Predicting Chief Complaints at Triage Time in the Emergency Department

Yacine Jernite, Yoni Halpern
New York University
New York, NY
{jernite,halpern}@cs.nyu.edu

Steven Horng
Beth Israel Deaconess Medical Center
Boston, MA
shorng@bidmc.harvard.edu

David Sontag
New York University
New York, NY
dsontag@cs.nyu.edu

Abstract

As hospitals increasingly use electronic medical records for research and quality improvement, it is important to provide ways to structure medical data without losing either expressiveness or time. We present a system that helps achieve this goal by building an extended ontology of chief complaints and automatically predicting a patient's chief complaint, based on their vitals and the nurses' description of their state at arrival.

1 Introduction

While recent years have seen an increase in the adoption of Electronic Medical Records (EMRs) and an interest in using them to improve the quality of care in hospitals, there is still considerable debate as to how best to capture data for both clinical care and secondary uses such as research, quality improvement, and quality measurement. Unstructured data (free text) is preferred by clinicians because it is more expressive and easier to input. Structured data is preferred by researchers and administrators because it can easily be used for secondary analysis.

In the emergency department, a patient's chief complaint represents their reason for the visit. It has the potential to be used to subset patients into cohorts, initiate decision support, and perform research. However, it is routinely collected as free text. The need to collect chief complaints as structured data has been advocated for by every Emergency Medicine organization [8]. However, an appropriate chief complaint ontology will consist of over 2,000 terms, making manual input of structured data difficult, if not impossible.

In this extended abstract, we present a novel use of natural language processing and machine learning that is able to utilize already collected unstructured clinical data to make collection of structured chief complaint data more efficient and reliable.

1.1 Clinical problem

When a patient arrives at the Emergency Department (ED), they are processed at the triage station by a nurse who writes a note summarizing their state (e.g. medical history, symptoms) and a chief complaint used to assign them to the right pathway. We focus on the latter step in this work, building a system that learns to predict the chief complaint automatically from the summary of the patient's state (the triage note), and building an extended ontology to support it.

Since we want our system to be used in a practical setting, the two following requirements must be met: the user must feel that the software actually saves them time, and that its results can be trusted. The first item leads to a requirement that the program runs instantaneously and that the user interface be well-designed. The second item necessitates that the system be correct most of the time, and that it never give shocking results (we will give an example of such an unwanted behavior in Section 2).

An important benefit of the system is that it transforms the chief complaints field in the EMR from free text to a categorical variable in a way that actually saves time (as opposed to simply having the nurse choose the complaint from an extended list), and makes the chief complaints easier to use in other systems at later stages of the patient's stay in the ED.

1.2 Related work

The present work is set in a context of growing interest for the applications of medical Natural Language Processing. A variety of software such as cTAKES [15], MedLee [6], NegEx [2] and MedConcepts [9] perform numerous NLP tasks, such as dependency parsing, negation detection or concept recognition, specifically on medical text. We focus here on applying NLP and machine learning methods to predicting chief complaints. This will allow us to improve the quality of chief complaints, by enabling use of a large standardized coding system in a practical way.

Chief complaints are widely used for a variety of applications. For example, in syndromic surveillance, Chapman et al. [3] used them to monitor the 2002 Winter Olympic Games, and Mandl et al. [12] proposed to take advantage of chief complaints for early detection of fast-spreading diseases. Another application of chief complaints is for improving diagnosis and triage, since they can be used as variables within prediction algorithms and to initiate clinical pathways. For example, Aronsky and Haug [1] use chief complaints in a Bayesian network for diagnosis of community-acquired pneumonia, and Goldman et al. [7] rely on them to predict myocardial infarctions in ED patients. Finally, chief complaints are used to retrospectively analyze clinical data for research purposes, such as to study the prevalence of pain in the ED, as in Cordell et al. [4], or the factors that lead to missing diagnoses of myocardial infarction, as in McCarthy et al. [13].

In contrast to the typical work on using chief complaints, which focuses on natural language processing of chief complaints, we completely change the workflow, providing context-specific algorithms to enable rapid natural language-based entry of coded chief complaints. This is related to the work of Pakhomov et al. [14] on mapping diagnoses to ICD-9 (International Classification of Diseases) codes, and that of Larkey and Croft [11], who assign ICD codes to discharge summaries, although the structures of the ontologies and machine learning techniques used are quite different from ours.

2 Approach

We decided to formalize the task of learning to predict the chief complaints for a patient as a multiclass learning problem; indeed, although a patient may come to the ED for more than one reason and thus have multiple labels, more than 4 in 5 actually have a single chief complaint. To that end, we chose to train a linear Support Vector Machine (SVM) on a bag-of-words representation of the triage notes. One useful feature of the linear SVM is that it makes it easy to see which words were most important in the decision, and makes analysis of the results much easier.

For each concept in our new ontology we specified an example chief complaint (e.g. '*SEIZURE*') which resulted in labeled examples to use for training. In order for the SVM to work as intended, we had to deal with the following two issues. First, we realized that some chief complaints appeared too rarely for the SVM to learn to predict them. This was fixed by extending our ontology of chief complaints to have all descriptions of the same concept (such as *'SEIZURES'*, *'S/P SEIZURE'*, *'S/P SZ'*, *'SZ'*, *'SEIZURE'*) linked to a single label (*SEIZURE*).

Second, we observed a few errors that we believed would hurt the credibility of the system when used in a practical setting. One example is the following note:

> pt here with complains severe sudden onset abd pain, nausea and vomiting, blood in emesis, **no black or bloody stools**

The chief complaints for this note were *['N/V', 'ABD PAIN']*, but our system predicted the 5 most likely labels to be *[***'BLOOD IN STOOL/MELENA'***, 'ABD PAIN', 'ST', 'ABDOMINAL PAIN', 'H/A']*. Predicting a chief complaint that is explicitly negated in the text seems to be an egregious mistake to a human, but it is a direct consequence of the bag-of-words assumption of the model. Because of this, we had to add a pre-processing step of negation detection, which we describe and report results for in Section 3.

3 Experiments

3.1 Experimental setup

We developed our system on a dataset of 97000 triage notes with an average 1.2 chief complaints reported per note. We separated this data into a training set of 58000 notes and a validation set (to choose the regularization parameter of the SVM) and test set of 19500 notes each.

To address some of the failings of the bag-of-words assumption, we applied the following pre-processing steps to our data. First, we detected and aggregated significant bi-grams, such as "mental status" or "shortness of breath". We then looked at the performance of three negation detection systems: NegEx [2], a NegEx-like system to which we added a few rules tailored to our data, and a perceptron classifier trained to predict the scope of a negation. The perceptron performed best, as shown in Table 1, and we applied it as a second pre-processing step. In addition to the performance gain, the main advantage of the perceptron is that it can easily be re-trained to adapt to a new hospital without needing an expert to design their own set of rules to complement the NegEx system.

	NegEx	added rules	perceptron
Precision	0.699	0.833	0.901
Recall	0.875	0.982	0.925
F1	0.777	0.901	0.913

Table 1: Performance of the different negation detection algorithms on 200 test sentences.

We then used the improved bag-of-words representation of the text, as well as vital signs measured at triage (temperature, blood pressure, etc.), within two learning systems. The first treats the problem as a multiple label prediction task and tries learning a binary SVM classifier [10] for each of the chief complaints, comparing their outputs to sort the labels from most to least likely. The second consisted of a single multiclass SVM [5] which automatically provides such a ranking.

Table 2 compares the performance of both systems according to two measures. The Best-n accuracy measures how often the list of n most likely predicted labels actually contained all of the true chief complaints, and DCG stands for the Discounted Cumulative Gain, which measures the quality of the whole ranking. Both measures show that the multiclass SVM performs much better, resulting in our choosing it to build our final system.

	many-to-one		multiclass SVM	
negation detection	none	perceptron	none	perceptron
Best-5	0.496	0.511	0.753	0.757
Best-10	0.615	0.620	0.819	0.825
DCG	0.381	0.393	0.601	0.613

Table 2: Performance of the linear SVMs on chief complaint prediction.

3.2 Live application

Our initial goal was to propose for each patient a set of 10 possible chief complaints for the nurses to select from. However, the user might feel that the software doesn't actually help if they have to go through the list of proposals and still input the right answer manually whenever the system fails. To remedy this situation, we decided to only propose the 5 best guesses of the system, and to additionally set up an intelligent auto-complete for the case when the nurse still wants to input their

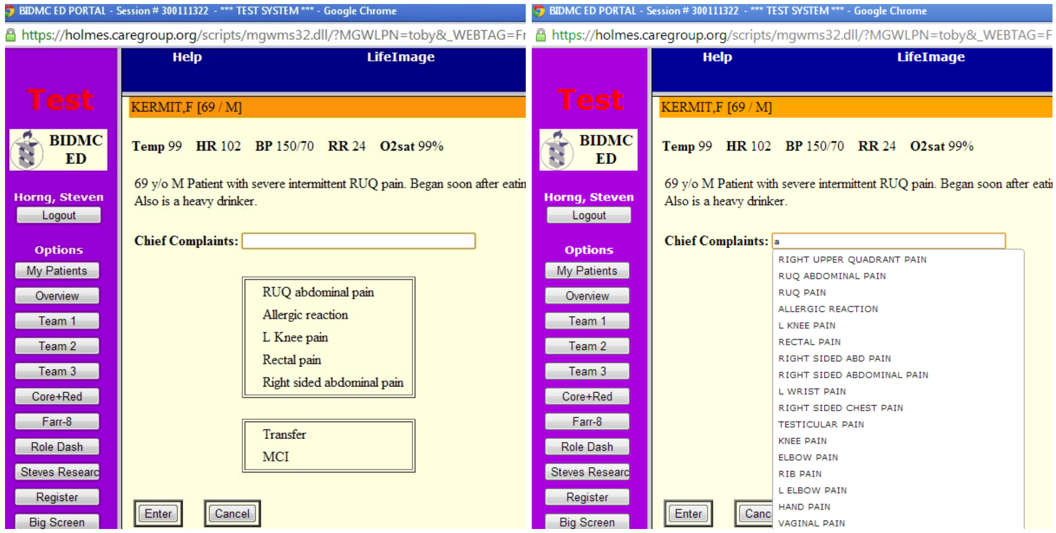

Figure 1: Screenshots of the system now running at BIDMC hospital on note : *69 y/o M patient with severe intermittent RUQ pain. Began soon after eating bucket of ice cream and cupcake. Also is a heavy drinker.* **Left:** the system correctly proposes both 'RUQ abdominal pain' and 'Allergic reaction' as possible chief complaints. **Right:** If the nurse does not see the label they want, they can start typing and see a list of suggested auto-completes. Again, the four most likely labels describe 'RUQ abdominal pain' and 'Allergic reaction'.

answer, based on the ranking of chief complaints output by the SVM. An example of the interface is presented in Figure 1 (the patient name is anonymized).

Getting our system to be usable in a practical setting required two further improvements. First, the initial system took about 5 seconds per note, far longer than the triage nurses' patience. Using Python's `shelve` package to store the SVM weights as a persistent dictionary brought this time down to about 200 ms. We also discovered that a small set of patients are taken in without a triage note, but still need to be assigned a chief complaint. We added the absence of text as a feature for the SVM, which allowed for a better ranking of chief complaints for the auto-complete interface.

4 Conclusion

In this work, we proposed a system to predict a patient's chief complaints based on a description of their state. Applied in a real-world setting, this provides us with a useful classification of patients which can be used for other tasks, without slowing down the triage process.

While our algorithm already provides results that are good enough to be of use in practice, we hope to add some new features in the future. One notable direction that would have benefits similar to those of negation detection is time resolution, and it is an issue we are planning to address next.

Finally, recall that while we built the current system on noisily annotated data, where we had to manually transform some of the labels, its use will create a much cleaner dataset, which we plan to use in many downstream applications.

References

[1] D. Aronsky and P. J. Haug. Diagnosing community-acquired pneumonia with a bayesian network. In *Proceedings of the AMIA Symposium*, page 632, 1998.

[2] W. W. Chapman, W. Bridewell, P. Hanbury, G. F. Cooper, and B. G. Buchanan. A simple algorithm for identifying negated findings and diseases in discharge summaries. *Journal of biomedical informatics*, 34(5):301–310, 2001.

[3] W. W. Chapman, L. M. Christensen, M. M. Wagner, P. J. Haug, O. Ivanov, J. N. Dowling, and R. T. Olszewski. Classifying free-text triage chief complaints into syndromic categories with natural language processing. *Artificial intelligence in medicine*, 33(1):31–40, 2005.

[4] W. H. Cordell, K. K. Keene, B. K. Giles, J. B. Jones, J. H. Jones, and E. J. Brizendine. The high prevalence of pain in emergency medical care. *The American journal of emergency medicine*, 20(3):165–169, 2002.

[5] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *The Journal of Machine Learning Research*, 2:265–292, 2002.

[6] C. Friedman. Medlee-a medical language extraction and encoding system. *Columbia University, and Queens College of CUNY*, 1995.

[7] L. Goldman, E. F. Cook, D. A. Brand, T. H. Lee, G. W. Rouan, M. C. Weisberg, D. Acampora, C. Stasiulewicz, J. Walshon, G. Terranova, et al. A computer protocol to predict myocardial infarction in emergency department patients with chest pain. *New England Journal of Medicine*, 318(13):797–803, 1988.

[8] S. W. Haas, D. Travers, J. E. Tintinalli, D. Pollock, A. Waller, E. Barthell, C. Burt, W. Chapman, K. Coonan, D. Kamens, et al. Toward vocabulary control for chief complaint. *Academic Emergency Medicine*, 15(5):476–482, 2008.

[9] P. Jindal and D. Roth. Using knowledge and constraints to find the best antecedent. In *Proceedings of International Conference on Computational Linguistics (COLING)*, pages 1327–1342, 12 2012.

[10] T. Joachims. Training linear svms in linear time. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 217–226. ACM, 2006.

[11] L. S. Larkey and W. B. Croft. Automatic assignment of icd9 codes to discharge summaries. *Center for Intelligent Information Retrieval Technical Report*, 1995.

[12] K. D. Mandl, J. M. Overhage, M. M. Wagner, W. B. Lober, P. Sebastiani, F. Mostashari, J. A. Pavlin, P. H. Gesteland, T. Treadwell, E. Koski, et al. Implementing syndromic surveillance: a practical guide informed by the early experience. *Journal of the American Medical Informatics Association*, 11(2):141–150, 2004.

[13] B. D. McCarthy, J. R. Beshansky, R. B. D'Agostino, and H. P. Selker. Missed diagnoses of acute myocardial infarction in the emergency department: results from a multicenter study. *Annals of emergency medicine*, 22(3):579–582, 1993.

[14] S. V. Pakhomov, J. D. Buntrock, and C. G. Chute. Automating the assignment of diagnosis codes to patient encounters using example-based and machine learning techniques. *Journal of the American Medical Informatics Association*, 13(5):516–525, 2006.

[15] G. K. Savova, J. J. Masanz, P. V. Ogren, J. Zheng, S. Sohn, K. C. Kipper-Schuler, and C. G. Chute. Mayo clinical text analysis and knowledge extraction system (ctakes): architecture, component evaluation and applications. *Journal of the American Medical Informatics Association*, 17(5):507–513, 2010.