University of Melbourne

2018 CIS MASTERS -COM02081

Software Development Project

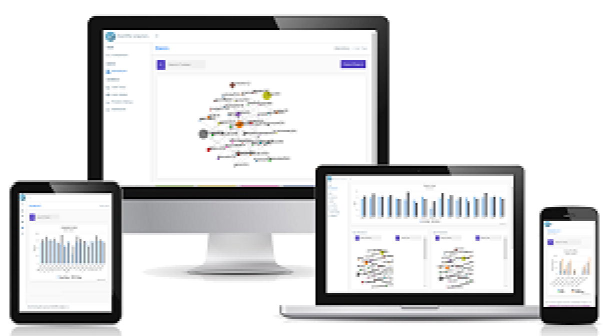
SWARM Analytics Dashboard

-Chat Visualization and Sentiment Analysis

Author:
Muhammad Umair, 863579

Supervisor:
Richard Sinnott (Director
eResearch, Melbourne University)

October 28, 2018



THE UNIVERSITY OF
MELBOURNE

Contents

Acknowledgements I would like to thank *Richard Sinott (Director eResearch, Melbourne University)*, for supervising the project and assisting in every possible way to make this project a success. *Yao Pan (Software Developer, eResearch, Melbourne University)* for providing SWARM data-sets and schematic elaborations, as well as SWARM specific use-case guidance. *Ariel Kruger (Post Doctoral Fellow -CREATE SWARM, Melbourne University)* and *Elizabeth Silver (Post Doctoral Fellow -CREATE SWARM, Melbourne University)* for helping me understand the principles behind 'SWARM Online Platform' and its core context.

SWARM Analytics Dashboard Website `http://115.146.93.213/swarm/`

Youtube Link (Demo) `https://youtu.be/3lR8yiCnnH4`

Youtube Link (Deployment) `https://youtu.be/DeHD-PgK_bk`

Github Link (Code) `https://github.com/mumairj/SWARM.git`

Abstract

Data is the new drug. With its ever increasing volume, it has enabled the fields of data mining, analytics and deep learning, to excel. These domains have pushed paradigms such as evidence based reasoning (EBR), to a promising extent. EBR is the process of thinking about something in a logical way in order to form a conclusion or judgment. The Smartly-assembled Wiki-style Argument Marshalling project lets team-based groups of users to produce evidence-based reasoning.'SWARM Analytics Dashboard' is a derivative of 'SWARM Online Platform'. Its core functionalists include sentiment analysis and interaction pattern visualizations over chat performed on the platform.

1 Introduction

The enormous influx of data over past few decades, has lead to an extensive research in the field of machine learning, artificial intelligence and advanced data mining. This in turn yielded fruitful results to derive knowledge and understanding by leveraging large scale infrastructures and computational paradigms such as cloud computing or high performance computing. However, the ability for reasoning has advanced at a relatively slower pace.

The Smartly-assembled Wiki-style Argument Marshalling project (SWARM), is one such online platform, which allows groups to generate evidence-based reasoning [1]. As such, the data accumulated as a result of SWARM platform utilization, can provide tremendous insights. These insights can be used to not only improve the platform, as a whole, but also decipher user behaviour and interaction patterns. This project 'SWARM Analytics', aims to facilitate users in studying these patterns via a web application. After discussing a brief background on existing works and SWARM itself, the architecture of analytics dashboard has been elaborated. Following which, some scenarios which can be visualized through dashboard, have been presented, with a critical analysis on them. Finally, the deployment automation and future works have been presented, along with a conclusion.

2 Background

Advances in the field of data-mining and knowledge discovery have proved to be the pinnacle of innovation in twenty first century [2]. This has given rise to modern research in statistics, databases, machine learning, and artificial intelligence. Under the umbrella of these domains, evidence-based reasoning (EBR) is another emerging dimension, which is under the spotlight lately [3]. In recent times, EBR is bieng specifically targeted to facilitate the demands of 'Intelligence Community' [4]. The SWARM platform aims to capture this paradigm by enabling users to interact with each other, discuss on a problem, provide evidences and come up with a feasible solution [1]. Hence, the platform naturally generates user interaction patterns.

A vast amount of research effort has been put into the field of human computer interaction (HCI) [5], including studies leading towards interaction pattern analysis [6]. Not only this, but also scrutinizing a user's emotions or opinions i.e. sentiment analysis, has contributed to excel the application of behavioral pattern analytics [7]. Therefore,'SWARM Analytics Dashboard' administers towards the idea to analyze user-interaction over 'SWARM Online Platform', specifically, by determining the sentiments of users and their behaviours throughout the SWARM life-cycle. Following sections provide a deeper insight towards this eco-system.

3 SWARM

Project SWARM was commenced in 2017 as a part of the US Intelligence Advanced Research Projects Activity (IARPA) funded Crowdsourcing Evidence, Argumentation, Thinking and Evaluation (CREATE) Program.

3.1 SWARM in a 'Nutshell'

It is an online platform whereby groups of people are assigned a problem for which they devise a solution. A problem in SWARM is called 'question', whereas a solution to the problem, is typically a 'hypothesis', which results in a 'report'. Initially, the users draft their proposed responses or hypothesis individually, after which they open up their work for review. Once users publish their proposed solution to the problem, other users can comment or provide suggestions for improvements. It is an iterative team-based process, during which users can also contribute resources which facilitate team-work eg. diagrams, tables, or images, which may or may not form part of the final report. In SWARM, each problem has a life-cycle, at the end of which, the highest rated report is auto-selected to be submitted on team's behalf. Figure 1 presents an overview of the SWARM work-flow hierarchy. Team types in swarm are categorized into three domains.

- GP – General Public
- AA – Agency Analyst (A professional analyst participating on SWARM with their Agency)
- PA – Professional Analyst (A professional analyst participating on SWARM independently)

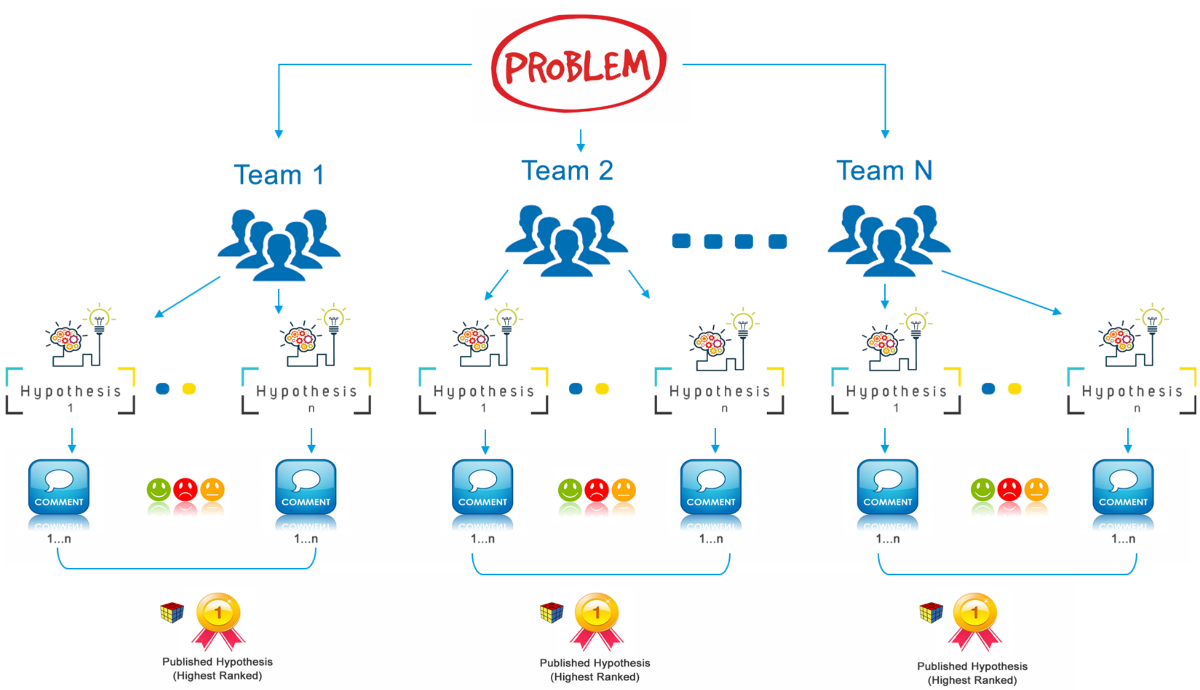


Figure 1: SWARM Work-Flow Overview

3.2 Internal Ratings

SWARM ratings are used to rate a report as per its readiness. A detailed specification for these ratings can be found at [8]. Teams use these ratings for the readiness of their report or solution. The readiness score id out of 100%, 75% of which constitutes *'Reasoning'*, and 25% of which constitues *'Communication'*. Reasoning is further an aggregate of five metrics i.e. *'Completeness'*, *'Correctness'*, *'Logic'*, *'Evidence'* and *'alternatives'*. Communication, on the other hand, is an aggregate of *'Clarity'* and *'Format'*. An example of rating readiness rating calculation is given below in table 1.

Table 1: Sample Ratings Division

Type of field	Field name	Score	Average for field type	Readiness score
Reasoning	Completeness	56	(56 + 88 + 70 + 81 + 20) / 5 = 63	(63 * .75) + (65 * .25) =63.5
	Correctness	88		
	Logic	70		
	Evidence	81		
	Alternatives	20		
Communication	Clarity	70	(70 + 60) / 2 = 65	
	Format	60		

3.3 External Ratings

Once the reports have been submitted, external reviewers check published reports provided by each team. These reports are rated by externals based on 8 criterion metrics, each of which is out of 4. Hence, the total points out which a report may be rated, is 32. This is to indicate the readiness of report from the perspective of other qualified individuals. These are generally PhD professionals.

4 Architecture

The architecture entails a typical three layer system, comprising the following.

- Presentation Layer
- Logic Layer
- Data Layer

4.1 Presentation Layer

The layer facilitates users to access 'SWARM Analytics Dashboard' via a website url on any standard web-browser. A bootstrap framework has been used for front-end development along with traditional HTML/CSS/Java-script languages. A JS library, 'Highcharts', as well as an open source library 'D3' has been used for graphs and charts. 'D3' has been specifically used for interaction graphs. 'Highcharts' is utilized for all other types of analytical illustrations such as bar-charts, line-charts, pie-charts, among others.

4.2 Logic Layer

This layers serves as an intermediary between front-end and data-source. An 'Apache' server, running PHP, hosts the web-application. It receives 'Ajax' calls from front-end, queries database, processes results and provides a response to front-end in JSON format. The reponse then gets interpreted by respective JS libraries for visualizations. It must also be noted that a python listener also sits on this layer. It listens for any incoming record, specifically on the chunk table, and processes it. The python script checks if the variant of record is a 'comment'. If so, it performs sentiment analysis on it, determines if any user was tagged, dumps data into respective tables i.e. tags, sentiments and status.

4.3 Data Layer

A PSQL database has been used for storage purpose. The database follows a schema which is an exact replica of production database i.e. SWARM online platform database. The only exceptions are three additional tables and views. The entire logic for analytics is abstracted out in views to maintain simplicity of the system.

4.4 Diagrammatic Representation

Figure 2 illustrates an overview of the system in detail. Starting from bottom the front-end (presentation layer) communicates with analytics server (logic layer), which in turn queries a psql database (data layer). While designing the architecture it has been kept in view that analytics server will run a replicated instance of production server database. To ensure that the production remains unimpacted, analytics server runs in complete isolation from production server. To reflect a consistent state at front-end, the python listener updates a table with time-stamp of the last record it processed. Hence, all views producing outputs on front-end, always check the last processed time-stamp to ensure consistent results.

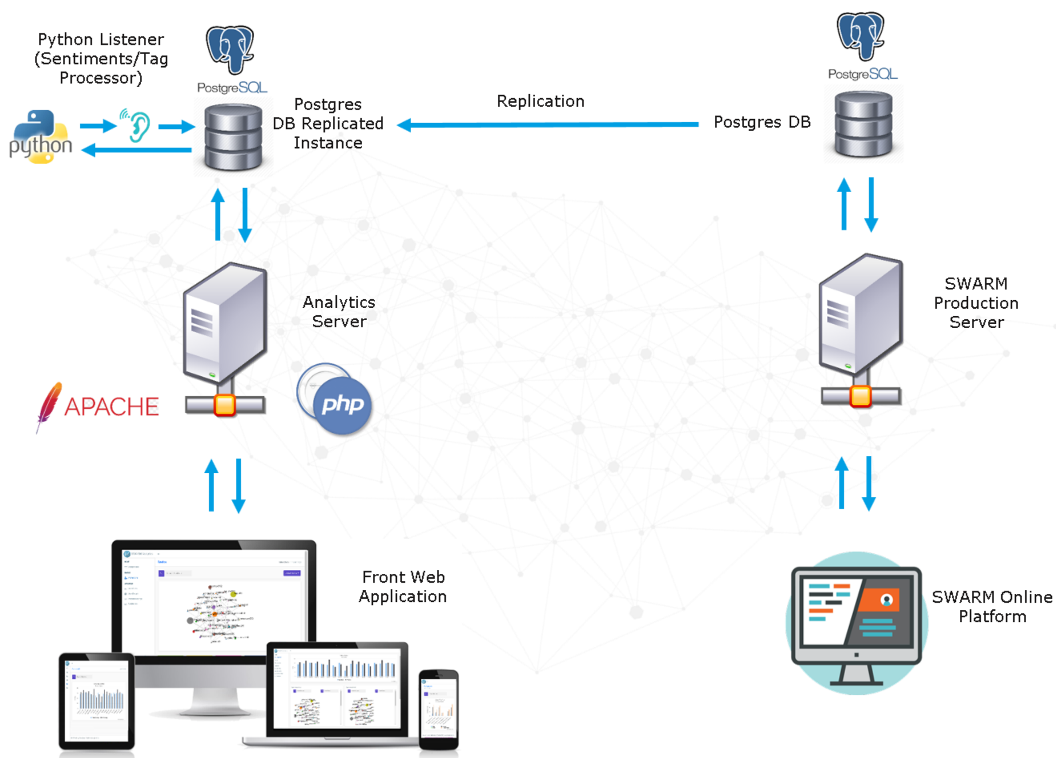


Figure 2: SWARM Analytics Dashboard Architecture

5 Analytics Dashboard

The natural flow of investigating the statistics produced by SWARM platform adheres to quite a few fundamental use-cases. The most prominent ones of which are top down and bottom up approaches. Sections 5.1 and 5.2 elaborate these approaches, after which, each web-page of the dashboard is presented in sections 5.3 to 5.8 in a top-down fashion (5.1). It must be noted that the flow for bottom-up analysis (5.2) is vice-versa.

5.1 Top Down Analysis

A problem presented in SWARM is assigned to multiple teams. As each team comes up with one or more possible solutions or hypothesis for the given problem, each of the hypothesis is then rated by respective teams, internally. Only the highest ranked hypothesis gets published, which is then rated by external markers. A classic analysis use-case would be to compare the internal ranking versus external ranking for each team, for a given problem. Team interactions can be further drilled down to check the number of people who tagged each other, most or least tagged person, number of positive, negative or neutral comments. Based on a particular user's behaviour, their profile can be viewed to observe the sentiments, comments, hypothesis, descriptions and tags description, along with their activity time-line.

5.2 Bottom Up Analysis

Similarly a typical bottom up analysis would initiate from a particular user. It can be determined that whether or not a particular user contributed in any problem in which they were assigned. Also, the team to which they belong in a particular problem, their activity time-line, posts distribution and sentiments. Their importance can also be measured by the weight of incoming pings to them, along with determining the people they tagged. Finally, the internal and external rating for the problem on which they were assigned, can be inspected.

5.3 Interactions

The 'Interaction' page serves as the homepage of 'SWARM Analytics' dashboard. Figure 3 shows a graph for interaction between all team members assigned to the problem 'How did Arthur Allen Die?'. Each node represents a team-member and the weight of node indicates the number of times a user was tagged by any of the other team member. The directional arrows show tag-direction for instance 'Bilby374' was tagged by 'crocodile738'.

It must be noted that the problem search input are is an intelligent predictive text indicator (Figure 3 top-left). Also, once a problem has been selected, the drop down menu located at the top-right position in Figure 3, gets auto-populated with teams that were assigned on the selected problem.

Similarly, figures 4 and 5 show the drop-down menu for team selection along with a filtered result of team interaction.

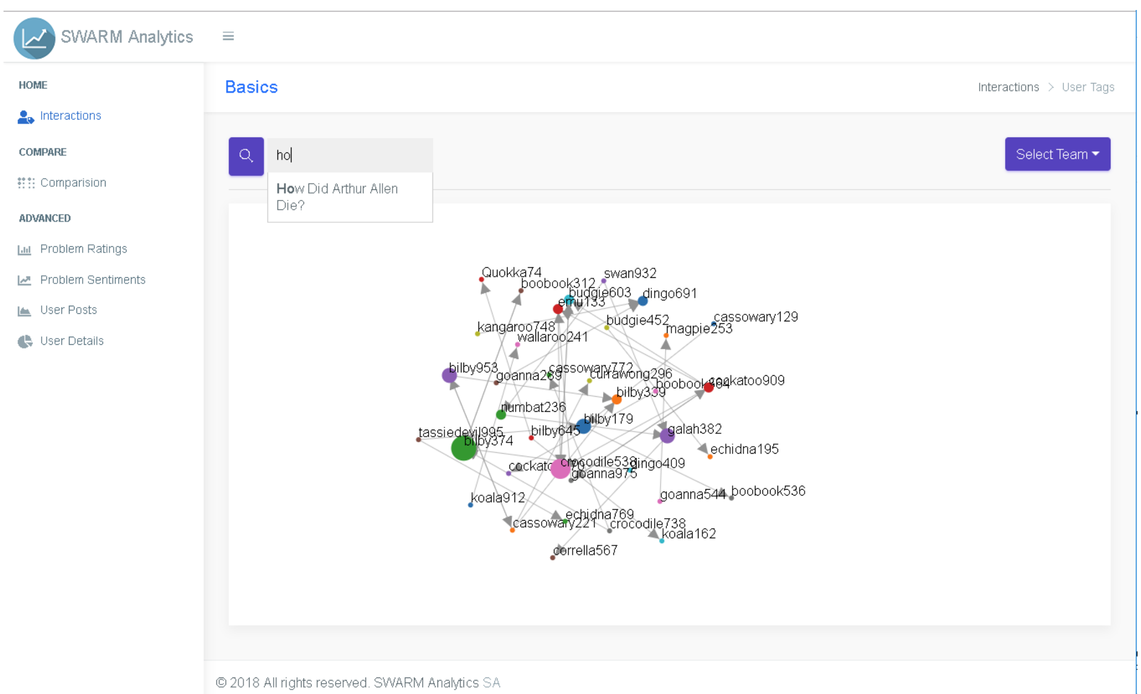


Figure 3: All Teams Interaction

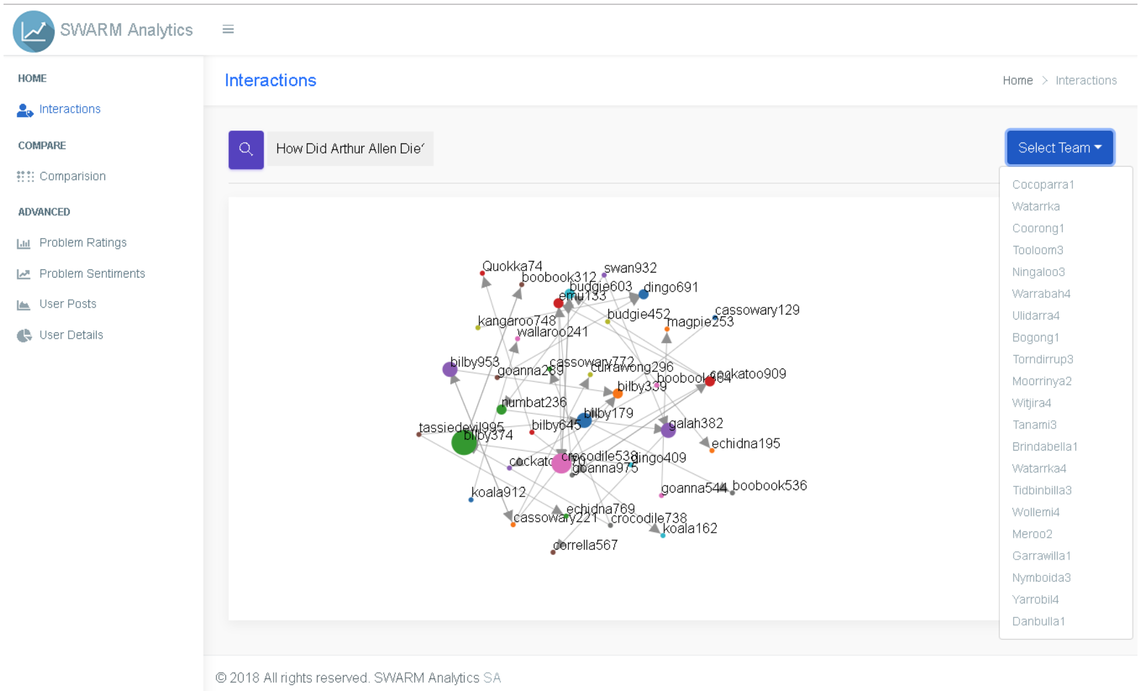


Figure 4: Drop-down Menue for Team Selection

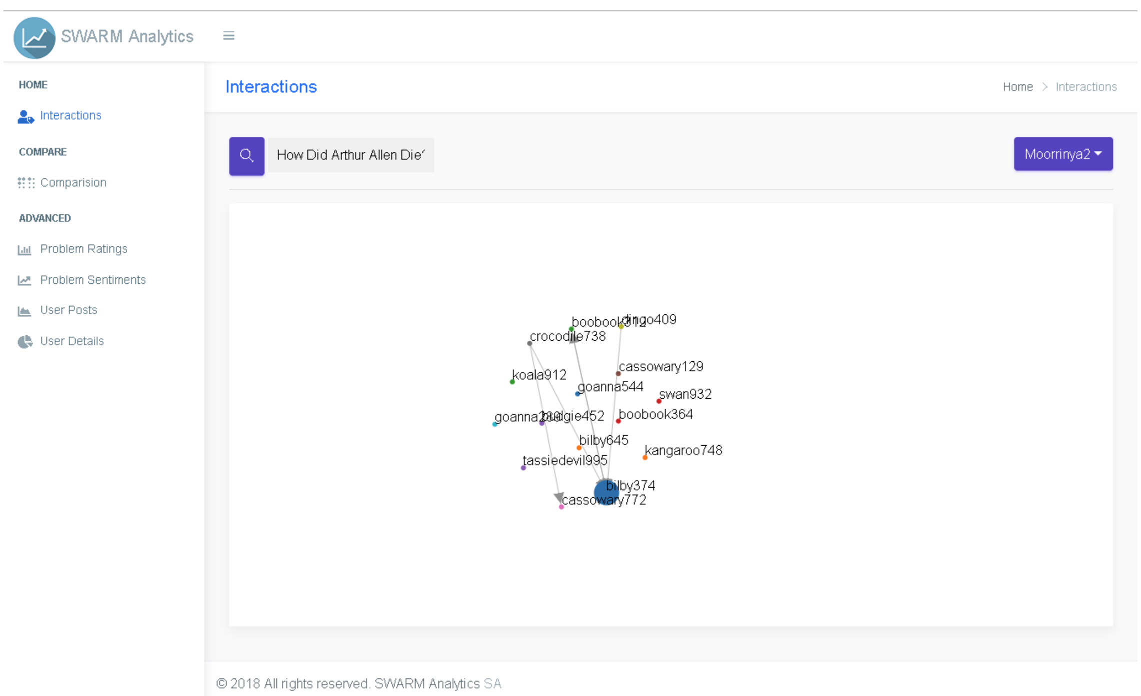


Figure 5: Team Filtered Interactions

5.4 Problem Ratings

Figure 6 shows a bar chart graph with teams on the x-axis and scores or ratings, scaled to 100 percent, on y-axis. It can be accessed on page 'Problem Ratings'. It is one of the most pivotal pages which can serve as the starting point of a top-down analysis, followed by 'Interactions' section 5.3 and sections 5.5 to 5.7. An internal rating is the one which given by teams, to their hypothesis, such that the highest rated hypothesis or solution gets published. External rating on the other hand is given by external professionals who rate the published hypothesis of a team based on another rubric. The details of these rubrics have been specified in section 3.

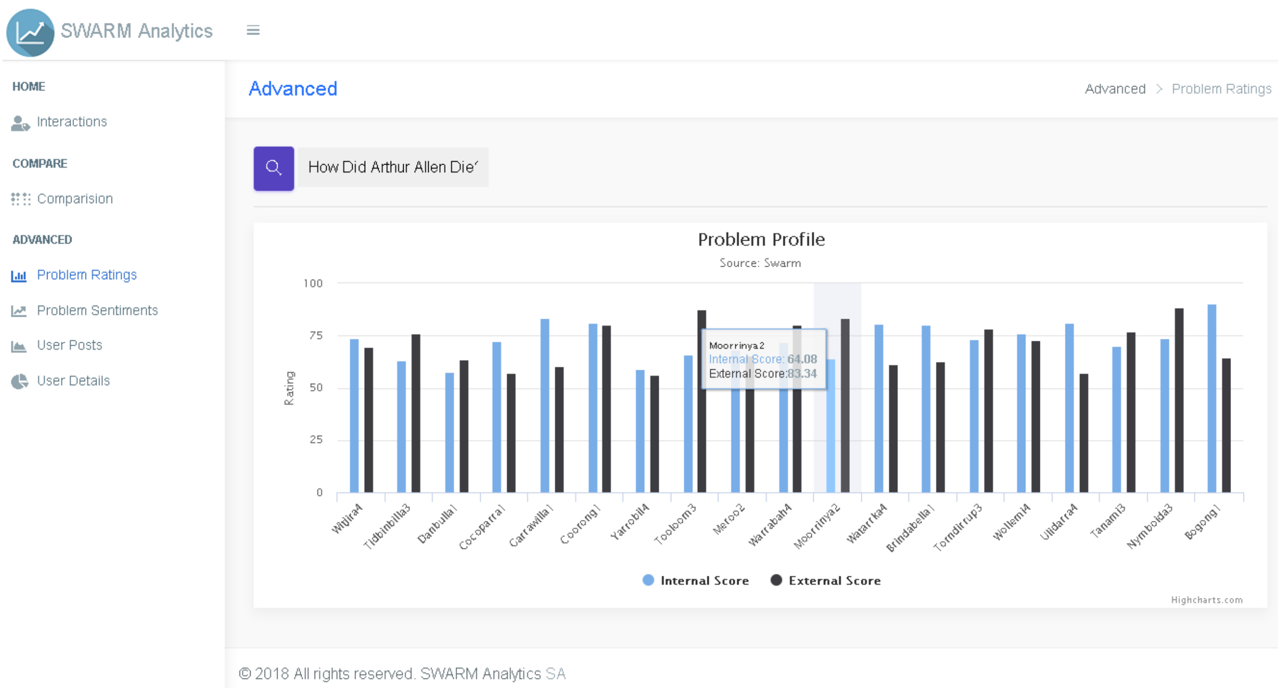


Figure 6: Problem Ratings (Internal vs External -Team-wise)

5.5 Problem Sentiments

A sentiment analysis has been performed on each comment put in by any user on a particular hypothesis. These can be observed in page 'Problem Sentiments'. A line chart, in figure 7, with date on x-axis and number of comments on y-axis is presented with lines showing number of positive, negative, neutral and total comments. The legend on left hand-side is interactive i.e. on-click it enables or disables the lines on chart. This makes it easier to compare the graph for certain type of sentiment versus another type.

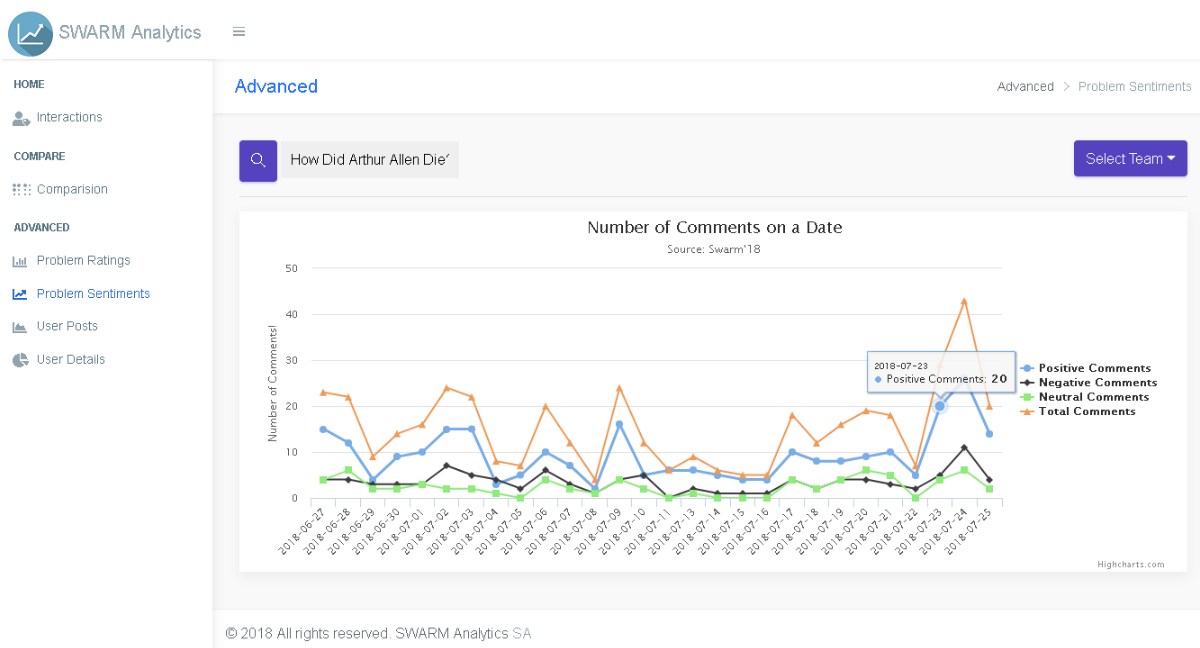


Figure 7: Problem Sentiments (Positive/Negative/Neutral/All)

Likewise figure 8 below presents a team-filtered view of the same graph. Once the problem is selected from top-left, the team drop-down is auto-populated. In the example above 'How Did Arthur Allen Die?' is selected as the problem, team 'Moorrinya2' is selected in the team filter and 'Neutral, Total Comments' have been disabled. A comparison of positive versus negative comments can be observed.

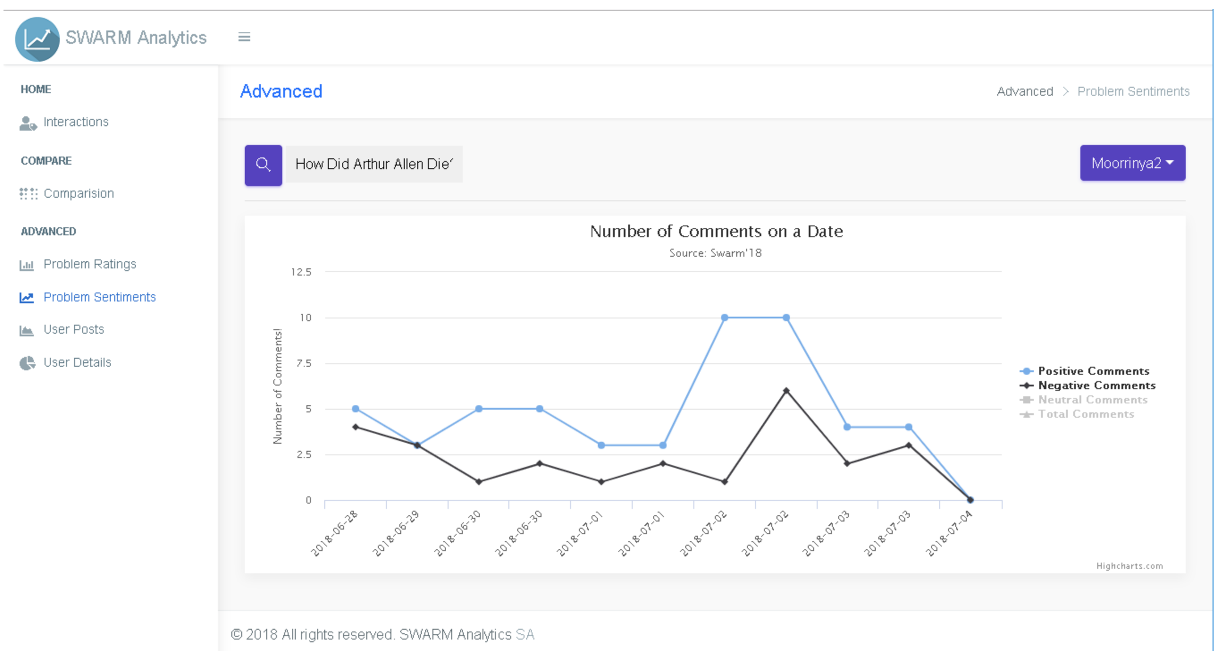


Figure 8: Team Filtered Problem Sentiments (Positive/Negative/Neutral/All)

5.6 User Posts

Section 5.3, 5.4 and 5.5 provides an overview of problem, team interactions, as well as team activity over time and sentiments. The page 'User Posts', however, gives a more detailed insight into a particular user's contribution in different problems. Figure 9 depicts user posts distribution over different problems. It provides a small translucent background box on hovering a specific problem that a user was assigned to. The user can be selected from top-left suggestive text input. For instance, in the screenshot below, it can be observed that 'Bilby374' came up with ten hypothesis, fifteen comments, twenty-five total posts (sum of comments, hypothesis and descriptions) and belonged to team 'Moorrinya2' in problem 'How Did Arthur Allen Die?'. It can also be noted that 'Bilby374' was quite active in other problems as well.

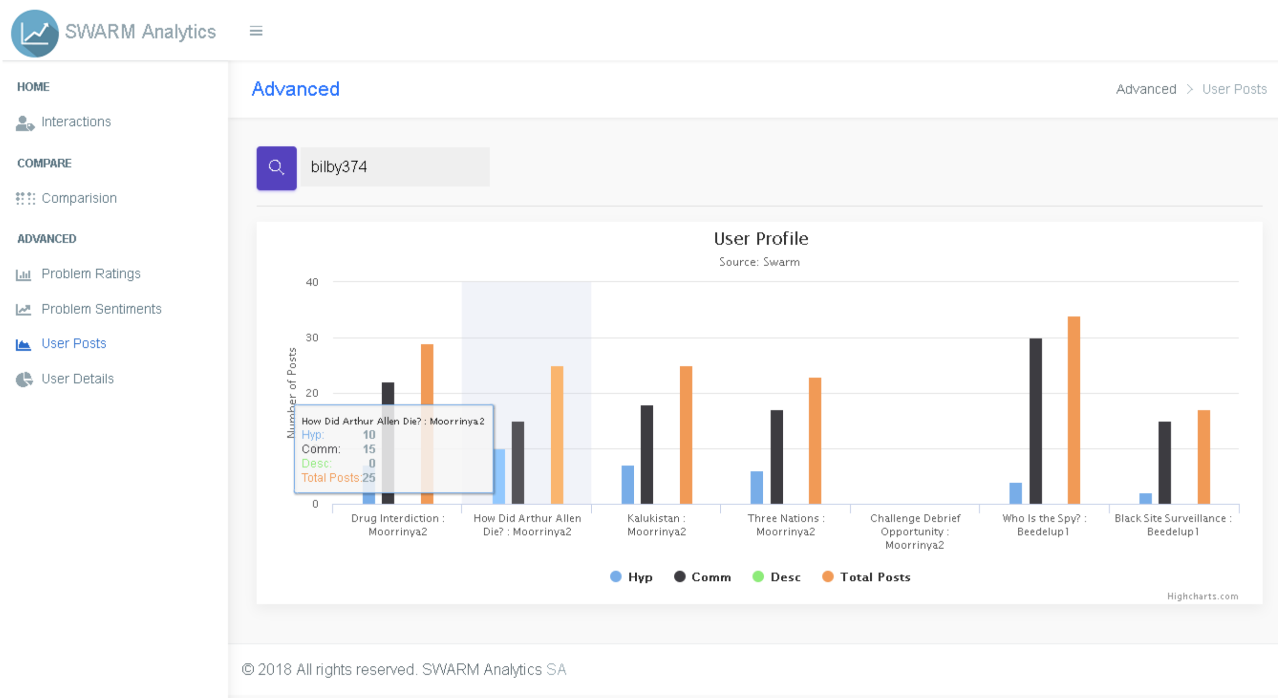


Figure 9: User Posts Distribution Over Different Problems

5.7 User Details

This page provides a more granular view of any user assigned on a particular problem. At 'User Details', once a user is selected via top-left search input, the corresponding problems user was ever assigned to, gets populated in top-right suggestive search input. The page only displays graphs after a problem is selected for the user in question. Figure 10 shows three pie-charts which depict posts, sentiments and tags distribution for user 'Bilby374' on problem 'How Did Arthur Allen Die?', along with the activity time-line for the user. Tags distribution bifurcates pie-chart into incoming and outgoing pings i.e. the number of time user was pinged versus the number of times user pinged someone, respectively.

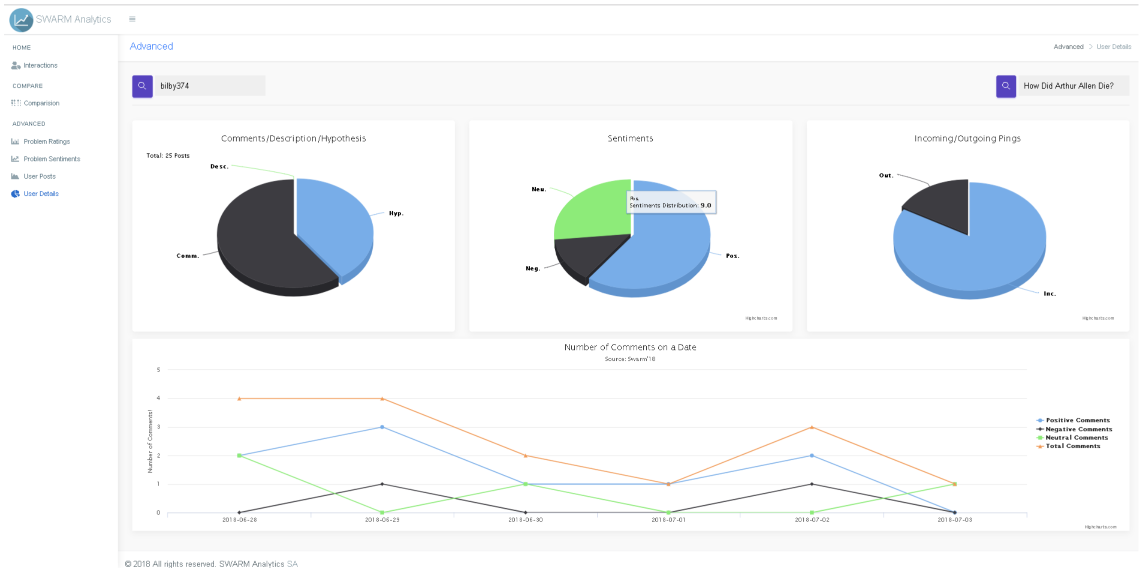


Figure 10: User Details -Post Distribution and Activity Time-line for a Problem

5.8 Comparison

'Comparison' page is a combination of all the pages in a logical flow. It must be noted that while all pages can be viewed in different tabs on a browser, for comparison's sake, this page makes it much easier to contrast different problems, teams or users. Figure 11 shows the team-wise internal/external ratings along with two interaction graphs i.e. for 'Moorrinya2' (left) and 'Bogong1' (right) on problem 'How Did Arthur Allen Die?'. On hovering problem ratings, the blue and grey bars, one can not only see the score in a pop-up, but also these are click-able and interactive. A click on any of the two bars for a team, changes the interaction sections alternatively to reflect the repective team's interaction graph.

It can be observed that the interaction pages are scroll-able i.e. figure 12 illustrates the activity between these two teams on a problem.

Similarly, figure 13 allows to search for a user and drill-down its profile, problem-wise, in detail. 'Bilby374' is selected on both sides and its contribution in different problems can be noticed. Similar to the behaviour delineated by interactive bars in problem ratings, the bars shown in figure 13 are interactive as well. Since the x-axis here represents 'problem:team', on-click of a bar, the graphs in team comparison section get populated.

Figure 14 shows a combined and scrolled screen-shot for each of the graphs in this user profile section. Left hand side of 14 shows user Bilby374's profile in problem 'How Did Arthur Allen Die?'. Whereas right hand side represents portrays the same for user Bilby374's profile in problem 'Drug Interdiction'.

Figure 15 shows team Moorrinya2's activity in two diffrenet problems ('How Did Arthur Allen Die?' versus 'Drug Interdiction').

A detailed critical analysis pertaining to scenario discussed on this section is described in section 6 ('Critical Analysis').

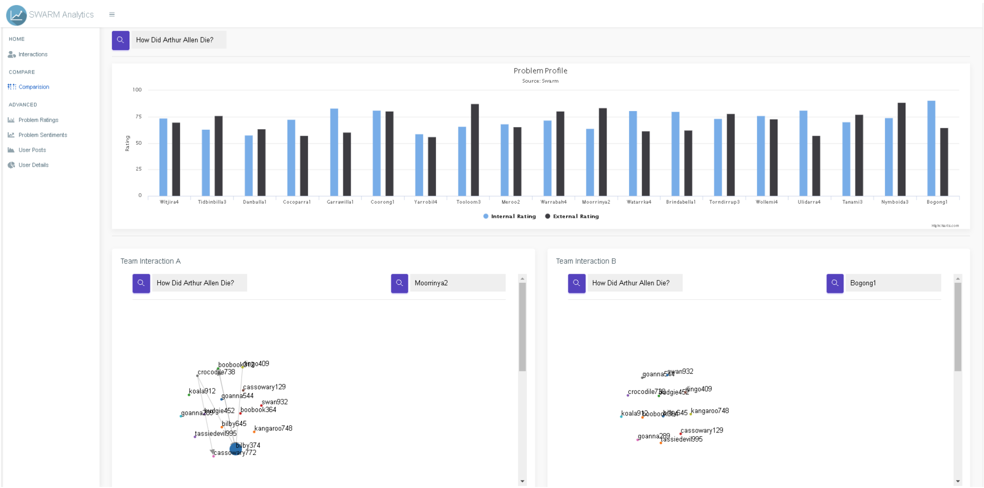


Figure 11: Team Interaction Comparison

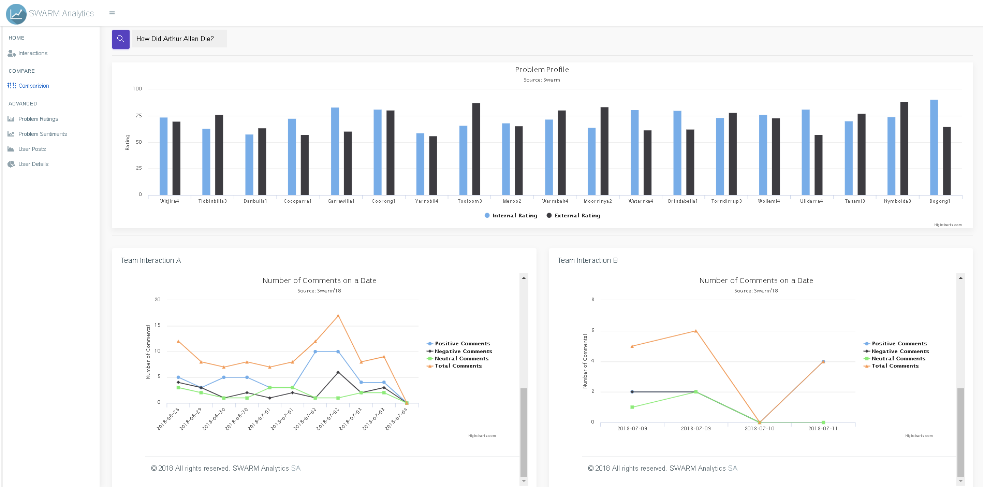


Figure 12: Team Activity Comparison

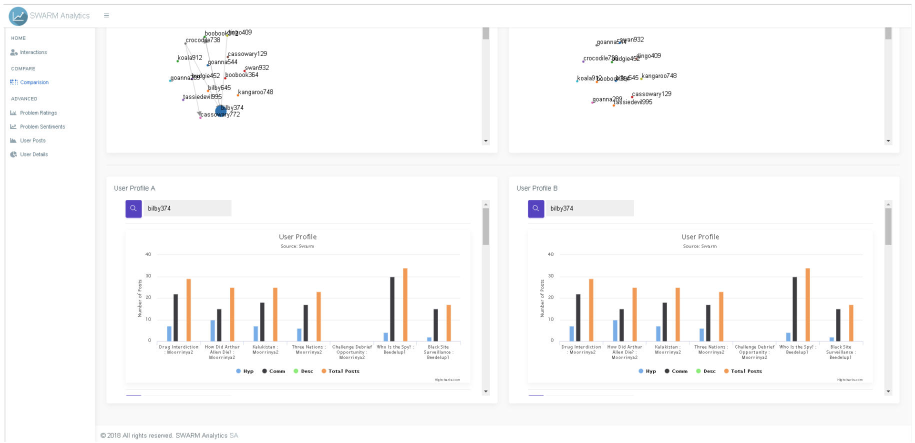


Figure 13: User Comparison over different Problems (1)

Figure 14: User Comparison over different Problems (2)

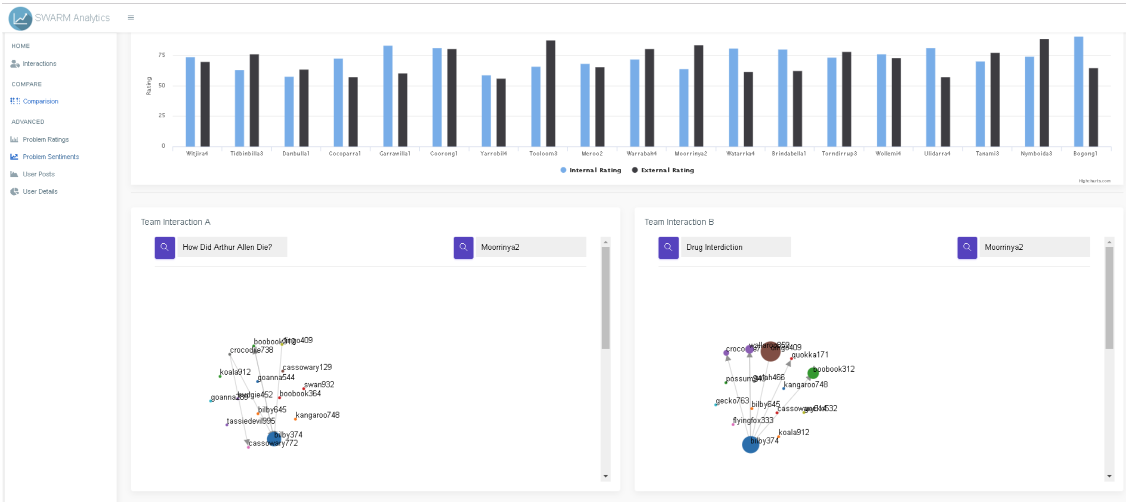


Figure 15: Team Interaction Comparison over different Problems

6 Critical Analysis

The ‘SWARM Analytics Dashoboard’ allows users to visualize interactions in multifarious combinations. Section 5.8, figure 11 shows team-wise, internal/external ratings, for problem ‘How Did Arthur Allen Die?’. Team ’Moorrinya2’ rated its highest hypothesis or solution as 64.08% on the overall quality metric. However, the external ratings for this team’s hypothesis touched up to 83.34%. On the other hand, notice team ‘Bogong1’, which has the opposite case. It rated its best hypothesis to be of a 90.05% quality, whereas externals rated it as 64.59%.

Going one step deeper, both teams interaction can be examined. Clearly, ’Moorrinya2’ was more active as compared to ‘Bogong1’, in terms of not only the interaction graph but also activity time-line, depicted in figures 11 and 12 respectively.

Hence, it can be inferred that to some extent, a highly active team is less confident about their quality of work, as compared to lesser functioning team. Moreover, even though an active team considered its hypothesis’ quality to be lower, it was ranked higher externally. This in turn indicates, that ‘Moorrinya2’ performed thorough discussions, after coming up with their hypothesis, unlike ‘Bogong1’. This can be validated in figure 12.

Analysis at user level also bring forth some useful insights. It can be seen in figure 15 that user ‘Bilby374’ was tagged the most in ‘How Did Arthur Allen Die?’ as well as the second-most tagged user in ‘Drug Interdiction’ problem. Such an insight points towards the fact that ‘Bilby374’ was not only an active user, but also an entity with good domain knowledge in given problems. Figures 13 and 14, also highlight this user’s profile in other problems, along with a side-by-side comparison for ‘How Did Arthur Allen Die?’ (left) and ‘Drug Interdiction’ problem (right).

One of the most imperative observations can be seen in most of the activity time-line graphs (figure 7), that number of comments increase towards the deadline i.e. teams are in general more active when a problem is about to be closed.

7 Deployment

For the purpose of building of 'SWARM Analytics Dashboard' web-application, a test driven development approach has been employed. The architecture is designed to facilitate a flexible deployment model. Figure 16 explains all deployment steps in order of execution. A few pre-requisites for the script to work are required. These include an Ubuntu server with git installed and project 'SWARM' cloned via 'https://github.com/mumairj/SWARM.git'. The script 'Deployminimum.sh' in project folder './SWARM/Deploy' performs the designated steps and outputs a URL after success full deployment.

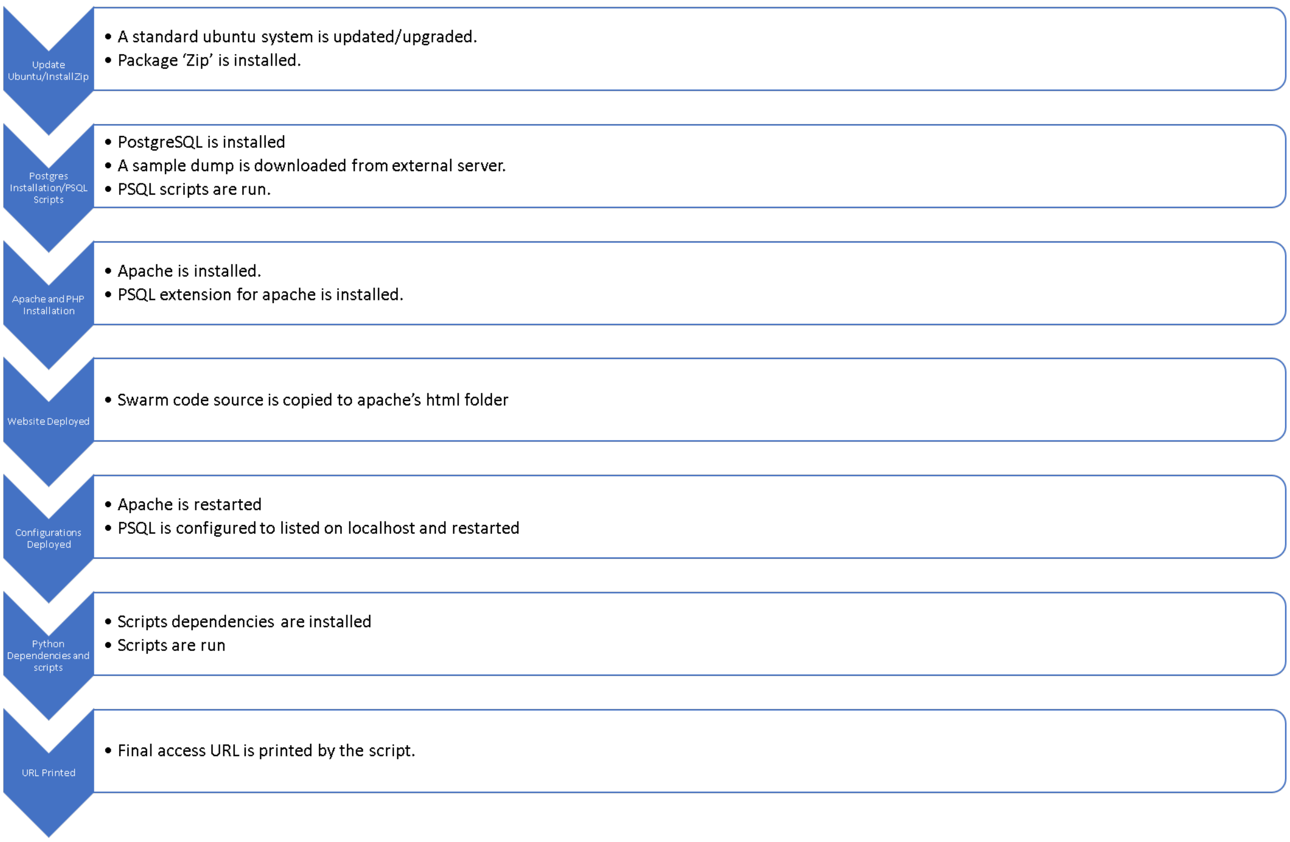


Figure 16: Deployment Steps (In Order of Execution)

8 Future Works

SWARM data can prove to be a valuable asset for scientific community in terms of future developments. A classic use-case for its application could be in the area of predictive analytics. The techniques from data mining, statistics, modeling, machine learning, and artificial intelligence, can be used to analyze current SWARM data for providing meaningful insights and predictions.

8.1 Classification Attempt

The project 'SWARM Analytics Dashboard', started off with an inception to move in a direction, similar to the one discussed above. The scenarior functions in a manner such that different teams provide an internal ratings to their solutions, which are then rated externally. A good utility for users could be, to get performance indicators during the SWARM life-cycle. Based on the team-activity i.e. features generated during the process, a team could get indications as to whether or not, their particular hypothesis or solution, will be rated highly by externals as well. Thereby, providing them an insight for a winning-hypothesis before hand.

However, due to limited availability of data, given SWARM's inaugural run, the idea was found to be in-feasible. A few of these limitations are discussed below.

Limitations An analysis of frequency distribution on data was performed, initially, to determine the whether or not to proceed with data modeling. However, the results were not quite promising. For instance, the core table for SWARM platform, named 'chunk', which comprises of a all the problems, hypothesis, comments, descriptions and suggested changes, consisted of 4198 records. Apart from this the data-set comprised of a total of seven problems, out of which, the results for four problems were received. A sample of external ratings received for each problem are given below. Table below shows problem name and number of teams which published a solution that was rated externally.

Problem	Teams Rated
Three Nations	19
How Did Arthur Allen Die?	19
Drug Interdiction	18
Kalukistan	20

Table 2: Problem versus Number of Teams Rated by Externals

8.2 Proposed Track

For future, this work can be utilized to develop essential features from a larger data-set. For example, a viable set of features could be number of team members active in a team, count of positive, negative, neutral comments, number of hypothesis a team came up with, frequency of team-members contribution over weeks or months etc. These features can be evaluated to include or exclude useful one. As a part of feature engineering process, new features can be determined. Classification algorithms can then be applied to predict, for instance, if a hypothesis exhibits the properties of a 'winning hypothesis'.

9 Conclusion

Conclusively, the SWARM platform in general, paves numerous ways for advanced developments. Such progressions are not limited to evidence-based reasoning, but in-fact, expand towards a greater circle of scientific research. The analytics dashboard developed as a part of this project, lead towards interesting insights. Whether it is the behaviour of teams and their sentiments over time for solving a problem, or their interaction patterns, resulting in a certain threshold of internal or external ratings, such insights remain an area for further explorations.

10 References

[1] "The Design and Development of a Cloud-based Platform Supporting Team-oriented Evidence-based Reasoning: SWARM Systems Paper". unpublished. N.D.

[2] Usama M Fayyad et al. "Advances in knowledge discovery and data mining". In: (1996).

[3] Nathaniel JS Brown et al. "The evidence-based reasoning framework: Assessing scientific reasoning". In: *Educational Assessment* 15.3-4 (2010), pp. 123–141.

[4] Alan Barnes. "Making intelligence analysis more intelligent: Using numeric probabilities". In: *Intelligence and National Security* 31.3 (2016), pp. 327–344.

[5] Serengul Smith-Atakan. *Human-computer interaction*. Cengage Learning EMEA, 2006.

[6] Gihan Osman and Susan C Herring. "Interaction, facilitation, and deep learning in cross-cultural chat: A case study". In: *The Internet and Higher Education* 10.2 (2007), pp. 125–141.

[7] Bo Pang, Lillian Lee, et al. "Opinion mining and sentiment analysis". In: *Foundations and Trends® in Information Retrieval* 2.1–2 (2008), pp. 1–135.

[8] *SWARM Zen-desk: How Readiness ratings are calculated*. `https://swarm-help.zendesk.com/hc/en-us/articles/360003932892-How-Readiness-ratings-are-calculated`. Accessed: 2018-10-28.

11 Appendix

11.1 Code Structure Dependency

The following table contains a list of web pages being used in the 'SWARM Analytics Dashboard' and the corresponding associated views and tables.

Page	Source	Dependency
index.php	-	-
tags_questions.php	get_team_interaction.php	***View*** : *- view_tags*
tags_questions.php	get_questions.php	***Table:*** *- chunk*
tags_questions.php	get_tags.php	***View*** : *- view_tags*
tags_questions.php	get_teams.php	***Tables:*** *- chunk* *- chunk_user_group_relation* *- perm_user_group*
user_profile_new.php	get_users.php	***Table:*** - user
user_profile_new.php	get_user_profile.php	***View:*** *- team_members* *- chunk_swarm* ***Tables:*** *- chunk_user_group_relation* *- chunk_parent_relation* *- chunk* *- chunk_parent_relation*
user_profile_details.php	get_users.php	***Table:*** - user
user_profile_details.php	get_user_problem.php	***View:*** *- team_members* ***Table:*** *- chunk* *- chunk_user_group_relation*

Figure 17: Code Dependency Structure (1)

Page	Source	Dependency
user_profile_details.php	get_profile_details.php	***View:*** *- team_members* *- chunk_swarm* *- view_comments* *- view_tags* ***Tables:*** *- chunk_user_group_relation* *- chunk_parent_relation* *- chunk* *- chunk_parent_relation*
problem_profile.php	get_questions.php	***Table:*** *- chunk*
problem_profile.php	get_problem_profile.php	***View:*** *- view_quality_prob_hyp_team_avg* *- swarm_scores* ***Table:*** *- chunk*
sentiments_series.php	get_teams.php	***Tables:*** *- chunk* *- chunk_user_group_relation* *- perm_user_group*
sentiments_series.php	get_questions.php	***Table:*** *- chunk*
sentiments_series.php	get_problem_sentiments.php	***Veiw:*** *- view_comments_sentiments_problemwise* *- view_comments_sentiments*
comparision_teams.php	get_questions.php	***Table:*** *- chunk*
comparision_teams.php	get_questions.php	***Table:*** *- chunk*

Figure 18: Code Dependency Structure (2)

Page	Source	Dependency
comparision_teams.php	get_problem_profile.php	***View:*** *- view_quality_prob_hyp_team_avg* *- view_published_hyp* ***Table:*** *- chunk* *- swarm_scores*
comparision_teams.php	tags_questions_A.php	***PHP Page:*** *- get_questions.php* *- get_tags.php* *- get_teams.php* *- get_team_interaction.php*
comparision_teams.php	compare_user_profile_details.php	***PHP Page:*** *- get_users.php* *- get_profile_details.php* *- get_user_problem.php* *- get_user_profile.php*

Figure 19: Code Dependency Structure (3)

11.2 Database Views Dependency

The following table contains a list of views being used in the 'SWARM Analytics Dashboard' and the corresponding tables.

View	Dependent	Type	Used Directly	Comments
view_tags	view_prob_hyp_comm_desc_team	View	Yes	-
view_tags	swarm_user_tag_chunk	Table	Yes	To be populated by python script
view_prob_hyp_comm_desc_team	Swarm Tables	Table	No	-
view_comments	chunk_sentiment	Table	Yes	To be populated by python script. All others are swarm tables.
team_members	Swarm Tables	View	Yes	-
chunk_swarm	Swarm Tables	View	Yes	-
view_quality_prob_hyp_team_avg	Swarm Tables	View	Yes	Using quality metric
swarm_scores	N/A	Table	Yes	To be populated manually
view_comments_sentiments	view_comments	View	Yes	Details above
view_comments_sentiments_problemwise	view_comments_sentiments	View	Yes	Details above
view_published_hyp	DB link extension	View	Yes	-

Figure 20: Database Views Dependencies