**Bugopedia**

**Problem Domain**

System Integration and Validation, Defect Management

**Problem Description**

In SIV, due to intensive validation a huge number of defects are filed every week which needs to be addressed with minimal turnaround time. With huge number of incoming defects, their efficient management such as de-duplication, triaging, and mapping defect to a correct failing usecase are often resource and time consuming. The most challenging part in identifying a duplicate/similar bug or to map to a correct failing usecase is the free unstructured text which makes the data highly noisy. On top of it, most of the times the labeled data is very less and manually labelling them is a humongous task in terms of time and resources consumption. Thus, we need an efficient unsupervised or a semi-supervised approach for the following two tasks I) identifying similar defects from the corpus and II) classifying the correct failing usecase of a defect.

~~The number of client platforms and the complexity of features are increasing rapidly. System integration and validation teams today need to deal with multiple platforms, a vast array of hardware and software components, and several operating system releases. As a result of intensive validation, huge number of defects are filed every week which needs to be addressed with minimal turnaround time. A typical PC platform provides a variety of usecases for an end-user ranging from audio playback, video playback, streaming, power management scenarios, communications, sensors, camera, security and much more. A failure in any of the usecases is considered as a defect. To map the correct defect to a particular failing usecase, we need to extract the information from the defect data and accurately identify the failing usecases.~~

**Problem Impact**

In SIV X% of filed bugs are duplicate of existing bugs and a huge number of bugs are identified into incorrect failing usecase. As per our estimate, it takes around $X­1K for successful resolution of each bug. Thus, with the bugs flowing into our system and the achievable accuracy, approximately $X2 million can be saved with the proposed solution and avg TAT can be reduced by X4 %.

~~If we can accurately classify defects into usecases, the solution could be used to identify failing usecase themes and help debug and triage team reduce the turnaround time, identify top failing usecase themes, use the solution with existing pipeline to find similar defects across platforms and map impacted testcases and feature requirements for all the defects filed. This in turn will help reduce huge amount of debug and triage turnaround time and help identify similar and duplicate defects which take up lot of resources and time.~~

**Dataset Description**

**1) Defect Similarity Data:**

This data set consists of 100,000 defects titles. Each defect has the following fields

* *Title*
* *ID*
* *Platform*
* *Domain*

The *Title* field consists title of the defect on avg 2-3 sentences or around 25 words. *ID* is unique identification of a defect. *Platform* field denotes the platform in which the defect is filed and *Domain* denotes the domain in which the defect is filed. The data is gathered from bug repository known as HSDES. It has defects from last 4 years of all the major platforms in this time range . There are around 10% defects that are labeled with it duplicate defects IDs. There can be more than one duplicate for a defect. The labelled can be used as a validation set in case of unsupervised training or a part of it can be used a seed training in case of semi supervised training.

2) **1) Defect Usecase Data:**

The data used for this task consists of the defect’s titles labelled with corresponding usecase labels. All the field are same as the previous data. In addition it has a *Usecase* field that maps the defects to its failing usecase. These usecase are defined and manually labelled by a team of domain experts. There are 50 distinct usecases and around 25000 defects have been labelled with the corresponding usecase labels.

**Current Solution**

Currently we are using a hybrid model for defect similarity task which is a combination of first identifying the top usecases and then finding the similar bug within the identified usecases. We use a traditional supervised model for usecase prediction and a vector space model to find the similar defects.

**Desired Solution**

As we have a very less data for a supervised learning in defect similarity task and the labelling defects leads a heavy cost in term of time and resources thus we are looking for sustainable un-supervised/semi-supervised learning solution which require minimal to no manual intervention.

The metric the we are using in task 2 of finding the usecase is accuracy. Currently, the model accuracy is around 75%. For this task we are looking to achieve an accuracy of >80%

**Solution Deployment**

We currently have the system implemented with our baseline solution, and we will be able to deploy the new model if it achieves the needed accuracy.