

Towards User-centered Corpus Development: Lessons Learnt from Designing and Developing MedTator

Time Flies Like a Banana: Advancing Natural Language Processing Techniques
S107

Huan He, PhD

Mayo Clinic

Twitter: #AMIA2022



Disclosure



I and my spouse/partner have no relevant relationships with commercial interests to disclose.

Learning Objectives

After participating in this session the learner should be better able to:

- Learn the challenges of developing a text annotation tool
- Learn the design and our experience in tool development

Outline

1. Background and challenges
2. Architecture design
3. Lessons we learned

Background - NLP in healthcare

NLP has been widely applied in healthcare-related domains

Uncovering **Hidden Symptoms** Identify **Eligible Lung Cancer** Screening Patients

Extracting Biomedical Factual **Knowledge** Identify Reasons for **Statin Non-Adherence**

Examiner Note **Extraction** Sampling **Adverse Drug Events** Identifying the **Risk for Hospitalizations**

Identify **Social Needs** Unraveling the “Other” Diagnosis of **Suspected Suicide Attempts**

Analyze ADRD Caregivers’ **Mental** Health Needs Clinical **Relation Extraction** Confirm MGUS **Diagnoses**

Build Disease-specific **Symptom** Vocabulary Identifying **Transgender** and Gender **Diverse**

Clinical Trial Eligibility **Criteria Representation** Predict Functional **Pain Score** Temporal **Reasoning**

Predict **Oral Cancer Risk** Identifying **Menopausal Status** Gleaning **Prediabetic** Risk and Factors

Extracting **Cancer Phenotypes** Analyzing **COVID-19 Vaccination** Sentiment Dynamics

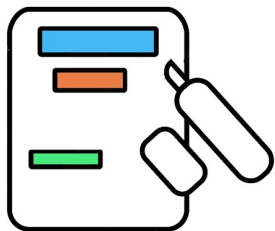
NLP in AMIA 2022 Annual Symposium

Background - Gold Standard Corpus



Gold Standard
Corpus (GSC)

High-quality GSC is used for developing NLP tools, training machine learning models, fine-tuning models, and providing the basis for evaluation to test the performance of NLP systems



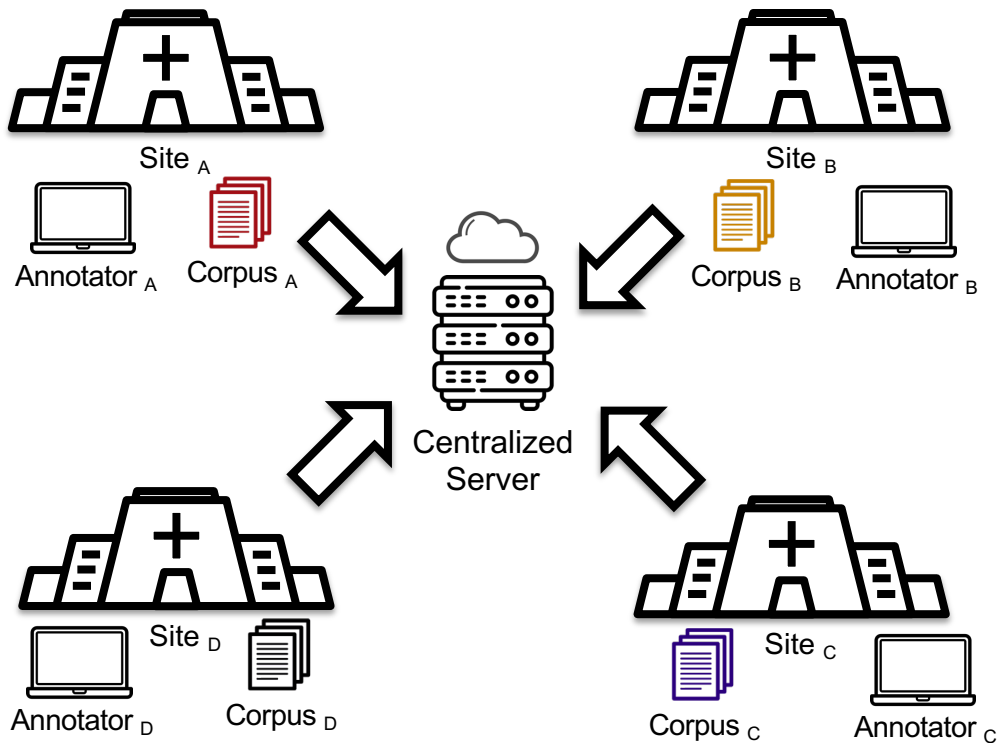
Text Annotation Tool

Building such a high-quality corpus requires considerable time and expertise to go through the whole annotation process with a comfortable annotation tool to facilitate the annotation process.

Background - Multi-site Annotation

Centralized annotation

Each site uploads raw corpus to a centralized server, and annotators can log in to this server to annotate.



Major concerns

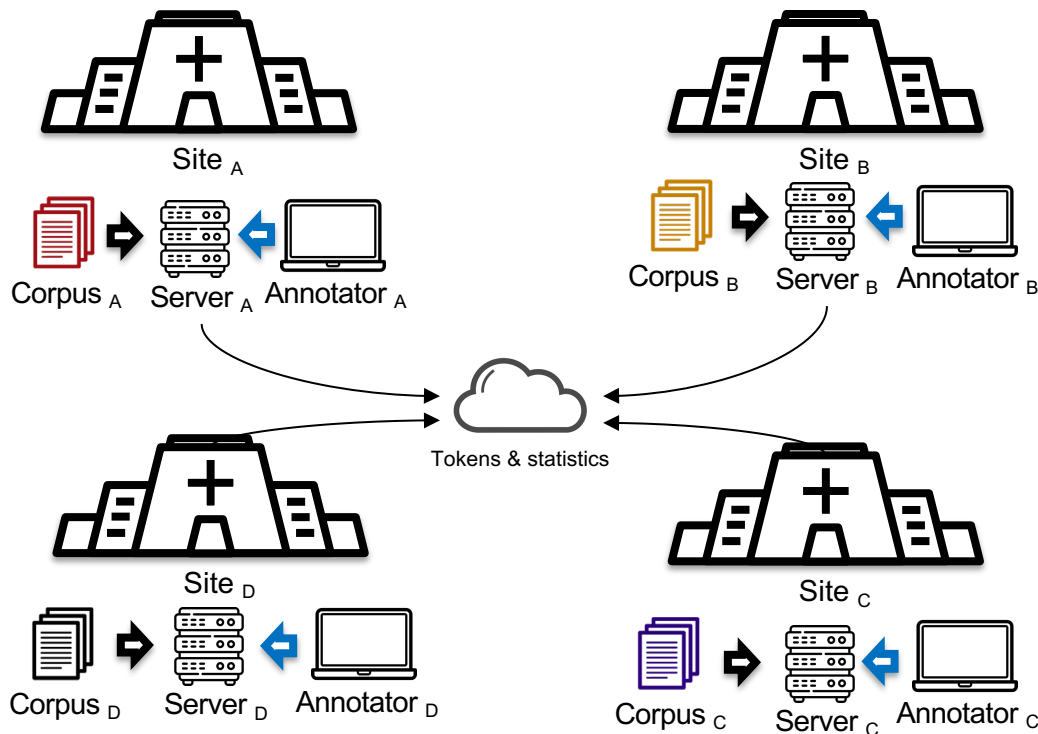
Protected health information (PHI) concerns.

Permission, de-identification, format conversion, etc.

Background - Multi-site Annotation

Own annotation servers

Each site builds its own annotation server and uploads raw corpus to its own server.



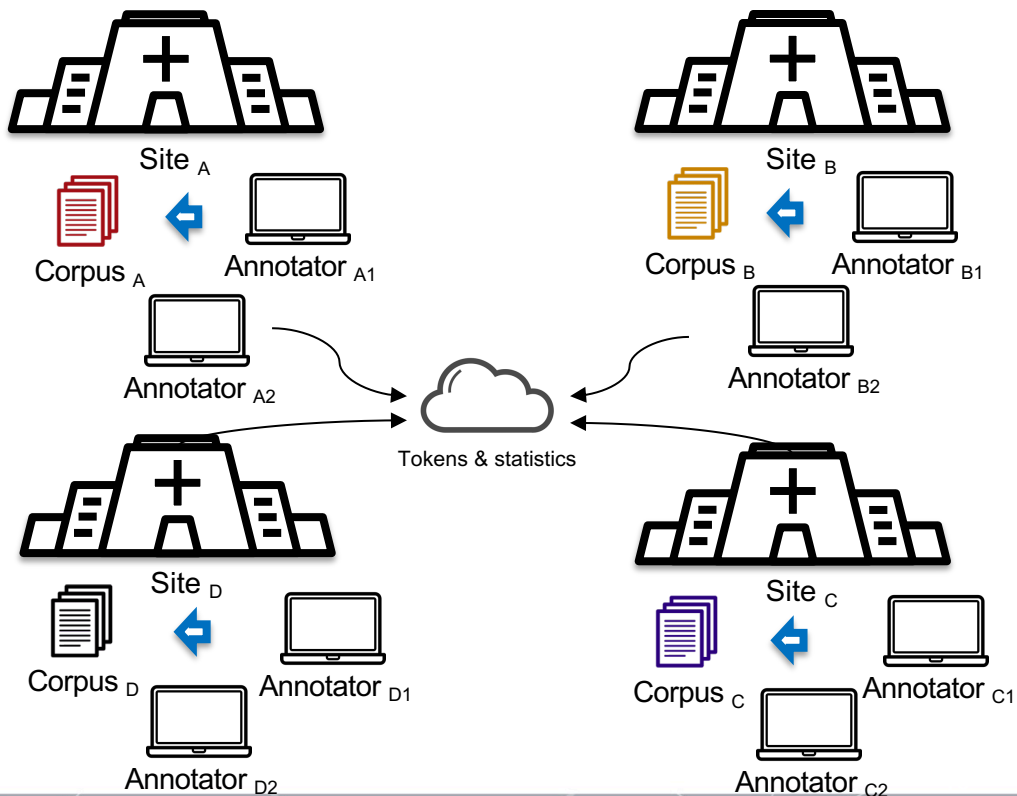
Major concerns

Internal server setup, maintenance, management, network permissions, data import and export, etc.

Background - Multi-site Annotation

Standalone tool

Each annotator installs the standalone tool on their local machine and annotates the corpus locally.

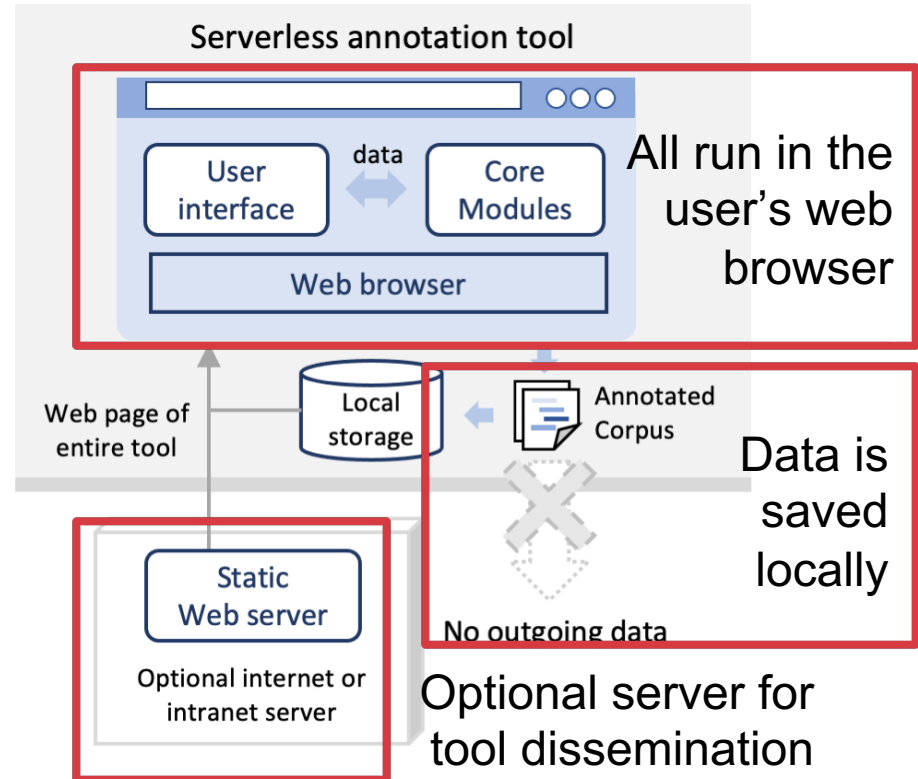


Major concerns

Permission of tool installation, runtime management, tool upgrade, environment upgrade, cross-platform compatibility, etc.

Background - Challenges of multi-site

1. Privacy concerns. Corpus with PHI should never be sent out to any external systems.
2. Concerns related to server resources, runtime setup, deployment, permissions, access controls, etc.
3. Concerns related to local runtime installation, management, upgrade, licenses, etc.



Multi-site Annotation based on MedTator

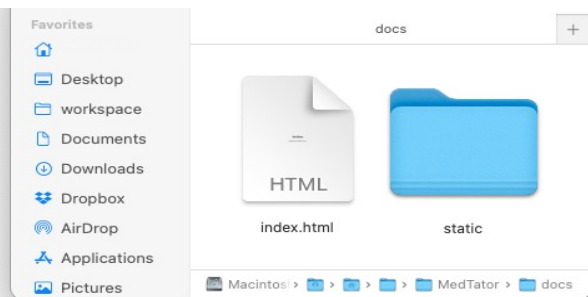
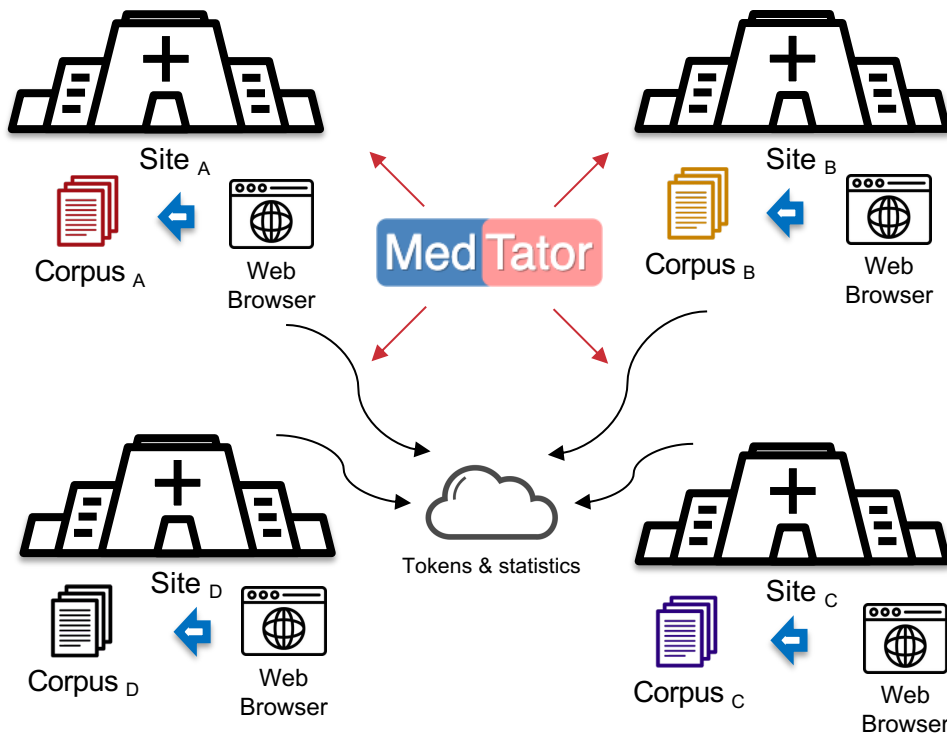
Serverless MedTator

1. No data is sent out:

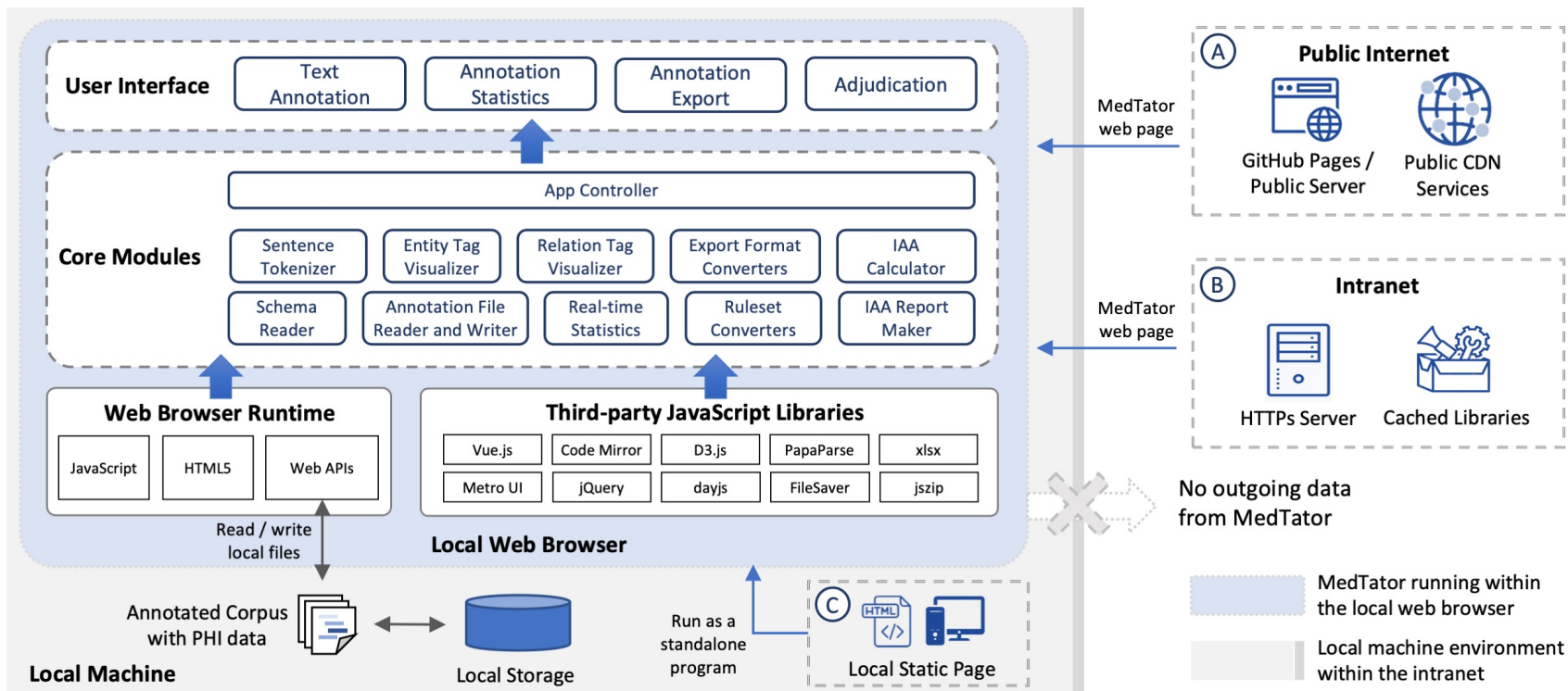
- Run locally in user's web browser
- Operate text files on locally

2. No "install":

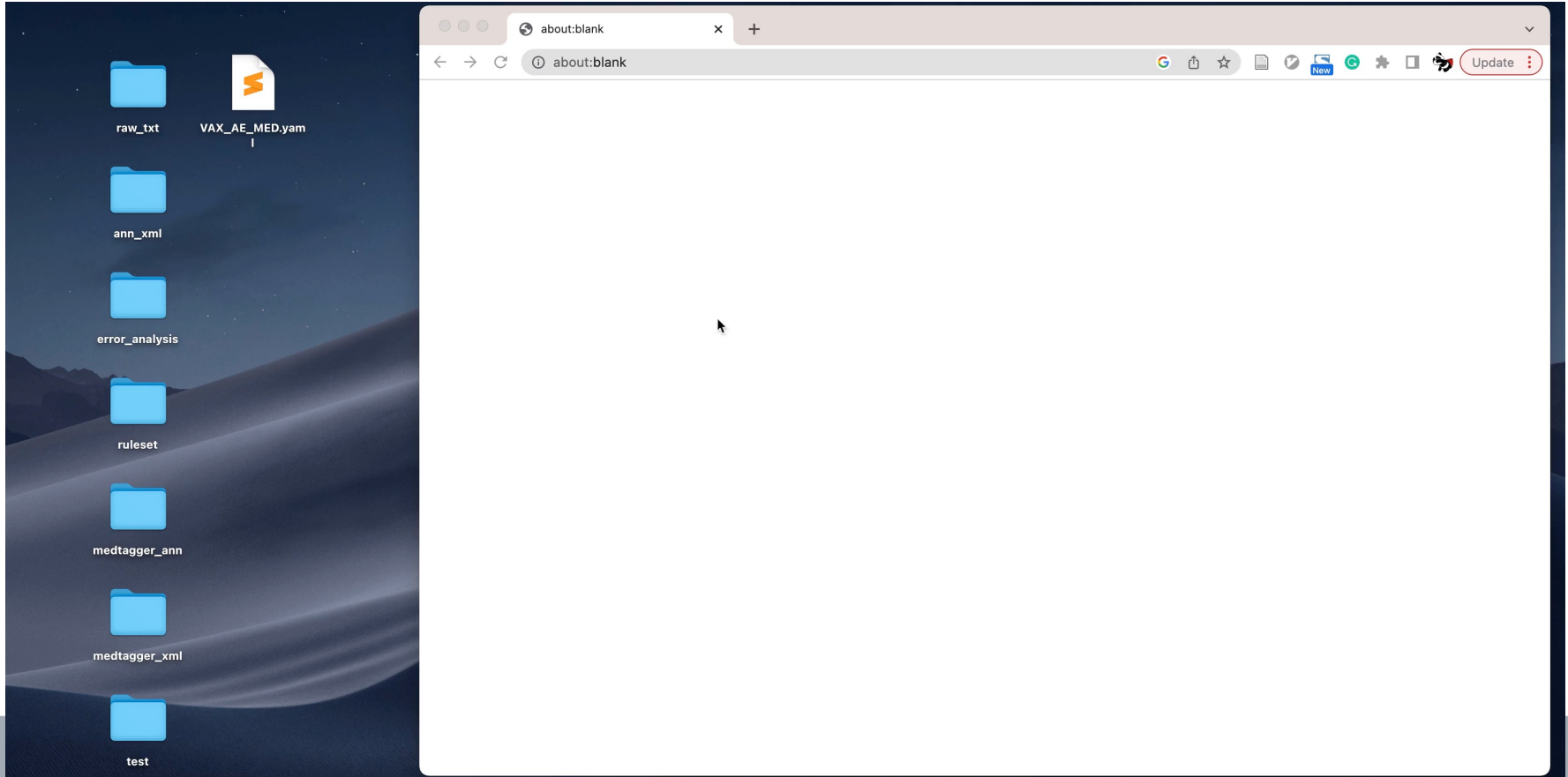
- No need to install any runtimes (e.g., Java, Python, Nginx, etc.)



MedTator: Serverless Architecture Design



MedTator - Run as a web App



MedTator - Run as a local App



MedTator - Tool Comparison

In terms of prerequisites for installation, MedTator shows a significant advantage

Tool	Source Code	Prerequisites for server installation on server
BioQRator	https://github.com/dongseop/bioqrator	MacOS or Linux; Ruby, Ruby on Rails, MySQL
brat	https://github.com/nlplab/brat	Linux; Apache 2, GeniaSS, Python 2.5
Catma	https://github.com/forTEXT/catma	Java, Maven, Jetty
Djangology	https://sourceforge.net/projects/djangology/	Python, Django, database server
ezTag	https://github.com/ncbi-nlp/ezTag	Ruby, Ruby on Rails, MySQL
FLAT	https://github.com/proycon/flat	Apache / Nginx, Python, foliadocserve, pynlpl, Django
MAT	http://mat-annotation.sourceforge.net/	Python, Java
PDFAnno	https://github.com/paperai/pdfanno	Node.js
TextAE	https://github.com/pubannotation/textae	Node.js
WAT-SL	https://github.com/webis-de/wat	Java, Docker
WebAnno	https://webanno.github.io/webanno/	Java, Apache Tomcat, MySQL

Neves M, Ševa J. An extensive review of tools for manual annotation of documents. Brief Bioinform. 2021 Jan 1;22(1):146–63.

Lessons we learned



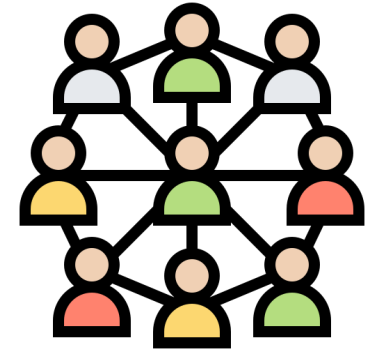
Lesson 1

Apply a user-centered design approach to meet the needs of various users



Lesson 2

Non-functional requirements are critical for adoption



Lesson 3

Community participation makes it thrive

Lesson 3: Community participation

Iterative improvement based on community feedback



OHNLP



National
COVID
Cohort
Collaborative

*Collaborative Analytics
Workstream*
**Subgroup of
Natural Language
Processing**



Oct. 30 - Nov. 3, 2021
San Diego, CA

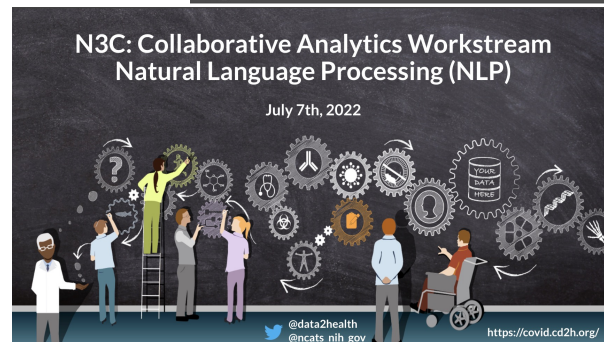


Informatics Summit
March 21-24, 2022 |
Chicago, IL



National
COVID
Cohort
Collaborative

N3C NLP: Focusing on Concept Extraction



N3C: Collaborative Analytics Workstream
Natural Language Processing (NLP)
July 7th, 2022

@data2health
@ncats_nih.gov

<https://covid.cd2h.org/>

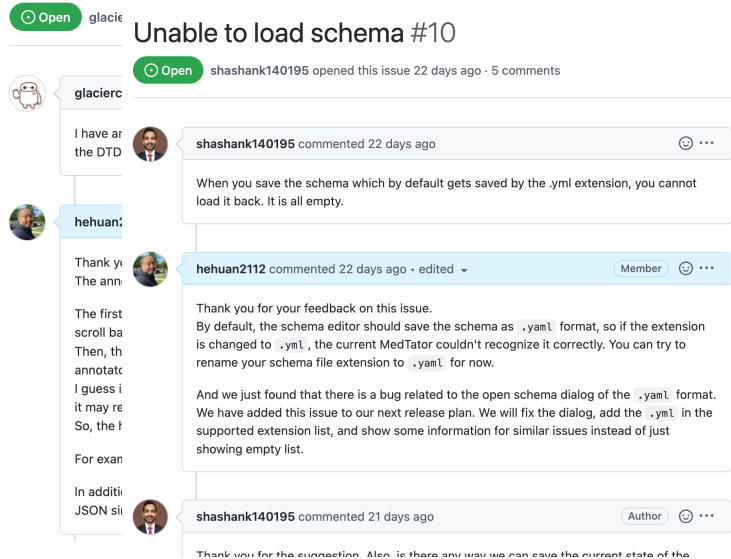


Office hours and forums

Lesson 3: Community participation

Participants from open-source community

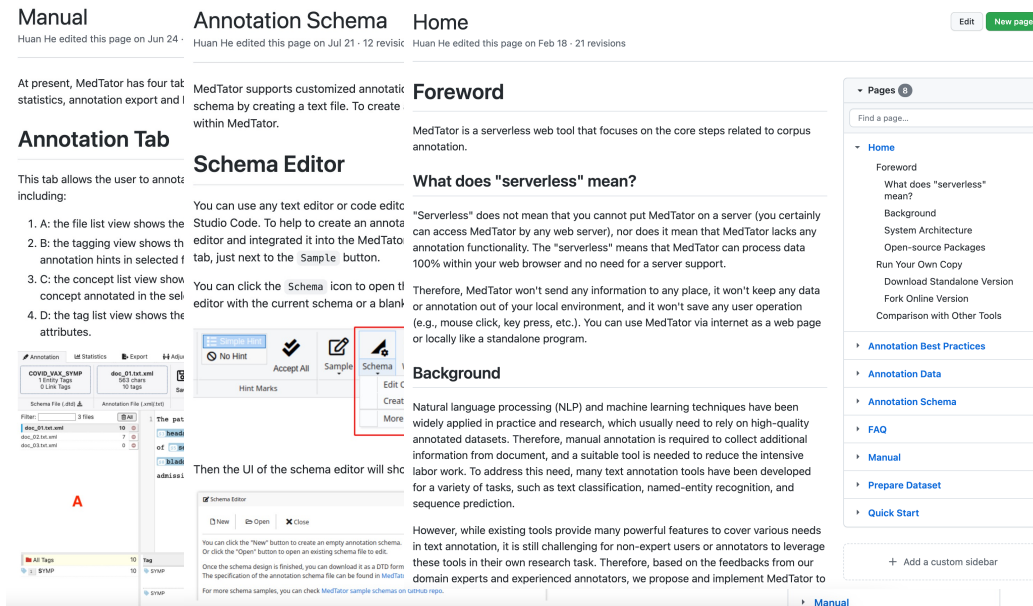
On the maximum length of "_ VALUE_TYPE_" #4



The screenshot shows a GitHub issue thread. The issue title is "Unable to load schema #10". The issue was opened by shashank140195 22 days ago. Comments include:

- glacierrc: "I have ar the DTD"
- shashank140195: "When you save the schema which by default gets saved by the .yaml extension, you cannot load it back. It is all empty."
- hehuan2112: "Thank you for your feedback on this issue. By default, the schema editor should save the schema as .yaml format, so if the extension is changed to .yml, the current MedTator couldn't recognize it correctly. You can try to rename your schema file extension to .yaml for now. And we just found that there is a bug related to the open schema dialog of the .yaml format. We have added this issue to our next release plan. We will fix the dialog, add the .yml in the supported extension list, and show some information for similar issues instead of just showing empty list."
- shashank140195: "Thank you for the suggestion. Also, is there any way we can save the current state of the"

Report bugs and issues



The screenshot shows the MedTator web application interface. The top navigation bar includes "Manual", "Annotation Schema", and "Home". The "Annotation Schema" page is active, showing a "Foreword" section and a "Background" section. The "Background" section discusses Natural Language Processing (NLP) and machine learning techniques. A red box highlights the "Schema" button in the "Schema Editor" section. Below the screenshot, there is a list of navigation links: "Annotation Best Practices", "Annotation Data", "Annotation Schema", "FAQ", "Manual", "Prepare Dataset", and "Quick Start".

Wiki for documentations (e.g., manual, annotation schema design, best practice, etc.)

Lesson 3: Community participation

Continuous updating bi-weekly for fixing issues and adding features

1.3.8 (2022-10-27)

- Added schema of .yaml extensio
- Added auto-save (experimente
- Fixed scheme editor open bug

1.3.7 (2022-09-29)

- Update the UI for IAA

1.3.6 (2022-09-15)

- Updated a data sample

1.3.5 (2022-08-18)

- Added label panel for er
- Added error sankey dia
- Added token scatter for
- Added heatmap of erro
- Added bar charts of err
- Added tag list and men
- Updated scripts for text
- Fixed IOB2/BIO empty e
- Fixed exporter window height bug
- Fixed BioC format export annotation text encoding bug
- Fixed BioC format export attribute text encoding bug

1.3.2 (2022-08-04)

- Added functions for er
- Added Math.js for stati
- Added ECharts for visu
- Added a sample datase
- Updated the UI design

1.3.5 (2022-08-18)

- Added label panel for error l
- Added error sankey diagram
- Added token scatter for erro
- Added heatmap of error dist
- Added bar charts of error st
- Added tag list and menu iter
- Updated scripts for text em
- Fixed IOB2/BIO empty expor
- Fixed exporter window heigh
- Fixed BioC format export an
- Fixed BioC format export att

1.3.0 (2022-07-21)

- Refactored file reading w
- Refactored annotation fil
- Refactored code structur
- Designed JSON/YAML fo
- Added tokenization exce
- Added pagination for larg
- Added loading dataset ar
- Added drag and drop fol
- Added 1.2.48 (2022-07-1)
- Added function fo
- Added changelog
- Added sorting for corpus token list in st
- Added tag details for corpus summary i
- Added filter options for token statistics
- Added filtered count in token summary
- Added a remote corpus sample
- Updated all sample datasets
- Updated the visual design for corpus st
- Updated the color encoding for annotat
- Updated scripts for experimental tasks
- Fixed ribbon menu resize bug

1.2.41 (2022-06-23)

- Added meta-data structure for annotatio
- Added color label to annotation file
- Added functions for updating annotator f
- Added download all tags for adjudication
- Added sorting l
- Added MedTag
- Added UI for cc
- Added folder di
- Added folder di
- Updated IAA dr
- Updated code i
- Updated wiki p
- Fixed meta cop

1.2.30 (2022-05-12)

- Added seperate file d
- Added annotator labe
- Added adjudication
- Fixed edit bug

1.2.29 (2022-04-28)

- Added the sort for I/A
- Updated iaa result bo
- Updated sample dataset
- Fixed duplicated tag in adjudicated results
- Fixed ajax request type bug
- Fixed adjudicated result download bug

1.2.27 (2022-04-14)

- Added sample schema dropdown for editor to selec
- Added sample schema files for standalone version
- Added help to schema editor with wiki
- Updated the tooltip information for annotation menu
- Updated scripts for creating sample DTD files
- Fixed the initial pr
- Fixed schema edi
- Fixed path missin

1.2.24 (2022-03-)

- Added Cohen's K
- Added dropdown
- Added confusio
- Added schema ec
- Added helper fun
- Added schema ec
- Updated annotati
- Updated dtd strin
- Updated sample i
- Updated browser
- Updated the size
- Fixed the messag
- Fixed the messag

1.2.15 (2022-03-17)

- Added sort button and menu fr
- Added dynamic sort indicator f
- Added a button for quickly dele
- Fixed concept name overflow l
- Fixed ribbon menu layout bug

1.2.13 (2022-03-03)

- Added using attribute IAA calcul
- Added menu for attribute selec
- Added dropping file in the fileli
- Updated the drop zone size
- Updated the schema and IAA s
- Updated the the JSON files of
- Fixed switch button status whe

1.2.10 (2022-02-17)

- Added a new section in the sta
- Added a new statistics report i
- Added colored cells to the statistics report i
- Added importing annotation files by dropping fold
- Added memory check in settings
- Updated the statistics summary with more items
- Updated the XML zip file name with timestamp

1.2.6 (2022-02-03)

- Added the adjudication sheet in the IAA report xlsx file
- Updated the included tags in the IAA report xlsx file
- Updated the colors in IAA report xlsx file
- Updated wiki pages

1.2.5 (2022-01-20)

- Added a setting option for showing annotated tags
- Added fixed header for the "All Tags" in concept list
- Updated the context menu position calculation
- Fixed the hover bug dismiss bug when deleting tag

1.2.3 (2022-01-06)

- Added IAA report download (as Excel XLSX format)
- Added backgrc
- Added search and clear search results in editor
- Added highlight in tag list
- Added script for standalone version export
- Updated menu item for exported rulesets
- Updated tool tips
- Updated samples for entity and relation annotation

1.1.0 (2021-11-28)

- Added a new standalone version for local usage without internet access
- Added build parameters for building standalone version
- Added local cache of all libraries for local standalone version
- Updated references configuration for standalone version
- Updated embedded sample data for standalone version
- Updated building output information
- Fixed drag & drop bug when reading local dtd file
- Fixed embedded sample JSON encoding bug

1.2.0 (2021-12-)

- Added hover b
- Added show ac
- Added downlo
- Added fixed ta
- Added downlo
- Added sentence splitting / tokenization method for fast splitting.
- Added a side bar for managing settings.
- Added entity tag locating. Users could click on the spans to locate the entity tag in the editor. The editor will jump to the line where the clicked tag is and highlight that tag.

1.0.1 (2021-10-15)

- Added sample data for AMIA 2021 workshop.

Acknowledgments

Co-authors and team members



Sunyang Fu



Liwei Wang



Andrew Wen



Sijia Liu



Sungrim Moon



Kurt Miller



Hongfang Liu



Donna Ihrke



Katelyn Cordie



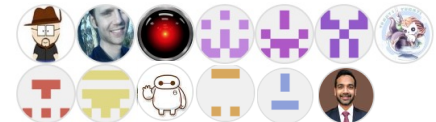
Samuel McKinven

Community Participants



National
COVID
Cohort
Collaborative

Collaborative Analytics
Workstream
Subgroup of
Natural Language
Processing



This work was supported by the National Center for Advancing Translational Sciences of the National Institutes of Health under award number U01TR002062.

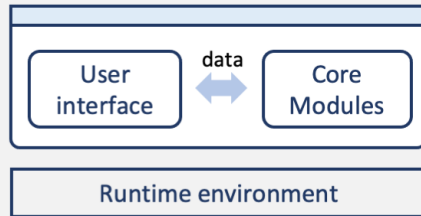
Thank you!

Email me at: He.Huan@mayo.edu



Appendix - Architecture Comparison

(A) Standalone annotation tool



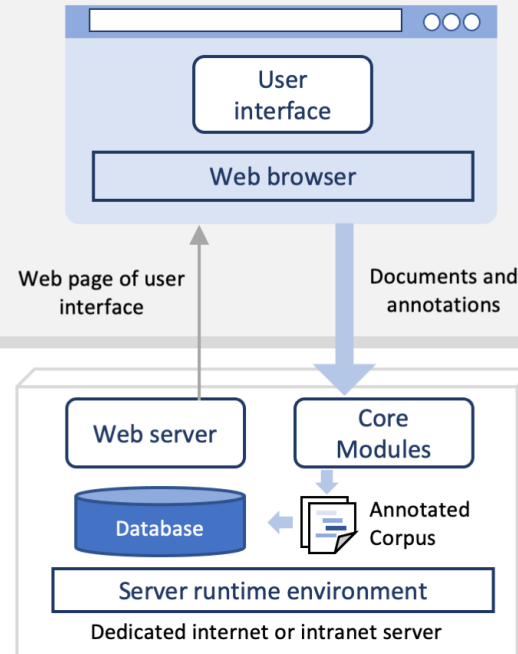
Local Machine

No outgoing data

Local machine environment within the intranet

Server environment on the internet or intranet

(B) Typical web-based annotation tool



Web page of user interface

Documents and annotations

Web server

Core Modules

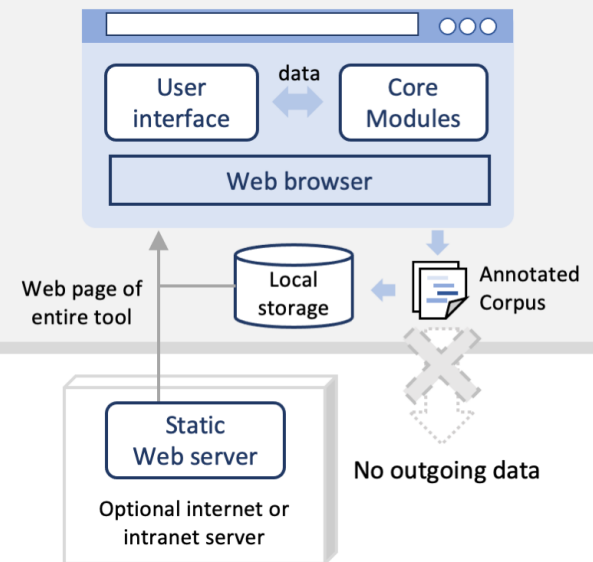
Database

Annotated Corpus

Server runtime environment

Dedicated internet or intranet server

(C) Serverless annotation tool



Web page of entire tool

No outgoing data

Static Web server

Optional internet or intranet server

Web browser file system access API is needed to support core features

mdn web docs _ References Guides 

References > Web APIs > File System Access API

RELATED TOPICS

File System Access API

Interfaces

- [FileSystemHandle](#)
- [FileSystemFileHandle](#)
- [FileSystemDirectoryHandle](#)
- [FileSystemWritableFileStream](#)

Methods

- [window.showOpenFilePicker\(\)](#)
- [window.showSaveFilePicker\(\)](#)
- [window.showDirectoryPicker\(\)](#)
- [DataTransferItem.getAsFileSystem](#)

File System Access API

Secure context: This feature is available only in [secure contexts](#) (HTTPS), in some or all [supporting browsers](#).

The File System Access API allows read, write and file management capabilities.

Concepts and Usage

This API allows interaction with files on a user's local device, or on a user-accessible network file system. Core functionality of this API includes reading files, writing or saving files, and access to directory structure.

Most of the interaction with files and directories is accomplished through handles.

A parent [FileSystemHandle](#) class helps define two child classes:

[FileSystemFileHandle](#) and [FileSystemDirectoryHandle](#), for files and directories respectively.

These handles represent the file or directory on the user's system. You must first gain access to them by showing the user a file or directory picker. The methods which allow this are [window.showOpenFilePicker](#) and [window.showDirectoryPicker](#). Once these are called, the file picker presents itself and the user selects either a file or directory. Once this happens successfully, a

IN THIS ARTICLE

Concepts and Usage


- Interfaces
- Examples
- Specifications
- Browser compatibility
- See also

Browser compatibility

[Report problems with this compatibility data on GitHub](#)

	Desktop					Mobile						
	Chrome	Edge	Firefox	Internet Explorer	Opera	Safari	WebView Android	Chrome Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet
FileSystemHandle 	 86	 86	 No	 No	 72	 15.2	 No	 86	 No	 No	 15.2	 14.0
isSameEntry 	 86	 86	 No	 No	 72	 15.2	 No	 86	 No	 No	 15.2	 14.0
kind 	 86	 86	 No	 No	 72	 15.2	 No	 86	 No	 No	 15.2	 14.0
name 	 86	 86	 No	 No	 72	 15.2	 No	 86	 No	 No	 15.2	 14.0
queryPermission  	 86	 86	 No	 No	 72	 No	 No	 86	 No	 No	 No	 14.0
requestPermission  	 86	 86	 No	 No	 72	 No	 No	 86	 No	 No	 No	 14.0

 Full support  No support  Experimental. Expect behavior to change in the future.

 Non-standard. Check cross-browser support before using.