JAVA LABELING SYSTEM

Software Requirement Analysis
V1.0

Lead Software Developers

Mahmut Hilmi ARIKMERT

Muhammed Enes AKTÜRK

Yunus Emre ERTUNÇ

Rabiul ISLAM

Asaf Talha GÜLTEKİN

Hamza TÜRKMEN

Kerim BOYACI

Customers

Murat Can GANİZ

Lokman ALTIN

Introduction

The purpose of this project is creating a data labeling system for different NLP problems using a console based program. Industry standard Object Oriented Designing and Programming will be implemented for this project. The chosen language is Java and agile software development methodology will be followed throughout the project.

Overview

Data labeling means assigning the several predetermined labels (class labels, categories, tags) to a group of instances (samples, examples, records, documents). In this system, user credentials , dataset and current dataset informations are hold in config.json file. Datasets contain set of labels , instances , labeling properties of instances and predetermined users to label specific dataset . All files are kept in json file format. There are two different labeling mechanism. If user enters this system by using his/her credentials , then improved labeling mechanism will be used to label instances whereas if user just enters system without using username and password , then system will automatically use random labeling mechanism. In each case, this system will allow user to label same instance more than once to detect the consistency of user. After each labeling operation, all assignments are updated to prevent losing information if system collapses. Thanks to this feature , this system can be stopped at any time. If same dataset will be used another run of this system , all previous assignments will be considered while calculating metrics. All system can be used via command line interface.

Functional Requirements

- **Usability Requirements**

    a. Number of user can be changed via config.json.
    b. The dataset can be determined via config.json.
    c. The dataset should be readable and easily accessible.
    d. Users that label dataset can be adjusted via dataset input files.
    e. The user informations will be predefined in json file.
    f. The mechanism to label instances will be changed according to the user type.
    g. User consistency can be calculated by using relabeling mechanism.
    h. The final label for instance can change at each label assignment.
    i. Each label assignments will be collected and used for calculating metrics.
    j. The actions should be logged and printed on the command line.
    k. The output file and the resulting report will be recorded in json files and also can be seen on the command line.

l. The output file and resulting report should be produced correctly when the program is terminated.
m. This system can be used in English language.
n. Type of error handling will be error message.

- **Implementation Requirements**

a. The program will be written in Java with OOP concept.
b. The program will take json files as input
c. The program will take json file as configuration file.
d. There will be no database integrity for this project

- **Physical Requirements**

a. The program can run with/without login.
b. The program is configured for adding users or datasets by the customers.
c. The program should be light and easy to operate through console.

Non-Functional Requirements

- **Performance Requirements**

a. The program should give the correct results for different cases as excepted. The program should response quickly.
b. After configurations ,program should run without any error.
c. If any problem or exception detected, it should be handled in a short time and recover the system.
d. Memory usage should be organized well and unnecessary parts in the code should be eliminated so that the program can run faster.

- **Supportability Requirements**

a. The program should be extensible and changeable according to customer's expectations.
b. The program should be adaptive for different kind of situations.
c. The program should have compatibility for different OS.
d. The program should be testable and if any error or bug is detected in the system, it should be caught in test method and fixed.
e. The program should be reusable.

- **Reliability Requirements**

    a. The program should perform as expected from command line.
    b. The program should produce results as accurate as possible.
    c. The program shouldn't lag or throw error halfway of the operation.

Use – Case

- **Scenario 1**

Step	**Actor**	**Action Description**
1	System	The login page appears
2	User	Users enters the following informations • UserName • Password
3	System	System checks user credentials. If they are correct, then system allows user to enter the system. Then system asks user to select labels for corresponding instance.
4	User	User selects labels by typing ids of each label with ",".
5	System	System labels corresponding instance with these labels.

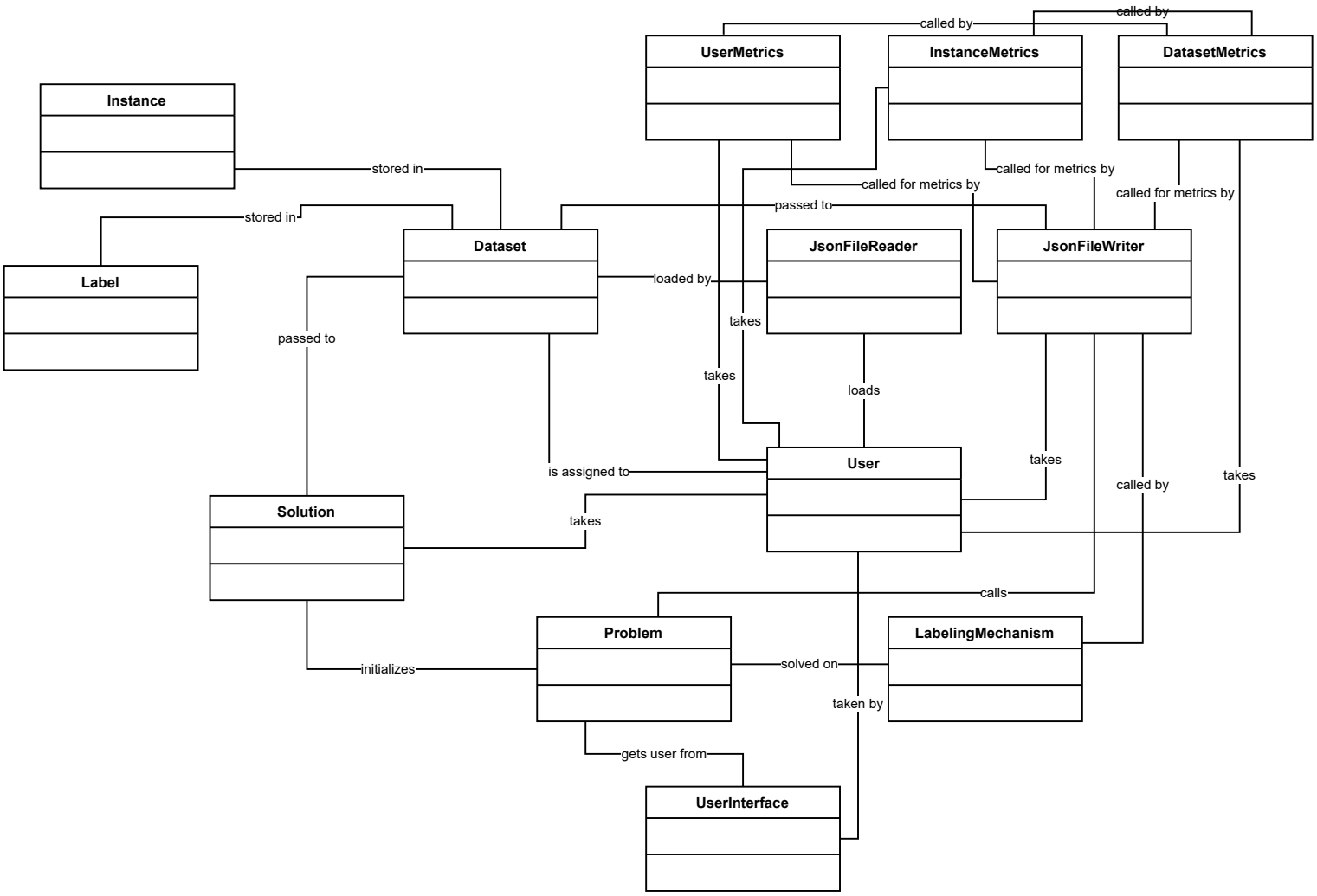
- **Scenario 2**

Step	Actor	Action Description
1	System	The login page appears
2	User	Users enters the following informations • UserName • Password
3	System	System checks user credentials. If they are not correct, then system asks UserName and Password again.

- **Scenario 3**

Step	Actor	Action Description
1	System	The login page appears
2	User	Users enters the following informations • UserName • Password
3	System	System checks user credentials. If user left both UserName and Password blank , then system checks bot type and starts labeling according to the mechanism of corresponding bot type.

Domain Model



Glossary

NLP – Natural Language Processing
Sentiment Analysis – Understanding human language's emotion
Label – Categorizing human language
Console – Command Line or a black window that is seen in windows/linux operating system. User needs to type through keyboard to provide necessary instruction to the program instead of normal mouse click and navigation button.
Json – JSON stands for JavaScript Object Notation. JSON is a lightweight format for storing and transporting data