Testing Plan

Design, Build, and Enrich Machine Learning Model for Natural Language Processing

Sponsored by DXC Technology

Chandra Kamalakantha

Jonathan Lawrence
Jeremiah Ramilo
Jacob Wilson

ABSTRACT

This document covers our plan for testing our project *Design, Build, and Enrich Machine Learning Model for Natural Language Processing*. This document will discuss our test cases, the traceability of test cases, and our techniques for test generation.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

The purpose of this document is to provide documentation for test of our project "*Design, Build, and Enrich Machine Learning Model for Natural Language Processing".* Our test cases, traceability, and techniques for test case generation will be covered in this document.

REQUIREMENTS/SPECIFICATIONS-BASED SYSTEM LEVEL TEST CASES

TC-0: To test the functional requirements we will submit several client tickets written in plain english and test what is returned. We will judge the knowledgebase articles returned based on relevance to keywords and relevance to the test client ticket. This test case will also judge how the system handles natural language and if the system improves over time.

TC- 1: The database will be tested for its ability to handle NoSQL injections

TC- 2: Support personnel will search for a knowledgebase article and await the response time of the database.

TC- 3: The database and search enrichment model will be tested for documentation and ability to increase or decrease functionality

TC- 4: Support personnel will search for a knowledgebase article and view for the relevancy of the returned articles

TC- 5: The system will be monitored regularly and have system data to measure uptime or the presence of any failures or crashes

TRACEABILITY OF TEST CASES TO USE CASES

	TC-0	TC-1	TC-2	TC-3	TC-4	TC-5
UC-0					X	
UC-1			X			
UC-2	X					

TECHNIQUES FOR TEST GENERATION

To generate test cases we looked at how the system would be used on a day to day basis and tried to make test cases that would reflect that usage by basing them of use cases. We also made a few test cases to test technical requirements, such as uptime. Most of our test cases are black box test cases because the most important outcome for this project is how it functions for the end user. We did however generate a few white box test cases to test non-functional requirements like security and up time. Our criteria for judging test cases was whether or not they covered a functional, non-functional requirement, or a use case. If the test case did cover one of those it was considered a useful test case.

EVIDENCE THE TEST CASES, DOCUMENT HAVE BEEN PLACED UNDER CONFIGURATION MANAGEMENT

We are using Github for requirements configuration management.

The link to our Github is: https://github.com/orgs/Senior-Design-F19/teams/dev.

REFERENCES

N/A