UNT UNIVERSITY OF NORTH TEXAS

Natural Language Processing TEXT ANNOTATOR

My Coder Academia

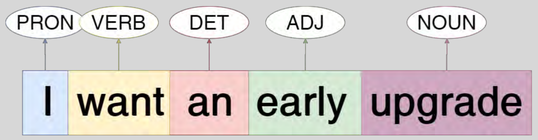
James Arnold, Jon Kleehammer, Lupe Rivera, Phillip Merritt

UNT UNIVERSITY OF NORTH TEXAS

Project Overview

Natural language processing is a rapidly expanding field as companies work to develop ever more sophisticated audio and text-based interfaces for their AI assistant and customer service systems. In order to train these new systems well annotated datasets are needed, but current NLP annotation software (like GATE) have limitations in portability and lack many quality of life features for annotators.

Annotation Example:



Creating a completely web-based NLP annotator app will make it possible to annotate from anywhere in the world on any system with a basic web browser. By working closely with real experienced NLP annotators to include sorely sought-after annotation tools, the productivity of annotators could also be improved. This web-based system will also allow NLP annotators to access/manage the annotated datasets they create from anywhere they choose.

Features

- File Management
  - Upload, edit, and save files
- Pre-processing
  - Can invoke tokenizers, sentence splitters, POS taggers, NER tagger, Dependency tagger
- Document Annotation
  - Can load documents and use a custom editor to annotate it.
- Dataset Analysis
  - Can visualize and compare annotated datasets between older versions and annotations by different users.
- Dataset Management
  - Can create and remove datasets as well as adding documents, removing documents, and moving documents between datasets.
  - Can share access to datasets via group membership.

Design

Our sponsor requested a webapp instead of a desktop app because it's easy is to start using and more stable, especially with updates rolling out as we make changes like bug fixes and content updates to the product.

Current tools that exist are extremely clunky and so we made improvements to the user interface to allow for it to be easy to get started, quick to annotate, and more enjoyable to do.

Testing

For testing, the approach was dividing all tests into unit, integration, and functional testing. Unit tests were written by the developers before feature development to adhere to Test Driven Development. Integration testing was used for testing modules and were written by associated developers. Functional tests were written as a team to ensure quality. Testing was also included in the Continuous Integration pipelines to assure quality as features were rapidly introduced to the codebase.

Technologies Used

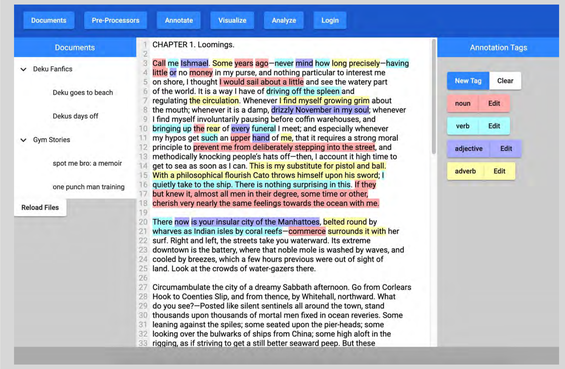
Frontend

For the web app's frontend we chose to use Angular for its quick implementation, modularity, and scalability over using pure HTML/CSS/JS.

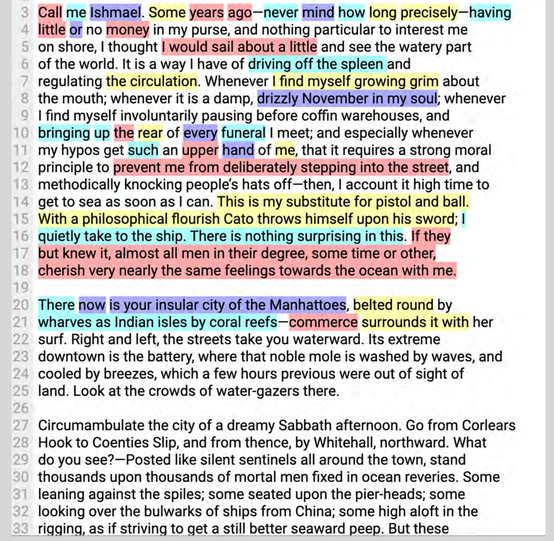
Backend

For the backend we chose to use SQLalchemy and PostgreSQL which is served to the frontend through a Flask REST api. We chose these since they provide a uniform python environment for the server and a SQL based database for file storage and management.

Screenshots



The main view the user will be using to annotate text. Showing an example document, file tree, annotation buttons, and menu bar with other tools in the product



A zoomed in screenshot of the annotations. Showing how the user can annotate a single word, a single character, or as many words/sentences/paragraphs they want.