



Human-In-The-Loop Generative AI Lab Training Session

Dia Trambitas, Head of Product, John Snow Labs

Outline

1. LLM evaluation projects

2. De-identification

3. HITL medical document annotation

4. HCC coding

- <https://github.com/databricks-industry-solutions/medicare-risk-adjustment>

- <https://app.swapcard.com/event/healthcare-nlp-summit-2025/planning/UGxhbm5pbmdfMjU3Njg5Mw==>

5. PDF annotation

DEMO - resources

<https://genailab.demo.johnsnowlabs.dev/>

User/pass → shared via email (also check the spam folder)

Generative AI Lab Instance will be available for 2 weeks

Projects available for import here.

- Download zip archive and remove .txt extension from filename
- Import into Generative AI Lab

Documentation available here.

Let's get started!

Why is Data Annotation Crucial for NLP?

- Transforming unstructured data into **structured** format.
 - Providing "ground truth" for model training and validation.
 - Enhancing the performance of NLP models.
 - Ensuring models understand context, sentiment, entities, etc.
 - A way of transferring annotators knowledge and experience to models.
-
- **The resulting models can only be as good as the training data.**

Role of Annotation Tools

- Facilitate **systematic and organized annotation**.
- Allow for **collaborative annotation** projects.
- Ensure standardization and **consistency**.
- Offer **quality control** and **review** functionalities.
- Provide **automation** and **AI-assisted features**.

Key Requirements for Annotation Tools

- **High productivity** – keep annotators "in the zone"
 - ✓ Intuitive, easy to learn
 - ✓ Keyboard shortcuts
 - ✓ How many clicks? How many eye movements?
 - ✓ Automatic transition between documents
- **Teamwork**
 - ✓ Support for projects, teams, and role-based permissions
 - ✓ Customizable workflows: validation, review, model training, etc.
 - ✓ Collaboration: compare versions, share guidelines
 - ✓ Versioning & audit trails
- **Improve over time**
 - ✓ New features and constant improvements
 - ✓ Learn from annotators actions
 - ✓ High-accuracy, configurable deep learning backend
 - ✓ "Closed-loop" automation: pre-train, annotate, retrain, measure

Project Creation Flow



Outline

1. Introduction to Text Annotation
- 2. Choosing your Models, Rules, and Designing Prompts**
3. Annotation Projects Setup in Gen AI Lab
4. Pre-Annotation
5. Annotation Guidelines
6. Model Training
7. Model Testing
8. Generating Augmented & Synthetic Tasks

The Hub Of Resources

Integration with Models Hub

- ✓ Check benchmarking information
- ✓ Download any resource with one click

Models

- ✓ Private repository of models
- ✓ Trained with Gen AI Lab, Manually uploaded, Downloaded from Models Hub

Rules

- ✓ Private repository of rules
- ✓ Rules editing interface

Prompts

- ✓ Private repository of prompts
- ✓ Prompt editing interface

Choosing a Model

The screenshot shows the John Snow LABS HUB interface. On the left, a sidebar menu includes 'Projects', 'Hub' (selected), 'Local Models', 'Test Suites', 'Embeddings', 'Rules', 'Prompts', and 'Pipeline'. Below that are 'Settings', 'Help?', and a user profile for 'jessica'. The main area is titled 'HUB Online Models' and has a search bar with 'Medical'. It displays a grid of six model cards:

- Extract medical devices and clinical department mentions (Voice of the Patient)**
Model Name: ner_vop_clinical_dept_wip
Edition: Healthcare NLP 4.4.2
Task: Named Entity Recognition
May 19, 2023 English
- Extract Pharmacological Entities from Spanish Medical Texts**
Model Name: ner_pharmacology
Edition: Healthcare NLP 4.0.2
Task: Named Entity Recognition
August 13, 2022 Spanish
- Extract Negation and Uncertainty Entities from Spanish Medical Texts**
Model Name: bert_token_classifier_negation_uncertainty
Edition: Healthcare NLP 4.0.2
Task: Named Entity Recognition
August 11, 2022 Spanish
- Extract Pharmacological Entities From Spanish Medical Texts...**
Model Name: bert_token_classifier_pharmacology
- Detect Adverse Drug Events (MedicalBertForTokenClassification)**
Model Name: bert_token_classifier_ade_tweet_binary
- Detect Disease Mentions (MedicalBertForTokenClassification)**
Model Name: bert_token_classifier_disease_mentions_tweet

Filtering options at the top right include 'Edition 1', 'Named Entity Recognition', and 'All Languages'. At the bottom, it says 'Showing 1-15 of 27 Models' with page navigation buttons.

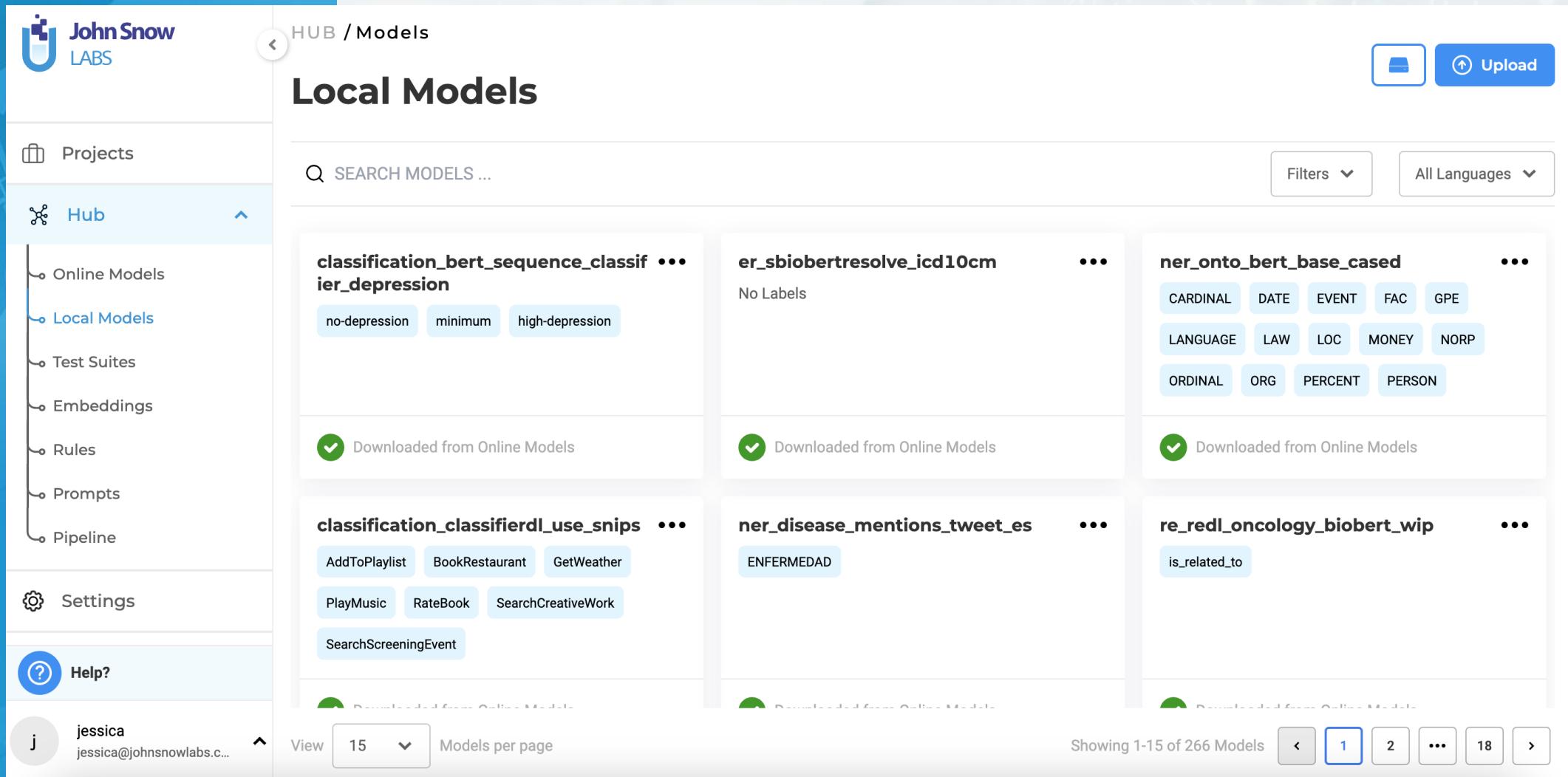
Reading Model Benchmarks

Benchmarking

name	f1	fn	fp	precision	recall	total	tp
AdmissionDischarge	0.91	5	1	0.97	0.85	34	29
ClinicalDept	0.89	34	41	0.88	0.90	326	292
MedicalDevice	0.75	88	72	0.77	0.73	332	244
macro_avg	0.85	127	114	0.87	0.83	692	565
micro_avg	0.82	127	114	0.83	0.82	692	565

View | 15 | Models per page | Close | Showing 1-15 of 27 Models

Your Downloaded Models



The screenshot shows the John Snow LABS HUB / Models interface. The left sidebar includes sections for Projects, Hub (with Local Models selected), Test Suites, Embeddings, Rules, Prompts, Pipeline, Settings, Help?, and a user profile for jessica. The main area displays a grid of local models:

- classification_bert_sequence_classif ...** (with three dots) - Labels: no-depression, minimum, high-depression. Status: Downloaded from Online Models.
- er_sbiorbertresolve_icd10cm** - Labels: No Labels. Status: Downloaded from Online Models.
- ner_onto_bert_base_cased** - Labels: CARDINAL, DATE, EVENT, FAC, GPE, LANGUAGE, LAW, LOC, MONEY, NORP, ORDINAL, ORG, PERCENT, PERSON. Status: Downloaded from Online Models.
- classification_classifierdl_use_snips** - Labels: AddToPlaylist, BookRestaurant, GetWeather, PlayMusic, RateBook, SearchCreativeWork, SearchScreeningEvent. Status: Downloaded from Online Models.
- ner_disease_mentions_tweet_es** - Labels: ENFERMEDAD. Status: Downloaded from Online Models.
- re_redl_oncology_biobert_wip** - Labels: is_related_to. Status: Downloaded from Online Models.

At the bottom, there are navigation controls for View (15), Models per page, and a pagination bar showing pages 1 through 18.

Rules - Introduction

Rules in NER

- **Definition:** Predefined conditions/patterns for entity identification.
- **Implementation:** Regular expressions or logical conditions.
- **Function:** Alternative that complements NER model outputs when NER model entities are not available
- **Example:** Rule requires “Dr. Smith” to be followed by “MD” to validate entity.

Rules - Regex

Define a new Rule

Entity Name *	Rule Scope ⓘ
quantity	Sentence
Match Scope ⓘ	
Sub-Token	
Set Type ⓘ	Complete Match Regex ⓘ
<input checked="" type="radio"/> Regex	<input type="radio"/> Dictionary
<input type="checkbox"/> <code>\d+[V]\?\d*</code>	
Context Length *	Exception distance ⓘ
50	15
Prefix Keyword ⓘ	
Enter prefix keywords	
Suffix Keyword ⓘ	
<input type="button" value="tablespoon x"/> <input type="button" value="tablespoons x"/> <input type="button" value="teaspoon x"/> <input type="button" value="teaspoons x"/> <input type="button" value="kg x"/> <input type="button" value="pounds x"/> <input type="button" value="gr x"/> <input type="button" value="ml x"/> <input type="button" value="cup x"/> <input type="button" value="cups x"/> Enter suffix keywords	
Context Exception Keyword ⓘ	
<input type="button" value="minutes x"/> <input type="button" value="degree x"/> <input type="button" value="degrees x"/> Enter context exception	
<input type="button" value="Save Rule"/>	

- Users can define a regex (regular expression) that will be used to label all possible hit chunks and label them as the target entity.
- For example, for labeling **height** entity the following regex can be used `[0-7]'((0?[0-9])|(1(0|1)))"`.
- All hits found in the task's text content that match the regex are pre-annotated as *height*.

Rules - Dictionary

Define a new Rule

Entity Name * Rule Scope

Match Scope

Set Type Regex Dictionary Case Sensitive Complete Match Regex

Drag and drop your CSV file or click for import

spices.csv

Context Length * Exception distance

Prefix Keyword

Suffix Keyword

Context Exception Keyword

Save Rule

- Users can create and upload a CSV dictionary of keywords that cover the list of chunks that should be annotated as a target entity.
- For example, for the label **female**, all occurrences of strings *woman*, *lady*, and *girl* within the text content of a given task will be pre-annotated as *female*.

Prompts - Introduction

A **zero-shot prompt** is an instruction given to a model that asks it to perform a task without prior examples or specific training. The model generates a response based solely on its understanding of the prompt and general knowledge, rather than on learned patterns from labeled data.

Key Characteristics:

- **No Prior Examples:** The model isn't provided with demonstrations of the task.
- **Generalization:** It relies on the model's ability to generalize from its training data.
- **Flexibility:** Zero-shot prompts allow models to tackle various tasks without retraining for each case.

Example Use Cases:

- **Text Classification:** Asking a model to label a text as "positive" or "negative" without any labeled examples.

Creating Prompts

The screenshot shows the John Snow LABS HUB / Prompts interface. The left sidebar includes sections for Projects, Hub (selected), and Settings. The Hub section contains links for NLP Models HUB, Models, Embeddings, Rules, and Prompts. The main content area displays a single prompt card titled "PROBLEM". The card shows "NER | Created by: admin on just now" and "Reference LM Model: HEALTHCARE". It lists two numbered questions: "1 What is the disease?" and "2 What is his symptom?". At the bottom of the page, there are navigation controls for "View" (set to 15), "Prompts per page", and a footer message "Showing 1-1 of 1 Prompts" with page numbers 1 and 2.

Reuse Resources in your projects

1 Content Type ————— 2 Define what to annotate ————— 3 Reuse Resources ————— 4 Customize Labels

Model **Rules** **Prompts**

USE LABELS FROM AVAILABLE MODELS

Select any class/label from an existing model and add it to your project

Select All

SEARCH LABELS...

ASSERTION_JSL_AUGMENTED (8) ▾

CLASSIFICATION_CLASSIFIERDL_USE_EMOTION (4) ▾

NER_CLINICAL (3) ▾

TREATMENT PROBLEM TEST

NER_DL (4) ▾

NER_ISI (70) ▾

+ ADD TO PROJECT CONFIGURATION

PREVIEW WINDOW

Preview your taxonomy and annotate a sample task to see the obtained output.

Age Gender Employment Date Medical_Problem Modifier Vital_Sign
Vital_Sign_Result Drug Present Absent Possible Family VS_Finding
Vital_Signs_Header Drug_BrandName PatientName Email PhoneNumber Name

The patient is a pleasant 17-year-old gentleman who was playing basketball today in gym. Two hours prior to presentation, he started to fall and someone stepped on his ankle and kind of twisted his right ankle and he cannot bear weight on it now. It hurts to move or bear weight. No other injuries noted. He does not think he has had injuries to his ankle in the past. He was given adderall and accutane.

Gender

Female Male Unknown

Kind of Text

Clusters

1. Up-to-date information on the deployed servers
2. Computation resource management
3. Access deployed playground

Settings / Clusters

Clusters

Auto Refresh

License Info

1 floating license is available. Each floating license allows one healthcare, finance, or legal training and/or pre-annotation job according to its scope. There are no restrictions for running open-source parallel jobs as long as your system has enough resources and has been configured to allow multiple parallel jobs during setup.

ID	Server Name	License Used/Scope	Usage	Status	Deployed By	Deployed At	Action
3	playground	Floating License Legal:Inference, Legal:Training, Finance:Inference, Finance:Training, Ocr:Inference, Ocr:Training, Healthcare:Inference, Healthcare:Training	Playground	Busy	dia	1 minute ago	

Outline

1. Introduction to Text Annotation
2. Choosing your Models, Rules, and Designing Prompts
- 3. Annotation Projects Setup in Gen AI Lab**
4. Pre-Annotation
5. Annotation Guidelines
6. Model Training
7. Model Testing
8. Generating Augmented & Synthetic Tasks

Project Setup

1. Project creation steps for Classification project
 - Configuration, choices, hotkeys
2. Project creation steps for NER project
 - Configuration, choices, hotkeys
3. Task import
4. Task assignment
5. Workflow details

Project Creation/NER

- 1.Login into Generative AI Lab
- 2.Once logged in,
in the projects page, select "New Project"



3.In the Project Description page, choose a name for your project and click Next. The name must be unique, you will get a warning in case there is already a project with the name you choose.

4.In the Add team Members page, use the "Search team Members" box to lookup usernames you want added to the project

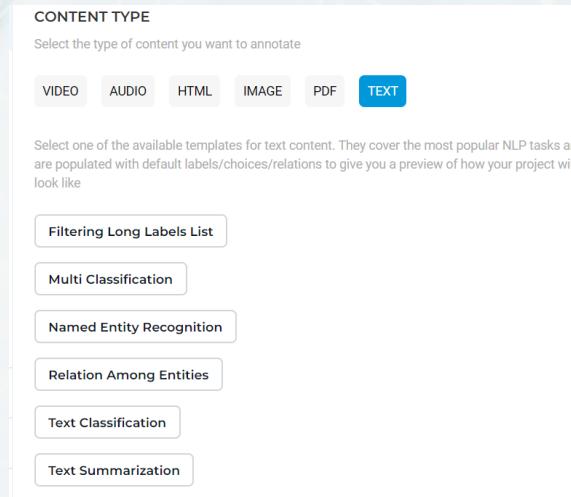
Note: the users must be created and added to the system prior to assigning them to a project

Click Next. This takes you to the Configuration page. Configuring your project is a guided experience and has 4 steps displayed in the top horizontal bar:

Project Creation/NER

For the training session today, we will create 2 different project types : Text Classification and Named Entity Recognition

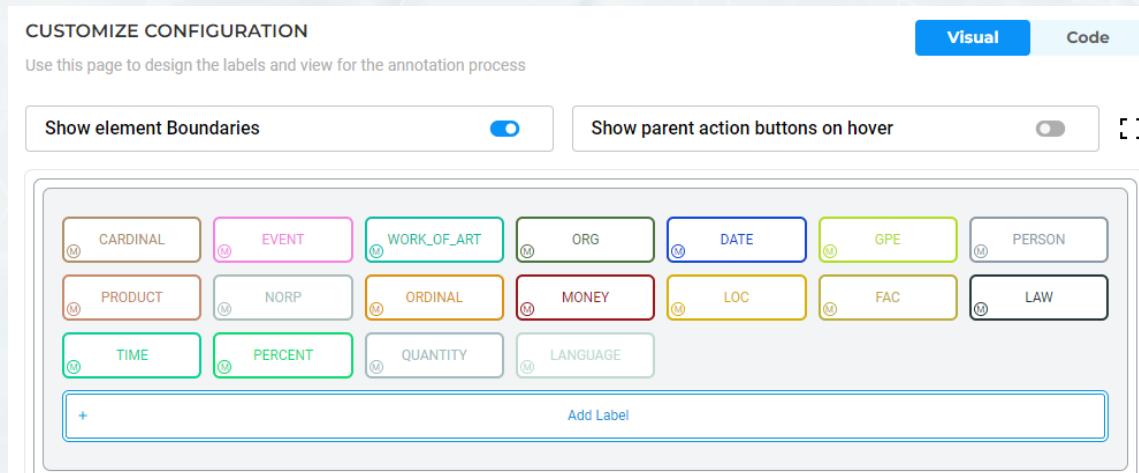
5. Select Named Entity Recognition and click Next.



6. In Step 2 and 3 leave the default settings and click Next.

Project Creation/NER-cont

7. In Step 4 "Customize Labels", by default, few labels are already populated for you.



Add yours then remove existing ones then click the Save Config button. This takes us to the Import page where tasks will be imported into the project.

Project Creation/NER-cont

- Tip: Remove unwanted labels by editing the xml config file. You can do this by switching modes between Visual to Code

CUSTOMIZE CONFIGURATION

Use this page to design the labels and view for the annotation process

Show element Boundaries

Show parent action buttons on hover

CARDINAL	EVENT	WORK_OF_ART	ORG	DATE	GPE	PERSON
PRODUCT	NORP	ORDINAL	MONEY	LOC	FAC	LAW
TIME	PERCENT	QUANTITY	LANGUAGE			

+ Add Label

```
xt">
<Label value="Age" background="#ff3333"/>
<Label value="Gender" background="#4c753d"/>
<Label value="Employment" background="#0036f8"/>
<Label value="Date" background="#ffb800"/>
<Label value="Disease Syndrome Disorder" background="#ba2b2b"/>
<Label value="Modifier" background="#b4b731"/>
<Label value="VS Finding" background="#fb16c8"/>
<Label value="Test Result" background="#13fa7c"/>
<Label value="Drug Ingredient" background="#7a6109"/>
<Label value="Present" background="#0b7a09" assertion="true"/>
<Label value="Absent" background="#0b7a09" assertion="true"/>
<Label value="Possible" background="#0b7a09" assertion="true"/>
<Label value="Family" background="#0b7a09" assertion="true"/>
</Labels>
<Text name="text" value="$text"/>
</View>
```

Project Creation/NER-cont

- Creating hot keys.

Hot keys are keyboard shortcuts you can set for increasing productivity in the labeling process.
To setup hot keys, switch to Code mode and add hotkey tag:

```
1 <View orientation="horizontal">
2   <Labels name="label" toName="text">
3     <Label value="Age" background="#ff3333" hotkey="A"/>
4     <Label value="Gender" background="#4c753d" hotkey="G"/>
5     <Label value="Employment" background="#0036f8" hotkey="E"/>
6     <Label value="Date" background="#ffb800"/>
7     <Label value="Disease Syndrome Disorder" background="#ba2b2b"/>
8     <Label value="Modifier" background="#b4b731"/>
9     <Label value="VS Finding" background="#fb16c8"/>
10    <Label value="Test Result" background="#13fa7c"/>
11    <Label value="Drug Ingredient" background="#7a6109"/>
12    <Label value="Present" background="#0b7a09" assertion="true"/>
13    <Label value="Absent" background="#0b7a09" assertion="true"/>
14    <Label value="Possible" background="#0b7a09" assertion="true"/>
15    <Label value="Family" background="#0b7a09" assertion="true"/>
16  </Labels>
17  <Text name="text" value="$text"/>
18 </View>
```

Project Creation/NER-cont

➤ Assertion Labels

Assertion labels are tags with contextual information that are assigned to NER extractions. The typical example of assertion status detection is negation identification: in the sentence “the patient has no history of diabetes”, the word “diabetes” is annotated using both an NER label (Disease) and an assertion label (Absent).

These labels are defined using the configuration file as a “regular” label + assertion=“true”

```
2   <Label value="Present" background="#0b7a09" assertion="true"/>
3   <Label value="Absent" background="#0b7a09" assertion="true"/>
4   <Label value="Possible" background="#0b7a09" assertion="true"/>
5   <Label value="Family" background="#0b7a09" assertion="true"/>
```

Using the Annotation Lab, you can annotate assertion in a very simple way. First, choose a NER label and select the part of the text that you want to extract, and then choose an assertion label and select that same part of the document.

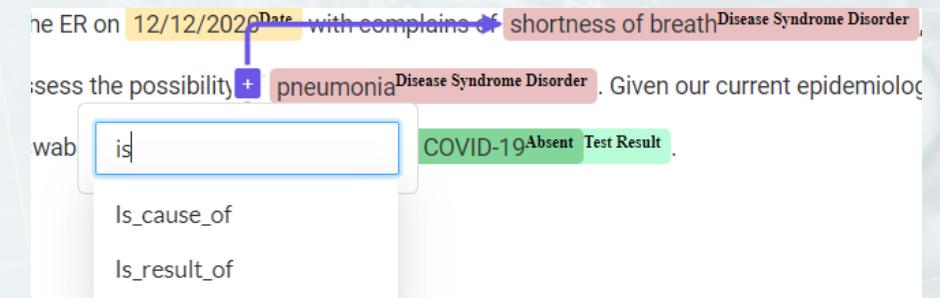
Read more about assertion labels here - <https://www.johnsnowlabs.com/tips-and-tricks-on-how-to-annotate-assertion-in-clinical-texts/>

Project Creation/NER-cont

➤ Creating relations

Relations can be defined using the configuration file by adding a Relation section as depicted in the screenshot below.

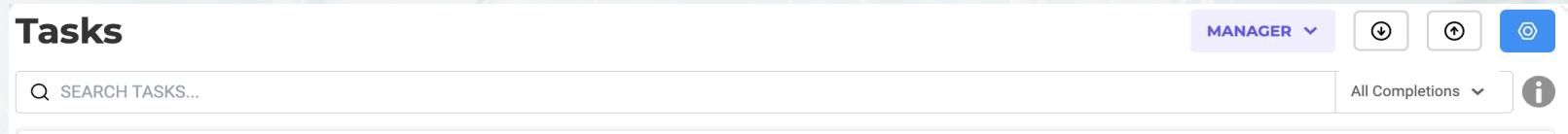
```
5   </Labels>
6
7   <Relations>
8       <Relation value="Is_cause_of"/>
9       <Relation value="Is_result_of"/>
10  </Relations>
```



When creating the relation, the name defined in the configuration file is displayed in the UI.

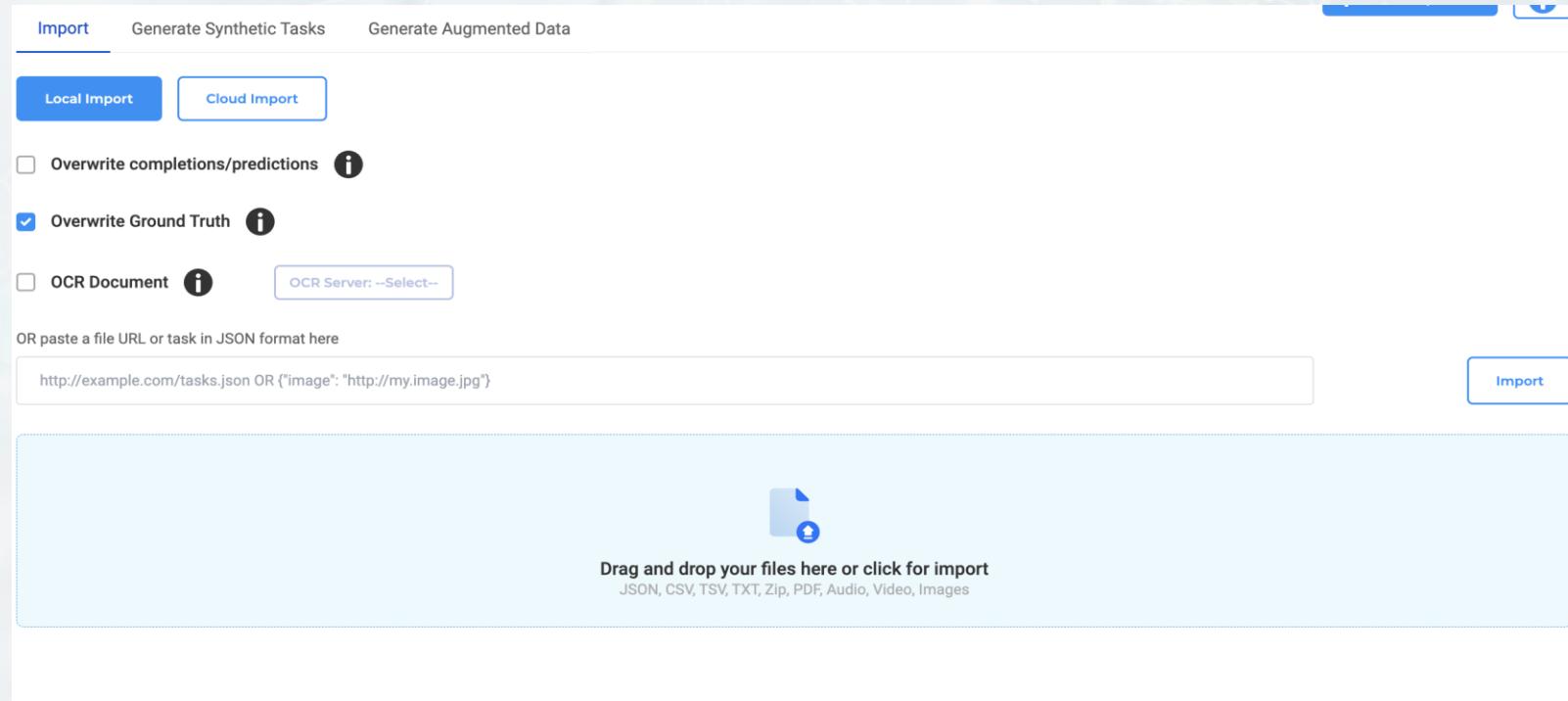
Task Import and Assignment

When project setup is done, you will land on the Task page. The page is empty, and to import your tasks, click on the Import button – top right.



Task Import and Assignment

There are multiple ways for importing tasks,: using a zip archive with your text files, importing tasks using a csv file, using json file, etc



The screenshot shows the 'Import' tab of the John Snow LABS interface. It includes options for 'Local Import' and 'Cloud Import'. Under 'Local Import', there are checkboxes for 'Overwrite completions/predictions' (unchecked), 'Overwrite Ground Truth' (checked), and 'OCR Document' (unchecked). An 'OCR Server' dropdown is set to '--Select--'. Below these, there's a text input field containing a URL: 'http://example.com/tasks.json OR {"image": "http://my.image.jpg"}'. To the right of the input field is a blue 'Import' button. At the bottom, there's a large dashed rectangular area with a blue file icon and the text 'Drag and drop your files here or click for import'. Below this, a list of supported file types is shown: JSON, CSV, TSV, TXT, Zip, PDF, Audio, Video, Images.

Exercise: Project Creation/Classification

1. Login into Generative AI Lab
2. Once logged in,
in the projects page, select "New Project"
3. In the Project Description page, choose a name for your project and click Next.
The name must be unique, you will get a warning in case there is already a project with the name you choose.
4. In the Add team Members page, use the "Search team Members" box to lookup usernames you want added to the project

Projects

Import Project

+ New Project

Note: the users must be created and added to the system prior to assigning them to a project

Click Next. This takes you to the Configuration page. Configuring your project is a guided experience and has 4 steps displayed in the top horizontal bar:

1 Content Type ————— 2 Define what to annotate ————— 3 Reuse Resources ————— 4 Customize Labels

Select Text Classification and click Next.

Exercise: Project Creation/Classification

CONTENT TYPE

Select the type of content you want to annotate

VIDEO AUDIO HTML IMAGE PDF **TEXT**

Select one of the available templates for text content. They cover you a preview of how your project will look like

Filtering Long Labels List
Multi Classification
Named Entity Recognition
Named Entity Recognition with Assertion
Relation Among Entities
Text Classification
Text Summarization

5. Select Text Classification and click Next.

6. In Step 2 and 3 leave the default settings and click Next.

Exercise: Project Creation/Classification-cont

7. In Step 4 "Customize Labels", by default, few choices are already populated for you.

Select Emotion

- surprise
- sadness
- fear
- joy

+ Add Choice

Add yours then remove existing ones then click the Save Config button.
We will be classifying tweets in this project so the choices we add are "positive", "negative" and "neutral"

Select Emotion

- positive
- negative
- neutral

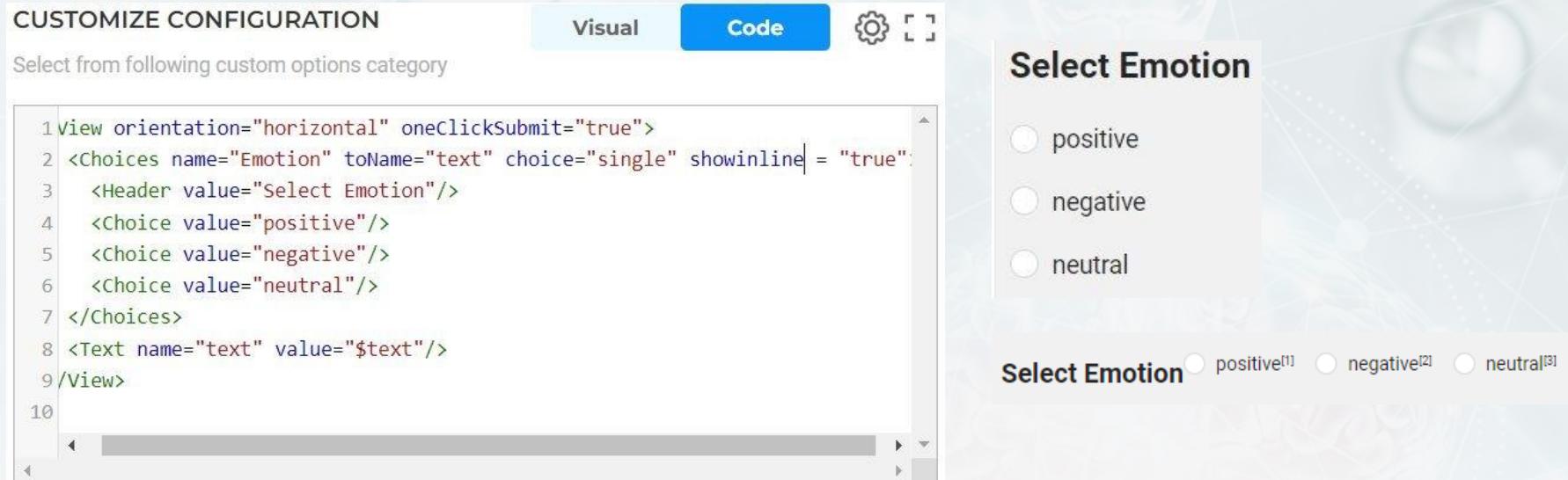
+ Add Choice

This takes us to the Import page where tasks will be imported into the project.

Exercise: Project Creation/Classification-cont

➤ Customize how you view your choices

By default, the choices are listed vertically. If you prefer to list them horizontally, switch to Code view and add “showInLine” tag set to True



The screenshot shows a "CUSTOMIZE CONFIGURATION" interface. At the top, there are tabs for "Visual" and "Code", with "Code" being the active tab. Below the tabs is a section titled "Select from following custom options category". On the left, a code editor displays the following XML code:

```
1<View orientation="horizontal" oneClickSubmit="true">
2  <Choices name="Emotion" toName="text" choice="single" showinline = "true">
3    <Header value="Select Emotion"/>
4    <Choice value="positive"/>
5    <Choice value="negative"/>
6    <Choice value="neutral"/>
7  </Choices>
8  <Text name="text" value="$text"/>
9</View>
10
```

To the right of the code editor is a preview area titled "Select Emotion" which displays three radio buttons labeled "positive", "negative", and "neutral". Below the preview area is another "Select Emotion" label followed by three radio buttons with corresponding superscripted numbers: "[1]", "[2]", and "[3]".

Task Import and Assignment

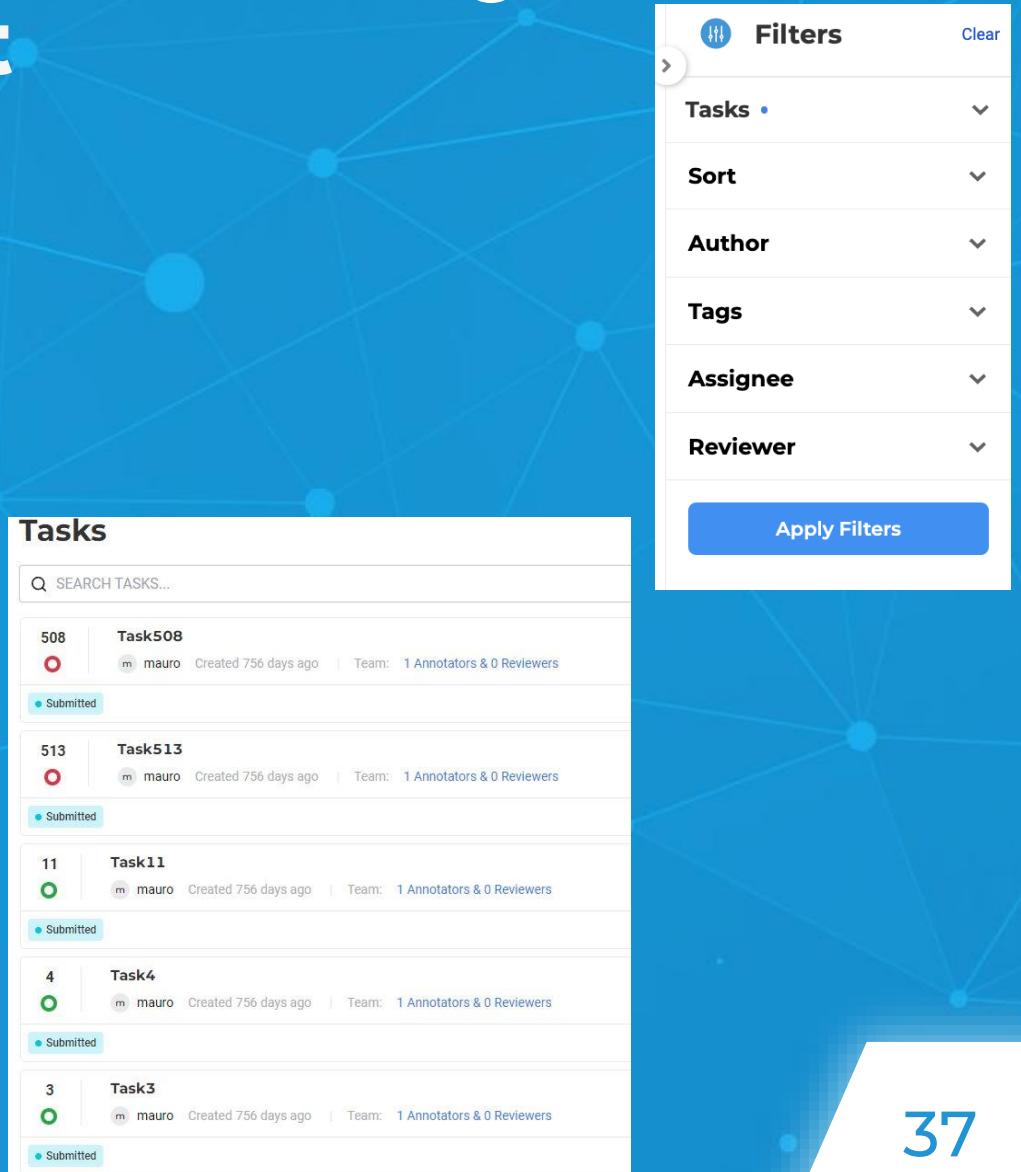


- The screenshot below shows the import page:

The screenshot shows the 'Import' tab selected in the top navigation bar, along with 'Generate Synthetic Tasks' and 'Generate Augmented Data'. Below the tabs, there are two buttons: 'Local Import' (highlighted in blue) and 'Cloud Import'. There are three checkboxes: 'Overwrite completions/predictions' (unchecked), 'Overwrite Ground Truth' (checked), and 'OCR Document' (unchecked). An 'OCR Server' dropdown menu is set to '--Select--'. A text input field contains the URL 'http://example.com/tasks.json OR {"image": "http://my.image.jpg"}'. To the right of the input field is a blue 'Import' button. Below the input field is a large, light-blue rectangular area with a dashed border, containing a small icon of a document with a plus sign and the text 'Drag and drop your files here or click for import'. At the bottom of this area, it specifies supported file types: 'JSON, CSV, TSV, TXT, Zip, PDF, Audio, Video, Images'.

Task Import and Assignment

- Once you import the tasks, choose to Explore tasks; this takes you to the Tasks page. As a project owner you will start assigning tasks to Annotators.
- select one or more tasks, click on Assign drop down and select the name of the user to assign selected tasks. Repeat this step to continue assigning tasks.

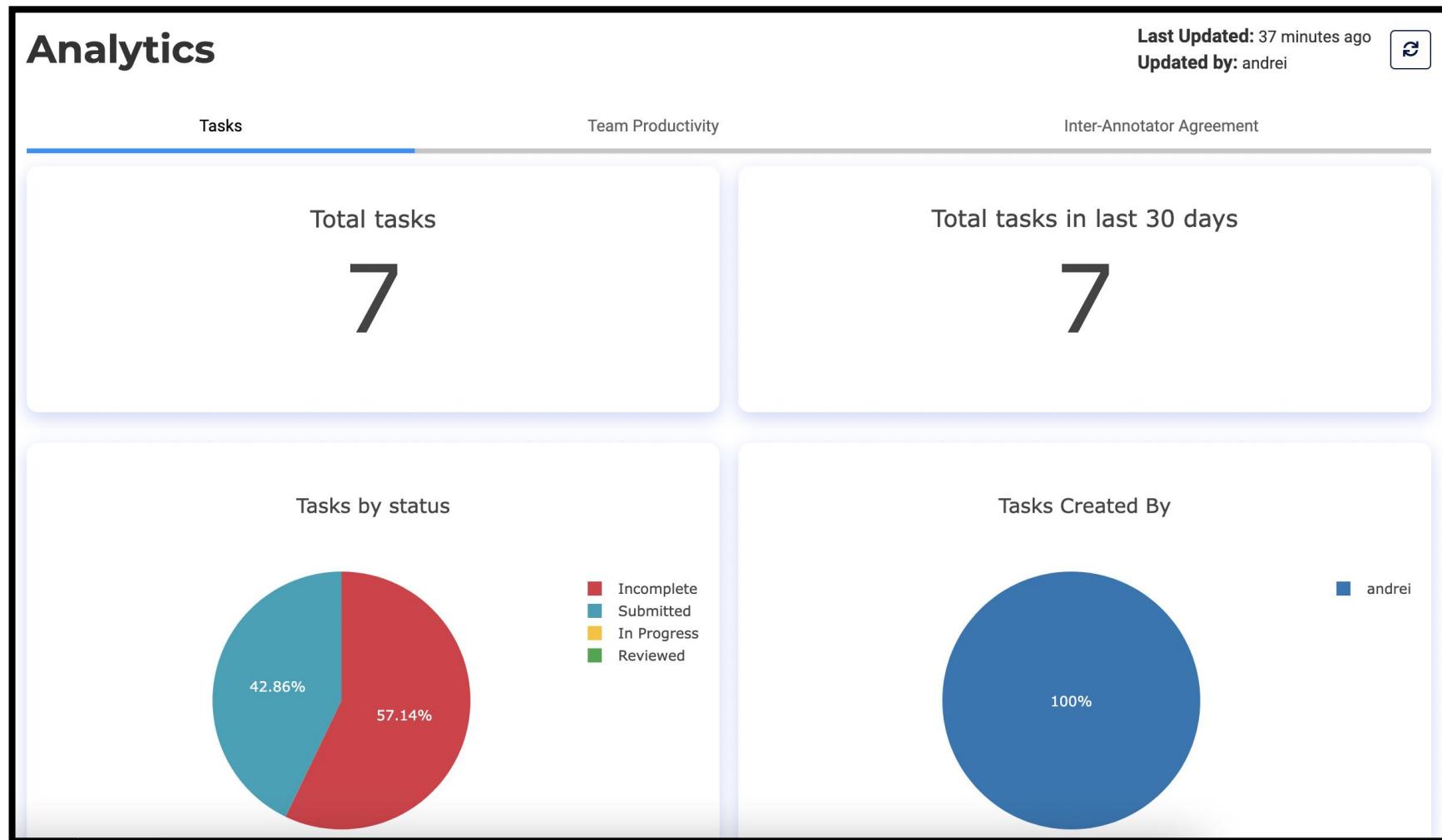


The screenshot shows the 'Tasks' page interface. On the left, there is a sidebar with filter dropdowns for 'Tasks', 'Sort', 'Author', 'Tags', 'Assignee', and 'Reviewer', each with a 'Clear' button. Below the filters is a blue 'Apply Filters' button. The main area is titled 'Tasks' and contains a search bar with placeholder 'SEARCH TASKS...'. A table lists six tasks:

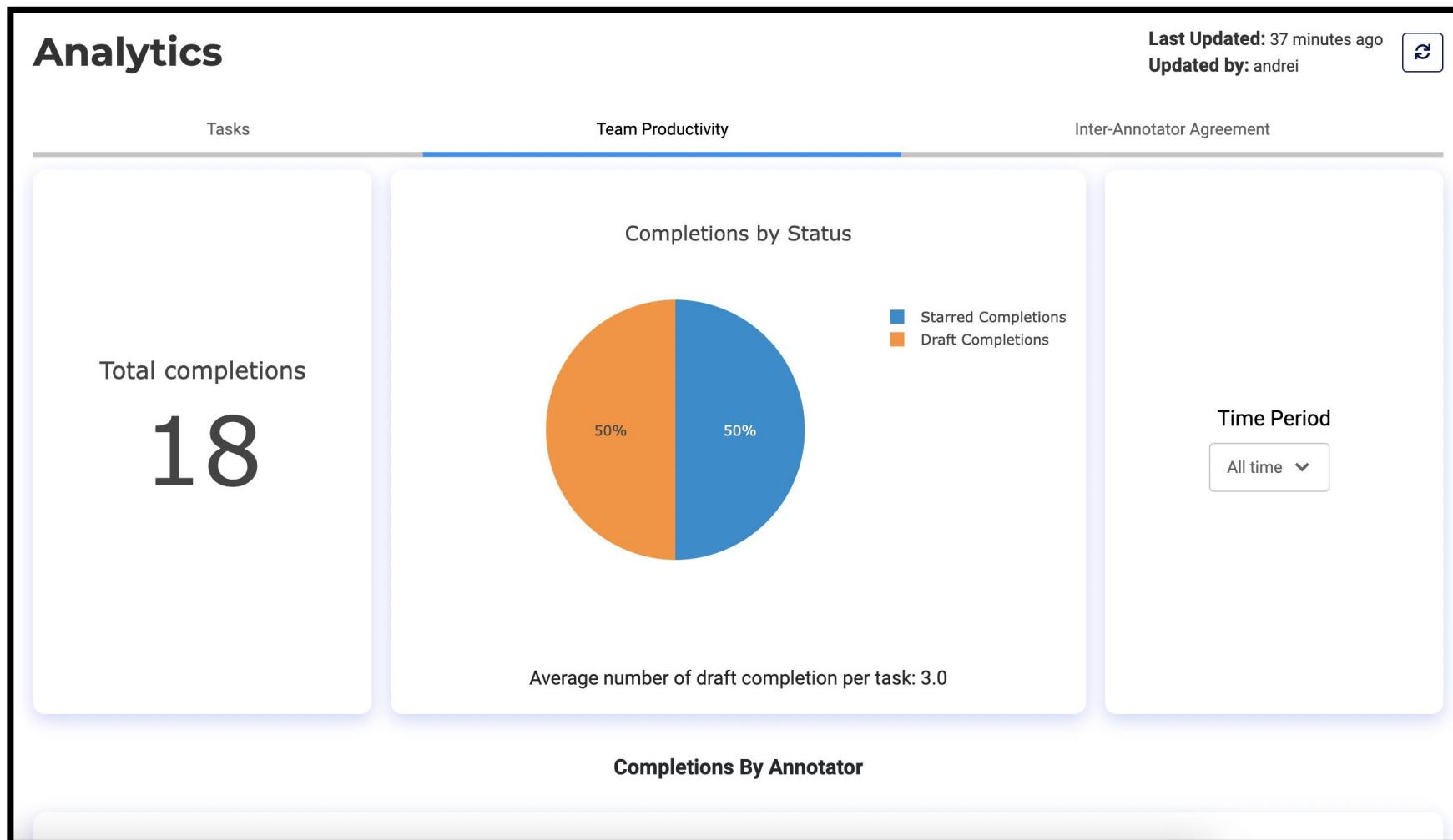
ID	Name	Owner	Created	Team
508	Task508	mauro	Created 756 days ago	1 Annotators & 0 Reviewers
513	Task513	mauro	Created 756 days ago	1 Annotators & 0 Reviewers
11	Task11	mauro	Created 756 days ago	1 Annotators & 0 Reviewers
4	Task4	mauro	Created 756 days ago	1 Annotators & 0 Reviewers
3	Task3	mauro	Created 756 days ago	1 Annotators & 0 Reviewers

Each task row includes a status indicator 'Submitted'.

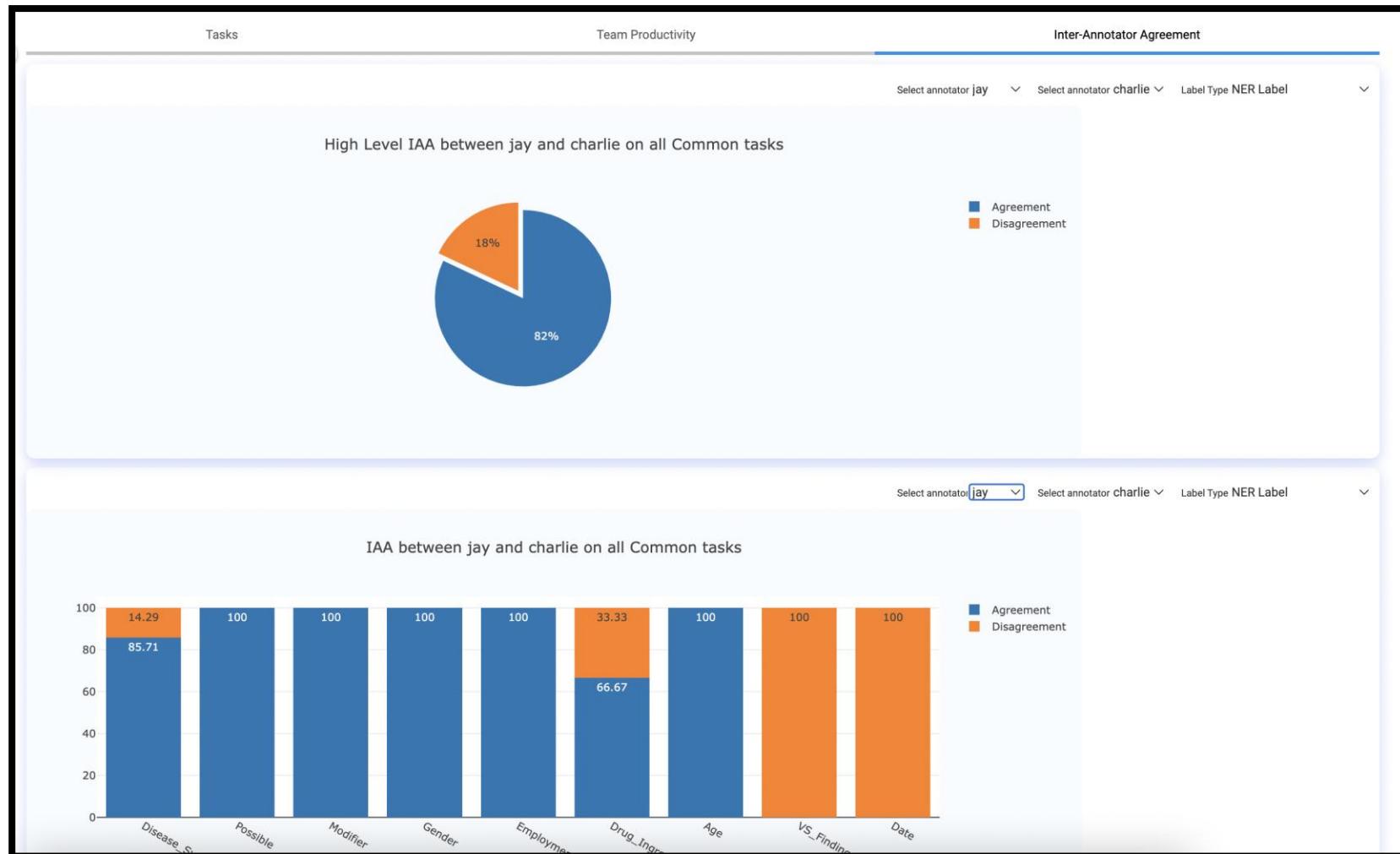
Analytics of Generative AI - Tasks



Analytics of Generative AI Lab - Productivity



Analytics of Generative AI Lab – IAA



Outline

1. Introduction to Text Annotation
2. Manual Annotation
- 3. Annotation Guidelines**
4. Project Setup and Management
5. Preannotations
6. Conclusions and further resources

Contents

- 1. Stakeholders and Roles**
- 2. Structure and Content of AG**
- 3. Best Practices**
- 4. Annotation Errors**
- 5. Project Analysis**



Stakeholders and Roles



Stakeholders and Roles



Annotator Lead

- oversees the entire project - > guidelines align with the project's objectives
- keeps track of changes with a Change Log
- manager role in projects

Annotators

- usually medical professionals (MDs, nurses etc.)
- assess/curate data/annotate
- annotator/reviewer role in projects

Stakeholders and Roles



Data Scientist

- technical stakeholders who will use the annotated data for training
- provide requirements for annotation granularity and specificity

Quality Assurance (Reviewers)

- ensure the quality and consistency of the annotations and AGs
- usually, an experienced annotator from the team is assigned as reviewer



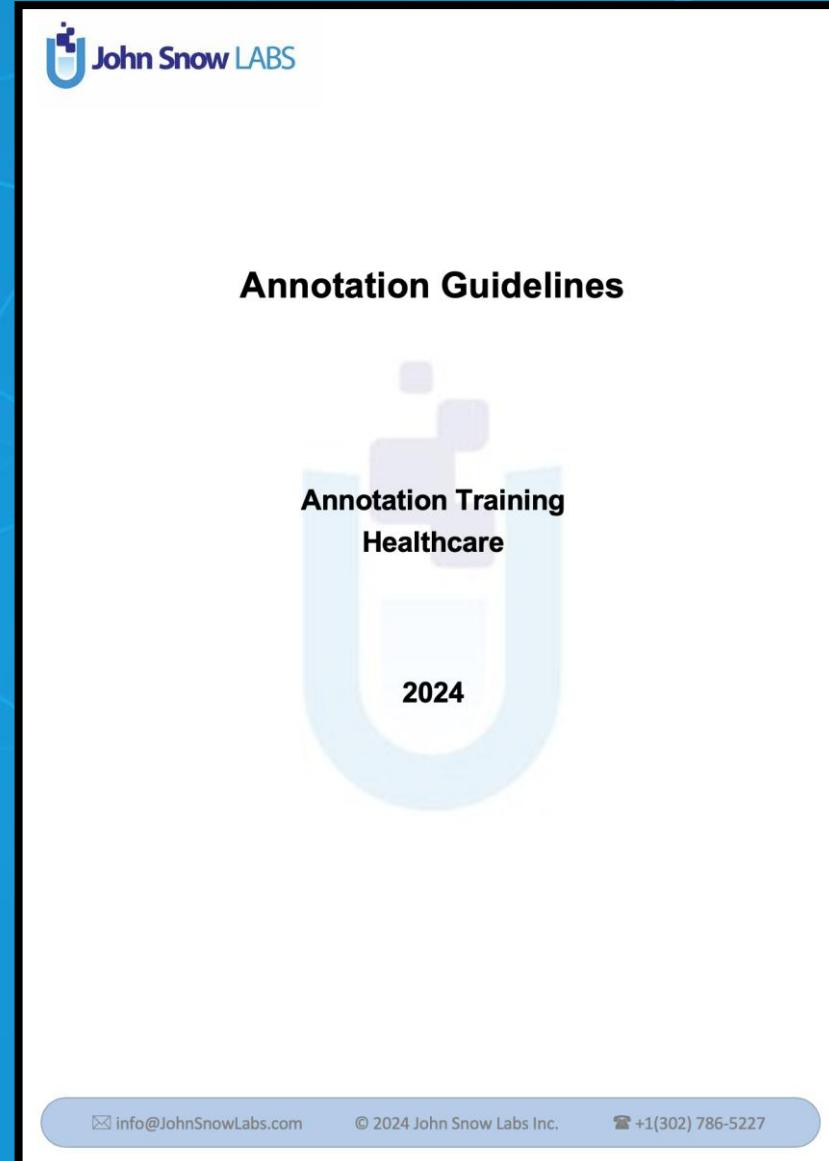
Structure and Content of AG

Contents

1. Cover Page
2. Table of Contents
3. Introduction
4. Taxonomy
5. NER Labels
6. Assertion Labels
7. Relations
8. Text Classification
9. Appendix
10. Changelog

Contents

1. [Cover Page](#)
2. Table of Contents
3. Introduction
4. Taxonomy
5. NER Labels
6. Assertion Labels
7. Relations
8. Text Classification
9. Appendix
10. Changelog



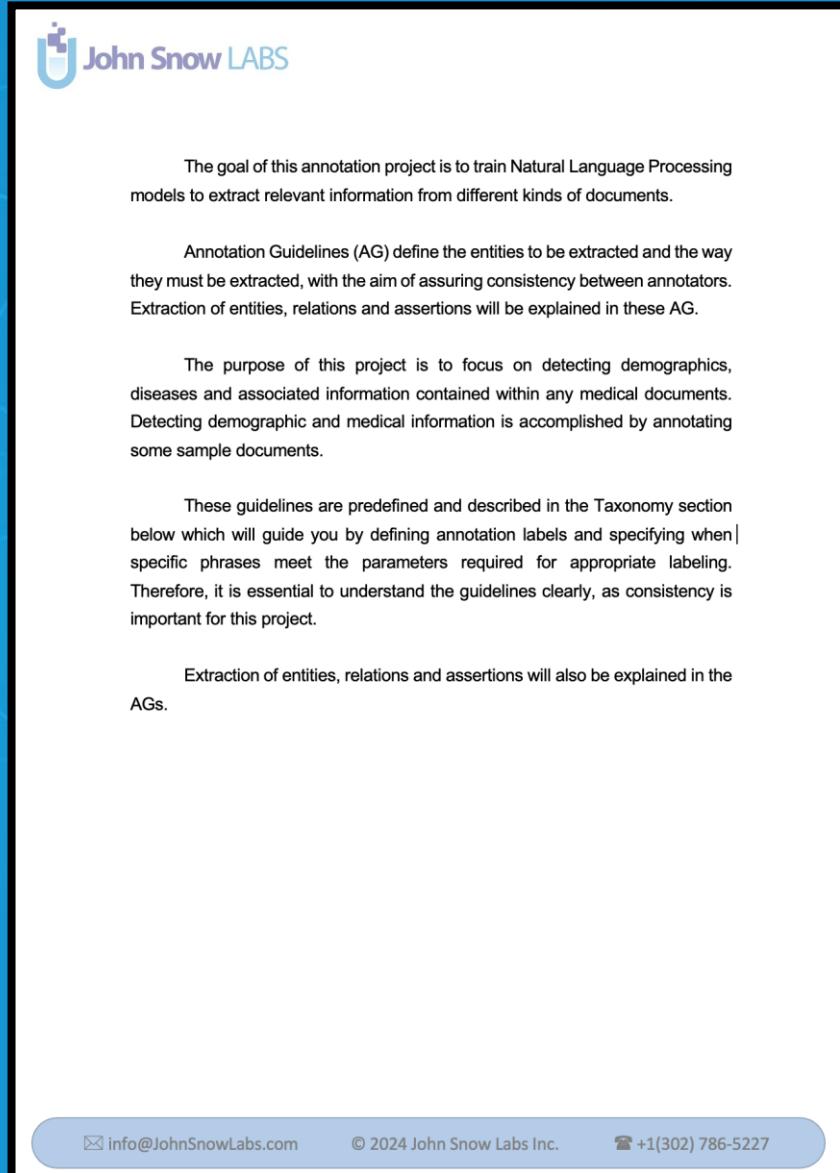
Contents

1. Cover Page
2. **Table of Contents**
3. Introduction
4. Taxonomy
5. NER Labels
6. Assertion Labels
7. Relations
8. Text Classification
9. Appendix
10. Changelog

	John Snow LABS
Content	
Introduction.....	3
Taxonomy	3
Entity Labels	4
Age	4
Gender	4
Employment.....	5
Date.....	5
Disease_Syndrome_Disorder.....	6
Modifier.....	6
VS_Finding	7
Test_Result.....	7
Drug_Ingredient.....	7
Assertion Labels.....	9
Present	9
Absent	9
Possible	10
Family	10
Assertion Table.....	11
Relations	12
Is_diagnosis_date_of	12
Is_modifier_of.....	12
Is_cause_of	13
Is_result_of.....	13
Relation Table.....	14
Text Classification	15
Gender.....	15
Type of Text.....	15

Contents

1. Cover Page
2. Table of Contents
3. Introduction
4. Taxonomy
5. NER Labels
6. Assertion Labels
7. Relations
8. Text Classification
9. Appendix
10. Changelog



The goal of this annotation project is to train Natural Language Processing models to extract relevant information from different kinds of documents.

Annotation Guidelines (AG) define the entities to be extracted and the way they must be extracted, with the aim of assuring consistency between annotators. Extraction of entities, relations and assertions will be explained in these AG.

The purpose of this project is to focus on detecting demographics, diseases and associated information contained within any medical documents. Detecting demographic and medical information is accomplished by annotating some sample documents.

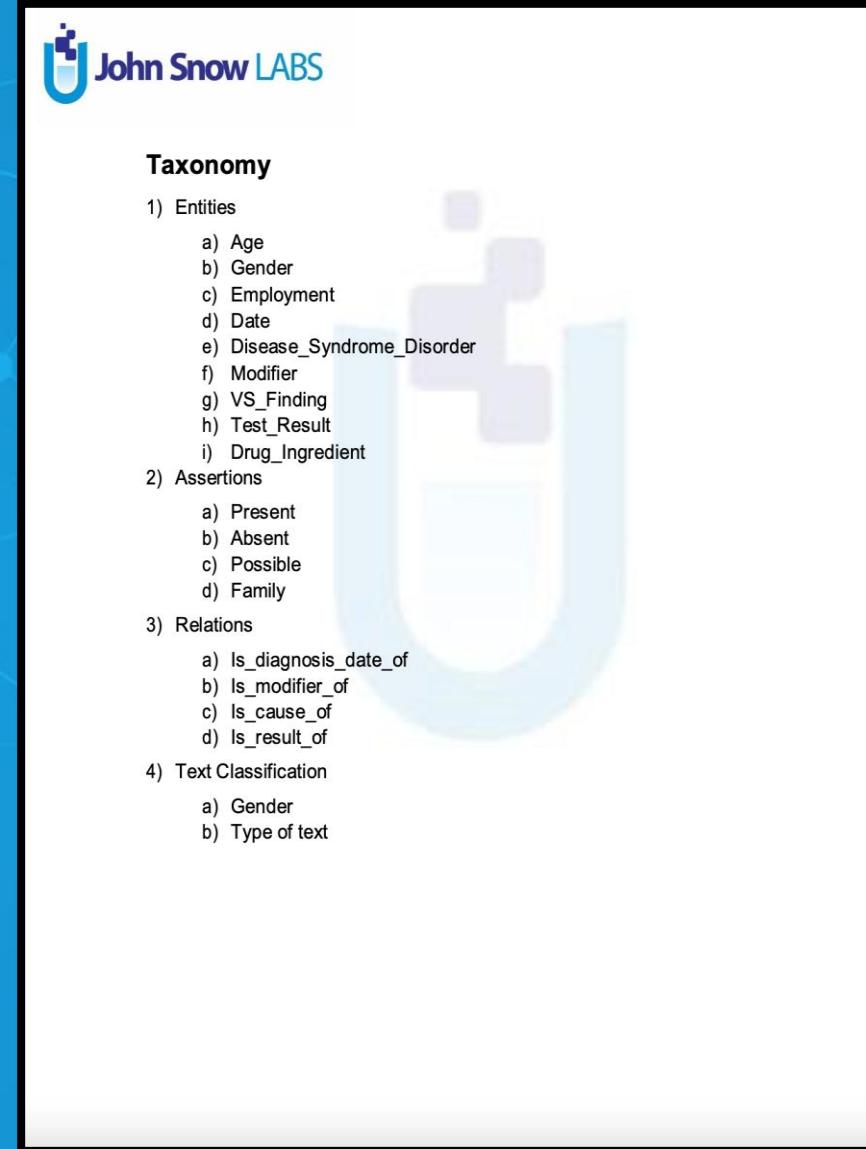
These guidelines are predefined and described in the Taxonomy section below which will guide you by defining annotation labels and specifying when specific phrases meet the parameters required for appropriate labeling. Therefore, it is essential to understand the guidelines clearly, as consistency is important for this project.

Extraction of entities, relations and assertions will also be explained in the AGs.

✉ info@JohnSnowLabs.com © 2024 John Snow Labs Inc. ☎ +1(302) 786-5227

Contents

1. Cover Page
2. Table of Contents
3. Introduction
4. **Taxonomy**
5. NER Labels
6. Assertion Labels
7. Relations
8. Text Classification
9. Appendix
10. Changelog

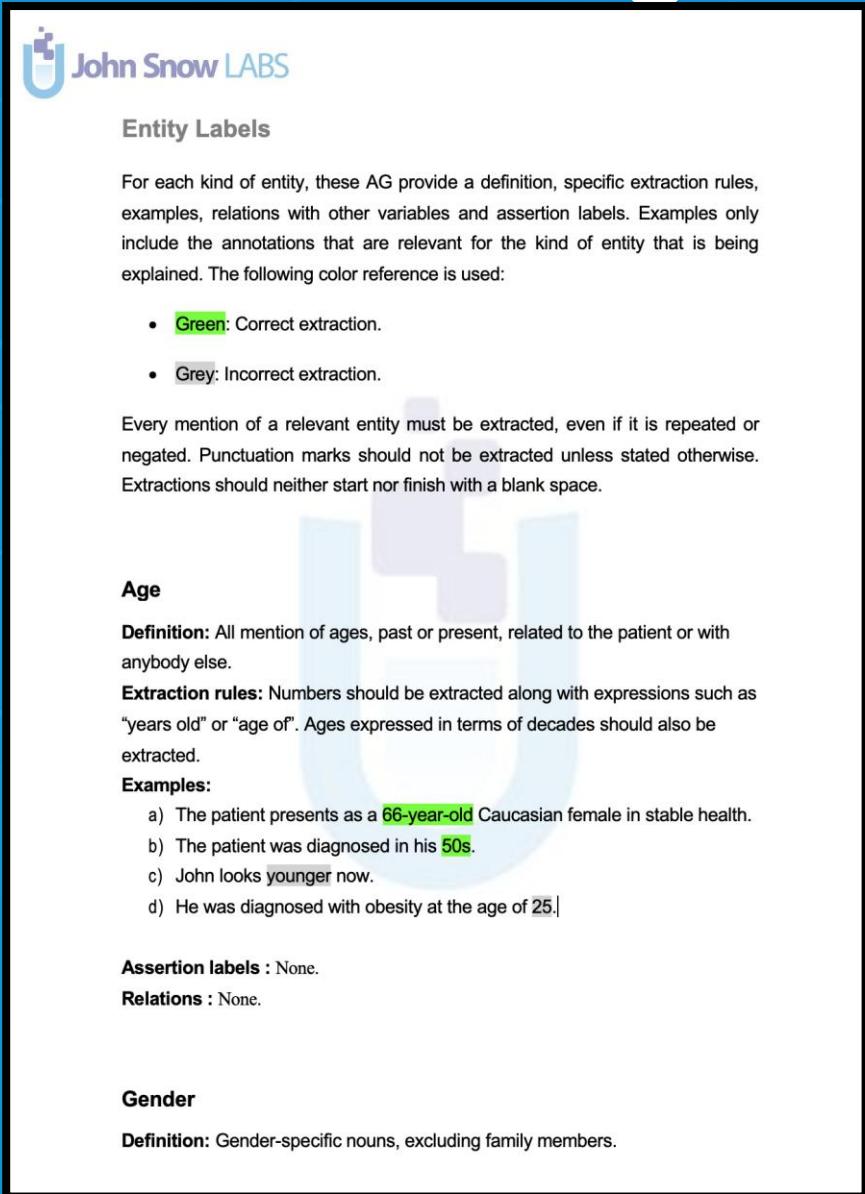


The screenshot shows a slide from a presentation titled "Taxonomy". The slide features the John Snow LABS logo at the top left. The main content is organized into four numbered sections: 1) Entities, 2) Assertions, 3) Relations, and 4) Text Classification. Each section contains a list of specific categories or labels.

Taxonomy	
1)	Entities
a)	Age
b)	Gender
c)	Employment
d)	Date
e)	Disease_Syndrome_Disorder
f)	Modifier
g)	VS_Finding
h)	Test_Result
i)	Drug_Ingredient
2)	Assertions
a)	Present
b)	Absent
c)	Possible
d)	Family
3)	Relations
a)	Is_diagnosis_date_of
b)	Is_modifier_of
c)	Is_cause_of
d)	Is_result_of
4)	Text Classification
a)	Gender
b)	Type of text

Contents

1. Cover Page
2. Table of Contents
3. Introduction
4. Taxonomy
5. **NER Labels**
6. Assertion Labels
7. Relations
8. Text Classification
9. Appendix
10. Changelog



Entity Labels

For each kind of entity, these AG provide a definition, specific extraction rules, examples, relations with other variables and assertion labels. Examples only include the annotations that are relevant for the kind of entity that is being explained. The following color reference is used:

- **Green**: Correct extraction.
- **Grey**: Incorrect extraction.

Every mention of a relevant entity must be extracted, even if it is repeated or negated. Punctuation marks should not be extracted unless stated otherwise. Extractions should neither start nor finish with a blank space.

Age

Definition: All mention of ages, past or present, related to the patient or with anybody else.

Extraction rules: Numbers should be extracted along with expressions such as "years old" or "age of". Ages expressed in terms of decades should also be extracted.

Examples:

- a) The patient presents as a 66-year-old Caucasian female in stable health.
- b) The patient was diagnosed in his 50s.
- c) John looks younger now.
- d) He was diagnosed with obesity at the age of 25.|

Assertion labels : None.

Relations : None.

Gender

Definition: Gender-specific nouns, excluding family members.

Contents

1. Cover Page
2. Table of Contents
3. Introduction
4. Taxonomy
5. NER Labels
6. Assertion Labels
7. Relations
8. Text Classification
9. Appendix
10. Changelog

Assertion Labels

Assertion labels are used to indicate an attribute of an entity. The assertion label is placed on top of the entity label, example, Entity Assertion.

The following are considerations when adding assertions:

- Entities should only be assigned one assertion label only.
- For the annotation of Assertion, it should be considered only the information found in the sentence that includes the asserted entity.

Not all the combinations of entity and assertion are possible. A table of all the entities and possible assertions is included at the end of this section.

Present

Definition: Entities referring to the patient that are currently present and not negated.

Extraction rules: Use this assertion label only for entities extracted as Disease_Syndrome_Disorder.

Example:

a) He is a 60 years old gentleman with diabetes Present Disease_Syndrome_Disorder. (Disease_Syndrome_Disorder + Present Assertion).

Absent

Definition: Label added to negated entities.

Extraction rules: Absent entities are found in phrases that include words such as *no*, *without*, *lack*, etc.

Examples:

a) The ultrasound showed that the patient does not have psoriasis Absent Disease_Syndrome_Disorder. (Disease_Syndrome_Disorder Entity + Absent Assertion).

b) She is neither Diabetic Absent Disease_Syndrome_Disorder nor diagnosed with Obesity Absent Disease_Syndrome_Disorder. (Disease_Syndrome_Disorder Entity + Absent Assertion).

Assertion Table

	Present	Absent	Possible	Family
Age	No	No	No	No
Gender	No	No	No	No
Employment	No	No	No	No
Date	No	No	No	No
Disease_Syndrome_Disorder	Yes	Yes	Yes	Yes
Modifier	No	No	No	No
VS_Finding	No	No	No	No
Test_Result	No	No	No	No
Drug_Ingredient	No	No	No	No

Contents

1. Cover Page
2. Table of Contents
3. Introduction
4. Taxonomy
5. NER Labels
6. Assertion Labels
7. **Relations**
8. Text Classification
9. Appendix
10. Changelog

 John Snow LABS

Relations

Relations are used to link two related entities. To create relations between entities, use the Create Relation button of the annotation tool. Relations are NOT created for entities present in different sentences, or are 2 or more sentences apart. Some relations require a relation label that is found in the relation section of the annotation tool. Also, some relations require assignment of direction that is represented by an arrow in the relation section of the annotator tool. A table with all the possible relations is included at the end of this section.

Is_diagnosis_date_of

Definition: This relation is used to associate a Disease_Syndrome_Disorder entity and a Date entity.

Extraction rules: The Disease_Syndrome_Disorder entity and the relevant date associated with it are extracted and related using the relation label **is_diagnosis_date_of** only when the date refers to the diagnosis of the medical problem.

Examples:

a) She was diagnosed with **hypertension** in **1987**. **Hypertension** | (Disease_Syndrome_Disorder entity) and **1978** (Date entity) are related with **is_diagnosis_date_of** label.

Is_modifier_of

Definition: This relation is used to associate a Disease_Syndrome_Disorder entity and a Modifier.

Extraction rules: The Disease_Syndrome_Disorder and the relevant modifier are extracted and related using the relation label **is_modifier_of**.

Examples:

a) He has been experiencing **chronic migraine** for five years. **Migraine** (Disease_Syndrome_Disorder entity) and **Chronic** (Modifier entity) are related with **is_modifier_of** label.

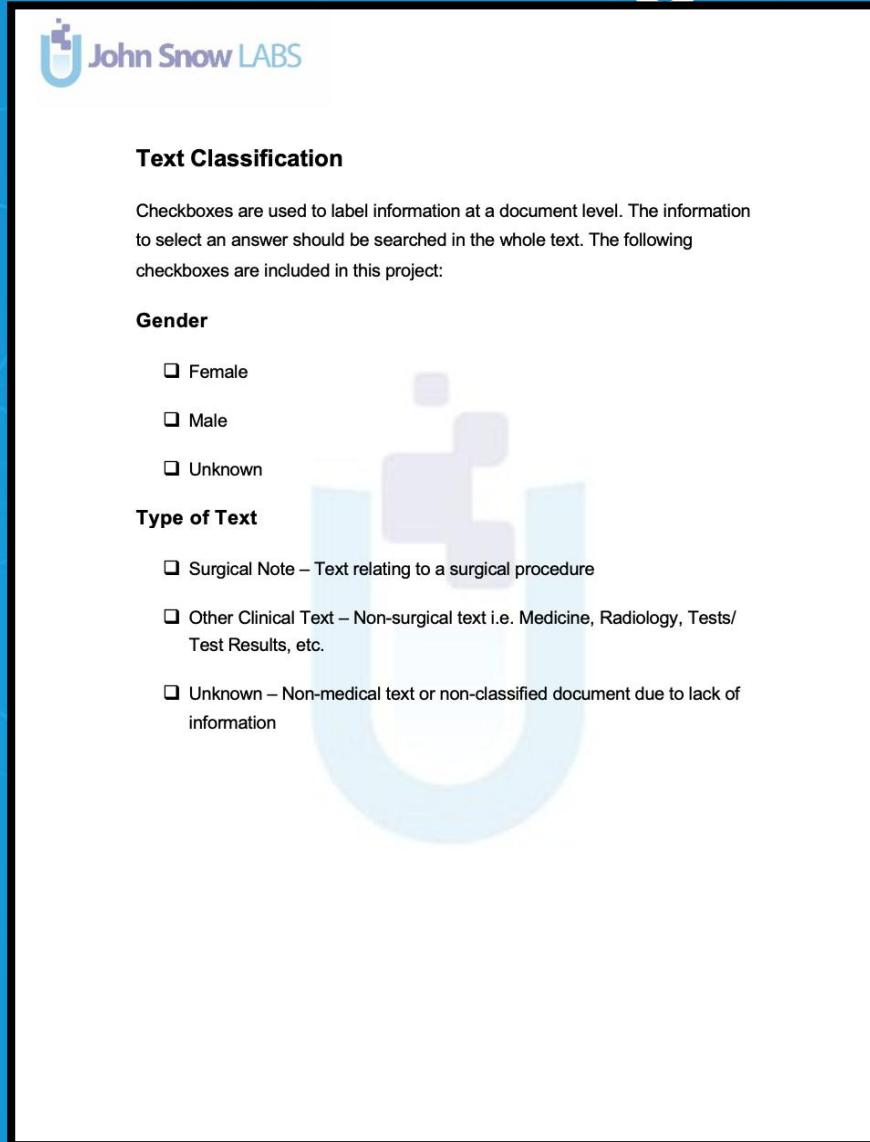
 John Snow LABS

Relation Table

	Entity 1	Entity 2	Label Needed	Direction Needed
Is_diagnosis_date_of	Disease_Syndrome_Disorder	Date	Yes	No
Is_modifier_of	Disease_Syndrome_Disorder	Modifier	Yes	No
Is_cause_of	Disease_Syndrome_Disorder	Disease_Syndrome_Disorder	Yes	Yes
Is_result_of	VS_Finding	Test_Result	Yes	No

Contents

1. Cover Page
2. Table of Contents
3. Introduction
4. Taxonomy
5. NER Labels
6. Assertion Labels
7. Relations
8. **Text Classification**
9. Appendix
10. Changelog



The screenshot shows a section of the John Snow LABS documentation titled "Text Classification". It includes a brief description of how checkboxes are used to label information at a document level, mentioning that answers should be searched in the whole text. It lists checkboxes for "Gender" (Female, Male, Unknown) and "Type of Text" (Surgical Note, Other Clinical Text, Unknown), each with a descriptive subtitle.

Text Classification

Checkboxes are used to label information at a document level. The information to select an answer should be searched in the whole text. The following checkboxes are included in this project:

Gender

Female

Male

Unknown

Type of Text

Surgical Note – Text relating to a surgical procedure

Other Clinical Text – Non-surgical text i.e. Medicine, Radiology, Tests/ Test Results, etc.

Unknown – Non-medical text or non-classified document due to lack of information

Contents

1. Cover Page
2. Table of Contents
3. Introduction
4. Taxonomy
5. NER Labels
6. Assertion Labels
7. Relations
8. Text Classification
9. Appendix
10. [Changelog](#)

 John Snow LABS

Change Log

Version	Revision Date	Revision Description	Responsible for AG Updates
1.0	June 23 rd , 2024	Updates to implement following the consensus meeting – 22.07.2024	@Annotator 1
2.0	June 27 th , 2024	Updates to implement following the consensus meeting – 27.07.2024	@Annotator 2

[✉ info@JohnSnowLabs.com](mailto:info@JohnSnowLabs.com) © 2024 John Snow Labs Inc. [☎ +1\(302\) 786-5227](tel:+1(302)786-5227)

Examples #1

Age

Definition: All mention of ages, past or present, related to the patient or with anybody else.

Extraction rules: Numbers should be extracted along with expressions such as “years old” or “age of”. Ages expressed in terms of decades should also be extracted.

Examples:

- a) The patient presents as a 66-year-old Caucasian female in stable health.
- b) The patient was diagnosed in his 50s.
- c) John is a 65 years old male.
- d) He was diagnosed with obesity at the age of 25.

Assertion labels : None.

Relations : None.

Examples #2

Employment

Definition: Mentions of jobs or occupations included in the text.

Extraction rules: Extract terms that are related to any specific jobs or employment, whether related to the patient or not. Do not extract words such as "works", "working" or "employed".

Examples:

- a) She is an office manager for a gravel company.
- b) She will also see a nutritionist and a social worker.
- c) He works as a financial officer.

Assertions: None.

Relations: None.

Examples #3

Disease_Syndrome_Disorder

Definition: Extract all the diseases, syndromes and any relevant condition mentioned in the document.

Extraction rules: Extract all mentions of medical conditions and diseases, including those related to the patient or to a family member. Do not include in the extraction modifiers such as "chronic", "mild" or "severe" (this kind of words should be extracted using the label Modifier).

Examples:

- a) The patient has **Alzheimer** diagnosed back in 2012.
- b) He was diagnosed with **colon cancer**.
- c) A diagnosis of **chronic kidney disease** was established in the past.
- d) He was diagnosed twice with **chronic depression**.

Modifier

Definition: Terms that modify the medical problem.

Extraction rules: Extract words that indicate severity (such as "mild" or "severe"), duration (such as "chronic" or "acute") or any other feature of the entities.

Examples:

- a) He has been experiencing **chronic** back pain for five years.
- b) Patient with history of **recurrent** angina.



Best Practices

Best Practices - Annotation Rules



Centralized

one document should include all the rules
(the AGs)



Consistent

contradiction should be avoided



Specific

ambiguity should be avoided



Cross-check Annotations

Use multiple annotators for the same data
("seed corpus")



Clear Objectives

objectives will shape your annotation
guidelines



Explicit

i.e written (even if in detail!)

Best Practices - AG Development



Define goals clearly

What information is needed from texts?

e.g.: "identifying all gender-related words
(NER)"

vs.

"classifying the document based on gender"
(Text Classification +/- NER)



Communication is key

Define communication channels (Slack,
Teams etc.)

Keep track of decisions made (Change Log)

Encourage annotators to avoid private
messaging



Iterative process

Test your AGs

Seed Corpus annotations (pilot)

data scientists/NLP engineers

Best Practices - NER



clinical problem of persistent coughing Clinical_Problem

VS

clinical problem of persistent Modifier coughing Symptom

Define the level of **granularity** of your taxonomy

e.g.: **Clinical_Problem** label vs. **Modifier + Symptom**

Treatment will involve the drug Tarceva Cancer_Treatment

VS

Treatment will involve the drug Tarceva Drug

Each label should have **clear boundaries** (avoid overlapping)

e.g.: “Treatment will involve the drug Tarceva”

(**Cancer_Treatment** vs **Drug**)

Best Practices - NER

concerning evidence of metastasis to the liver^{Metastasis}

a primary symptom indicative of a potential respiratory issue^{Possible Clinical_Problem}

vs

diagnosed with lung disease, specifically non-small cell lung can
 concerning evidence of metastasis^{Metastasis} to the liver^{Site}

Gender

Female^{LJ} Male^{LJ 1.00} Unknown^{LJ}

Extractions should not include more than 2-3 words!

Combine NER with other NLP features

- Consider merging/dropping/splitting entities

assertion labels

e.g.: “*There is metastasis in the lung*” (*Metastasis* >
Metastasis + Site)

relations

text classification

Best Practices - Assertion

Keep Assertion Taxonomy simple!

the patient was diagnosed with lung disease^{Present} Disease

vs

the patient was diagnosed with lung disease^{Disease}

medical history reveals an allergy to penicillin^{Allergen} Drug

- Assign a default assertion status

e.g.: **"The patient has cancer". (Disease + Present assertion label by Default)**

How to decide between multiple possible assertion labels

e.g.: **"No family history of cancer". (Disease + Absent/Family?)**

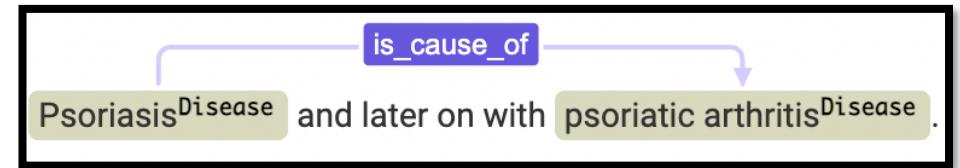
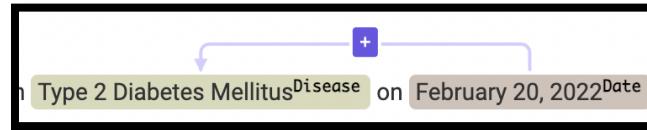
Create your own assertion labels

e.g.: **"medical history reveals an allergy to penicillin" (Drug NER label + Allergen assertion label)**

- Define each assertion label

Best Practices - Relations

Define **entities** that can be related!



- Add Relation Labels Names only when there are different ways of relating two entities

Disease & Date (1 relation): no need to add a Label
is_date_of_diagnosis

- Drug & Date (3 relations): label needed*
(is_start_date_of, is_stop_date_of, is_generic_date_of)

Add direction only if necessary

is_cause_of (Disease & Disease) needs direction to differentiate cause and consequence

- Define the **distance** between entities to use relation label



Frequent Annotation Errors

Cause of Error

1. Accidental Human Error

Example: "female" extracted as Age

2. Wrong Extraction

Example: "young" extracted as Age (if AG specify that only numeric values should be extracted).

3. Normal Disagreement

Example: different granularity to extract "late 40s" as Age.

Kinds of NER Errors

1. Missing Extraction

“The patient is 30 years old” [0]

2. Wrong Label

“The patient is 30 years old” [Gender]

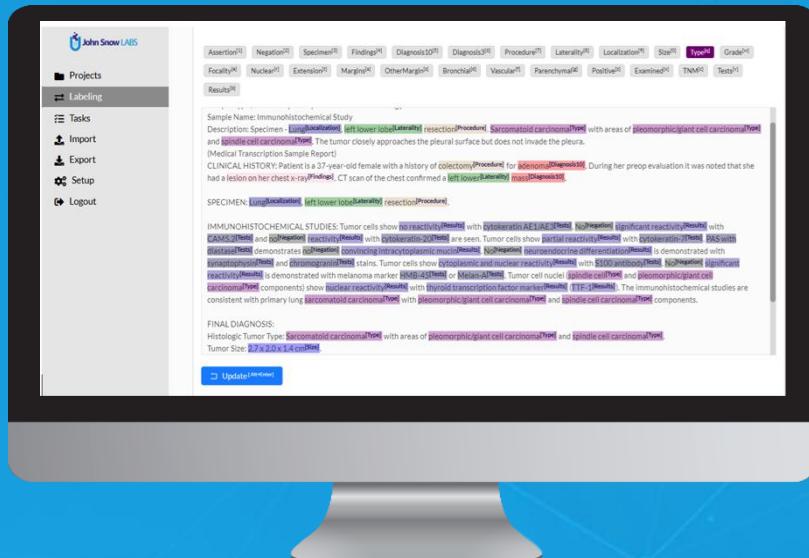
3. Wrong Granularity

“The patient is 30 years old” [Age]

4. Wrong Extraction

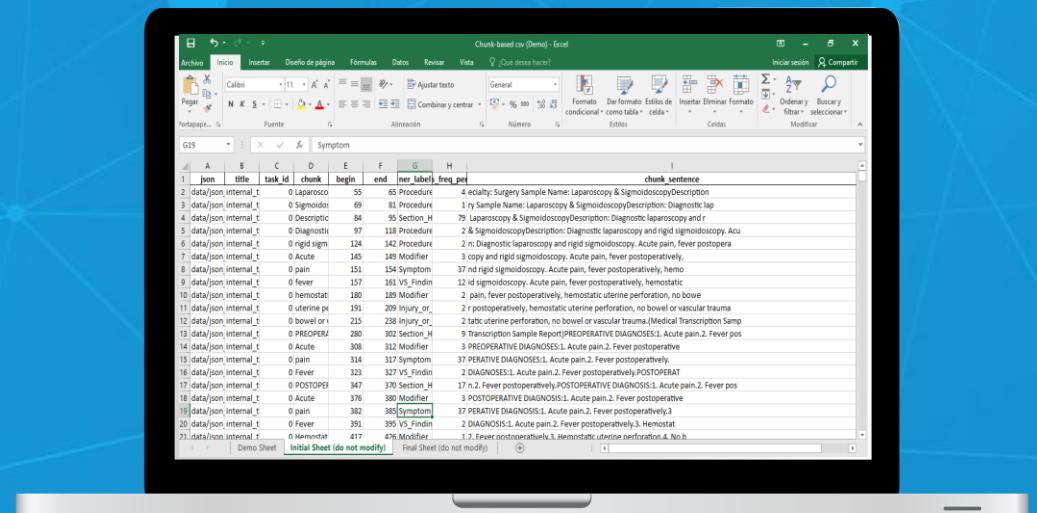
“The patient is young” [Age] (redundant if AG specify “only numeric values”)

Data Review



The screenshot shows the John Snow LABS labeling interface. On the left, a sidebar lists 'Projects', 'Labeling' (which is selected), 'Tasks', 'Import', 'Export', 'Setup', and 'Logout'. The main area displays a medical report for a 'Left lower lobectomy resection' specimen. The report includes sections for 'CLINICAL HISTORY', 'IMMUNOHISTOCHEMICAL STUDIES', and 'FINAL DIAGNOSIS'. It also notes 'Histologic Tumor Type: Sarcomatoid carcinoma with areas of pleiomorphic giant cell carcinoma and spindle cell carcinoma'. A blue 'Update' button is at the bottom.

Manual Reviews



The screenshot shows a Microsoft Excel spreadsheet titled 'Chunk-based cov (Demo) - Excel'. The data is organized into columns labeled A through H. Column A contains labels such as 'pom', 'data/json/internal_t', 'data/json/internal_t', etc. Column B contains titles like 'title', 'internal_t', 'internal_t', etc. Column C contains task IDs like 'task_id', 'Laparoscop', 'Sigmoidoi', etc. Column D contains 'chunk' values like 'chunk', '0', '0', etc. Column E contains 'begin' and 'end' coordinates, and column F contains 'ner_label'. Column G contains 'freq' and 'per' values, and column H contains 'chunk_sentence'. The data includes various medical terms and codes, such as '4 ectomy', '1/r Sample Name: Laparoscopy & Sigmoidoscopy/Description: Diagnostic laparoscopy and sigmoidoscopy', and '2 n: Diagnostic laparoscopy and rigid sigmoidoscopy. Acute pain, fever postopera'. The last row is a 'Demo Sheet'.

CSV Reviews



Project Analytics



Model Training



Projects / Healthcare_Standard_October_Training_Public / Training & Active Learning

Training & Active Learning

1. TRAINING SETTINGS

The settings chosen below also apply to Active Learning.

Training Type: ner

License Type: Open source

Embeddings: glove_100d

2. TRAINING PARAMETERS

Epoch: 25 | Learning Rate: 0.001 | Learning Rate Decay: 0.005

Dropout: 0.5 | Batch: 16

Train / Test data

Split dataset using Test/Train tags

Random Split Data Set | Validation Split: 0.2

[Optional] Provide a snippet of text you've used during training, in case you decide to share this model on the Online Models page later.

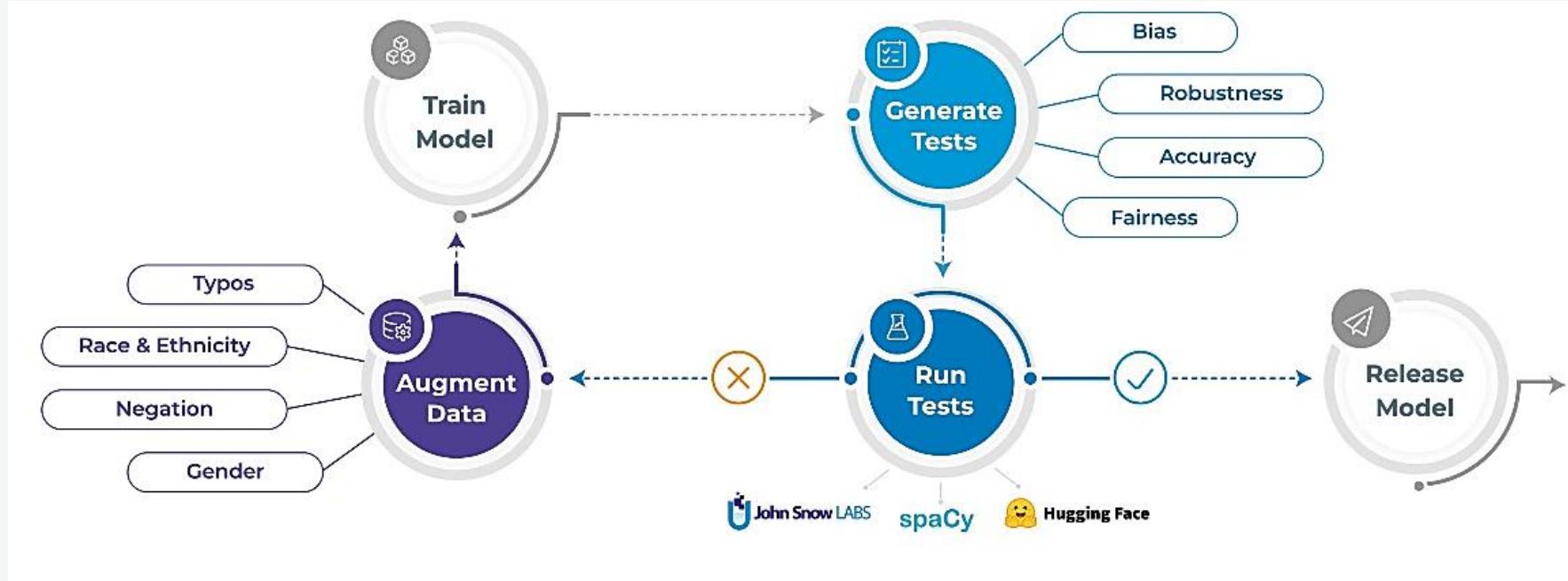
Sample Text

3. ACTIVE LEARNING

Active Learning feature will automatically train a new model when the selected number of new completions is reached. The training will be triggered only when a license is available and the model server count is within the limit.

Completion Count: 50

Save | Train Model | Click here to go to Training Resource Management



Model Testing Flow

- Pick a model to test. Generative tests. Run tests. Improve system until the tests pass.
- Automated test generation & execution is based on the LangTest library.

Robustness Tests

Are your model's predictions stable under small perturbations of the input?

Contextual changes	Punctuation manipulation	Randomization	Semantic alterations
add_abbreviation	add_punctuation	randomize_age	add_slangs
add_context	strip_all_punctuation		number_to_word
add_contraction	strip_punctuation		

Spelling variations	Text casing	Typographical errors	Word and entity swaps
american_to_british	lowercase	add_ocr_typo	adjective_antonym_swap
british_to_american	titlecase	add_speech_to_text_typo	adjective_synonym_swap
	uppercase	add_typo	dyslexia_word_swap
			swap_entities

Bias Tests

Does your model change its answer if a person's name is changed to be female, Muslim, or Hispanic?

Gender bias	Economic bias	Racial and ethnic bias	Religious bias
replace_to_male_pronouns	replace_to_high_income_country	replace_to_white_firstnames	replace_to_muslim_names
replace_to_female_pronouns	replace_to_low_income_country	replace_to_black_firstnames	replace_to_hindu_names
replace_to_neutral_pronouns	replace_to_lower_middle_income_country replace_to_upper_middle_income_country	replace_to_hispanic_firstnames replace_to_asian_firstnames replace_to_white_lastnames	replace_to_christian_names replace_to_jain_names replace_to_parsi_names
		replace_to_black_lastnames	replace_to_buddhist_names
		replace_to_hispanic_lastnames	replace_to_sikh_names
		replace_to_asian_lastnames	
		replace_to_native_american_lastnames	
		replace_to_inter_racial_lastnames	

Fairness Tests

Is your model accurate for men and for women?

Classification models	Question Answering and Summarization	Question Answering with LLM-eval
max_gender_f1_score	max_gender_rouge1_score	max_gender_llm_eval
min_gender_f1_score	max_gender_rouge2_score	min_gender_llm_eval
	max_gender_rougeL_score	
	max_gender_rougeLsum_score	
	min_gender_rouge1_score	
	min_gender_rouge2_score	
	min_gender_rougeL_score	
	min_gender_rougeLsum_score	

Representation Tests

Are there enough cases of each gender, ethnicity, religion, and country to even calculate fairness?

Representation Count	Representation Proportion
min_country_economic_representation_count	min_country_economic_representation_proportion
min_ethnicity_name_representation_count	min_ethnicity_name_representation_proportion
min_gender_representation_count	min_gender_representation_proportion
min_label_representation_count	min_label_representation_proportion
min_religion_name_representation_count	min_religion_name_representation_proportion

More Test Types

Accuracy

Add Test Suite

Accuracy (6) 1 Tests selected

Accuracy tests assess the correctness of a model's predictions.

Min F1 Score

Description : This test checks the f1 score for each label. Test is passed if the f1 score is higher than the configured min score.

min_score
Minimum pass rate to pass the test.

Add Model Labels:

Pathogen	x	Medicine	x
0.8		0.75	
MedicalCondition			
0.75			

Min Macro F1 Score

Description : This test checks the macro-f1 score. Test is passed if the macro-f1 score is higher than the configured min score.

Speed

Edit Test Suite

Accuracy (6)

Bias (24)

Fairness (2)

Performance (1) 1 Tests selected

Performance tests assess the efficiency and speed of a language model's predictions.

Speed

Description : Evaluate the speed of model execution based on tokens.

min_pass_rate
Minimum pass rate to pass the test.

unit
tokens/sec

Representation (8)

Create Tests with the Test Page Wizard

The screenshot shows the Pacific AI Test Page Wizard interface, specifically the 'Projects' page. The top navigation bar includes the Pacific AI logo, a back arrow, the title 'Projects', and buttons for 'Import' and '+ New'. A search bar labeled 'SEARCH PROJECT ...' is positioned above a grid of project cards. The sidebar on the left offers links to 'Projects', 'Hub', and 'Settings', along with a 'Help?' button. The bottom navigation bar features user information ('admin'), a 'View' dropdown set to '15 Projects per page', and a page indicator 'Showing 1-3 of 3 Projects'.

Project Name	Last Update	Tasks	Actions
TestAnalyticsPage	Sep 13, 2024	149 Tasks	⋮
LungCancer	Sep 13, 2024	1254 Tasks	⋮
Hospital_Analysis	Sep 11, 2024	71 Tasks	⋮

Use Generative AI to Auto-Generate Test Cases

The screenshot displays the Pacific AI platform's interface for managing test suites. The left sidebar, titled "Projects", is expanded to show the "Hub" section, which contains categories like Online Models, Local Models, Test Suites, Embeddings, Rules, Prompts, and Pipeline. Below the sidebar is a "Help?" button. The main area is titled "HUB / Test Suite" and features a search bar labeled "SEARCH TEST SUITES ...". A prominent orange banner at the top states, "Continuous Server Usage Incurs Costs! Please check the deployed server". On the right side of the header are buttons for "Go To Cluster Page", "New", and "Import". The central content area lists four test suites:

- Hospital_Evaluation_Test_Suite** (Created by: admin on Sep 12, 2024) - Model Type: ner
- AccuracyUpper** (Created by: admin on Sep 10, 2024) - Model Type: ner
- Test2** (Created by: admin on Aug 19, 2024) - Model Type: ner
- Test** (Created by: admin on Aug 19, 2024) - Model Type: ner

At the bottom, there are navigation controls for "View" (set to 15), "Test Suites per page", and a page indicator showing "Showing 1-4 of 4 Test Suites" with a page number "1".

Run Tests and Visually Review Results

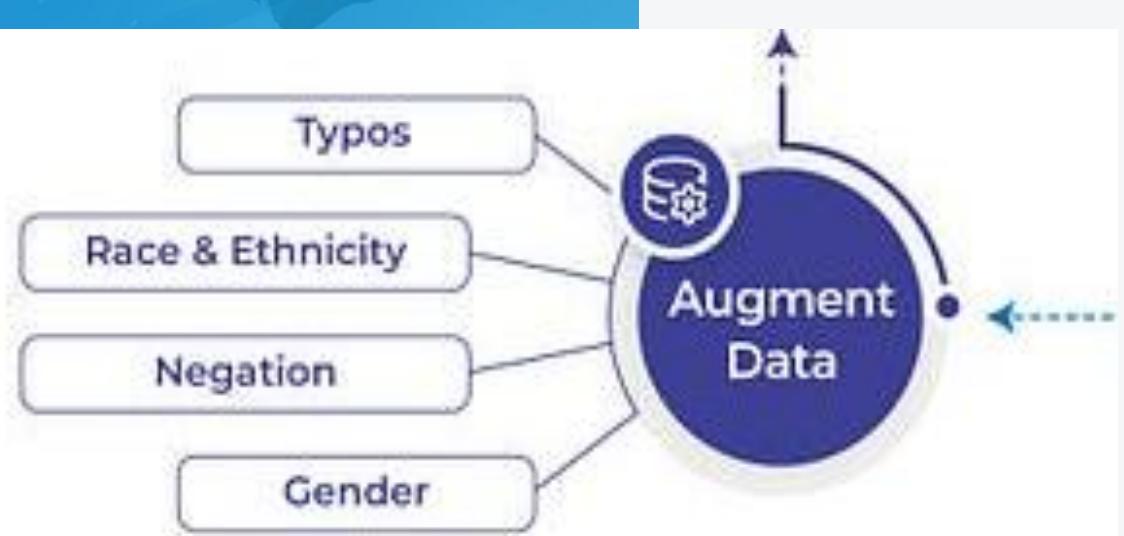
The screenshot shows the Pacific AI platform's HUB/Test Suite interface. The left sidebar navigation includes 'Projects' (Hub, Online Models, Local Models, Test Suites, Embeddings, Rules, Prompts, Pipeline), 'Settings', and 'Help?'. The main area displays a list of test suites:

- Hospital_Evaluation_Test_Suite** (Created by: admin on Sep 12, 2024) - Model Type: ner
- AccuracyUpper** (Created by: admin on Sep 10, 2024) - Model Type: ner
- Test2** (Created by: admin on Aug 19, 2024) - Model Type: ner
- Test** (Created by: admin on Aug 19, 2024) - Model Type: ner

Top navigation bar: Continuous Server Usage Incurs Costs! Please check the deployed server, Go To Cluster Page, + New, Import.

Bottom navigation: View 15, Test Suites per page, Showing 1-4 of 4 Test Suites, Page 1.

Augmented Data

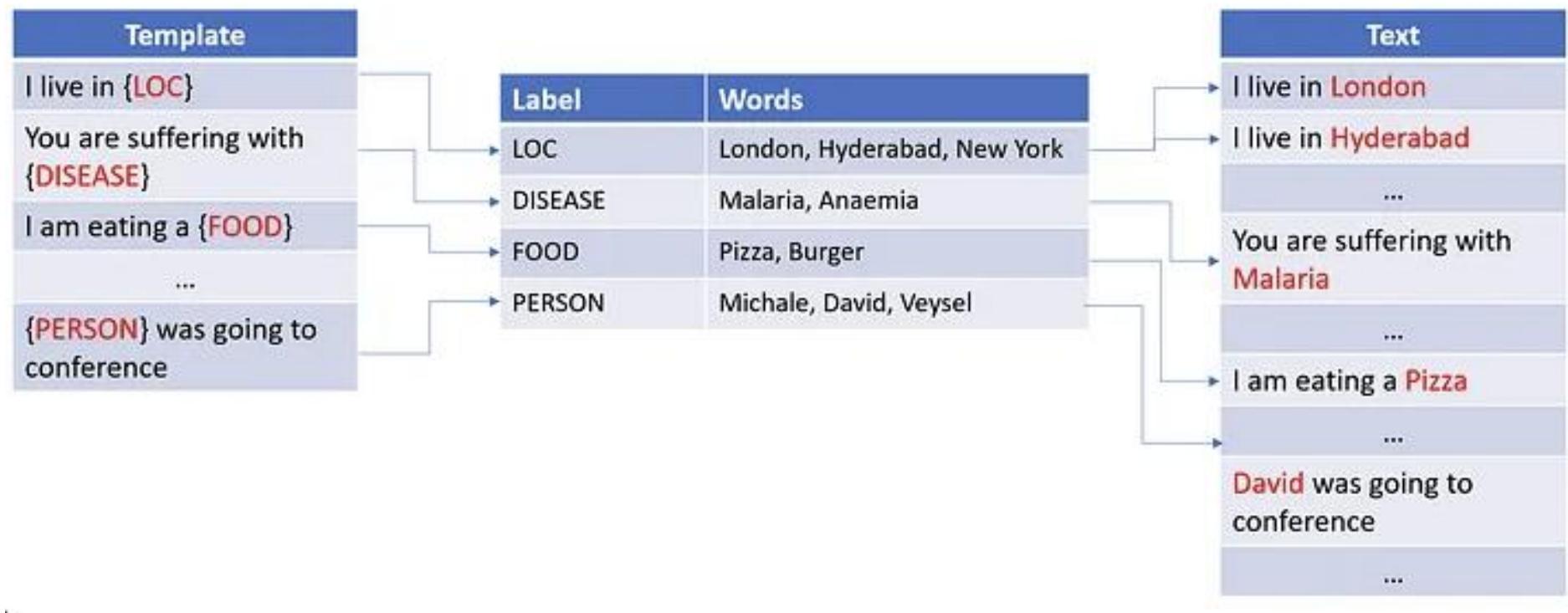


Why Use Augmented Data?

1. HIPPA compliant
2. Designed to fill holes in original dataset
3. Generate additional possible cases the model can use to improve itself

Data Augmentation: Don't Just Test It, Fix It!

Expand the model's training dataset by generating templatic variations of existing data (e.g., rotations, noise addition, translations) to help the model learn more diverse patterns.



Synthetic Data – Use Open AI

Create net-new templates and data using our chatgpt integration.

The screenshot shows a user interface for generating synthetic data. At the top, there are three navigation links: "Import", "Generate Synthetic Tasks" (which is underlined in blue), and "Generate Augmented Data". Below these, a sub-header reads: "Generate Synthetic text sentences, documents in batches, by providing a prompt as input." A "Settings" gear icon is located in the top right corner of this section.

Write Prompt i

As a versatile and creative text generator, you have the ability to generate texts that are over 1000 words long. Please generate a text that meets the following criteria:
The text should be written from the perspective of a patient discussing a healthcare-related issue on an online health forum.
- Each text must mention at least one allergen.
- Include at least one medical treatment or medical test the patient has experienced.
- Use an informal writing style, similar to oral language or a tweet.
- Be creative in your text generation.
- To help achieve the word count, consider discussing some of the following topics in your response:
 Describe the patient's initial symptoms and how they discovered their allergy.
 Discuss the patient's experience navigating the healthcare system, including any tests or treatments they've undergone.
 Share the patient's feelings and concerns about living with the allergy.
 Provide any tips, advice, or resources the patient has found helpful in managing their allergy.
 Talk about any challenges the patient has faced in adapting their lifestyle to accommodate the allergy.
- Each generated text MUST be AT LEAST 1200 chars long and contain at least tree paragraphs. (IMPORTANT: WORD COUNT MINIMUM OF 1200 chars for each Text)

Set Parameters

Integration
Provide details for integration endpoint and secret key
Input field: `https://api.openai.com/v1/chat/comple`
Input field: Enter secret key... x

Validate

Select Temperature i
A slider is set to 0.5, with 0.00 on the left and 1.00 on the right.

Select number of text to generate
A slider is set to 1, with 1 on the left and 50 on the right.

Generate Result

i 1 distinct text results will be generated in this run, use the configuration section if you need to adjust this number.

Tag your results i +

Sentence

No Synthetic Tasks yet!

Create Data Augmented Tasks



Pacific AI

Continuous Server Usage Incurs Costs! Please check the deployed server

Go To Cluster Page | X

HUB / Test Suite

+ New | Import

Projects

Hub

- Online Models
- Local Models
- Test Suites
- Embeddings
- Rules
- Prompts
- Pipeline

SEARCH TEST SUITES ...

Hospital_Evaluation_Test_Suite ...
Created by: admin on Sep 12, 2024
Model Type: ner

AccuracyUpper ...
Created by: admin on Sep 10, 2024
Model Type: ner

Test2 ...
Created by: admin on Aug 19, 2024
Model Type: ner

Test ...
Created by: admin on Aug 19, 2024
Model Type: ner

View 15 Test Suites per page

Showing 1-4 of 4 Test Suites

Help? | admin



Retrain the model based on the new tasks

Projects / Hospital_Analysis / Import

Import Generate Synthetic Tasks Generate Augmented Data

+ Add Sample Task i

Projects ^

Hospital_Analysis

- Analytics
- Tasks
- Train
- Test
- Setup

Hub

Settings

Help? ?

admin ^

Proportional Augmentation Templatic Augmentation

This option improves data quality and model strength by making small changes based on selected robustness and bias tests. The augmented dataset can then be used to retrain a model to make it more effective across a wide range of NLP scenarios than its previous version. [Learn more](#)

Bias i

Max Proportion
Default Increase (30%) Apply to all

Robustness i

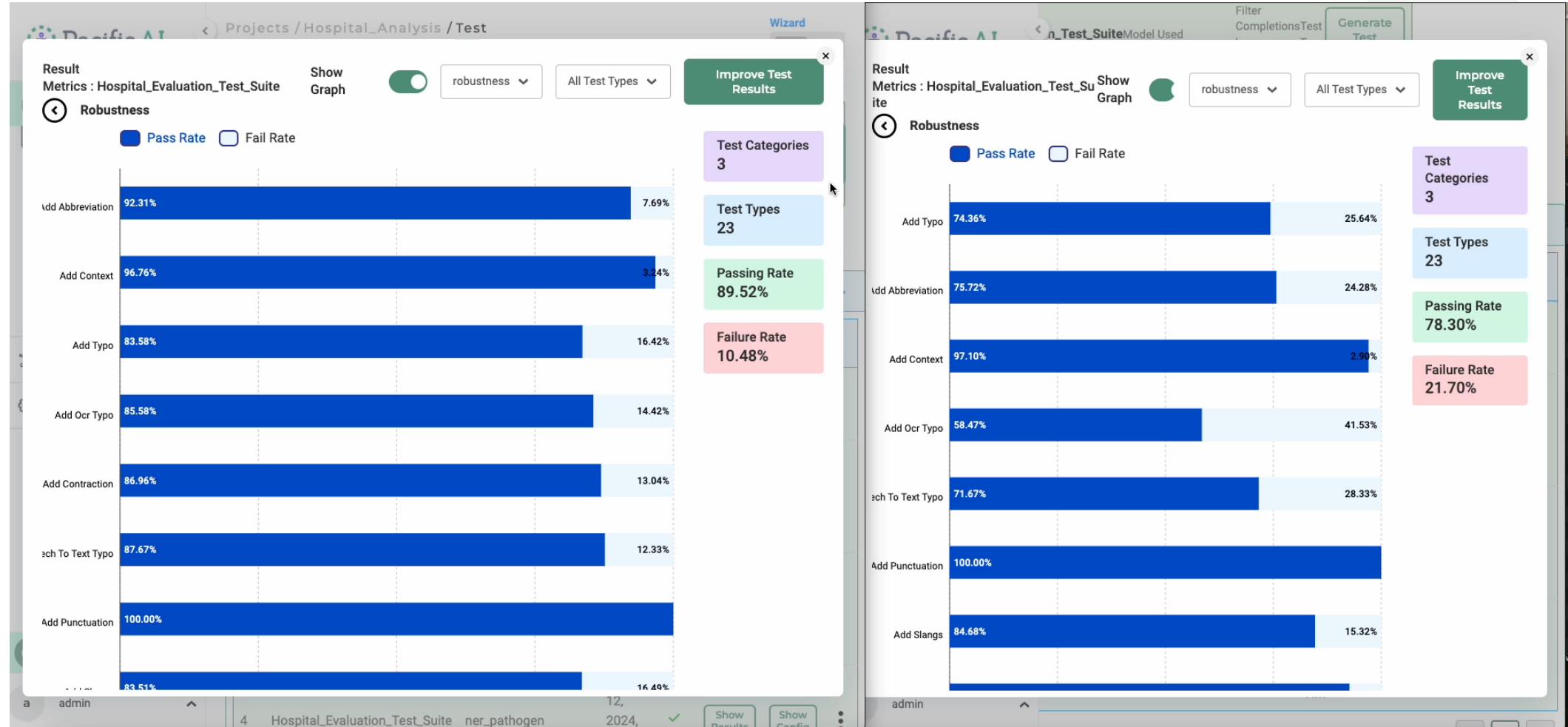
Max Proportion
Default Increase (30%) Apply to all

Generate Results

Tag your results i + Import

#	Sentence	Select All
1	COLI ^{Pathogen} .	<input type="checkbox"/>
2	So, John Had This Crazy Case Of Bacterial Pneumonia ^{MedicalCondition} Last Year.	<input type="checkbox"/>
3	Charles ^{MedicalCondition} 11	<input type="checkbox"/>
4	Last year Jake had to deal with a pesky little bug called E ^{Pathogen}	<input type="checkbox"/>

Visually Compare Improvements



Outline

- 1. Introduction to Text Annotation**
- 2. Manual Annotation**
- 3. Annotation Guidelines**
- 4. Project Setup and Management**
- 5. Preannotations**
- 6. Conclusions and further resources**

Conclusions – NLP Lab Facilitates



- 1 Manual annotation process
- 2 Best practices for efficient and coherent annotations
- 3 Project templates and project configuration to suite your project
- 4 Reuse pretrained resources for faster and more efficient results
- 5 Detect annotation errors with analytics
- 6 Train models to learn common tasks

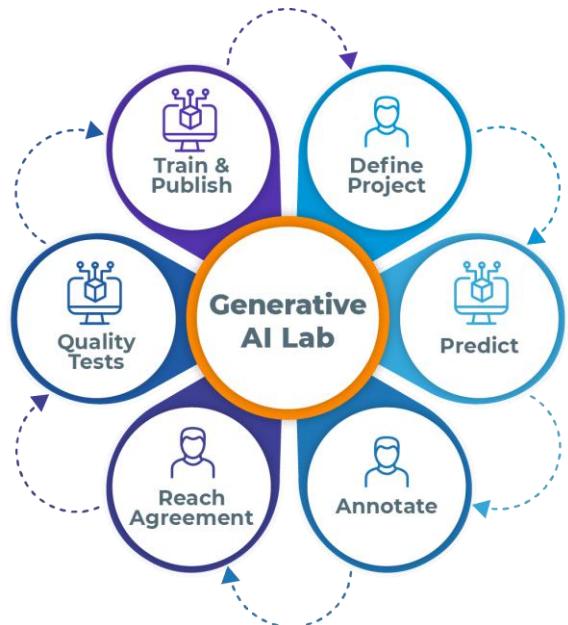
Quick Install

- [AWS Marketplace](#)
- [Azure Marketplace](#)
- [On-Premise](#)

Support

- [Slack](#)
- [Email](#)

Learning Resources



Docs

<https://nlp.johnsnowlabs.com/docs/en/alab/quickstart>

Tutorials

https://nlp.johnsnowlabs.com/docs/en/alab/step_by_step_tutorials

Blog

<https://www.johnsnowlabs.com/nlp-lab-blog/>

Questions?

dia@johnsnowlabs.com