Automated concept-level information extraction to reduce the need for custom software and rules development

Leonard W D'Avolio,[1,2,3] Thien M Nguyen,[1] Sergey Goryachev,[1] Louis D Fiore[1,4,5]

[1]Massachusetts Veterans Epidemiology Research and Information Center (MAVERIC) Cooperative Studies Coordinating Center, VA Boston Healthcare System, Jamaica Plain, Massachusetts, USA
[2]Center for Surgery and Public Health, Brigham and Women's Hospital, Boston, Massachusetts, USA
[3]Department of Medicine, Division of Aging, Brigham and Women's Hospital and Harvard Medical School, Boston, Massachusetts, USA
[4]Boston University School of Public Health, Boston, Massachusetts, USA
[5]Boston University School of Medicine, Boston, Massachusetts, USA

Correspondence to
Dr Leonard W D'Avolio, 150 S Huntington Ave, MAVERIC (151 MAV), VA Boston Healthcare System, Jamaica Plain, MA 02130, USA;
leonard.davolio@va.gov

Received 14 February 2011
Accepted 24 May 2011
Published Online First
22 June 2011

ABSTRACT

Objective Despite at least 40 years of promising empirical performance, very few clinical natural language processing (NLP) or information extraction systems currently contribute to medical science or care. The authors address this gap by reducing the need for custom software and rules development with a graphical user interface-driven, highly generalizable approach to concept-level retrieval.

Materials and methods A 'learn by example' approach combines features derived from open-source NLP pipelines with open-source machine learning classifiers to automatically and iteratively evaluate top-performing configurations. The Fourth i2b2/VA Shared Task Challenge's concept extraction task provided the data sets and metrics used to evaluate performance.

Results Top F-measure scores for each of the tasks were medical problems (0.83), treatments (0.82), and tests (0.83). Recall lagged precision in all experiments. Precision was near or above 0.90 in all tasks.

Discussion With no customization for the tasks and less than 5 min of end-user time to configure and launch each experiment, the average F-measure was 0.83, one point behind the mean F-measure of the 22 entrants in the competition. Strong precision scores indicate the potential of applying the approach for more specific clinical information extraction tasks. There was not one best configuration, supporting an iterative approach to model creation.

Conclusion Acceptable levels of performance can be achieved using fully automated and generalizable approaches to concept-level information extraction. The described implementation and related documentation is available for download.

INTRODUCTION

Over 40 years of empirical evidence generated from individual studies as well as community-wide shared task challenges have demonstrated the potential of natural language processing (NLP) to facilitate the secondary use of electronic medical record data. Despite repeated demonstrations of capable performance, few systems have migrated beyond the laboratories of their creators. As a result, clinically meaningful use of NLP technologies has been mostly limited to the collaborators of NLP developers and researchers. Meanwhile, the supply of electronic medical record data continues to grow,[1] as do the demands for secondary uses of it for important goals such as quality measurement,[2] comparative effectiveness,[3] evidence-based medicine,[4] and phenotyping for genomic analysis.[5]

In response, the informatics team at the Massachusetts Veterans Epidemiology Research and Information Center has focused not on algorithms that perform optimally for a single information extraction task, but on algorithms and end-to-end workflows capable of performing well across a number of tasks while minimizing the burden on system developers and end users. In a previous study, we demonstrated the potential of algorithms that combine the results of an open-source NLP pipeline with open-source supervised machine learning classifiers to perform document-level information retrieval. The algorithm and end-to-end workflow were implemented as the open-source proof of concept, Automated Retrieval Console (ARC).[6]

In this study, we evaluate algorithms for achieving fully automated and generalizable concept-level information retrieval. If successful, this approach will enable researchers to move beyond the identification of relevant documents (eg, discharge summaries of patients with post-traumatic stress disorder) to the identification of relevant concepts (eg, symptoms of post-traumatic stress disorder) while limiting the end-user burden to the supply of relevant examples. As a result, we hope to reduce the need for custom software or rules development, enabling more widespread diffusion of these capabilities. Evaluations of these algorithms are performed using data and metrics from the concept extraction portion of the Fourth i2b2/VA Shared Task Challenge, which focused on the extraction of medical symptoms, treatments, and tests from the free text of medical records.[7]

BACKGROUND

The potential of machines to help parse through volumes of unstructured clinical free text in search of important information has been systematically explored since at least 1967.[8] Since that time, several different approaches have yielded positive results, including the creation of custom-written rules to detect specific patterns in text, the development of grammatical systems[9 10] that provide both shallow and deep parsing of text, and the use of machine learning to support several aspects of information extraction.[11 12] Yet despite at least 40 years of promising performance from several different approaches and dramatic advances in computing capabilities and methods, very few clinical NLP and information-extraction systems have diffused beyond the laboratories of their creators. While surely a range of barriers have prevented the widespread adoption of such

technologies, the inability of such systems to generalize beyond a specific task is a significant factor. In their review of 113 automated clinical coding and classification system-related articles, Stanfill *et al* concluded that, 'Automated coding and classification systems themselves are not generalizable, nor are the results of the studies evaluating them.'[8]

Several recent advances have made it possible to achieve strong performance without custom software or rules development. The increased availability of open-source NLP, information retrieval, text mining, and machine learning software has made it possible for researchers and systems developers to more quickly develop and effectively share task-specific modules such as part of speech taggers and negation detectors. Open-source frameworks such as the Generalized Architecture for Text Engineering[13] and the Unstructured Information Management Architecture (UIMA)[14] have provided a chassis onto which developers can bolt task-specific modules or create general clinical NLP pipelines such as the Clinical Text Analysis and Knowledge Extraction System (cTAKES).[15] Also important are the shared task challenges[7 16–19] which have enabled apples-to-apples comparisons of various approaches and have spurred advances in our understanding of information retrieval. Among the lessons learned from such competitions is the potential of the combination of robust feature sets and supervised machine learning classifiers to work well across a number of tasks.[7 17 20] The release of data sets to the NLP and data-mining communities[21–23] has recently made it possible for researchers and developers to evaluate the potential of their approaches more quickly and effectively and for their findings to be validated. Finally, the annotation of data sets is increasingly performed using the open-source Knowtator,[24] which is built on the open-source ontology engine Protégé,[25] allowing for easier sharing of annotations and making it possible to automate the import of reference sets.

We hypothesize that the use of NLP-derived features combined with supervised machine learning can perform effectively across tasks. The classifier used in this experiment is an open-source implementation of Conditional Random Fields (CRFs). A CRF is an undirected graphical model with edges representing dependencies between variables.[26] Peng and McCallum showed that CRFs can outperform Support Vector Machines,[27] and Wellner *et al* showed the ability of CRFs to achieve high levels of performance in the deidentification of personal health identifiers.[28] The features used for classification are supplied by the open-source clinical NLP pipeline, cTAKES, which maps free text to over 90 different data types. UIMA exposes all user-defined data types and the expected input and output of modules within UIMA pipelines (ie, aggregate analysis engines), making it possible to drag-and-drop UIMA pipelines into ARC and expose a pipeline's results as features for classification.

In our previous work on automated document-level classification,[6] all documents were processed using cTAKES, resulting in over 90 different data types. These data types and their associated values were used as features in supervised machine learning classification. Two different feature-selection algorithms were evaluated; one that combines top scoring individuals features and negated features, and another that evaluates all combinations of the top five individual performing features. The latter, referred to for convenience as the 'Blast' algorithm, performed well and is used again in this study and described in more detail in the Methods section. This approach has since been used to identify cancer in pathology and imaging reports,[6] psychotherapy in the mental-health notes of PTSD patients,[29] and breast cancer-related records from 67 community hospitals. While document-level retrieval has proven useful in a variety of tasks, some information-extraction exercises require the more granular identification of concepts such as specific symptoms or treatments present in a patient's record. In this study, we build upon this foundation and capitalize on previously described advances to explore automated and generalizable approaches for concept-level information extraction.

METHODS

Automated retrieval console

The ARC was designed to facilitate rapid development, evaluation, and deployment of information-retrieval pipelines.

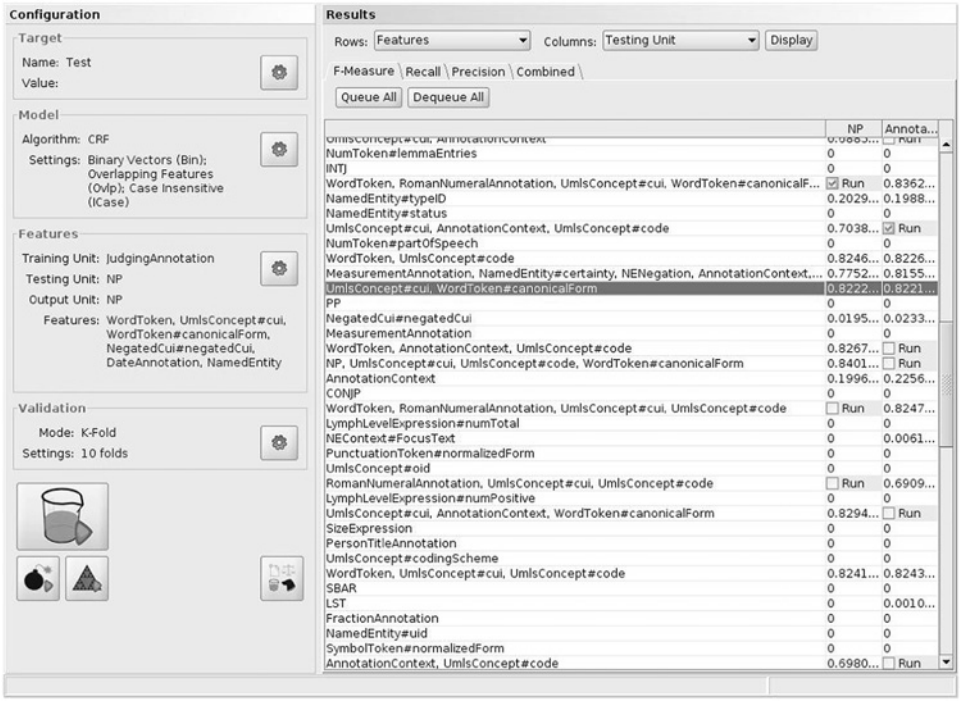

Figure 1 Screenshot of Automated Retrieval Consoles Lab interface where users can configure, conduct, and review experiments.

Researchers familiar with the processes of information extraction and NLP can design, manage, and compare performance of experiments using ARC's 'Lab' graphical user interface. Using Lab, feature types can be selected from those exposed by UIMA pipelines, classification models can be applied, and evaluations can be performed with fixed sample sizes or different fold-sizes of n-fold cross validation. The most recent version (v2) includes an H2 database for storing results, improved modularization to drive API-level integration with any parts of ARC, drag-and-drop import of Knowtator project files for importing reference sets, and more sophisticated project and workspace management. All experiments described in this study were performed using ARC's Lab interface, which is shown in figure 1.

The 'Do-It-Yourself' (DIY) interface is an attempt to provide information retrieval to relatively non-technical end users. It reduces the end-user burden in using concept-level information retrieval to four steps. First, the user imports an existing reference set formatted as a Knowtator project file. Second, the user points to the location of the documents referenced in the project file. Third, the user selects the target of interest from the available annotation types in the Knowtator project. Finally, the user clicks the DIY button which launches the algorithm described and evaluated below. The user is then presented top-performing models and given the option of deploying that model on a larger document corpus. A screenshot of the DIY interface is shown in figure 2. More information about ARC as well as documentation and video tutorials are available at our website.[30] Thanks to the generous cooperation of G Savova, J Willis, and the National Library of Medicine, it is possible to download ARC+cTAKES+the customized subset of the UMLS used by cTAKES at our website as well.

Algorithm design

Our focus is on achieving a balance between maximizing performance in concept-based information extraction and minimizing the burden on the end user. The goal of reducing the burden on the end user for concept-level information retrieval introduces a number of challenges otherwise avoided in document-level classification. The different aspects of the process that must be solved include mapping human annotations to machine-interpretable training and testing units of analysis, the selection of relevant feature types from NLP output to include in supervised machine learning, and the choice of what grammatical unit is most appropriate as the output of concept classification tasks. Each of these configurations is exposed to the user through ARC's Laboratory Interface. These challenges as well as proposed solutions for them are described in more detail in the following section. An overview of the algorithm design evaluated in this study is provided in figure 3.

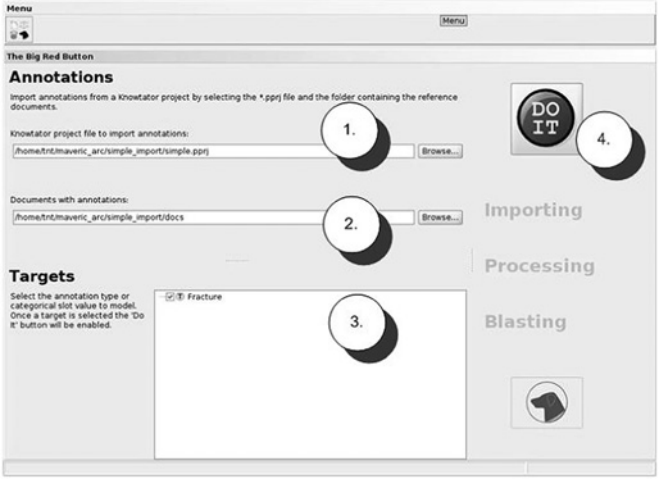

Figure 2 Screenshot of Automated Retrieval Consoles 'Do-It-Yourself' interface which allows non-technical end users to perform information retrieval in four steps: (1) annotated (Knowtator) project file selection; (2) document corpus selection; (3) select target or targets from those available in the list of annotations and; (4) launch the process.

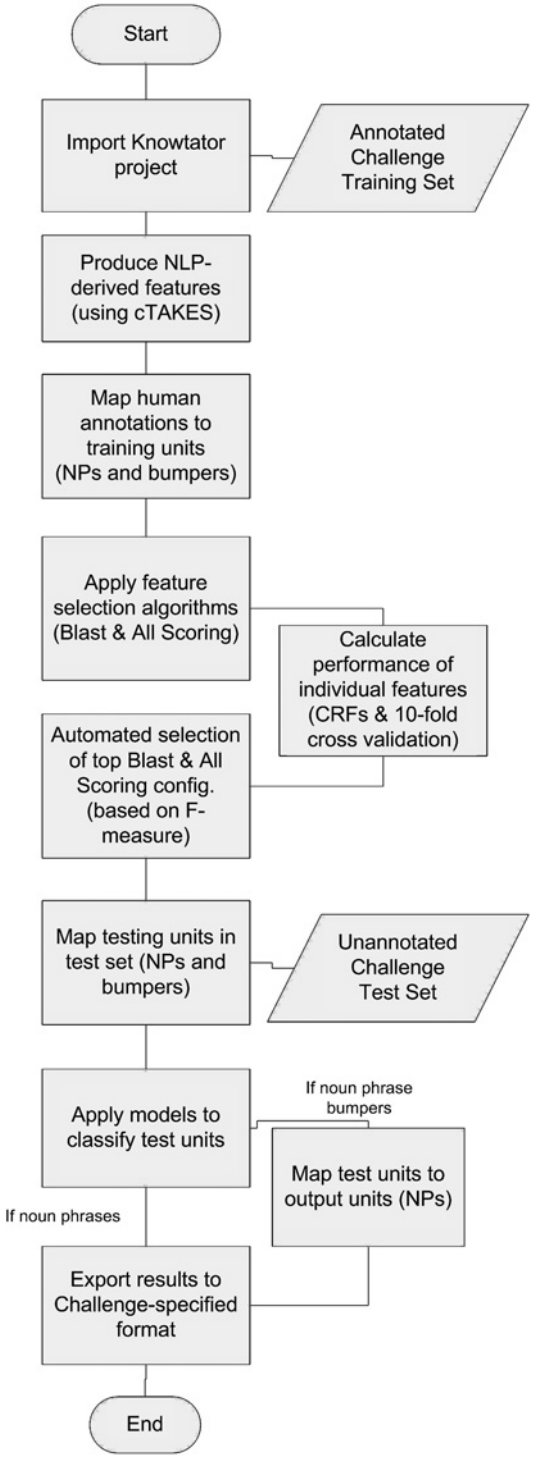

Figure 3 Overview of the algorithms used in this study. CRFs, Conditional Random Fields; cTAKES, Clinical Text Analysis and Knowledge Extraction System; NLP, natural language processing; NPs, noun phrases.

Units of analysis

In supervised machine learning classification, examples of relevant targets are used to 'train' a model. Those models are then tested and eventually deployed on unlabeled potential targets. The training examples given should represent, as closely as possible, the targets one hopes to classify. A challenge that arises in concept-level information retrieval is the selection of appropriate units of analysis for both training and testing models. Whereas in document-level information retrieval the entire document is considered for training and testing, in concept-level information retrieval a human annotator may label a single word (eg, 'infarction') or a named entity (eg, 'myocardial infarction'), or span parts or all of a sentence or phrase (eg, 'history of myocardial infarction'). While human annotated training units can vary in length, an automated system cannot arbitrarily alter the units it considers for classification (eg, phrases, trigrams, sentences, etc).

This challenge is traditionally handled in information extraction by defining patterns that are specific to the task at hand. For example, in extracting tumor stage from a pathology report, one can define, in advance, a series of tokens or characters most likely to narrow features of interest. In our attempt to create a single approach to concept extraction that can be applied across many tasks, it becomes necessary to decide on which logically structured sources of features to consider. If one maps a human annotation to a machine-interpretable training unit that is too narrow (eg, a single noun), there is the potential to limit the number of relevant features used to create a classification model with a likely detrimental effect on recall. The selection of too broad a feature source (eg, sentences) may improve recall at the expense of precision.

For this study, we explored the potential of two types of training units that we hypothesized to be most capable of providing a favorable signal-to-noise ratio for concept extraction, noun phrases and units we refer to as 'noun-phrase bumpers.' For noun phrases, the minimum training unit to which a human annotation is mapped is a single noun phrase. If a human annotation does not span a noun phrase, it is expanded to encompass an entire noun phrase. If the human annotation spans multiple noun phrases, each of the noun phrases spanned is included in a single training unit. For example:

Original sentence 'The patient had a heart attack'	
Human annotation	Noun phrase-based training unit
'Heart attack'	(Heart attack)
'Heart'	(Heart attack)
'Patient had a heart attack'	(Patient), (heart attack)

Concerned that the selection of noun phrases may leave behind an important context that can be useful for classification, we added a data type and analysis engine to our UIMA pipeline designed to capture words between identified noun phrases as part of the training unit. To address this issue, we created data types we refer to as noun-phrase bumpers. The modular design of UIMA pipelines allows access to the products of each component or 'analysis engine.' We designed our noun-phrase bumper engine to consume each noun phrase identified by cTAKES and then extend in both directions of the text to either the preceding or next noun phrase or the start or end of a sentence, whichever comes first. For example:

Original sentence 'The patient had a heart attack'	
Human annotation	Noun phrase bumper-based training unit
'Heart attack'	(Had a heart attack)
'Heart'	(Had a heart attack)
'Patient had a heart attack'	(The patient had a), (had a heart attack)

The term 'testing units' is used to describe the logically structured unit passed by the system to the classifier in attempts to classify targets of interest. The testing units used in our experiments were set to match the training units. For example, in the first iteration we mapped human annotations to noun phrases as part of processing the training set. Consequently, in our test set, all identified noun phrases were considered for classification. In the second iteration, we used noun-phrase bumpers as our training unit and all noun-phrase bumpers were considered for classification in our testing set.

Feature type selection

Once a training unit has been defined, the next task in using supervised machine learning is to determine which data types or feature types (eg, bag-of-words, noun phrases, UMLS concepts, etc) to use for classification. The balance in feature type selection, as in the choice of training units, is to select units that maximize the signal while reducing the noise. The choice of feature types to include may also have performance implications, as documents must be preprocessed to derive feature types, and the number of attributes considered in models has an effect on their performance. Whereas this decision is also traditionally made in consideration of a specific clinical domain and task in mind, our intention is to develop a single approach capable of generalizing across multiple clinical domains and tasks.

The feature types available for inclusion in our system were derived from the first release of cTAKES, which produces 90-plus feature types (eg, nouns, verbs, noun phrases, UMLS concepts, tokens, etc). The subset of the UMLS used by that version of cTAKES were taken from the 2008AB release of the UMLS.

Two algorithms for selecting appropriate feature types and their related values were evaluated in this study. The first algorithm used was the 'Blast' algorithm used in our previous study of document-level classification. Using the performance of individual features, Blast creates permutations of the top five performing features in combination to produce 26 combinations multiplied by the number of classification models used. More specifically, the Blast algorithm uses the following configurations:

1	Highest F-measure
2	Config. 1+2nd Highest F-measure
3	Config. 2+3rd Highest F-measure
4	Config. 3+Highest recall not already included
5	Config. 4+Highest precision not already included

The second algorithm explored is referred to as the 'All Scoring' algorithm. Like the Blast algorithm, the All Scoring algorithm also starts by calculating the performance of each individual feature using 10-fold cross-validation. To test and potentially benefit from the ability of supervised machine learning classifiers such as CRFs to detect patterns in highly dimensional data, our second algorithm incorporated all feature types and related values that scored greater than 0 in their individual assessments.

Output units

As with training and testing units, we cannot arbitrarily decide at run time the most appropriate unit to present as a result. Therefore, ARC's workflow defines an 'output unit.' For our attempt to determine a single approach for generalizable concept-level extraction, we chose noun phrases as the output unit. For iterations in which noun phrases were already the training and testing unit, the delivery of noun phrases as output units was straightforward. Each noun-phrase bumper in a noun-phrase bumper testing unit has a single noun phrase. Therefore, for iterations using noun-phrase bumpers, the noun phrase 'anchor' in the bumper was presented as the output unit. For example, if the noun-phrase bumper classified was 'had a heart attack,' the output unit would be 'heart attack.'

Data set and evaluation

The data set, annotations, and evaluation algorithm from the 2010 Fourth i2b2/VA Shared Task Challenge were used to evaluate the performance of our approach. Most uses of ARC to date have been in support of specific tasks such as the identification of a cohort of patients with specific criteria. The Challenge data set and metrics therefore provided a valuable opportunity to evaluate the performance of our approach on a rather difficult and general task with the added advantage of comparing performance against others focused on achieving the best possible performance.

The 2010 Challenge featured three tasks; extraction of concepts, assertions, and relations from clinical text. Our evaluation was focused only on the extraction of clinical concepts, which in the challenge were narrowed to those representing medical problems, treatments, and tests. The Challenge used stand-off annotations with the raw text of medical records as input. ARC's output format was modified to produce the expected system output entries of the form, c='concept text' line:token(s) offset || t='concept type.' For example:

```
c='cancer' 5:8 5:8 || t='problem'
c='chemotherapy' 5:4 5:4 || t='treatment'
c='chest x-ray' 6:12 6:13 || t='test'
```

The data set contains deidentified discharge summaries and progress notes from Partners Healthcare, Beth-Israel Deaconess Hospital, and the University of Pittsburgh Medical Center. Data-use agreements were signed with each of the institutions in order to use the dataset. The challenge divided the set into 349 training reports and 477 test reports, and the annotation was performed by professional chart reviewers using arbitration on disagreements. The breakdown of records supplied by each hospital is provided in table 1.

Table 1 Distribution of record types and institutions in the i2b2/Veterans Affairs fourth shared task challenge used in this study

n	Report type	Institution
349 training reports		
97	Discharge summaries	Partners Healthcare
73	Discharge summaries	Beth-Israel Deaconess
98	Discharge summaries	University of Pittsburgh Medical Center
81	Progress notes	University of Pittsburgh Medical Center
477 test reports		
133	Discharge summaries	Partners Healthcare
123	Discharge summaries	Beth-Israel Deaconess
102	Discharge summaries	University of Pittsburgh Medical Center
119	Progress notes	University of Pittsburgh Medical Center

Our results were submitted to the Challenge judges, and our performance was calculated using the software created for the Challenge. Performance in the concept extraction task was measured for both exact match and inexact match using micro-averaged precision, recall, and F-measure for all concepts together (ie, problems, treatments, tests). For 'exact match' calculation, all phrase boundaries and concept types must match exactly to receive credit.

For example, a returned result of:

c = 'has cancer' 5 : 7 5 : 8 || t = 'problem'

would not be counted as a successful match with:

c = 'cancer' 5 : 8 5 : 8 || t = 'problem'

The above example would be considered a successful match for 'inexact match' scoring, where it is required that the result overlaps with the ground truth concept at least in part. Importantly, any result span can count only once toward a match, preventing results with long spans from counting multiple hits as successful matches. Our goal of 'off the shelf' information extraction precluded tailoring of output for this specific task and led us to employ already-available units from cTAKES as output units. As a result, we considered the Challenge's inexact scoring algorithm a more appropriate metric and used it for our evaluation.

A total of 22 teams participated in the concept extraction portion of the competition. The top-scoring entry for inexact concept mapping with all three tasks combined was submitted by deBruijn *et al* with recall, precision, and F-measure scores of 0.92, 0.93, and 0.92 respectively. The mean and median scores of all 22 teams' top entries were 0.84 and 0.87 with an SD of 0.07. Detailed descriptions of top-scoring approaches are described elsewhere.[7]

Experiments

Competitors in the Challenge had access to training data with which to improve their systems for the specified tasks 3 months in advance of the release of test data. In contrast, we used ARC to import the Knowtator project files representing the training set, selected the target from the project files, and deployed the produced models on the test set with no further interventions.

Two variations on our algorithm were explored: (1) the use of noun phrases versus noun-phrase bumpers as both training and testing units; and (2) the use of the blast versus the all scoring algorithms for the selection of which feature types to include in our models. This resulted in four variations per task which, multiplied by three tasks (ie, problems, treatments, tests), led to a total of 12 experiments as described in table 2.

RESULTS

Each of the 12 iterations took approximately 5 h to run on a standard desktop computer. The total time in configuring and launching experiments using ARC's Lab Interface was less than 5 min per experiment. The top F-measures for extracting medical problems, treatments, and tests rounded to the nearest hundredth were 0.83, 0.82, and 0.83, approximately one point behind the mean performance for all 22 entrants. The top recall, precision, and F-measure scores for each task and configuration are shown in table 2.

For all 12 experiments, recall scores were considerably lower than precision. For medical test extraction, the use of noun phrases versus noun-phrase bumpers as training/testing units did not have a significant impact on the recall or precision. This, however, was not the case in the extraction of medical problems

Table 2 Top-scoring iterations for extracting medical problems, treatments, and tests in terms of F-measure

Units of analysis	Feature-selection algorithm	R Value	p Value	F Value
Medical problems				
Noun phrase	Blast	0.7271	0.9350	0.8181
Noun phrase	All scoring	0.7453	0.9338	0.8290
Noun-phrase bumper	Blast	0.7625	0.8657	0.8108
Noun-phrase bumper	All scoring	0.7712	0.8759	0.8202
Medical treatments				
Noun phrase	Blast	0.6735	0.9094	0.7739
Noun phrase	All scoring	0.6353	0.9170	0.7506
Noun-phrase bumper	Blast	0.7335	0.8915	0.8048
Noun-phrase bumper	All scoring	0.7555	0.8881	0.8164
Medical tests				
Noun phrase	Blast	0.7649	0.9012	0.8275
Noun phrase	All scoring	0.7117	0.9160	0.8010
Noun-phrase bumper	Blast	0.7689	0.8791	0.8203
Noun-phrase bumper	All scoring	0.7171	0.9149	0.8040

and treatments. For the extraction of both medical problems and medical treatments, recall was higher when noun-phrase bumpers were used, while precision was higher when noun phrases were used as training and testing units.

The differences in performance of the All Scoring versus Blast algorithms were also task-dependent. For medical problem extraction, the use of All Scoring versus Blast was comparable. For medical treatments, the use of All Scoring decreased the F-measure by two points when noun phrases were used but increased the F-measure slightly when noun-phrase bumpers were used. Finally, for medical test extraction, the Blast algorithm outperformed the All Scoring algorithm using both noun phrases and noun-phrase bumpers by two points in F-measure.

The top-scoring feature type combinations emerging from the Blast algorithm are listed in table 3. Word tokens and UMLS CUIs were featured prominently in most top-scoring runs. It is interesting to note that the noun-phrase bumper annotation which was added to provide logical training and testing units emerged as a useful feature for classifying noun phrases as medical tests. The Lookup Window annotation emerging as a contributing feature type for medical problem classification using bumpers is the span of text used by cTAKES in attempts to map to UMLS concepts.

Table 3 Top-scoring feature type combinations emerging from the Blast algorithm for each of the three i2b2/VA Challenge concept extraction tasks

Top-scoring blast feature-type combinations		
Training/testing units	**F-Measure**	**Top-scoring feature-type combinations**
Medical problems		
Noun phrase	0.8181	Word Token, UMLS CUI, Named Entity
Noun-phrase bumper	0.8290	Word Token, UMLS CUI, Canonical Form of Token
Medical test		
Noun phrase	0.7739	Word Token, Noun Phrase Bumper, UMLS CUI, Canonical Form of Token
Noun-phrase bumper	0.8048	Word Token, UMLS CUI, Canonical Form of Token
Medical treatments		
Noun phrase	0.8275	Word Token, Named Entity, UMLS CUI
Noun-phrase bumper	0.8203	Word Token, Lookup Window, Named Entity, Canonical Form of Token

DISCUSSION

The algorithms explored as part of this study achieved F-measures that were competitive with the average performance of i2b2/VA Challenge competitors (mean=0.84, median=0.87) without requiring the use of custom rules, custom software development, or any additions to a knowledge base based on the training set. Our recall scores reflected the challenge of finding all symptoms, treatments, and tests with no knowledge-base customization and suggest that alternative methods be explored for comprehensive mapping tasks (eg, find all symptoms). We expect that a modest improvement would have been achieved by simply mapping discovered shortcomings from the training set to a knowledge base. However, such mapping would have defeated the intent of this study. The precisions score of our top F-measure configurations hovered at 0.90, suggesting a strong F-measure performance in the extraction of more specific targets (eg, symptoms of PTSD as opposed to all symptoms). That hypothesis will be evaluated in future applications.

The findings suggest that no single feature type or unit of analysis is ideal for all retrieval tasks. For example, the performance of noun-phrase bumpers in medical treatment extraction bested noun phrases by a considerable margin, yet showed no real difference on medical problem or test extraction. This result may be indicative of longer phrases being considered relevant for treatment versus medical problems and tests which are more likely to be one or two word phrases.

Such findings support our current emphasis on the use of automation to iteratively discover the best-suited configuration. Properly designed, such an approach need not increase the burden on the end user, researcher, or developer as shown in this study. Even in the case of near-real-time results requirements (eg, biosurveillance), automation can be used to evaluate several configurations, and the decision of which to deploy can be made objectively on any number of criteria including performance as well as computational burden. For example, the discovery of word tokens as consistent top-performing feature types through the application of the Blast algorithm suggests that users might deploy a configuration that processes just tokens to achieve a faster turnaround time for point of care applications.

The comparable performance of the All Scoring versus Blast algorithm leads us to eliminate the All Scoring algorithm from further consideration. Our hypothesis that a larger feature space may contribute to better performance was not supported. In addition, the All Scoring algorithm provides a single experiment, hence a single result. In contrast, the 26+ iterations provided by the Blast Algorithm allows the user to choose from several models with different performance criteria, allowing the selection of one with greater recall or precision based on the task at hand.

There are several limitations to the proposed approach. First, while the approach described in this study has demonstrated potential for the extraction of categorical variables, it does not support the extraction of numeric attribute value pairs (eg, blood pressure=120/80). The proposed method could extract the phrase, but some level of postprocessing would be required to pull the exact value. In addition, some concept-level extraction problems call for the identification of a single semantic 'concept' that appears across multiple sentences or phrases. The need to identify a consistent unit of analysis for automation would require that paragraph or document level classification occur, which may not be ideal for such tasks.

CONCLUSION

The purpose of this research is to enable the widespread adoption of clinical NLP and information extraction. The results of

this study demonstrate that it is possible to deliver acceptable concept-level information extraction performance across different tasks with no custom software or rules development. To help foster continued advances in this area, the ARC is available for download as well as ARC+MALLET+cTAKES+the UMLS subset used by cTAKES.[30] Also available for download is a data importer for mapping various formats of raw text data into ARC.

In future work, we will continue to focus on methods capable of striking a balance between achieving acceptable performance and minimizing the burden on the end user. One specific area with the potential to reduce end-user burden is the exploration of algorithms in the area of active learning and relevance feedback that can reduce the number of annotations required to achieve acceptable performance. In addition, the incorporation of more advanced feature-selection algorithms is an attractive future direction for ARC.

Acknowledgments We thank G Savova, of Children's Hospital Boston and Harvard Medical School, and J Masanz, of the Mayo Clinic, as well as D Mimno and F Pereira, of the University of Massachusetts for their assistance in incorporating the open source tools cTAKES and MALLET. We also thank O Uzuner, S DuVall, B South, and S Shen, for their hard work in preparing and generosity in sharing the i2b2/VA Shared Task Challenge data and evaluation tools. We would also like to acknowledge the dedicated staff of MAVERIC for their assistance in this project.

Funding This work was supported by Veterans Affairs Cooperative Studies Program as well as the Veterans Affairs Health Services Research and Development grant, Consortium for Health Informatics Research (CHIR), grant # HIR 09-007.

Competing interests None.

Ethics approval Ethics approval was provided by the VA Boston Healthcare System.

Provenance and peer review Not commissioned; externally peer reviewed.

REFERENCES

1. **Hsiao CJ,** Hing E, Socey TC, *et al*. *Electronic Medical Record/Electronic Health Record Systems of Office-based Physicians: United States, 2009 and Preliminary 2010 State Estimates*. Hyattsville, MD, USA: National Center for Health Statitsics, 2010.
2. **US Department of Health and Human Services.** *About Healthy People*. http://www.healthypeople.gov/2020/about/default.aspx (accessed 14 Feb 2011).
3. **Committee on Comparative Effectiveness Research Prioritization.** *Initial Priorities for Comparative Effectiveness Research*. Washington DC: Institute of Medicine, 2009.
4. **Bates DW,** Kuperman GJ, Wang S, *et al*. Ten commandments for effective clinical decision support: making the practice of evidence-based medicine a reality. *J Am Med Inform Assoc* 2003;**10**:523—30.
5. **Murphy S,** Churchill S, Bry L, *et al*. Instrumenting the health care enterprise for discovery research in the genomic era. *Genome Res* 2009;**19**:1675—81.
6. **D'Avolio LW,** Nguyen T, Farwell WR, *et al*. Evaluation of a generalizable approach to clinical information retrieval using the automated retrieval console (ARC). *J Am Med Inform Assoc* 2010;**17**:375—82.
7. **Uzuner O,** South B, Shen S, *et al*. 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text. *J Am Med Inform Assoc* 2011;**18**:552—6.
8. **Stanfill MH,** Williams M, Fenton SH, *et al*. A systematic literature review of automated clinical coding and classification systems. *J Am Med Inform Assoc* 2010;**17**:646—51.
9. **Sager N,** Lyman M. Computerized language processing: implications for health care evaluation. *Med Rec News* 1978;**49**:20—1.
10. **Friedman C.** Sublanguage text processing—application to medical narrative. In: Grishman R, Kittridge R, eds. *Analyzing Language in Restricted Domains: Sublanguage Description and Processing*. Hillsdale, NJ: Lawrence Erlbaum, 1986:85—102.
11. **Riloff E.** Automatically constructing a dictionary for information extraction tasks. *Proceedings of the Eleventh National Conference on Artificial Intelligence*. Menlo Park, CA, USA: AAAI Press/Cambridge, MA, USA: MIT Press, 1993:811—16.
12. **McCarthy J,** Lehnert W. Using decision trees for coreference resolution. *Proceedings of the Fourteenth International Conference on Artificial Intelligence*. Montreal, Canada: Morgan Kaufmann, 1995:1050—5.
13. **Cunningham H.** GATE, a general architecure for text engineering. *Comput Humanit* 2004;**36**:223—54.
14. **Ferrucci D,** Lally A. UIMA: an architectural approach to unstructured information processing in the corporate research environment. *Nat Lang Eng* 2004;**10**:327—48.
15. **Savova G,** Kipper-Schuler K, Buntrok J, *et al*. UIMA-based clinical information extraction system. *Language Resources and Evaluation Conference, Morocco*. 2008.
16. **i2b2.** *First Shared Task for Challenges in Natural Language Processing for Clinical Data*. 2006. https://www.i2b2.org/NLP/.
17. **Uzuner O,** Luo Y, Szolovits P. Evaluating the state-of-the-art in automatic de-identification. *J Am Med Inform Assoc* 2007;**14**:550—63.
18. **National Institute of Standards and Technology.** *Overview of TREC*. 2009. August 1, 2000. http://trec.nist.gov/overview.html (accessed 31 Aug 2009).
19. **National Institute of Standards and Technology.** *Introduction to Information Extraction*. 2001. March 8, 2005 (cited 2006 August 11); Available from http://www-nlpir.nist.gov/related_projects/muc/.
20. **Sparck Jones K.** Further reflections on TREC. *Inform Process Manag* 2000;**36**:37—85.
21. **University of Pittsburgh BLULab Team.** *BLULab NLP Repository*. 2010. http://nlp.dbmi.pitt.edu/nlprepository.html (accessed 5 Feb 2011).
22. **Massachusetts Institute of Technology.** *MIMIC II*. 2010. http://mimic.physionet.org/ (accessed 5 Feb 2011).
23. **Computational Medicine Center.** *Natural Language Processing*. 2007. http://www.computationalmedicine.org/project/nlp.htm (accessed 5 Feb 2011).
24. **Ogren P.** Protege plug-in for annotated corpus construction. *Proceedings of the 2006 Conference of the North American Chapter of the Association for Computational Linguistics*. New York: Association for Computational Linguistics, 2006:273—5.
25. **Stanford Center for Biomedical Informatics Research.** *The Protege Ontology Editor and Knowledge Acquisition System*. 2010. http://protege.stanford.edu/ (accessed 5 Feb 2011).
26. **Lafferty J,** McCallum A, Pereira F. Conditional random fields: probabilistic models for segmenting and labeling sequence data. *International Conference on Machine Learning*. San Francisco: Morgan Kaufman, 2001:282—9.
27. **Peng F,** McCallum A. Information extraction from research papers using conditional random fields. *Inform Process Manag* 2006;**42**:963—79.
28. **Wellner B,** Huyck M, Mardis S, *et al*. Rapidly retargetable approaches to de-identification. *J Am Med Inform Assoc* 2007;**14**:564—73.
29. **Shiner B,** D'Avolio L, Nguyen T, *et al*. Automated classification of psychotherapy note text: implications for quality assessment in PTSD care. *J Eval Clin Pract* 2011. In Press.
30. **D'Avolio L.** *The Automated Retrieval Console*. 2011. http://research.maveric.org/mig/arc.html (accessed 27 Mar 2011).