Software Requirement Analysis For Data Labeling System

1. Overview

This document includes brief description, list of functional and non-functional requirements, approaches for the specification of software requirements of our Data Labeling Project. It is divided into five clauses:

In Scope part, we explained the purpose and content of the project.

In Glossary part, we listed the keywords that used in project, project guide or words associated with the project.

In Functional Requirements part we have listed requirements are essential to building our program.

In Non-Functional Requirements part we represent a set of standards that specify our operation of a system. But we should note that non-functional requirements can be just as important as functional ones.

In addition to these, we did Use Case and created Domain Model.

2. Scope

Scope of first iteration&general glance:

The purpose of our project that we want to do is to ensure that an organization's artificial intelligence categorizes the feedback sent by its customers as positive, negative or neutral. At the same time, we can use the same system for labeling by category such as articles/breaking news/interview/critisim from magazine or news website according to whether they are in such as culture, sports, economy, politics, art, health categories or not.

In a result, this project gives us the opportunity to avoid confusion in labeling and categorizing and we can manage to handle classification problems easily.

Scope of the second iteration:

The second iteration of our project aims adding reporting functionality for user performance and labeling operation for a particular dataset. We should collect statistics for users, compare users in the context of a particular dataset or globally and calculate metrics for instances in the dataset that are labeled with many users. In that way, we will obtain information about quality of the data labeling and the quality of the users.

Scope of the third iteration:

The third iteration of our project aims in addition to the bot users, we need to add a user interface for human users. These human user should be able to interact with our system to choose possible labels from a list to assign an instance. The old aims of our project is keeps going to be needed.

3. Glossary

Instances: Positive, negative or neutral comments-specified-.

Label: Tagging.

Dataset: Collection of datas.

Dataset ID: Dataset's unique identifier

Dataset Name: Specializes the data set to be processed.

Label ID: Put in order the instances

Label text: Sets connection with pointed instance.

Multi-User System: The system that used by more than one user.

Json file: An open standard file format and data interchange format that uses human-readable text to store and transmit data objects consisting of attribute-value pairs and array data types.

.jar: Jar is a package file format typically used to aggregate java class files and associated meta-data, text, images etc. İnto one file for distrubition.

Named Entity Recognition Problem: A subtask of information extraction that seeks to locate and classify named entities mentioned in unstructured text into pre-defined categories.

Classification: A process of categorization that ideas and obhects are recognized, differantiated and understood.

Classification Problem: Separating the data belonging to more than one given category and assigns them to different previously known groups.

Command line: User interface that navigated by writing commands at prompts instead of using a mouse.

Performance metrics: Figures and data representative of an organization's actions, abilities and overall quality.

4. Functional Requirements

- Our object oriented modeling must support multiple options in which words in a document can also be labeled.

- It must be a multi-user system.

- In every data sets, the entire text is labeled, as in the sentiment classification. So we have to consider the whole text.

- Program takes input from json file which includes meta data such as the set of labels even tough it comes randomly.

- We must consider that there are several labeling mechanisms.

- The dataset file must include the set of labels, max. number of labels to label for an instance, and a set of instances.

- A user can label many instances. We can understand it according to given input.

- An instance can be labeled by one or more users.

- Our model implements word or phrase tagging in the first iteration.

- Our code must be changeable.

Additional requirements arrive (naturally, requirements always goes and increases with new demands, respectively) with 2nd iteration:

-We must allow users to label a spesific instance more than once to measure user's consistency in a dataset.

-Our user.json has to be evaluated as a config.json file that must include 3 different users at least. We had 2 before.

-Our job must be flexible that it should be changed/added more users by using config.json and be easy to add datasets to config.json.

-It has to satisfy assigning any number of existing users in config.json to a spesific dataset for labeling condition.

-When our code is being stopped, it should update output files and log files after instance is labeled by a user so that current results can be seen. It means that we have to compute metrics after each label assignment occurrence.

- It has to satisfy the condition that only the indicated dataset (which is shown by currentDatasetID) can be labeled.

-Users have to be assigned to the datasets.

-There must be limitation about user's labeling in dataset. If one user labels all instances in a dataset, in following runs, he/she can not label instances in same dataset again.

-Datasets must be listed along with their completeness percentage. In order to determine the completeness percentage, we have to divide the number of labeled instances to the total number of instances in dataset.

With 3rd iteration, more additional requirements arrive:

- There must be many human users. We need to store human users' credentials in our config.json with ConsistencyCheckProbability.

- When our program runs, it must ask for a user name and password to entry. If the user name and password matches correctly one of the user credentials in our config.json, then our user will label instances in the current dataset one by one.

- If user name and password does not match any user credentials in our config.json, we display the error message says our user that wrong user name / password message and ask user name and password again.

- If there is no entry -blank- for user name and password, then our system must assign bot users to set for current dataset in our config.json will label the current dataset automatically.

- We must add more and different bot user types. In addition the our current random labeling mechanism and user, we must add a rule based labeling mechanism and user that labels sentiment classification instances based on our choice.

5. Non-Functional Requirements

- Design should be extensible in iterations.

- Our work should work with minimal changes.

- Our work should be easy to give a maintenance.

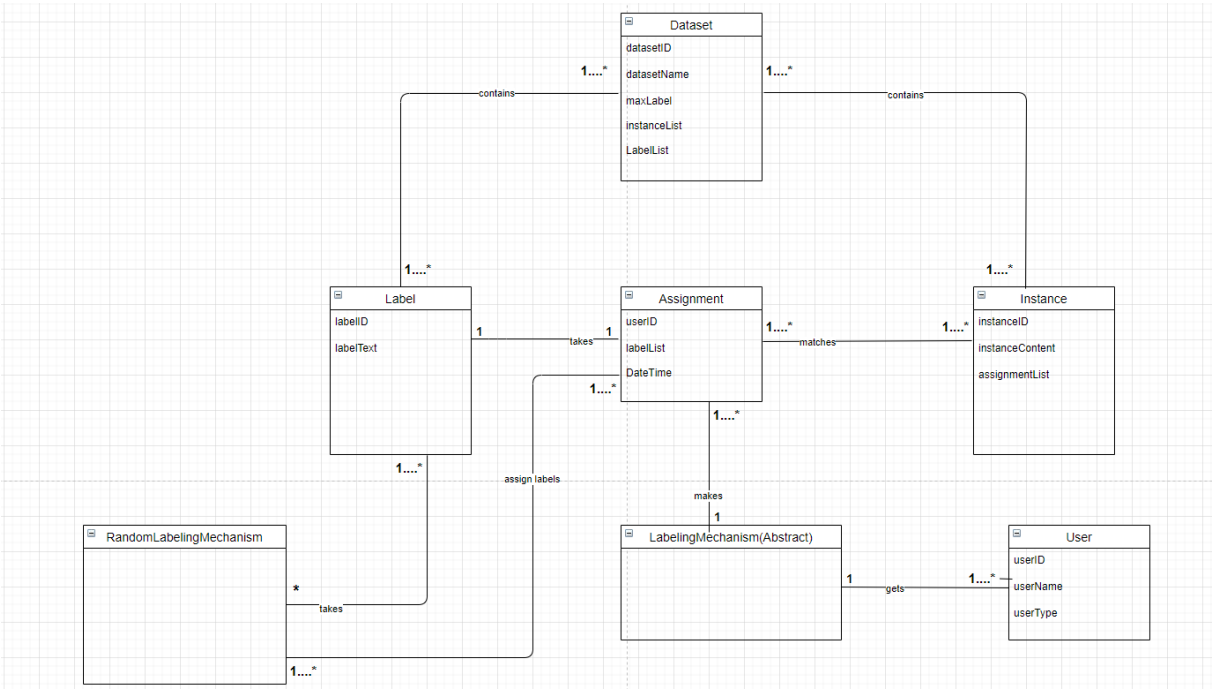
Use Case: E-Commerce Comment

Actors: Customer,System

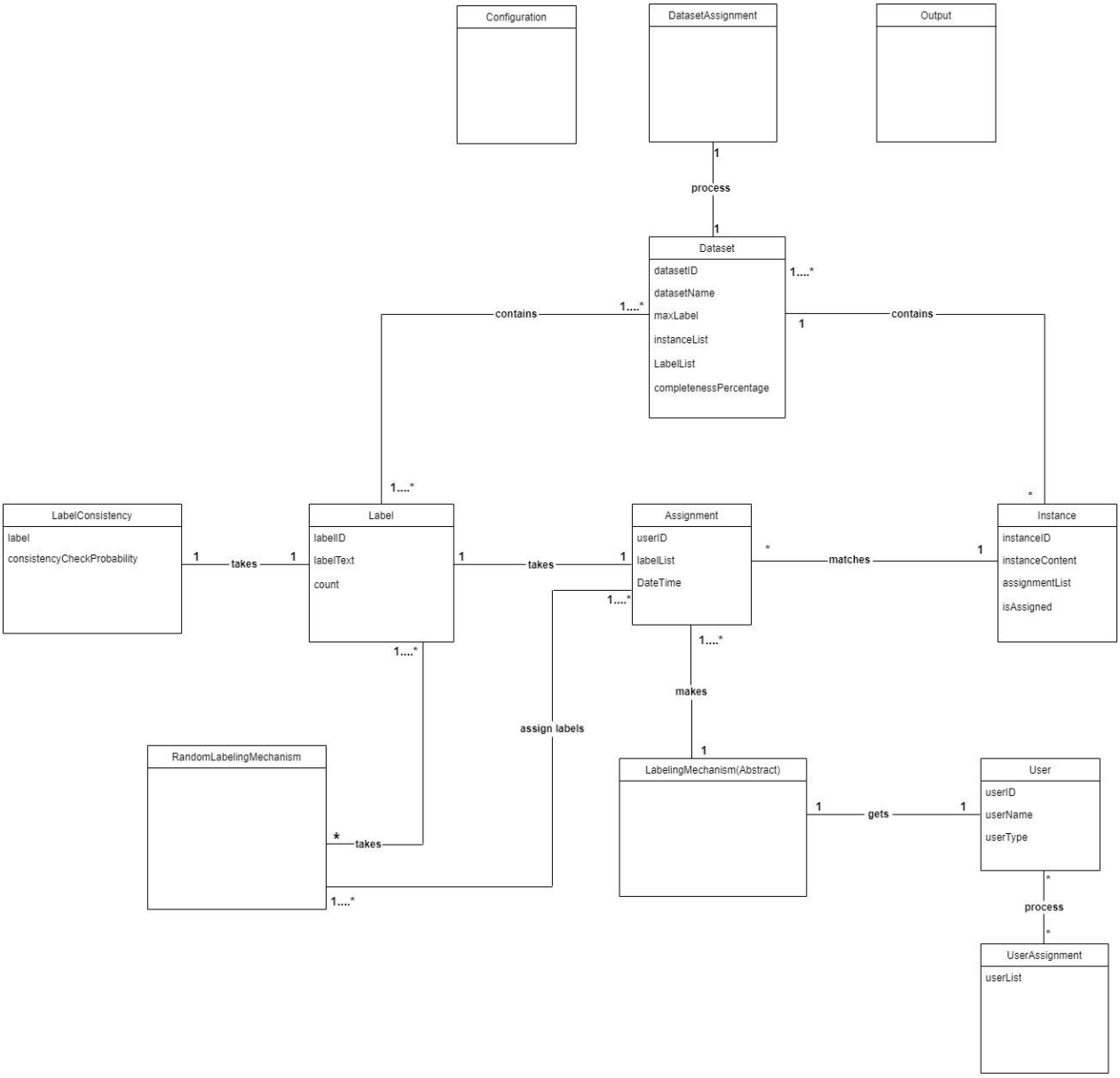
Successfull Scenario:

1. A service is provided to the customer.
2. Customer checks the service received.
3. Customer comments on the product he/she bought.
4. Customer shares this comment with the e-commerce site.
5. The system classifies these incoming comments as positive negative or neutral

Domain Model



With 2nd iteration, our domain model changes:



With 3rd iteration, again, our domain model changes: