Journal of the American Medical Informatics Association, 25(3), 2018, 331–336
doi: 10.1093/jamia/ocx132
Advance Access Publication Date: 24 November 2017
Brief Communication

Brief Communication

CLAMP – a toolkit for efficiently building customized clinical natural language processing pipelines

Ergin Soysal,[1] Jingqi Wang,[1] Min Jiang,[1] Yonghui Wu,[1] Serguei Pakhomov,[2] Hongfang Liu,[3] and Hua Xu[1]

[1]School of Biomedical Informatics, University of Texas Health Science Center at Houston, Houston, TX, USA, [2]Department of Pharmaceutical Care and Health System, University of Minnesota Twin Cities, Minneapolis, MN, USA and [3]Department of Health Sciences Research, Mayo Clinic, Rochester, MN, USA

Corresponding Author: Hua Xu, 7000 Fannin, Suite 600, Houston, TX 77030, USA. E-mail: Hua.Xu@uth.tmc.edu. Phone: 713.500.3924

Received 4 May 2017; Revised 28 September 2017; Editorial Decision 2 October 2017; Accepted 19 October 2017

ABSTRACT

Existing general clinical natural language processing (NLP) systems such as MetaMap and Clinical Text Analysis and Knowledge Extraction System have been successfully applied to information extraction from clinical text. However, end users often have to customize existing systems for their individual tasks, which can require substantial NLP skills. Here we present CLAMP (Clinical Language Annotation, Modeling, and Processing), a newly developed clinical NLP toolkit that provides not only state-of-the-art NLP components, but also a user-friendly graphic user interface that can help users quickly build customized NLP pipelines for their individual applications. Our evaluation shows that the CLAMP default pipeline achieved good performance on named entity recognition and concept encoding. We also demonstrate the efficiency of the CLAMP graphic user interface in building customized, high-performance NLP pipelines with 2 use cases, extracting smoking status and lab test values. CLAMP is publicly available for research use, and we believe it is a unique asset for the clinical NLP community.

Key words: natural language processing, machine learning, clinical text processing

INTRODUCTION

In the medical domain, clinical documents contain rich information needed for both clinical research and daily practice.[1] Natural language processing (NLP) technologies play an important role in unlocking patient information from clinical narratives. Several general-purpose NLP systems have been developed to process clinical text, including Clinical Text Analysis and Knowledge Extraction System (cTAKES),[2] MetaMap[3]/MetaMap Lite,[4] and Medical Language Extraction and Encoding System.[5] These systems can extract diverse types of clinical information and have been successfully applied to many information extraction tasks, such as detection of smoking status[6] and identification of respiratory findings[7] and suspicious breast cancer lesions.[8] In addition, many NLP systems have been developed to extract specific types of information from clinical text, eg, medication information extraction systems,[9] temporal information extraction tools, and deidentification systems.[10–13]

Despite the success of current NLP systems, studies have shown that it takes substantial effort for end users to adopt existing NLP systems.[14] Furthermore, users often report reduced performance when an existing system is applied without customization beyond its original purpose, eg, when moving to a different institution, different types of clinical notes, or a different application.[15] For machine learning–based NLP systems, statistical models may have to be retrained using target domain data to achieve desired performance, due to differences between the target domain and the original domain.[16] The effort of customizing existing NLP systems for individual applications is nontrivial and often requires substantial NLP knowledge and skills, which could be challenging in settings with

limited NLP expertise. This prevents the widespread adoption of NLP technologies in the medical domain.

To address this problem, we have developed a new clinical NLP toolkit called CLAMP (Clinical Language Annotation, Modeling, and Processing), which provides not only state-of-the-art NLP modules, but also an integrated development environment with user-friendly graphic user interfaces (GUIs) to allow users to quickly build customized NLP pipelines for individual applications.

METHODS

Architecture, components, and resources

CLAMP is implemented in Java as a desktop application. It builds on the Apache Unstructured Information Management Architecture™ (UIMA) framework[17] to maximize its interoperability with other UIMA-based systems such as cTAKES. CLAMP also supports the Apache UIMA Asynchronous Scaleout (AS) framework for asynchronous processing in a distributed environment. (UIMA AS is a flexible and powerful scaleout solution for NLP pipelines maintained by the Apache Foundation: https://uima.apache.org/doc-uimaas-what.html).

CLAMP follows a pipeline-based architecture that decomposes an NLP system into multiple components. Most CLAMP components are built on approaches developed in our lab and proven in multiple clinical NLP challenges, such as i2b2 (2009 and 2010, named entity recognition [NER] tasks, ranked no. 2),[18,19] Shared Annotated Resources/Conference and Labs of the Evaluation Forum (2013 Task 2, abbreviation recognition, ranked no. 1),[20] and SemEVAL (2014 Task 7, encoding to concept unique identifiers [CUIs] in the Unified Medical Language System [UMLS], ranked no. 1).[21] Various technologies, including machine learning–based methods and rule-based methods, were used when developing these components. A list of CLAMP's available components and their specifications follows:

- Sentence boundary detection: CLAMP provides both a machine learning–based sentence detector using OpenNLP[22] and a configurable rule-based sentence boundary detection component.
- Tokenizer: CLAMP contains 3 types of tokenizers: the machine learning–based OpenNLP tokenizer, a delimiter-based (eg, white space) tokenizer, and a rule-based tokenizer with various configuration options.
- Part-of-speech tagger: CLAMP implements a machine learning–based part-of-speech tagger that is retrained on clinical corpora.[23]
- Section header identification: CLAMP uses a dictionary-based approach for identifying section headers. We provide a list of common section headers collected from clinical documents. Users can extend/optimize the list based on the document types in their tasks.
- Abbreviation reorganization and disambiguation: CLAMP partially implements the clinical abbreviation recognition and disambiguation (CARD) framework.[20,24,25] Users can specify their own abbreviation list if needed.
- Named entity recognizer: CLAMP presents 3 different types of NER approach: (1) a machine learning–based NER component that uses the conditional random fields (CRF) algorithm in the CRFSuite library,[26] following our proven methods in the 2010 i2b2 challenge[19]; (2) a dictionary-based NER system with comprehensive lexicon collected from multiple resources such as the UMLS; users can provide their own lexicons and specify options for dictionary lookup algorithms, such as with or without stemming; and (3) a regular expression-based NER for entities with common patterns such as dates and phone numbers.
- Assertion and negation: CLAMP provides a machine learning–based approach for assertion detection that we developed in the 2010 i2b2 challenge, which determines 6 types of assertion: present, absent, possible, conditional, hypothetical, and not associated with the patient. In addition, the rule-based NegEx[27] algorithm is also implemented and users can specify additional negation lexicons and rules.
- UMLS encoder: After an entity is recognized, it can be mapped to UMLS CUIs using this component (also known as an entity linking task). The UMLS encoder is built on our top-ranked algorithm in the SemEval-14 challenge, which calculates the similarity of an entity and candidate concepts using the vector space model.[28]
- Rule engine: We implemented the Apache Ruta Rule Engine[29] in CLAMP, thus allowing users to add rules before or after the machine learning algorithms to fine-tune performance. Users can develop Ruta rules by either editing the rule files directly or using the interface for rule specification.

In addition to the above NLP components, we prepared 338 clinical notes with entity annotations derived from MTSamples,[30] a collection of different types of transcribed clinical note examples made for clinical documentation education, as a test corpus to be co-released with CLAMP.

GUI development

CLAMP's GUI was built on top of the Eclipse Framework, which provides built-in components for developing interactive interfaces. Figure 1 shows a screenshot of the main interface of CLAMP for building an NLP pipeline. Built-in NLP components are listed in the top-left palette, and the corpus management palette is in the left-middle area. User-defined NLP pipelines are displayed in the left-bottom palette. The details of each pipeline are displayed in the center area after users click a pipeline. A pipeline can be visually created by dragging and dropping components into the middle window, following specific orders (eg, tokenizer should be before NER). After selecting the components of a pipeline, users can click each component to customize its settings. For example, for regular expression-based or dictionary-based NER components, users can specify their own regular expression or dictionary files. For machine learning–based NER, users can swap the default machine learning model with models trained on local data.

To facilitate building machine learning–based NER modules on local data, CLAMP provides interfaces for corpus annotation and model training. We developed a fully functional annotation interface (by leveraging the brat annotation tool[31]), which allows users to define types of entity of interest and annotate them following guidelines (see Figure 2 for the annotation interface). After finishing annotation, users can click the training icon to build CRF models using the annotated corpus. The system will automatically report its performance based on user-specified evaluation settings (eg, 5-fold cross-validation). Figure 3 shows the popup window where users can select different types of features to build the CRF-based NER models.

Evaluation

CLAMP is currently available in 2 versions: (1) CLAMP-CMD, a command line NLP system to extract clinical concepts, built on

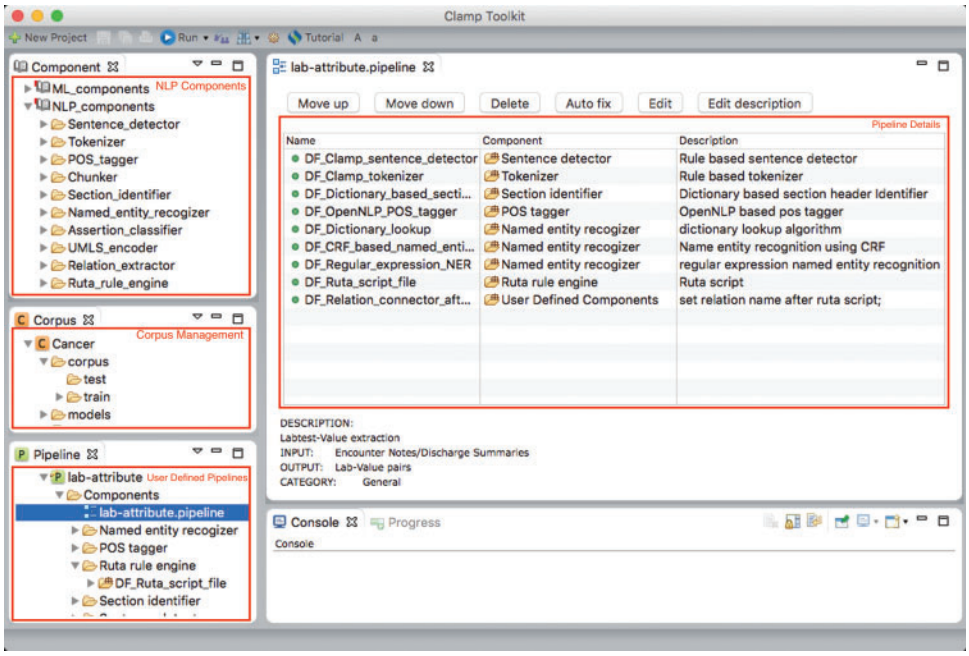

Figure 1. The user interface for building a pipeline in CLAMP.

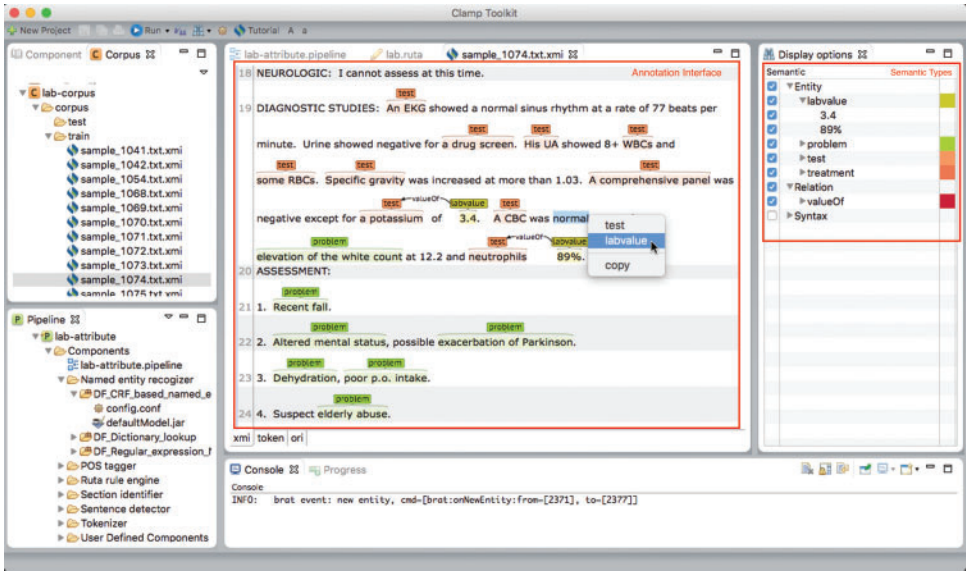

Figure 2. The interface in CLAMP for annotating entities and relations.

default CLAMP components, and (2) CLAMP-GUI, which provides the GUI for building customized NLP pipelines. We evaluated CLAMP-CMD on 2 NLP tasks: (1) an NER task to recognize problems, treatments, and lab tests, similar to the 2010 i2b2 challenge[32]; and (2) a UMLS CUI encoding task for diseases, similar to the 2014 SemEVAL Task 7.[28] For the NER task, we included 3 corpora annotated following the guidelines in the i2b2 challenge: (1) the discharge summaries used in the 2010 i2b2 challenge (i2b2, 871 annotated notes); (2) a new corpus of outpatient clinic visit notes from the University of Texas Health Science Center at Houston (UTNotes, 1351 notes); and (3) a new corpus of mock clinical documents from MTSamples, as described in the Methods section (MTSamples, 338 notes). We randomly selected 50 notes from each corpus as the test sets for evaluating CLAMP-CMD, and combined the remaining notes for training the CRF-based NER model. Standard measurements of precision (P), recall (R), and *F*-measure (F_1) were reported for each corpus using both the exact and relaxed matching evaluation scripts in the i2b2 challenge.[19]

For the UMLS CUIs encoding task, we compared CLAMP-CMD with MetaMap, MetaMap Lite, and cTAKES using the SemEVAL-2014 Task 7 corpus, which contains 431 notes. In order to compare these systems, we limited the CUIs to those in Systematized Nomenclature of Medicine – Clinical Terms only and slightly changed the evaluation criteria: if a CUI is identified by both the gold standard and a system within the same sentence, we treat it as a true positive, without considering offsets. As MetaMap and cTAKES sometimes

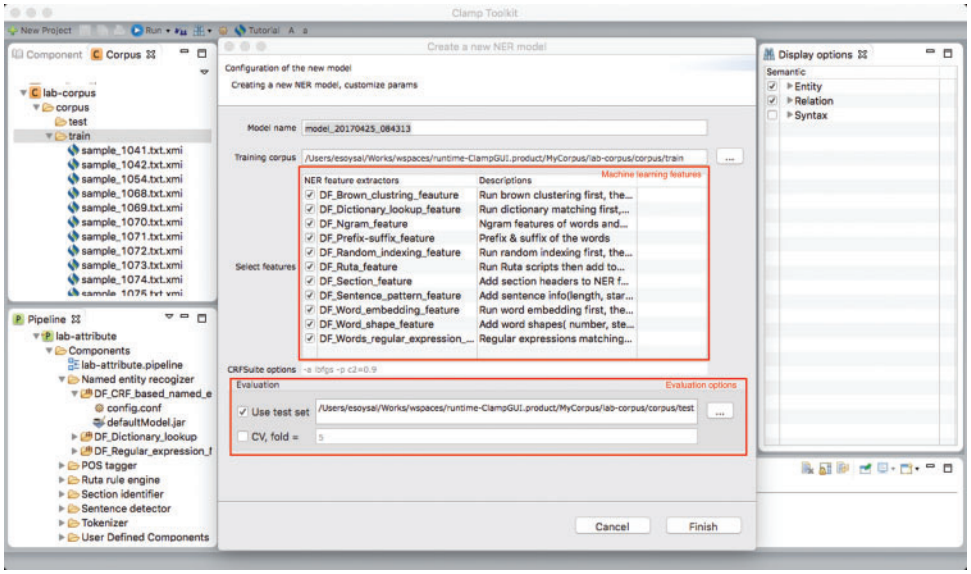

Figure 3. The interface for selecting features and evaluation options for building machine learning–based NER models using CLAMP.

Table 1. Performance of CLAMP-CMD on the NER task (problem, treatment, and test) across different corpora

Corpus	No. of entities	State-of-the-art F1 (exact vs relaxed)	Exact match			Relaxed match		
			Precision	Recall	F1	Precision	Recall	F1
i2b2	72 846	0.85/0.92[a]	0.89	0.86	0.88	0.96	0.93	0.94
MTSamples	25 531	N/A	0.84	0.81	0.83	0.92	0.89	0.91
UTNotes	124 869	N/A	0.92	0.90	0.91	0.96	0.94	0.95

[a]The best performance reported in the shared task.

output multiple CUIs for one concept, we either selected the one with the highest score or randomly picked one from CUIs with a tied score (or no score).

For CLAMP-GUI, we conducted 2 use cases to demonstrate its efficiency: (1) detect smoking status (past smoker, current smoker, or nonsmoker), similar to the task in,[33] and (2) extract lab test names and associated values. For smoking status detection, we annotated 300 sentences with a smoking keyword; 100 were used to develop the rule-based system and 200 were used to test the system. For lab test name/value extraction, we annotated 50 clinic visit notes and divided them into a development set (25 notes) and a test set (25 notes). Performance of the customized pipelines and development time using CLAMP-GUI was then reported using the test sets.

RESULTS

Table 1 shows the results of CLAMP-CMD on the NER task across different corpora. When the relaxed matching criterion was used, CLAMP-CMD could recognize clinical entities (problems, treatments, and tests) with reasonable *F*-measures >90% across different corpora. Table 2 shows the results of different clinical NLP systems on mapping disease entities to the UMLS CUIs on the SemEVAL-2014 corpus. CLAMP achieved *F*-measure superior to MetaMap and cTAKES, with faster processing speed, although MetaMap Lite achieved the best *F*-measure. In Supplementary Table 1, we also report evaluation results of individual CLAMP components (eg, tokenizer and sentence boundary detector) using existing corpora. The performance of each component was comparable to the state-of-the-art results reported by other systems.[2,34–36]

For smoking status detection, we quickly built a rule-based pipeline (~4 h) using CLAMP-GUI, which could detect patients' smoking status (nonsmoker, current smoker, past smoker) with accuracies of 0.95, 0.89, and 0.90, respectively. For lab test names/values, we built a hybrid pipeline that combines machine learning (eg, CLAMP's default NER model for lab names) and rules (eg, for extracting lab values and linking values to names) within approximately 12 h (8 h of annotation and 4 h of customizing components). The pipeline contains 8 different components (Sentence Boundary, Tokenizer, Section Header, POS Tagger, CRF-based NER, Regular Expression–based NER, Ruta Rule Engine, and Relationship Connector) and achieved *F*-measures of 0.98, 0.85, and 0.81 for recognizing lab test names, values, and their relations in the test set, respectively. The detailed processes for developing these 2 pipelines were recorded as videos, available at http://clamp.uth.edu/tutorial.php.

DISCUSSION

GUI-based NLP tools such as General Architecture for Text Engineering[37] have been developed and are widely used in the general domain, but few exist in the medical domain. The main advantage of CLAMP is that it provides GUIs to allow non-NLP experts to quickly develop customized clinical information extraction pipelines using proven state-of-the-art methods, eg, machine learning–based

Table 2. Performance of CLAMP (version 1.3), MetaMap (2016), MetaMap Lite (2016, version 3.4), and cTAKES (version 4) on extracting disease concepts to UMLS CUIs using the SemEval-2014 corpus

NLP System	No. of entities			Performance			Processing Time (s/doc)[a]
	Correct	Predict	Gold	Precision	Recall	F1	
CLAMP	7228	9329	13 555	0.775	0.533	0.632	0.95
MetaMap	5574	10 214	13 555	0.546	0.411	0.469	7.07
MetaMap Lite	8009	11 282	13 555	0.710	0.591	0.645	1.95
cTAKES	9126	19 713	13 555	0.463	0.673	0.549	2.27

[a]All evaluations were performed on a MacBook with 16G RAM and Intel i7 as CPU with 4 cores. For MetaMap, the default setting was used. For cTAKES, the fast dictionary lookup annotator was used. The performance of both MetaMap and cTAKES could be further improved by optimizing their settings.

NER models. Although there is an increasing trend of applying machine learning to clinical NLP systems,[38,39] widely used systems such as cTAKES and MetaMap do not provide easy ways to build machine learning–based models. To the best of our knowledge, CLAMP is the first comprehensive clinical NLP system that provides interfaces for building hybrid solutions (machine learning plus rules) for information extraction. However, such a GUI-based tool also comes with limitations; eg, some tasks are complex and are difficult to build through GUIs. To address such issues, we also provide application program interfaces for individual components in CLAMP, so that professional developers can build integrated systems by directly calling them.

To further facilitate building NLP solutions for end users, we are developing a library of NLP pipelines for diverse types of clinical information in CLAMP. For example, if a user wants to extract ejection fraction information from local text and there is a prebuilt pipeline for ejection fraction, he/she can just copy the prebuilt pipeline to his/her own workspace and start customizing each component of the pipeline based on local data, without starting from scratch, thus saving development time. So far we have developed >30 pipelines for extracting different types of information from clinical text, ranging from general pipelines (eg, medication and signature information) to specific pipelines (eg, smoking status).

CLAMP was developed with interoperability in mind. It can directly exchange objects with cTAKES via Apache UIMA interfaces. We have also developed wrappers for displaying MetaMap outputs in CLAMP and for integrating CLAMP with other NLP frameworks such as Leo.[40] Our future work includes developing more NLP components (eg, syntactic parsing and relation extraction), improving GUIs by conducing formal usability testing, normalizing its outputs to common data models such as those of the Observational Medical Outcomes Partnership,[41] as well as expanding the library of NLP pipelines for different information extraction tasks.

CLAMP is currently freely available for research use at http://clamp.uth.edu. To develop a sustainable model for continuing its development and maintenance, we are evaluating a paid licensing model for industrial use. Since its release in 2016, there have been >160 downloads by >120 academic institutions and industrial entities.

CONCLUSION

CLAMP integrates proven state-of-the-art NLP algorithms and user-friendly interfaces to facilitate efficient building of customized NLP pipelines for diverse clinical applications. We believe it will complement existing clinical NLP systems and help accelerate the adoption of NLP in clinical research and practice.

COMPETING INTEREST

The authors have no competing interests to declare.

FUNDING

This work was supported in part by grants from the National Institute of General Medical Sciences, GM102282 and GM103859, the National Library of Medicine, LM 010681, the National Cancer Institute, CA194215, and the Cancer Prevention and Research Institute of Texas, R1307.

CONTRIBUTORS

Study planning: SP, HL, HX. Software design and implementation: ES, JW, MJ, YW. Wrote the paper: ES, SP, HX. All authors read and approved the final manuscript.

SUPPLEMENTARY MATERIAL

Supplementary material is available at *Journal of the American Medical Informatics Association* online.

REFERENCES

1. Demner-Fushman D, Chapman WW, McDonald CJ. What can natural language processing do for clinical decision support? *J Biomed Inform*. 2009;42(5):760–72.
2. Savova GK, Masanz JJ, Ogren PV, *et al*. Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications. *J Am Med Inform Assoc*. 2010;17(5):507–13.
3. Aronson AR, Lang F-M. An overview of MetaMap: historical perspective and recent advances. *J Am Med Inform Assoc*. 2010;17(3):229–36.
4. Demner-Fushman D, Rogers WJ, Aronson AR. MetaMap Lite: an evaluation of a new Java implementation of MetaMap. *J Am Med Inform Assoc*. 2017;24(4):841–44.
5. Friedman C. Towards a comprehensive medical language processing system: methods and issues. *Proc AMIA Annu Fall Symp*. 1997:595–99.
6. Savova GK, Ogren PV, Duffy PH, *et al*. Mayo Clinic NLP system for patient smoking status identification. *J Am Med Inform Assoc*. 2008;15(1):25–28.
7. Chapman WW, Fiszman M, Dowling JN, *et al*. Identifying respiratory findings in emergency department reports for biosurveillance using MetaMap. *Medinfo*. 2004;11(Pt 1):487–91.
8. Jain NL, Friedman C. Identification of findings suspicious for breast cancer based on natural language processing of mammogram reports. *Proc AMIA Annu Fall Symp*. 1997:829–33.
9. Xu H, Stenner SP, Doan S, *et al*. MedEx: a medication information extraction system for clinical narratives. *J Am Med Inform Assoc*. 2010;17(1):19–24.
10. Tang B, Wu Y, Jiang M, *et al*. A hybrid system for temporal information extraction from clinical text. *J Am Med Inform Assoc*. 2013;20(5):828–35.

11. Uzuner O, Luo Y, Szolovits P. Evaluating the state-of-the-art in automatic de-identification. *J Am Med Inform Assoc*. 2007;14(5):550–63.
12. Dernoncourt F, Lee JY, Uzuner O, *et al*. De-identification of patient notes with recurrent neural networks. *J Am Med Inform Assoc*. 2017;24(3):596–606.
13. Stubbs A, Uzuner Ö. Annotating longitudinal clinical narratives for de-identification: The 2014 i2b2/UTHealth corpus. *J Biomed Inform*. 2015;58 (Suppl):S20–29.
14. Zheng K, Vydiswaran VG, Liu Y, *et al*. Ease of adoption of clinical natural language processing software: an evaluation of five systems. *J Biomed Inform*. 2015;58 (Suppl):S189–96.
15. Chapman WW, Nadkarni PM, Hirschman L, *et al*. Overcoming barriers to NLP for clinical text: the role of shared tasks and the need for additional creative solutions. *J Am Med Inform Assoc*. 2011;18(5):540–43.
16. Liu M, Shah A, Jiang M, *et al*. A study of transportability of an existing smoking status detection module across institutions. *AMIA Annu Symp Proc*. 2012;2012:577–86.
17. Ferrucci D, Lally A, Verspoor K, *et al*. Unstructured Information Management Architecture (UIMA) *Version 1.0*. 2008.
18. Uzuner O, Solti I, Cadag E. Extracting medication information from clinical text. *J Am Med Inform Assoc*. 2010;17(5):514–18.
19. Uzuner Ö, South BR, Shen S, *et al*. 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text. *J Am Med Inform Assoc*. 2011;18(5):552–56.
20. Wu Y, Tang B, Jiang M, *et al*. Clinical Acronym/Abbreviation Normalization using a Hybrid Approach. Proc CLEF Evaluation Labs and Workshop. 2013.
21. Tang YZJWB, Jiang YWM, Xu YCH. UTH_CCB: a report for SemEval 2014–task 7 analysis of clinical text. *SemEval* 2014. 2014:802.
22. Baldridge J. *The OpenNLP Project*. 2015. http://opennlp.apache.org. Accessed April 6, 2015.
23. Fan J-w, Yang EW, Jiang M, *et al*. Syntactic parsing of clinical text: guideline and corpus development with handling ill-formed sentences. *J Am Med Inform Assoc*. 2013;20(6):1168–77.
24. Murtola SS, Suominen H, Martinez D, *et al*. *Task 2: ShARe/CLEF eHealth Evaluation Lab*. 2013.
25. Wu Y, Denny JC, Trent Rosenbloom S, *et al*. A long journey to short abbreviations: developing an open-source framework for clinical abbreviation recognition and disambiguation (CARD). *J Am Med Inform Assoc*. 2017;24(e1):e79–86.
26. Okazaki N. *CRFsuite: a Fast Implementation of Conditional Random Fields (CRFs)*. 2007. http://www.chokkan.org/software/crfsuite/. Accessed July 20, 2017.
27. Chapman WW, Bridewell W, Hanbury P, *et al*. A simple algorithm for identifying negated findings and diseases in discharge summaries. *J Biomed Inform*. 2001;34(5):301–10.
28. Pradhan S, Elhadad N, Chapman W, *et al*. SemEval-2014 Task 7: analysis of clinical text. *SemEval 2014*. 2014;199(99):54.
29. Kluegl P, Toepfer M, Beck P-D, *et al*. UIMA Ruta: rapid development of rule-based information extraction applications. *Nat Language Eng*. 2016;22(01):1–40.
30. *Transcribed Medical Transcription Sample Reports and Examples – MTSamples*. 2015. www.mtsamples.com/. Accessed April 6, 2015.
31. Stenetorp P, Pyysalo S, Topić G, *et al*. BRAT: a web-based tool for NLP-assisted text annotation. *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, Avignon, France, April 23–27, 2012.
32. Weber GM, Kohane IS. Extracting physician group intelligence from electronic health records to support evidence based medicine. *PLoS One*. 2013;8(5):e64933.
33. Uzuner O, Goldstein I, Luo Y, *et al*. Identifying patient smoking status from medical discharge records. *J Am Med Inform Assoc*. 2008;15(1):14–24.
34. Fan JW, Prasad R, Yabut RM, *et al*. Part-of-speech tagging for clinical text: wall or bridge between institutions? *AMIA Annu Symp Proc*. 2011;2011:382–91.
35. Griffis D, Shivade C, Fosler-Lussier E, *et al*. A quantitative and qualitative evaluation of sentence boundary detection for the clinical domain. *AMIA Jt Summits Transl Sci Proc*. 2016;2016:88–97.
36. Dai HJ, Syed-Abdul S, Chen CW, *et al*. Recognition and evaluation of clinical section headings in clinical documents using token-based formulation with conditional random fields. *Biomed Res Int*. 2015;2015: 873012.
37. Cunningham H. GATE, a general architecture for text engineering. *Comput Hum*. 2002;36(2):223–54.
38. Boag W, Wacome K, Naumann T, *et al*. CliNER: A lightweight tool for clinical named entity recognition. *AMIA Jt Summits Clin Res Inform (poster)*. 2015.
39. Dernoncourt F, Lee JY, Szolovits P. NeuroNER: an easy-to-use program for named-entity recognition based on neural networks. *arXiv preprint. arXiv:170505487*. 2017.
40. Cornia R, Patterson O, Ginter T, *et al*. Rapid NLP development with Leo. AMIA Annu Symp Proc. 2014;2014:1356.
41. Overhage JM, Ryan PB, Reich CG, *et al*. Validation of a common data model for active safety surveillance research. *J Am Med Inform Assoc*. 2012;19(1):54–60.