# A Prototype Tool Set to Support Machine-Assisted Annotation

**Brett R. South[1,2], Shuying Shen[1,2], Jianwei Leng[2], Tyler B. Forbush[4], Scott L. DuVall[3,4], Wendy W. Chapman[5]**

Departments of [1]Biomedical Informatics, [2]Internal Medicine, and [3]Radiology University of Utah, Salt Lake City, Utah, USA

[4]IDEAS Center SLCVA Healthcare System, Salt Lake City, Utah, USA

[5]University of California, San Diego, Division of Biomedical Informatics, La Jolla, California, USA

brett.south@hsc.utah.edu,
shuying.shen@hsc.utah.edu, jianwei.leng@utah.edu,
tyler.forbush@utah.edu, scott.duvall@utah.edu,
wendy.w.chapman@gmail.com

## Abstract

Manually annotating clinical document corpora to generate reference standards for Natural Language Processing (NLP) systems or Machine Learning (ML) is a time-consuming and labor-intensive endeavor. Although a variety of open source annotation tools currently exist, there is a clear opportunity to develop new tools and assess functionalities that introduce efficiencies into the process of generating reference standards. These features include: management of document corpora and batch assignment, integration of machine-assisted verification functions, semi-automated curation of annotated information, and support of machine-assisted pre-annotation. The goals of reducing annotator workload and improving the quality of reference standards are important considerations for development of new tools. An infrastructure is also needed that will support large-scale but secure annotation of sensitive clinical data as well as crowdsourcing which has proven successful for a variety of annotation tasks. We introduce the Extensible Human Oracle Suite of Tools (eHOST) http://code.google.com/p/ehost that provides such functionalities that when coupled with server integration offer an end-to-end solution to carry out small or large scale as well as crowd sourced annotation projects.

## 1 Introduction

Supervised learning methods benefit from a reference standard that is used to train and evaluate the performance of Natural Language Processing (NLP) or Machine Learning (ML) systems for information extraction and classification. Ideally, generating a reference standard involves the review of more than one annotator with an accompanying adjudication step to resolve discrepancies (Roberts et al., 2007; Roberts et al., 2009). However, manual annotation of clinical, texts is time-consuming, expensive, and requires considerable effort. Reducing the time and costs required for manual annotation could be achieved by developing new tools that integrate methods to more efficiently annotate clinical texts and integrate a management interface that allows administration of large or small scale annotation projects. Such a tool could also integrate methods to pre-annotate entities such as noun phrases or clinical concepts mapped to a standard vocabulary. Efficiencies could be realized via reduction in human workload, modification of annotation tasks that could include crowd sourcing, and implementation of machine-assisted approaches.

Typically annotation of clinical texts requires human reviewers to identify information classes of interest called *"markables"*. These tasks may also require reviewers to assign attributes to those information classes and build relations between spans of annotated text. For each annotation task there may be one or many types of markables and each markable class may be associated with one or more spans of text and may include single or even multiple tokens. These tasks may occur simultaneously, or may also be done in different steps and by multiple reviewers. Furthermore, these activities require written guidelines that clearly explicate what infor-



*Proceedings of the 2012 Workshop on Biomedical Natural Language Processing (BioNLP 2012)*, pages 130–139,
Montréal, Canada, June 8, 2012. 

mation to annotate, specifics about each markable class, such as how much information to include in annotated spans, or syntactic rules to provide further guidance on annotated spans. Annotation tasks may benefit by incorporating rules or guidelines as part of the annotation task itself in the form of machine-assisted verification.

There are many annotation tools available, and the majority of them were designed for linguistic or gene annotation. Linguistic annotation tools such as Callisto and WordFreak are standalone clients suitable for small to medium scale tasks where collaborative effort is not emphasized. Functionality integrated with eHOST was inspired by existing features of these tools with the intent of providing a more efficient means of reference standard generation in a large collaborative environment. One annotation tool called Knowtator, a plug-in for Protégé (Musen, M.A., et al, 1995) developed by Ogren (2006) has been widely used to annotate clinical texts and generate reference standards. However, no standalone system exists that can provide end users with the ability to manually or semi-automatically edit, curate, and easily navigate annotated information. There are also specific functionalities that are missing from open source annotation tools in the clinical and biomedical domains that would introduce efficiencies into manual annotation tasks. These functionalities include: annotation of clinical texts along with database storage of stand-off annotations, the ability to interactively annotate texts in a way that allows users to react to either pre-annotations imported from NLP or ML systems or use exact string matching across an active corpus to identify similar spans of text to those already annotated. Additionally, these systems do not generally support crowd sourcing, machine-assisted pre-annotation or verification approaches integrated directly with the annotation tool.

This paper discusses development of a prototype open source system designed to provide functionality that supports these activities and offers an end-to-end solution when coupled with server integration to reduce both annotator and administrative workload associated with reference standard. We introduce the Extensible Human Oracle Suite of Tools (eHOST) created with these expectations in mind.

## 2 Background

Our goal for these development efforts was to build a prototype open source system that improves upon existing tools by including new functions and refining capabilities available in other annotation tools. The resulting GUI interface provides a means of visually representing annotated information, its attributes, and relations between annotated mentions. These efforts also focused integrating various machine-assisted approaches that can be used to easily curate and navigate annotated information within a document corpus, pre-annotate information, and also verify annotations based on rules checks that correspond with annotation guidelines or linguistic and syntactic cues.

The eHOST provides basic functionality including manual annotation of information representing markable classes and assignment of information attributes and relationships between markable classes. Annotations exported from eHOST are written using the XML format as Knowtator thus allowing integration of inputs and outputs to and from Knowtator and indirectly to Protégé 3.3.1. Coupling eHOST with an integrated server package such as the one under development by the VA Informatics and Computing Infrastructure (VINCI) called the Chart Administration Server for Patient Review (CASPR) provides one method of increasing efficiencies for small or large-scale annotation efforts that could also include crowd sourcing.

### 2.1 System Features Development

In the domains of computational linguistics and biomedical informatics various approaches that can be used to improve annotation efficiencies have been evaluated for a variety of tasks including information extraction and classification. While several methods may help reduce the time and costs required to create reference standards, one of the simplest approaches may include integrating machine-assisted methods to pre-annotate relevant spans of text allowing the annotator to add missing annotations, modify spans, or delete spurious annotations. Neveol (2011) evaluated use of automatic semantic pre-

annotation of PubMed queries. This study showed a significant reduction in the number of required annotations when using pre-annotations, reduction in annotation time with higher inter-annotator agreement. Pre-annotation using simple approaches such as regular expressions coupled with dictionaries (South et al., 2010a) based on the UMLS as a source of lexical knowledge (Friedman, 2001) and pre-annotation of information representing protected health information (South et al., 2010b). In both cases finding that annotators preferred particular types of pre-annotation over others, but improvements in reference standard quality occur when pre-annotation was provided. Others have explored the use of third party tools for the pre-annotation task for UMLS concepts (Savova, 2008) and pre-annotation using an algorithmic approach (Chapman, et al., 2007) combined with domain expert annotations reused for temporal relation annotation (Mowery, 2008). Savova (2008) suggests limited utility when a third party tool is used for pre-annotation and Mowery (2008) suggest that even with domain expert pre-annotations, additional features are required to discern temporality. Finally, Fort and Sagot (2008) evaluated using pre-annotation for part-of-speech tagging on the Penn Tree bank corpus and demonstrate a gain in quality and annotation speed even with a not so accurate tagger.

Semi-automated curation has been explored as a means to build custom dictionaries for information extraction tasks (Riloff, 1993). More recently this approach was spurred on by the BioCreative II competition (Yeh et al., 2003). Alex et al., (2008), explored the use of NLP-assisted text mining to speed up curation of biomedical texts. Settles et al., (2008) estimates true labeling costs and provides a review of active and interactive learning approaches as a means of providing labels and reducing the cost of obtaining training data (Settles, 2010). Although eHOST does not yet include an active learning module it does provide one means of interactive annotation so these are important considerations for future development efforts.

In the biomedical informatics domain crowd sourcing has been evaluated as part of the 2009 i2b2 Medication Challenge (Uzuner, 2010). Nowak and Ruger (2010) provide estimates of annotation reliability from crowd sourcing of image annotation. Hsueh et al., (2009) provide estimates of the quality of crowd sourcing for sentiment classification using both experts and non-expert annotators. In all three cases the resulting annotation set was of comparable quality to that derived from expert annotators. Wang et al., (2008) make general recommendations for best approaches to crowd sourcing that include closer interactions between human and machine methods in ways that more efficiently connect domain expertise with the annotation task.

Subsequent sections in this paper walk the reader through the various basic and advanced features eHOST provides. These features have been developed in a way that provides flexibility to add additional modules that support improvements in annotation workflow and efficiency for a variety of annotation scenarios applicable to computational linguistics and biomedical informatics. Some of these features may be useful for crowd-sourced efforts whereas others may simply represent an improvement in the way annotation is visualized or how manual effort can be reduced. Figures in this paper use a set of synthetic clinical documents and a demonstration annotation project based on the 2010 and 2011 i2b2/VA annotation tasks as examples available from http://code.google.com/p/ehost.

## 2.2 Systems Architecture

The eHOST is a client application that can run on most operating systems that supports Java including, most Microsoft Windows x86/x64 platforms, Apple Mac OS X, Sun Solaris, and Linux. The application uses standardized formats including a file folder system, and structured XML inputs and outputs. These capabilities also support integration with other open source tools for annotation and knowledge management including Knowtator and Protégé. An Extract-Transform-Load process (ETL) is used by the system to import concept information from different sources, such as XML or Protégé PINS files. These inputs sources are normalized for loading into eHOST. All data that exists in the data pool can be transformed into various output formats. Raw input data documents in a single text file or sequential text files in a file folder system.

Information representing an annotation in-

cluding concept attributes such as the annotated span, attributes, and relationships between annotations are inserted into a common data pool using a dynamic structured storage space. The data pool ensures that eHOST has capabilities to add new functions easily without making major changes to system architecture.

## 2.3 Annotation Project Workspace

In eHOST each project has its own user assigned workspace that includes an annotation schema and document corpus. Annotation schema can also be imported from an existing Protégé PINS file. Project settings can be inherited from existing projects for similar annotations tasks using eHOST. Other workspace functions include quickly switching between up to five of the most recently used workspaces. A workspace can be assigned for each annotation layer or document batch. In these situations, an annotator would receive a pre-compiled project that specifies all settings including any text documents and the annotation schema. Defining a workspace is a particularly useful function in situations where annotations may be crowd sourced and there may be multiple layers of annotation that are potentially fielded to many annotators.

### 2.3.1 Corpus Management

For any annotation task, the end user must manage the document corpus, which can originate from a server or a file folder system that contains individual text files. Using the stand-alone eHOST client tool, corpus management is accomplished via the current workspace (Figure 1). When the user initializes a new project, documents are placed in a "corpus" folder that is associated with the newly created annotation project. All text files, are copied to the "corpus" folder at the time of workspace assignment. Therefore, there is no risk of deleting the original documents associated with each new annotation project. This feature makes distribution of projects easier, because of the consistency between the workspace, corpus assignment and annotation output folders. For crowd-sourced projects eHOST can be integrated with a backend server via web services using an administrative module called CASPR.

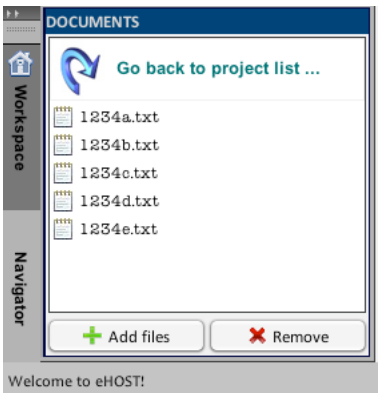

Figure 1. eHOST corpus management

### 2.3.2 Viewer/Editor Panels

Figure 2 shows an annotation for *"full body pain"*, (shown with black bar above and below the active annotation) and information for that annotation including the annotated span, the class assignment and an assertion for the 2010 and 2011 i2b2/VA Challenge annotation tasks (Uzuner et al., 2011 and Uzuner et al., 2012). The result editor tab and its associated panels serve as the central place for basic annotation features. These functionalities include: assigning an annotator, creating new annotations or adjusting annotated spans of text and assignment of attributes or creating relationships between annotated spans of text. Other functions in the results editor tab include navigation between documents in the active corpus, resizing the text displayed in the document viewer, and *"save"* and *"save as"* functions that assigns a path for XML output files. The end user can easily remove all annotations in a document or remove specific kinds of annotations by deleting a "markable" class as well as remove attributes, and relationships between all annotations.

From the navigator screen in the stand-alone eHOST client tool a user can build annotation schema specifying markable classes, their associated attributes, and any allowed relationships. The navigator interface allows the user to review all annotated spans either within the current document or across the entire document corpus, toggle the view of each class on or off, see counts for all unique annotations and all annotations for each class, and choose a class for a fast annotate mode.

An annotation editor panel allows the user to view more detailed information for each selected

annotation. This includes the time stamp of when the annotation was created, annotator assignment, comments on the annotation and class, attribute and relationship information.

Annotations can be created using several approaches from the result editor. In the normal mode, a class assignment window appears when the user selects a span of text, new annotations are generated by selecting any one of the markable classes. Activating a *"one click annotate"* mode is possible by checking the box next to a class of markables. Under this mode, any text selected is automatically annotated as that markable class. This feature improves task efficiencies when categories of markables are low or annotations of the same category cluster in small sections. Keyboard shortcuts have also been integrated with eHOST to reduce annotator click burden and dependence on a mouse. These shortcuts are available for tasks such as modification of spans, deletion of annotations, and navigation between annotations.

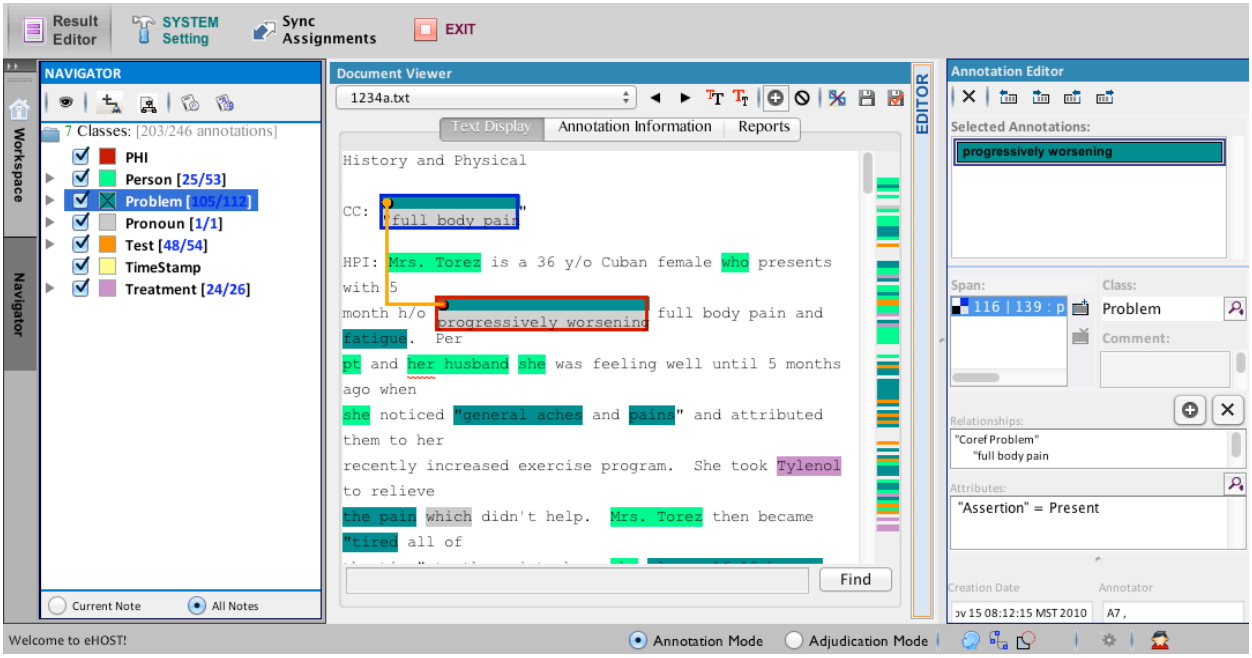

Figure 2. Example annotations using the eHOST interface

### 2.3.3 Server Integration

Annotation projects of any scale benefit from an automated means of building and distributing batches of texts to annotators, managing standoff XML files generated from annotation tasks or written directly to a database and getting and submitting assignments with minimal user input. Coupling eHOST with server components that comply with the web services API defined for eHOST allows these functionalities. The CASPR module under development by VINCI provides a means to automate the administration of annotation efforts that could include crowdsourced annotation projects.

Clicking on the sync assignments tab in the eHOST client (Figure 2) brings up a GUI that allows annotators to sync with a server location, enter credentials, see documents assigned, and designate documents as on hold, in process, or completed. When a user syncs and gets assignments from CASPR, a project folder is created that contains the annotation schema, text documents, and annotations sent from the server. The CASPR module allows an annotator to open the project and complete their task without needing to manage files or folders. Once completed, annotations can be synced to the server, and the next assignment will be loaded. The CASPR module allows iterative distribution of annotation batches without sending large sets of documents to annotators that may contain sensitive data, decreasing the risk of breaches in privacy and data security.

### 2.3.4 Additional Features

The document viewer panel employs visual cues to display relationships between annotations using color coding representing a parent and child node and line indicator between them showing the relationship. An "annotation profiler" to the right of the scroll bar shows the density of annotations color-coded to their categories, as well as relative to their positions in the document. This type of data visualization is useful to see the relative location of annotations within a single document or across an en tire document corpus.

An adjudication mode is also included in the stand-alone eHOST client that allows difference matching and side-by-side comparison of annotations for efficient adjudication of discrepancies between annotations. Standard reporting metrics can be calculated including Inter-Annotator Agreement (IAA), Recall, Precision and $F_1$-Measure.

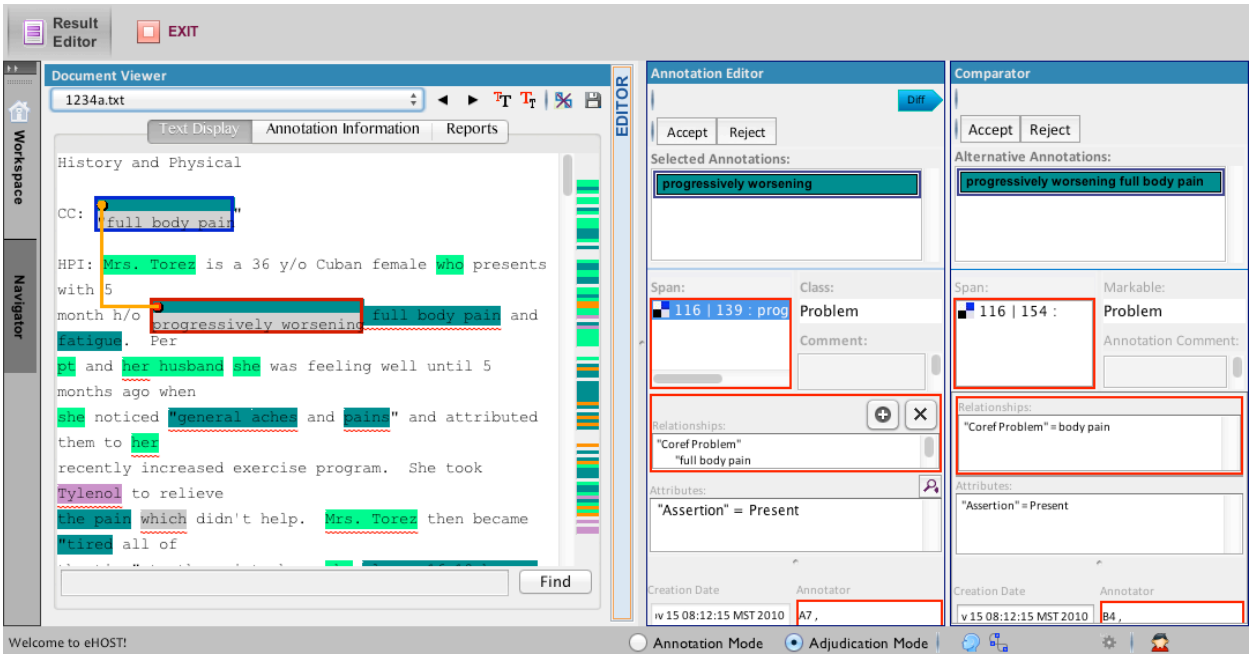


Figure 3. eHOST adjudication mode showing discrepant annotations between annotators A7 and B4

In Adjudication mode discrepant annotations are shown using a wavy red underline in the editor window and by a red bolded outline in a side by side two panel view between the annotation editor and comparator (Figure 3). These metrics and comparison tables between annotator results on the same documents can be output as HTML formatted reports that can be used by an adjudicator to quickly identify discrepancies between annotators (Figure 4). These reports and the editor window display can also be used to quickly train annotators on new clinical domains using a reference standard created by domain experts for training purposes. Using these features error analysis can also be done by importing outputs from an NLP system that have been converted into the XML format used by eHOST.

File: 1234a.txt
h 5 month h/o progressively worsening full body pain and fatigue. Per pt and her husband she was feeling well until 5 months ago when she noticed "general aches and pains" and attributed them to her recently increased exercise program. She took Tylenol to relieve the pain which didn't help. Mrs. Torez the

| | Annotator:[ A7 , ] | Annotator:[ B4 , ] |
|---|---|---|
| Annotation Text | "general aches | aches |
| Span | ( 251,265) | ( 260,265) |
| Class | Problem | Problem |
| Relationship | linked to "progressively worsening" with relationship: [Coref Problem] | linked to "progressively worsening full body pain" with relationship:[Coref Problem] |
| Attributes | Assertion = Conditional; | Assertion = Conditional; |

Figure 4. HTML Formatted report showing discrepant annotations between annotators A7 and B4

## 3 Advanced eHOST Features

There are also other more advanced features that have been integrated with eHOST. These include an "Oracle" mode that allows semi-automated annotation of similar spans of text across a document corpus, a means to easily and quickly curate annotated spans of text to create custom dictionaries, and machine-assisted pre-annotation integrated with the annotation tool itself.

### 3.1 Oracle Mode

Also implemented with eHOST is an "Oracle" mode which uses exact string matching allowing the user to annotate all spans of text that are identical to a new annotation. The oracle lists where these candidate annotations are found along with the surrounding context. The annotator can then accept or reject candidate spans annotated with the same markable class. Oracle mode can run within the current document or across the entire document corpus. This type of functionality is useful for annotation tasks that may involve identifying and marking spans of text that are repetitive or follow the same format For example, the 2011 i2b2/VA annotation task in which annotation of pronominal information was required for co-reference resolution (Figure 5).

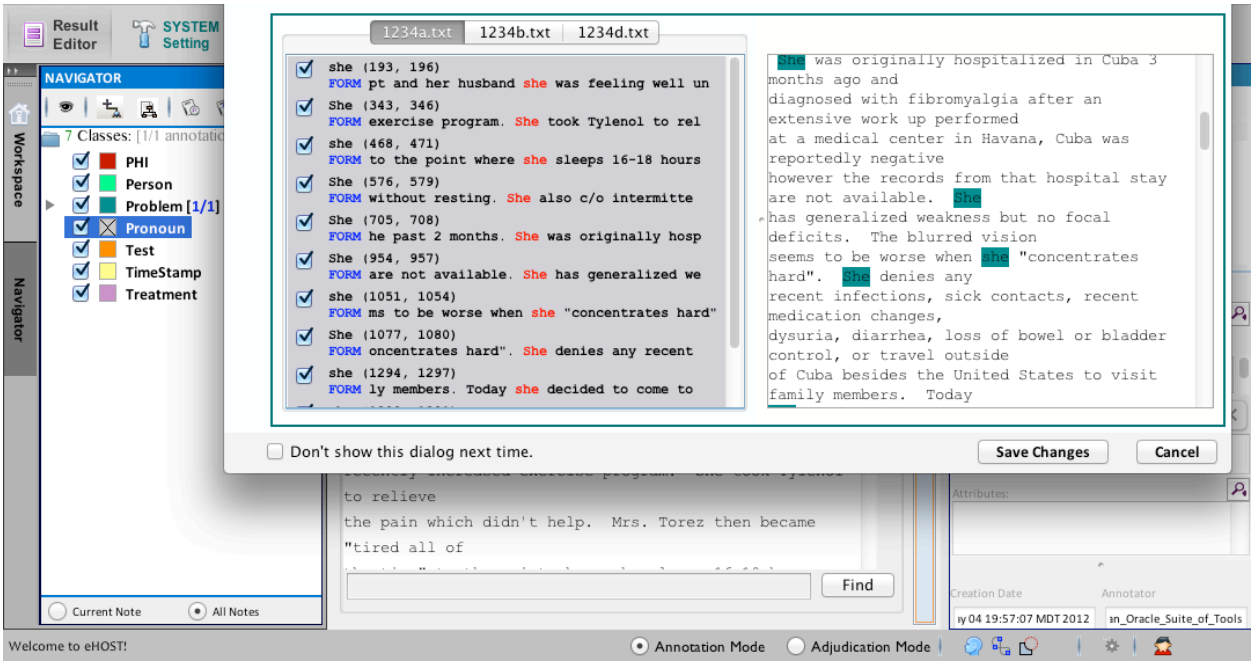


Figure 5. Example annotations generated using the eHOST "Oracle" mode

### 3.2 Semi-Automated Curation and Dictionary Management

Using the navigator window users can navigate to all annotations in either a single document or across an entire document corpus (Figure 6). The end user can curate annotations directly, create classes on the fly, or add attributes to annotations found from the navigator pane. These functions also allow users to easily identify spurious annotations introduced from machine-assisted approaches correct misclassification errors, and quickly curate all annotations within a single document or across an entire document corpus.

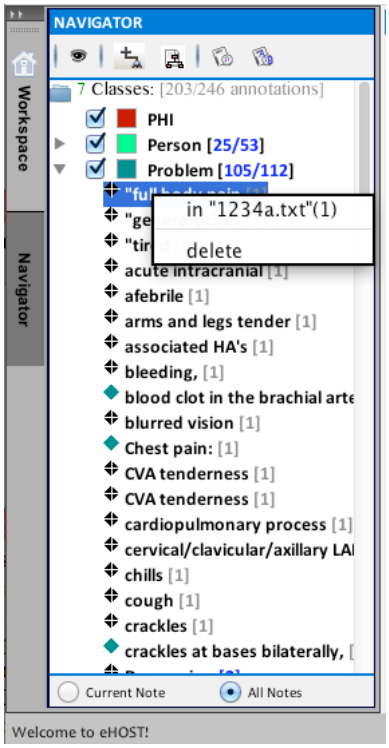


Figure 6. Semi-Automated curation within the document corpus

One task often associated with development of NLP systems involves manually creating or enhancing some existing representation of lexical knowledge that can be used as a domain specific dictionary. Using eHOST users can export annotations to create a dictionary of terms, phrases, or individual tokens that have been identified by human annotators and assigned to markable information classes. Once curated, annotated information can be exported as a new dictionary. User created dictionaries can be integrated with a database or exported and used in the creation of some ontologic representation of information using Protégé. Output from a dictionary manager is in the form of a delimited text file and can therefore be modified to fit any standardized information model or used to pre-annotate subsequent document batches.

### 3.3 Machine-Assisted Pre-Annotation

An interface is provided in eHOST that can be used for machine-assisted pre-annotation of documents in the active project corpus using either dictionaries or regular expressions based approaches. Users can import libraries of regular expressions or build their own regular expressions using a custom regular expression builder. Users can build and modify dictionaries created as part of annotation tasks that may include semi-automated curation steps. Dictionaries and regular expressions can also be coupled with the ConText algorithm (Chapman et al., 2007) to identify concept attributes such as negation, experiencer, and temporality. Pre-annotations derived from some external third party source such as an NLP system written as Knowtator XML outputs may also be imported into eHOST or passed to eHOST using CASPR.

Computational speed required for pre-annotation can be improved by selecting an option to use an internal statistical dictionary indexing function. This feature is particularly useful in situations where pre-annotation dictionaries are extremely large, such as where a subset of some standard vocabulary may be used to pre-annotate documents. Using the result editor and its associated functions annotators can add missed annotations, modifying existing annotations and delete spurious annotations. Handling pre-annotations in this way allows troubleshooting and error analysis of NLP system outputs imported into eHOST that can be shown to a reviewer in context and also facilitates interactive annotator training.

### 3.4 Machine-Assisted Verification

One of the more innovative features integrated with eHOST is the ability to verify and produce recommendations that help human annotators comply with syntactic and lexical rules that are specified by annotation task guidelines. Ma-

chine-Assisted verification is most useful when used on lexical or syntax rules to ensure that candidate phrases generated by automated systems are similar to those marked by humans. These rules rely more on adherence to patterns than on decision-making, so the strengths of human review with machine approaches to semi-automated verification can be leveraged. When identifying medical concepts, it is common that noun phrases are marked as candidates. The determination of how much of a noun phrase to mark (inclusion of articles, adjectives, noun-modifiers, prepositional phrases) and at what granularity (simple nouns or complex noun phrases) may vary with each project.

The verifier allows portions of an annotation guideline to be programmed into rules that check for consistency. Rules check whether a word appears within a user-defined window before and after an annotation. Each rule can be linked to text that describes why the annotation was flagged. Annotators are then provided suggestions on the correct span based on the rule. Using the surrounding text, the guideline text, and the suggestion, the annotator can determine the final span for an annotation. These machine-assisted verifier functions help support reference standard generation by providing the context of annotations that seem to fail syntactic and lexical rules while allowing human annotators to focus on domain expertise required to identify and classify information found in clinical texts.

## Conclusion

Our prototype system provides functionalities that have been created to more efficiently support reference standard generation including machine-assisted annotation approaches. It is our hope that these system features will serve as the basis for the further development efforts that will be part of an enterprise level system. Outputs of such an annotation tool could be used as inputs for pipeline NLP systems or as one component of a common workbench of tools used for clinical NLP development tasks.

We have implemented and tested eHOST for the 2010 and 2011 i2b2/VA challenge annotation tasks and annotation projects for the Consortium for Healthcare Informatics Research (CHIR). The stand-alone eHOST client tool is available from http://code.google.com/p/ehost along with a demonstration project, a users guide, API documentation, and source code. The eHOST/CASPR interfaces will be used to support a large-scale crowd sourced annotation task used for annotation of disorders, temporal expressions, uncertainty, and negation along with data standardization. These efforts will include more rigorous analysis and usability assessment of eHOST/CASPR for crowd sourcing and other small and large-scale annotation projects.

## Acknowledgments

Support and funding was provided by the VA Salt Lake City HealthCare System and the VA Consortium for Healthcare Informatics Research (CHIR), VA HSR HIR 08-374, the VA Informatics and Computing Infrastructure (VINCI), VA HIR 08-204, and NIH Grant U54 HL 108460 for integrating Data for Analysis, Annonymization and Sharing (iDASH), NIGMS 7R01GM090187.

## References

Beatrice Alex, Claire Grover, Barry Haddow, Mijail Kabadjov, Ewan Klein, Michael Matthews, Stuart Roebuck, Richard Tobin, and Xinglong Wang. 2008. Assisted curation: does text mining really help? In: *Proceedings of the Pacific Symposium on Biocomputing*.

Wendy W. Chapman, David Chu, John N. Dowling. 2007. ConText: An Algorithm for Identifying Contextual Features from Clinical Text. In: *ACL-07 2007*.

Carol Friedman, Hongfang Liu, Lyudmila Shagina, Stephen Johnson, George Hripcsak. 2001. Evaluating the UMLS as a source of lexical knowledge for medical language processing. In: *Proc AMIA Symp, 2001: 189-93.*

Karen Fort and Saggot B. 2010. Influence of Pre-Annotation on POS-tagged Corpus Development. In: *Proceedings of the Fourth Linguistic Annotation Workshop. ACL 2010: 56-63.*

Pei-Yun Hsueh, Prem Melville, Vikas Sindhwani. 2009. Data Quality from Crowdsourcing: A study of Annotation Selection Criteria. In: *Proceedings of the NAACL HLT Workshop on Active Learning for Natural Language Processing. June 2009: 27-35*.

Danielle Mowery, Henk Harkema, Wendy W. Chapman. 2008. Temporal Annotation of Clinical Text. In: *ACL-08 2008*.

Mark A. Musen, John Gennari, Henrik Eriksson, Samson W. Tu, and Angel R. Puerta. 1995. PROTEGE-II: computer support for development of intelligent systems from libraries of components. In: *Medinfo 1995*.

Aurélie Névéol, Rezarta Islamaj-Doğan, Zhiyong Lu. 2011. Semi-automatic semantic annotation of PubMed queries: a study on quality, efficiency, satisfaction. In: *J Biomed Inform. 2011 Apr; 44(2):310-8.*

Stefanie Nowak and Stefan Ruger. 2010. How Reliable are Annotations via Crowdsourcing? A Study about Inter-Annotator Agreement for Multi-label Image Annotation. In: *MIR 10 2010*.

Philip V. Ogren. 2006. Knowtator a protege plug-in for annotated corpus construction. In: *Proceedings of the 2006 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, 2006: 273-5.*

Philip V. Ogren, Guergana K. Savova, Christopher G. Chute. 2008. Constructing Evaluation Corpora for Automated Clinical Named Entity Recognition. In: *Proceedings of the sixth international conference on Language Resources and Evaluation LREC 2008: 3143-3150.*

Angus Roberts, Robert Gaizauskas, Mark Hepple, Neil Davis, George Demetriou, Yikun Guo, Jay Kola, Ian Roberts, Andrea Setzer, Archana Tapuria, Bill Wheeldin. 2007. The CLEF corpus: semantic annotation of clinical text. In: *AMIA Annu Symp Proc, 625-9.*

Angus Roberts, Robert Gaizauskas, Mark Hepple, George Demetriou, Yikun Guo, Ian Roberts, Andrea Setzer. 2009. Building a semantically annotated corpus of clinical texts. In: *J Biomed Inform, 42(5): 950-66.*

Ellen Riloff. 1993. Automatically constructing a dictionary for information extraction tasks. In: *Proceedings of the Eleventh National Conference on Artificial Intelligence, 811-816.*

Burr Settles, Mark Craven, and Lewis Friedland. 2008. Active Learning with Real Annotation Costs. In: *Proceedings of the NIPS Workshop on Cost-Sensitive Learning*. 2008.

Burr Settles. 2009. Active Learning Literature Survey. In: *Computer Sciences Technical Report 1648. University of Wisconsin-Madison. 2009.*

Brett R. South, Shuying Shen, F. Jeff Friedlin, Matthew H. Samore, and Stephane M. Meystre. 2010. Enhancing Annotation of Clinical Text using Pre-Annotation of Common PHI. In: *AMIA Annu Symp Proc. 2010.*

Brett R. South, Shuying Shen, Robyn Barrus, Scott L. DuVall, Ozlem Uzuner, and Charlene Weir. 2011. Qualitative analysis of workflow modifications used to generate the reference standard for the 2010 i2b2/VA challenge. In: *AMIA Annu Symp Proc. 2011.*

Özlem Uzuner, Imre Solti, Fei Xia, and Eithon Cadag. 2010. Community annotation experiment for ground truth generation for the i2b2 medication challenge. In: *J Am Med Inform Assoc, 2010. 17(5):519-23.*

Özlem Uzuner, Brett R. South, Shuying Shen, and Scott L. DuVall. 2011. 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text. In: *JAMIA 18(5): 552-556.*

Özlem Uzuner, Andreea Bodnari, Shuying Shen, Tyler B. Forbush, John Pestian, and Brett R. South. 2012. Evaluating the state of the art in coreference resolution for electronic medical records. In: *JAMIA doi: 10.1136/amiajnl-2011-000784.*

Aobo Wang, Cong Duy Vu Hoang, and Min-Yen Kan. 2010. Perspectives on Crowdsourcing Annotations for Natural Language Processing. In: *CSIDM Project No. CSIDM-200805.*

Alexander S. Yeh, Lynette Hirschman, and Alexander A. Morgan. 2003. Evaluation of text data mining for database curation: Lessons learned from the KDD challenge cup. In: *Bioinformatics, 19(Suppl 1): i331–339, 2003.*