigrams, respectively. Similarly, c1, c2, and c3 represent experiments including UMLS concept unigrams, bigrams, and trigrams, respectively. As can be observed in plot (a), the best combination of word n-grams is achieved by the experiment which considers only the word unigrams. Similarly, the results shown in plot (b) indicate that the best combination of UMLS concept n-grams corresponds to the experiment which includes all concept unigrams in the representation of feature vectors. When experimenting with word and concept n-grams, the best combination was achieved by considering both word and concept unigrams (experiment w1+c1 in plot (c)). Moreover, the system using the configuration of this experiment and LOOKBACK PERIOD = 1 achieved 76.46 F1-measure, which is the best performance result corresponding to a baseline configuration.

In (d) we investigated the impact of the assert feature when used in combination with some of the best baseline configurations. As illustrated, all of these experiments lead to results above 80 F1-measure for small values of the LOOKBACK PERIOD parameter indicating a significant improvement over the baseline results. The best result of 82.23 F1-measure was achieved when the assert feature was used in combination with word unigrams and the LOOKBACK PERIOD parameter was set to 1. In this plot, we also investigated how well the system performs when using only the assert feature. The best result of 81.36 F1-measure reached under this configuration indicates that the assert feature is the most salient feature used by our system.
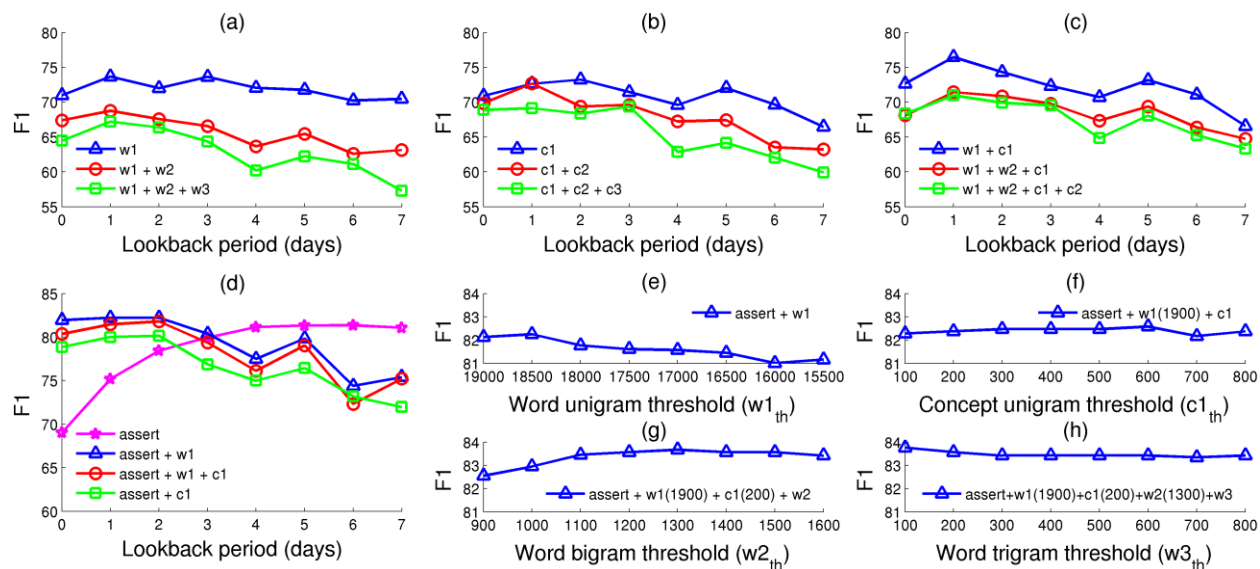
Figure 3 System performance results under different configurations. (a), (b) and (c) show the baseline results of the system when considering various combinations of word and UMLS concept n-grams. (d) shows the experiments performed when using the assertion feature as well as combinations of this feature with word and concept unigrams. Finally, (e), (f), (g), and (h) illustrate the steps of a greedy algorithm for searching the most optimal threshold values which indicate what word and concept n-gram features to be selected in the learning framework.

The experiments in the plots (a), (b), (c), and (d) of Figure 3 also investigate what is the most suitable value for the LOOKBACK PERIOD parameter. This is equivalent with determining the most appropriate value of recent past information to be used for representing each data instance. Most of our experiments reveal the fact that for the n-gram features it is enough to look in the past up to one ICU day (LOOKBACK PERIOD = 1). These empirical studies are also consistent with our intuition that the most recent information is more relevant for better predicting pneumonia. For instance, the fact that a patient identified as positive on day *X* was in normal clinical condition on day *X*–3 does not help in predicting this patient as positive on day *X*. On the other hand, the experiment using only the assert feature in (d) recommends to explore the entire past information for assertion values associated with the pneumonia concept. This is also intuitive since, once a patient is assessed by a clinician as being positive for pneumonia, the patient will remain positive until discharged from ICU.

In the next phase of our quest for the best combination of feature types and parameter configuration, we investigated how statistical feature selection can further improve the performance results. For this purpose, we employed a forward-backward greedy approach to find the best threshold values corresponding to each ranked list of features. The main steps of this greedy approach are shown in the plots (e), (f), (g), and (h) of Figure 3. Based on the observations discussed previously, in the initialization, we fixed the LOOKBACK PERIOD parameter to 1 for any n-gram feature type and to 7 for the assert feature. Also, in this initialization step, the only feature type considered was the assert feature. In the first forward step, the approach automatically added to the initial configuration one feature type at a time. For each such feature type, the approach automatically explored possible threshold values that can improve the best performance result achieved using the default configuration. As depicted in Figure 3(e), the best results in this step were achieved when using a threshold value of 1900 and 18500 for selecting the most relevant word unigrams. Since the total number of word unigrams extracted from our dataset is equal with 22086, this experiment corresponds to selecting almost the entire set of such features. In the next step – the backward step – the algorithm individually removes all the feature types which are different from the feature type added in the previous step in order to determine if any further improvements can be obtained. Since removing the assert feature in this step does not improve the best performing results, the algorithm performs another backward forward step in an iterative manner. Based on this mechanism, the algorithm managed to further improve the results (1) when adding the concept unigrams (the experiment using the assert feature, the most relevant 1900 word unigrams, and concept unigrams in Figure 3(h)), (2) when further adding word bigrams as shown in Figure 3(g), and finally, (3) when also considering word trigrams as illustrated in Figure 3(h).

Table 2 The best performing results for on-time prediction of pneumonia.

Configuration	TP	FP	FN	TN	P	NPV	R	Spec	Acc	F1
ICU day 0	19	6	7	68	76.00	90.67	73.08	91.89	87.00	74.51
ICU day 1	33	12	7	48	73.33	87.27	82.5	80.00	81.00	77.65
ICU day 2	39	10	3	48	79.59	94.12	92.86	82.76	87.00	85.71
ICU day 3	45	12	2	41	78.95	95.35	95.74	77.36	86.00	86.54
ICU day 4	47	16	2	35	74.60	94.59	95.92	68.63	82.00	83.93
ICU day 5	48	16	3	33	75.00	91.67	94.12	67.35	81.00	83.48
ICU day 6	48	15	3	34	76.19	91.89	94.12	69.39	82.00	84.21
ICU day 7	51	10	4	35	83.61	89.74	92.73	77.78	86.00	87.93
Aggregate	330	97	31	342	77.28	91.69	91.41	77.9	84.00	83.76

Acc, accuracy; F1, F1–measure; FN, false negatives; FP, false positives; NPV, negative predictive value; P, precision; R, recall; Spec, specificity; TN, true negatives; TP, true positives.

Table 2 lists the final results corresponding to the best feature combination and parameter configuration achieved the greedy approach we previously described. Specifically, these results correspond to a system configuration which includes: (1) the assert feature, (2) the first 1900 most significant word unigrams, (3) the first 200 most significant concept unigrams, (4) the first 1300 most significant word bigrams, and finally, (5) the first 100 most significant word trigrams. The best result of 83.76 F1-measure is also shown in Figure 3(h) for the first value of the word trigram threshold ($w3_{th}=100$). The first rows of this table list the results of the data instances associated with each timeline element. The worst results are obtained when evaluating the data instances of the ICU day 0 element due to the fact that these instances are constrained to use only the reports associated with this first element on the report timeline. In concordance with our intuition, when evaluating the data instances corresponding the ICU day 7 our system achieved the best results of 87.93 F1-measure. We believe this is because ICU day 7 represents the entire set of discharge summaries, which are special types of reports where clinicians provide the last conclusions about the state of the patients, may describe the diagnostic findings, and provide recommendations on discharge.

Conclusion

We presented a clinical information extraction system which is able to identify whether a patient is positive or negative for pneumonia at any point in time during the patient hospitalization period. The way we defined a data instance for solving this problem allowed us to investigate what is the most relevant clinical information encoded into the patient reports at prediction time. In our experiments, we found that the LOOKBACK PERIOD parameter plays a critical role in determining how much information to extract from reports as well as whether the information more recent to the prediction time is more informative than the information extracted from reports with an older timestamp.

Acknowledgements

References

1. Nadkarni PM, Ohno-Machado L, Chapman WW. Natural language processing: an introduction. J Am Med Inform Assoc. 2011 Sep-Oct;18(5):544-51.
2. Demner-Fushman D, Chapman WW, McDonald CJ. What can natural language processing do for clinical decision support? J Biomed Inform. 2009 Oct;42(5):760-72.
3. Ritchie MD, Denny JC, Crawford DC, Ramirez AH, Weiner JB, Pulley JM, Basford MA, Brown-Gentry K, Balser JR, Masys DR, Haines JL, Roden DM. Robust replication of genotype-phenotype associations across multiple diseases in an electronic medical record. Am J Hum Genet. 2010;86(4):560-72.
4. Ritchie MD, Denny JC, Zuvich RL, Crawford DC, Schildcrout JS, Bastarache L, Ramirez AH, Mosely JD, Pulley JM, Basford MA, Bradford Y, Rasmussen LV, Pathak J, Chute CG, Kullo IJ, McCarty CA, Chisholm RL, Kho AN, Carlson CS, Larson EB, Jarvik GP, Sotoodehnia N, Manolio TA, Li R, Masys DR, Haines JL, Roden DM. Genome- and Phenome-Wide Analysis of Cardiac Conduction Identifies Markers of Arrhythmia Risk. Circulation 2013. In print.

5. Liao KP, Kurreeman F, Li G, Duclos G, Murphy S, Guzman R, Cai T, Gupta N, Gainer V, Schur P, Cui J, Denny JC, Szolovits P, Churchill S, Kohane I, Karlson EW, Plenge RM. Associations of autoantibodies, autoimmune risk alleles, and clinical diagnoses from the electronic medical records in rheumatoid arthritis cases and non-rheumatoid arthritis controls. Arthritis Rheum. 2013 Mar;65(3):571-81.

6. Bejan CA, Xia F, Vanderwende L, Wurfel MM, Yetisgen-Yildiz M. Pneumonia identification using statistical feature selection. J Am Med Inform Assoc. 2012;19(5):817-23.

7. Bejan CA, Vanderwende L, Wurfel MM, Yetisgen-Yildiz M. Assessing pneumonia identification from time-ordered narrative reports. AMIA Annu Symp Proc. 2012;2012:1119-28.

8. Bejan CA, Vanderwende L, Xia F, Yetisgen-Yildiz M. Assertion modeling and its role in clinical phenotype identification. J Biomed Inform. 2013;46(1):68-74.

9. Fiszman M, Chapman WW, Aronsky D, Evans RS, Haug PJ. Automatic detection of acute bacterial pneumonia from chest x-ray reports. J Am Med Inform Assoc 2000;7:593–604.

10. Chapman WW, Fizman M, Chapman BE, Haug PJ. A comparison of classification algorithms to automatically identify chest x-ray reports that support pneumonia. J Biomed Inform 2001;34(1):4–14.

11. Mendonça EA, Haas J, Shagina L, Larson E, Friedman C. Extracting information on pneumonia in infants using natural language processing of radiology reports. J Biomed Inform 2005;38(4):314–21.

12. Haas JP, Mendonça EA, Ross B, Friedman C, Larson E. Use of computerized surveillance to detect nosocomial pneumonia in neonatal intensive care unit patients. Am J Infect Control 2005;33(8):439–43.

13. Quirk C, Choudhury P, Gao J, Suzuki H, Toutanova K, Gamon M, Yih W, Vanderwende L, Cherry C. MSR SPLAT, a language analysis toolkit. NAACL HLT 2012 demonstration session.

14. Aronson AR. Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. Proc AMIA Symp 2001:17–21.

15. Fan RE, Chang KW, Hsieh CJ, Wang XR, Lin CJ. LIBLINEAR: A library for large linear classification. J Mach Learn Res 2008;9:1871–4.

16. Uzuner Ö, South BR, Shen S, DuVall SL. 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text. J Am Med Inform Assoc 2011;18(5):552–6.