

# CROWDSOURCING FOR REQUIREMENTS GATHERING: STATE OF PRACTICE

MUHAMMAD FAHMI ISMAIL  
SABRINA TIUN  
NOOR HASRINA BAKAR

*Faculty of Information Science and Technology, National University of Malaysia*

## ABSTRACT

Requirements gathering plays vital role in software development, for software to be usable and likeable it must cater the best to communicate the needs, ideas and wishes from millions of stakeholders, hence always rely heavily on user perspectives. Thus, crowd involvement is becoming significant in requirements gathering phase of software development be it for software evolution or software development. Nowadays, requirement from user in the form of review of the software can be accessed in many reliable sources which is part of the crowdsourcing process. Therefore, this paper discusses the implementation of natural language text processing on crowdsource review and clustering algorithm approach such as K-Means to show result of the most frequent need of the user for a software requirement.

## INTRODUCTION

Requirements Elicitation or requirements gathering is defined as the progression of obtaining a complete understanding of stakeholder's requirements. It is the initial and main process of requirements engineering phase where "what is to be done" is elicited and modeled (Leite 87). Elicitation process usually involves interaction with stakeholders to obtain their actual needs: what and how they imagine the software to be developed will solve their current problems. The requirements elicitation activities help stakeholders to express their needs and expectations of the new system. (Masooma, Asger 2015). One of the traditional methods of requirement elicitation is interview. For many years, team projects interview users and stakeholders for developing a new system or improving existing systems. It is true that interviews can be very successful in many occasions and it does help developers to understand the users need. However, there are also some drawbacks of interviews such as interviewee did not reveal their actual needs due to incomplete understanding of their needs. Additionally, sometimes requirements captures are ambiguous and its scope are ill-defined. There are also cases of miscellaneous miscommunication issues during Requirement Engineering (RE) process. Most of the time, important stakeholders who are from managerial team usually are busy people. Their time for RE interviews can be very limited. This may result that some important information might not be delivered during the RE process. Due to fundamental problems during traditional RE activities, developers option for an easier RE activity, that is via online collaboration including mining user reviews from the crowd (crowdsourcing). This option can provide benefits for software evolution and new software upgrade releases.

## PROJECT MOTIVATION

Reuse of requirement is very tedious and error prone if done manually. Thus, an automated approach is needed to expedite the reuse process. Requirement documents from legacy systems are usually left in the archive, went wasted. Although improvements to the retired system are usually done from the weaknesses of previous system, very little work proposed on modernizing the new system through reuse of legacy documentations with crowd (potential users) involvement, for example through automated text extractions. The requirement engineering phase of traditional software development usually did not consider crowd involvement in gathering requirements. As a result, the developed software products sometimes did not meet crowd expectations. It is essential to encourage crowd involvement at the early stage of software development to ensure the software produced can meet potential users' expectations. Using unstructured data to represent requirements may cause lower precision values. Consequently, the existing approach failed to automatically cluster similar features that were extracted from unstructured textual data. Therefore, the objective of this project carry out an experiment to extract out or discover user's requirement on based on a crowdsorce data using NLP approach. The following section will describe how the experiment was conducted as well as the data that has been used in the experiment.

## REQUKM EXTRACTION EXPERIMENT

### REQUKM

Derived from its name, REQUKM are abbreviation from the "Requirement UKM" which means the process of gathering requirement from the user. This application is built from Python programming language. The objective of this application is to read user review and identify the software functionality from the selected review from the result of K-Means clustering that implement in this system. This application use to extract the features that reside from the selected reviews. One of its outputs is to produce features in forms of noun phrases extracted via text processing. Then, the term is being count to find the weight of the term and to obtain phrase relatedness via number of its occurrences, this show how frequent the term in the review. Next is applying the TFIDF to find the coordinate for all the phrases in the document space. Lastly K-Means process take place to clustering into top term by group the noun phrase together. Then be visualize via word cloud to project the most frequent term that are known as functionality of the software that user need.

REQUKM

Directory Files: D:/KERJA FAHMI/TEXT PROCESSING SYSTEM: Browse

Context:	Text Processing:	TFIDF:
I dislike that the User Interface is sometimes rather complicated.	User Interface, TFS, Visual, custom report, easy way, submit market requests, trackable dashboard, Slack, initial set, support team, nice design, mobile app, mobile friendly site, update information, certain projects, good way, Referred, Aha, Aha, top-level tasks, removes functionality, Kanban, current release, ~30 members, Aha, Requirements MUST, Todos, different fields, Aha, different way, user's permissions, flat organization, complete control, Aha, full control, usability issues, Aha, whole product, super intuitive, product teams, non developers, product managers, pretty awful, large backlog, Roadmapping, quirky things, product functionality, Product, mobile app, new users, small portion, overall functionality, capacity planning, project plans, Mgmt, shareable link, team member, UI, big clunky, Notifications, Off, engineering managers, learning curve, setup mistakes, Aha, Aha, can't tag members, various teams, sufficient privileges, Aha, can't figure, user interface, Aha, FogBugz, FogBugz, MAJOR, Big projects, Adjusting, duplicate use, feature board, numerous categories, company-specific designations, Looks, Unclear, screen slides, multiple times, Jira, admin account, Note	(0, 189) 0.7234658845825394 (0, 70) 0.6903601334413828 (1, 173) 1.0 (2, 193) 1.0 (3, 34) 0.75237276222703 (3, 134) 0.6587376007629054 (4, 47) 0.7798229746751543 (4, 194) 0.6260001023912325 (5, 161) 0.5773502691896257 (5, 87) 0.5773502691896257 (5, 137) 0.5773502691896257 (6, 182) 0.7071067811865476 (6, 36) 0.7071067811865476 (7, 151) 1.0 (8, 67) 0.7071067811865476 (8, 146) 0.7071067811865476 (9, 185) 0.7428947121126313 (9, 170) 0.6694082810326526 (10, 100) 0.7071067811865476 (10, 38) 0.7071067811865476 (11, 94) 0.6545455847044936 (11, 10) 0.7560225377221584 (12, 94) 0.4936449002859725 (12, 59) 0.6149450025903341 (12, 150) 0.6149450025903341 : : (177, 181) 0.5482934497983435 (177, 106) 0.5913435096913771 (177, 4) 0.5913435096913771 (178, 198) 1.0

Process

Figure 1: REQUKM interface

## Flowchart of REQUKM

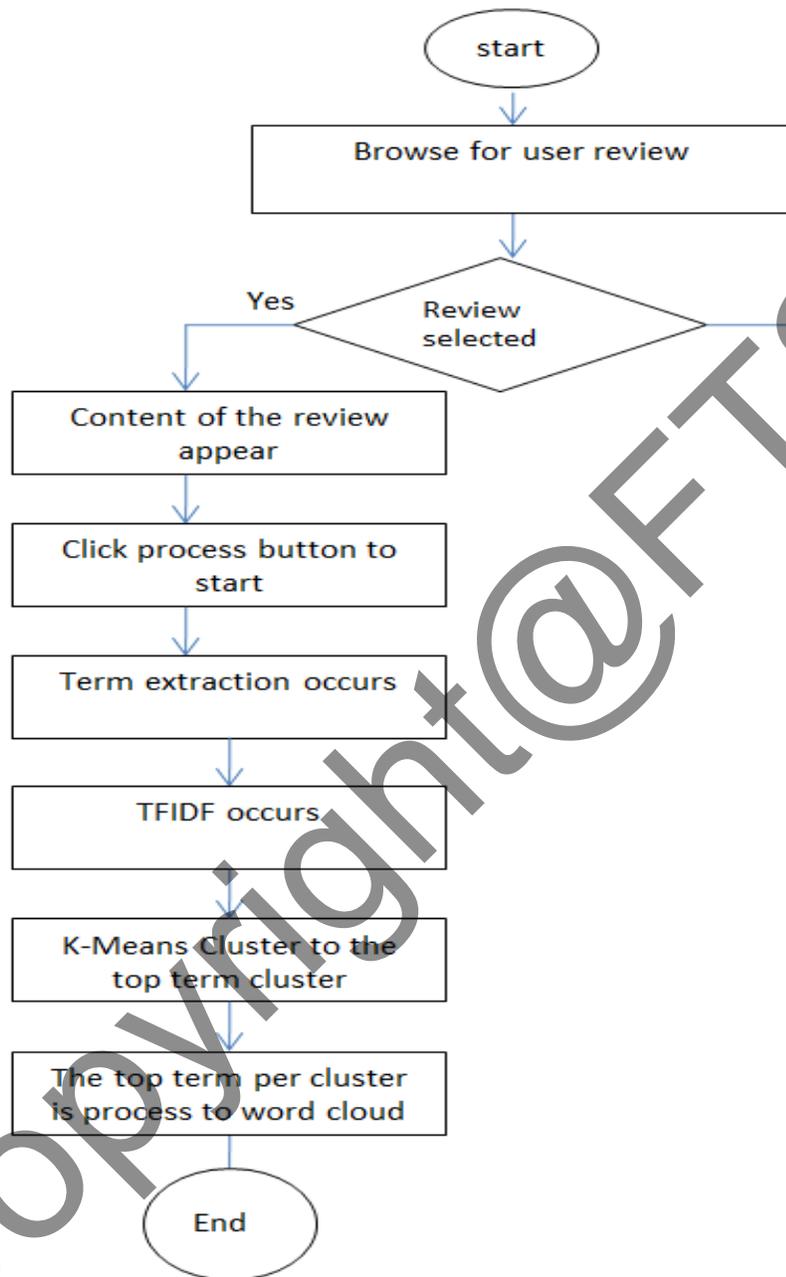


Figure 2: Flowchart of REQUKM

## DFD of REQUKM

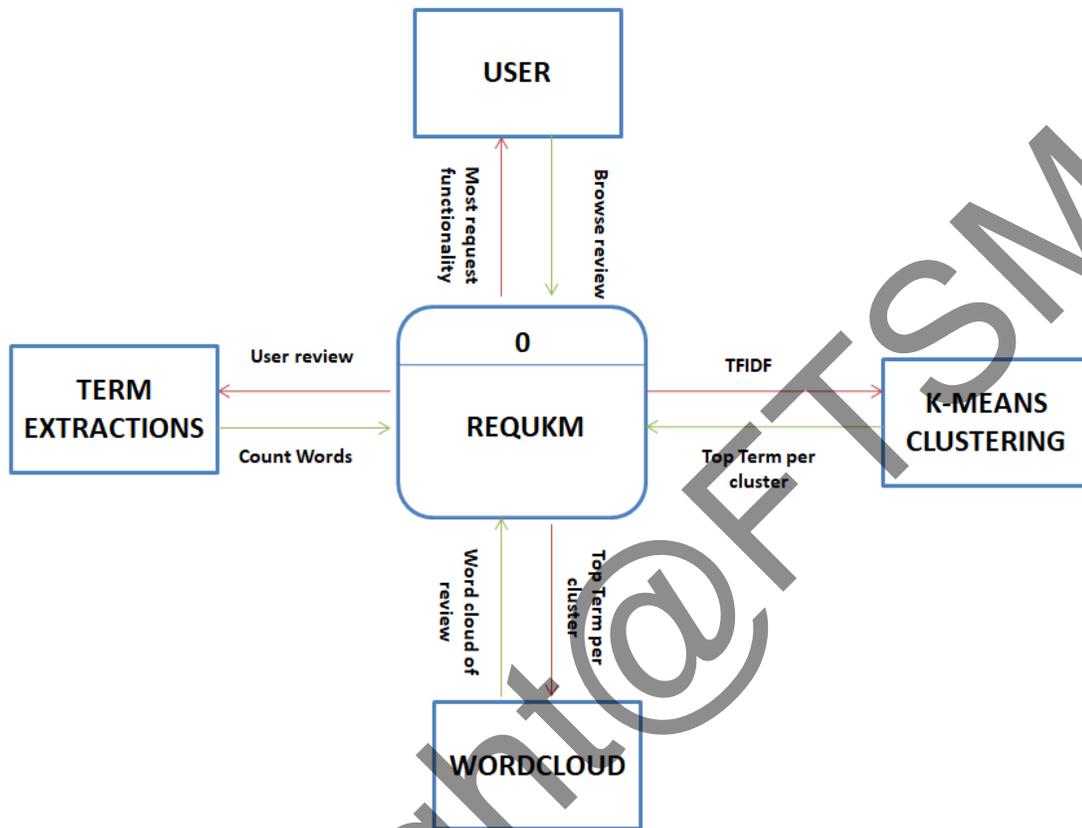


Figure 3: DFD of REQUKM

## INTERFACE DESIGN

Figure 4 show the GUI of REQUKM. In this interface, user can select data of the review by clicking the browse button. After that, the directory of the selected files will be display in the text box to show where the data is located. At the same time, the content of the selected review will be automatically read and display in the first text box. Second, user need to click the process button to start the text processing. Then, the term from user review and TFIDF will be display on the second and third text box to indicate the process that occurs is successful.

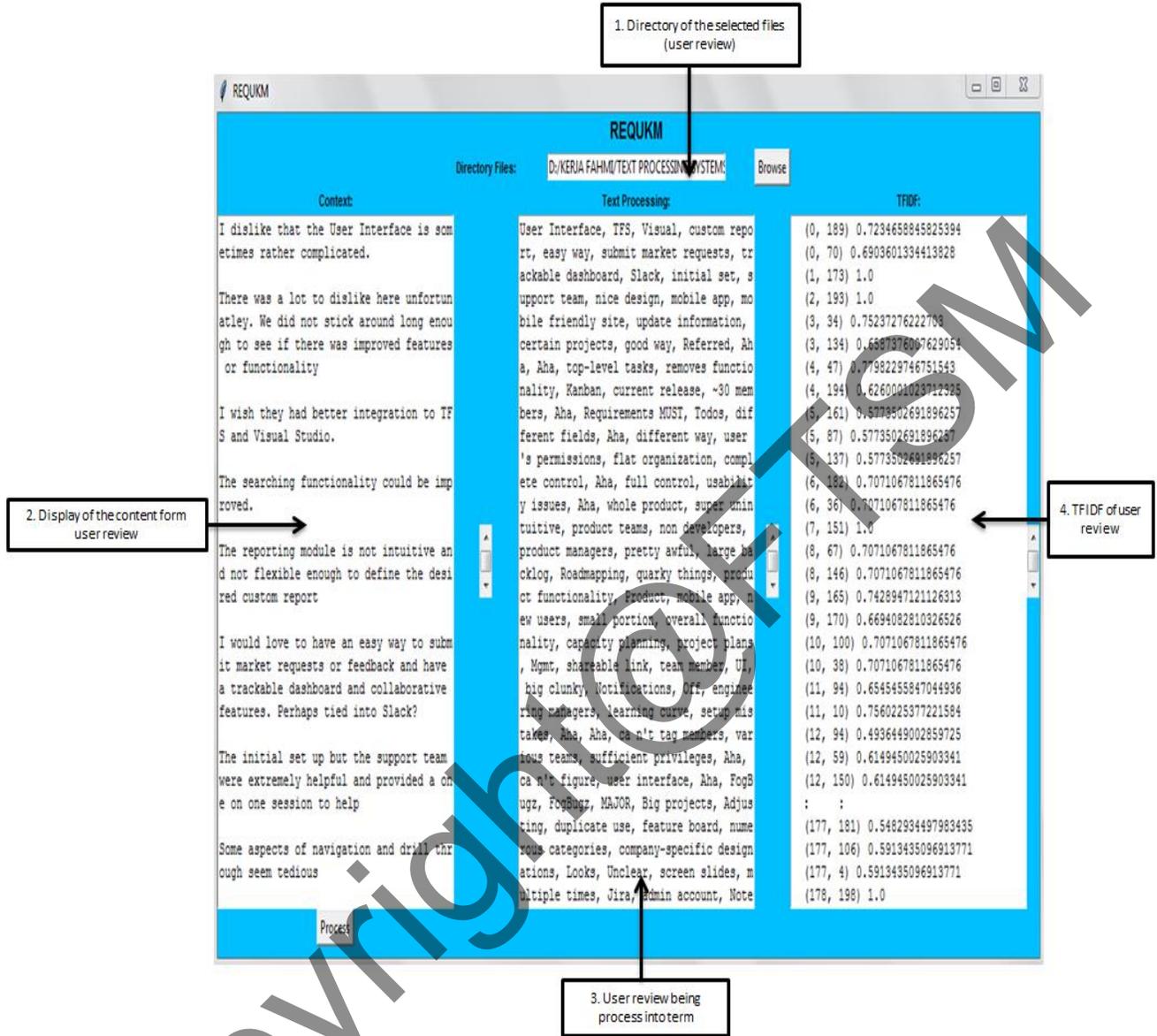


Figure 4: REQUKM GUI

## Dataset

Our objective is to identify and cluster the software modernization from the selected software review that publishes by the crowds from variety of online platform such as G2CROWD, GetApp and Capterra. Therefore, the dataset are from crowdsourcing activity which is the consumer of the software give information, critics, and requirement towards the modernization of software. The total of 28 software reviews are scrapped as a dataset for this project and this raw data are then stored in the text files to be process through REQUKM tools. Figure 5 shows the example of website software review.

Reviews   Product Information   Pricing   Features

---



**Malena Y**   
 Marketing Manager  
 Information Technology and  
 Services  
 Mid-Market  
 (51-200 employees)

Validated Reviewer   
 Verified Current User   
 Review Source 

★★★★★ Feb 25, 2018

[Copy Review URL](#)

**" Powerful product management tool"**

**What do you like best?**  
 Aha! This is a very full-featured road tool with a lot of multifunctional capabilities. Just to realize, what else needs to be done, and I can prioritize. In addition, I can see what my team members need to do, which simplifies the transition and support. In general, it is productive.

**What do you dislike?**  
 This service does not need absolute functionality, but the numerous functions of Aha! must be easily accessible in mobile devices, including messages, ideas and functions. In some areas, I would like the document to be done in a different way, but this is insignificant.

Recommendations to others considering the product  
[Show More](#)

---



**Delinda S**   
 Quality Specialist  
 Information Technology and  
 Services  
 Enterprise  
 (10,001+ employees)

Validated Reviewer   
 Verified Current User   
 Review Source 

★★★★★ Jan 29, 2018

[Copy Review URL](#)

**"Stay focused on what's important for your working process."**

**What do you like best?**  
 I guess the best at working days it is to have balanced scheduling which will make better discipline and with this platform, we have this chance. I'm glad that I found the faster and better management system. The software allows the bigger way for useful tools.

**What do you dislike?**  
 By luck, I never have problems at work during use that platform. Doesn't bring any trouble and don't rush our team.

Recommendations to others considering the product  
 Don't have any doubts about this platform. You will use the better strategy at work. You will have  
[Show More](#)

---



**Reynaldo C**   
 Project Manager  
 Human Resources

★★★★☆ Jan 14, 2018

[Copy Review URL](#)

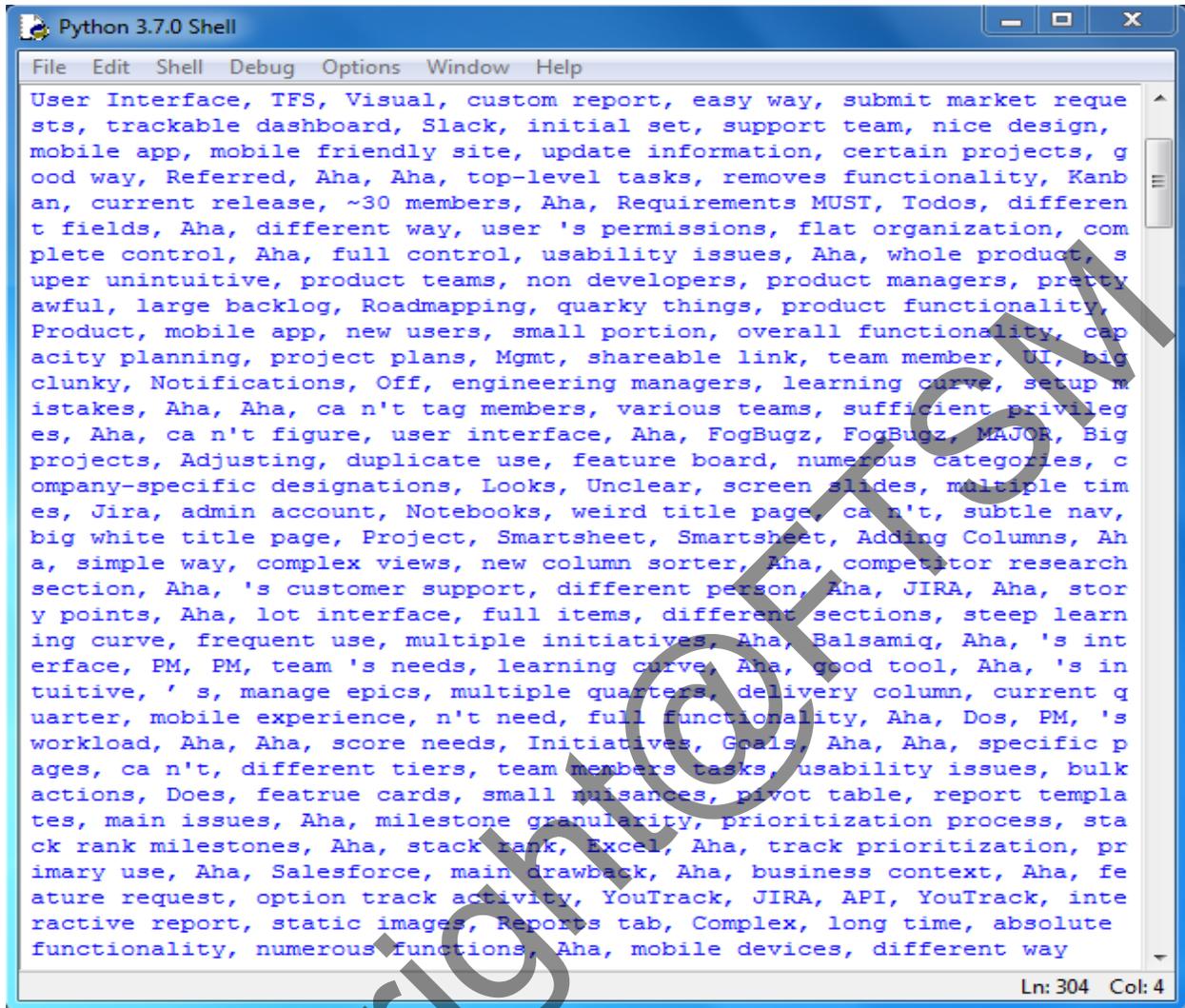
**"All information in one place makes your work easy."**

**What do you like best?**  
 I'm glad that I found the faster and better controlling system of my scheduling. Now, I can have an effective process of working reports. Also, it is a wonderful way to work on projects. The software give the big opportunity of useful tools.

Figure 5: G2CROWD website software review

## Term extractions

First of all, we clean the user review file via splitting into terms or word and handling punctuation and case, we clean the text to make it ready for modeling or cluster with machine learning. The step consists of split whitespace, remove punctuation, normalization, tokenization and split into words. Therefore the result show the sentence become like a term or word so we can progress to the next process to find the frequent and weight of those word that derived from user reviews.



```

Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
User Interface, TFS, Visual, custom report, easy way, submit market reques
sts, trackable dashboard, Slack, initial set, support team, nice design,
mobile app, mobile friendly site, update information, certain projects, g
ood way, Referred, Aha, Aha, top-level tasks, removes functionality, Kanb
an, current release, ~30 members, Aha, Requirements MUST, Todos, differen
t fields, Aha, different way, user 's permissions, flat organization, com
plete control, Aha, full control, usability issues, Aha, whole product, s
uper unintuitive, product teams, non developers, product managers, pretty
awful, large backlog, Roadmapping, quarky things, product functionality,
Product, mobile app, new users, small portion, overall functionality, cap
acity planning, project plans, Mgmt, shareable link, team member, UI, big
clunky, Notifications, Off, engineering managers, learning curve, setup m
istakes, Aha, Aha, ca n't tag members, various teams, sufficient privileg
es, Aha, ca n't figure, user interface, Aha, FogBugz, FogBugz, MAJOR, Big
projects, Adjusting, duplicate use, feature board, numerous categories, c
ompany-specific designations, Looks, Unclear, screen slides, multiple tim
es, Jira, admin account, Notebooks, weird title page, ca n't, subtle nav,
big white title page, Project, Smartsheet, Smartsheet, Adding Columns, Ah
a, simple way, complex views, new column sorter, Aha, competitor research
section, Aha, 's customer support, different person, Aha, JIRA, Aha, stor
y points, Aha, lot interface, full items, different sections, steep learn
ing curve, frequent use, multiple initiatives, Aha, Balsamiq, Aha, 's int
erface, PM, PM, team 's needs, learning curve, Aha, good tool, Aha, 's in
tuitive, ' s, manage epics, multiple quarters, delivery column, current q
uarter, mobile experience, n't need, full functionality, Aha, Dos, PM, 's
workload, Aha, Aha, score needs, Initiatives, Goals, Aha, Aha, specific p
ages, ca n't, different tiers, team members tasks, usability issues, bulk
actions, Does, featrue cards, small nuisances, pivot table, report templa
tes, main issues, Aha, milestone granularity, prioritization process, sta
ck rank milestones, Aha, stack rank, Excel, Aha, track prioritization, pr
imary use, Aha, Salesforce, main drawback, Aha, business context, Aha, fe
ature request, option track activity, YouTrack, JIRA, API, YouTrack, inte
ractive report, static images, Reports tab, Complex, long time, absolute
functionality, numerous functions, Aha, mobile devices, different way
Ln: 304 Col: 4

```

Figure 6: the term extractions

### Count Words

Count word approach is use to count the word that derive from user review to find the most frequent word thus give an insight on what the requirement the user most needed. Figure 7 show the result of print from count word process

```

Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Word Occurencies
-----
[user 1
interface 3
tfs 1
visual 1
custom 1
report 3
easy 1
way 4
submit 1
market 1
requests 1
trackable 1
dashboard 1
slack 1
initial 1
set 1
support 1
team 3
nice 1
design 1
mobile 5
app 2
friendly 1
site 1
update 1
information 1
certain 1
projects 2
good 2
referred 1
aha 32
top-level 1
tasks 2
removes 1
functionality 5
kanban 1
current 2
release 1
Ln: 44 Col: 18

```

Figure 7: Count Words

**TFIDF**

The use of Term Frequency Inverse Data Frequency (TFIDF) in this program is to find the frequency and calculate the weight of rare words for each user review. After the text processing is done, to make sure the process is accurate, we apply TFIDF and export it to the text file and compare with both method, which is analyze the text file and applying the K-means cluster to gain result. Figure 8 show the TFIDF process that been done to the user review.

```

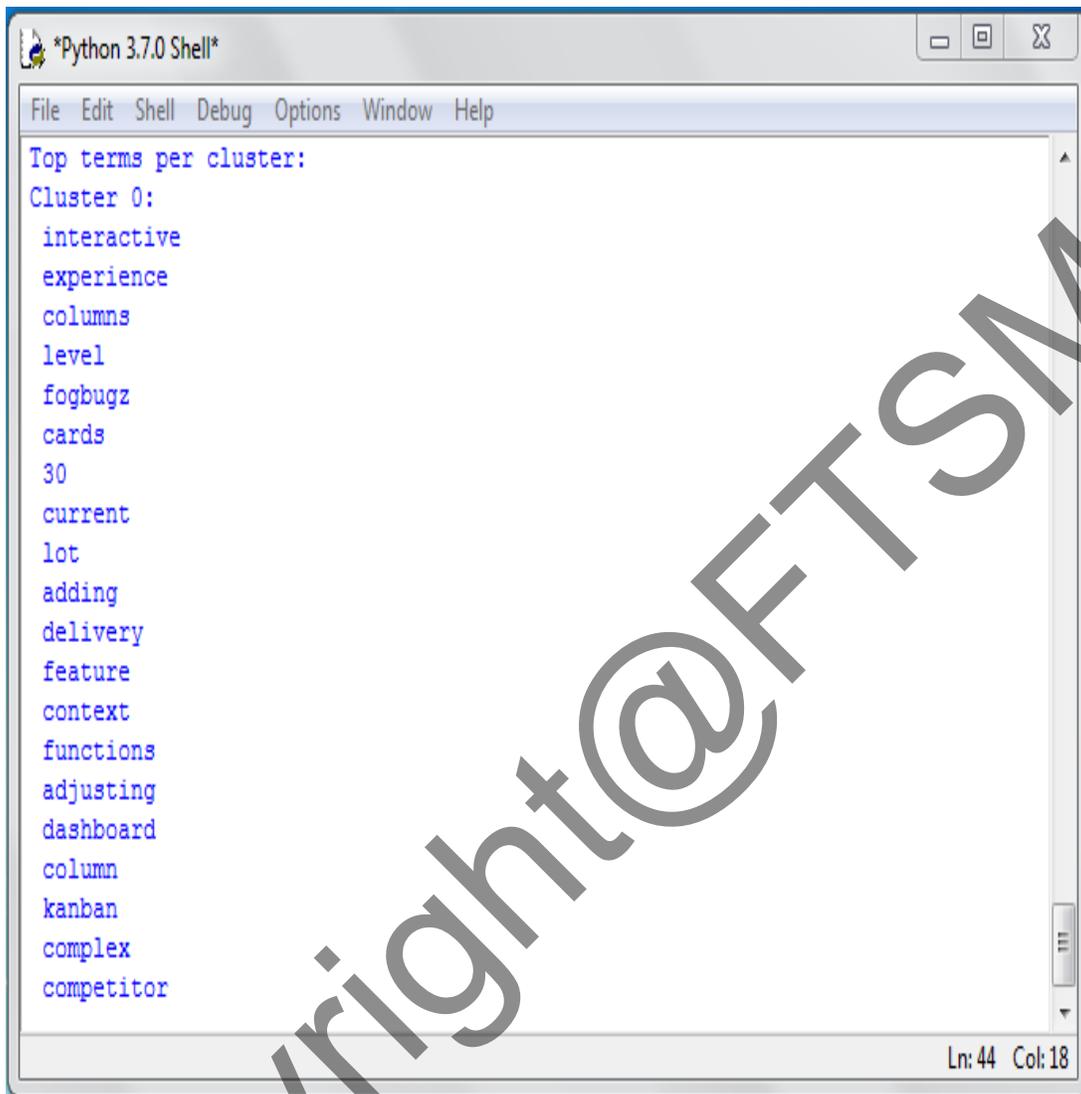
*Python 3.7.0 Shell*
File Edit Shell Debug Options Window Help
(0, 189)      0.7234658845825394
(0, 70)       0.6903601334413828
(1, 173)      1.0
(2, 193)      1.0
(3, 34)       0.75237276222703
(3, 134)      0.6587376007629054
(4, 47)       0.7798229746751543
(4, 194)      0.6260001023712325
(5, 161)      0.5773502691896257
(5, 87)       0.5773502691896257
(5, 137)      0.5773502691896257
(6, 182)      0.7071067811865476
(6, 36)       0.7071067811865476
(7, 151)      1.0
(8, 67)       0.7071067811865476
(8, 146)      0.7071067811865476
(9, 165)      0.7428947121126313
(9, 170)      0.6694082810326526
(10, 100)     0.7071067811865476
(10, 38)      0.7071067811865476
(11, 94)      0.6545455847044936
(11, 10)      0.7560225377221584
(12, 94)      0.4936449002859725
(12, 59)      0.6149450025903341
(12, 150)     0.6149450025903341
:
:
(177, 181)   0.5482934497983435
(177, 106)   0.5913435096913771
(177, 4)     0.5913435096913771
(178, 198)   1.0
(179, 74)    1.0
(180, 9)     1.0
(181, 198)   1.0
(182, 134)   0.6587376007629054
(182, 69)    0.75237276222703
(183, 158)   0.7071067811865476
(183, 65)    0.7071067811865476
(184, 135)   0.7071067811865476
(184, 166)   0.7071067811865476
(185, 29)    1.0
Ln: 44 Col: 18

```

Figure 8: TFIDF

### K-Means

This program use K-Means which is of one unsupervised machine learning algorithm. This approach will create a cluster automatically from the derived user review that been process through text processing, count word and TFIDF. We use step known as feature extraction because K-Means deal with numbers that derived from TFIDF process before, thus give us the statistical numerical for implementation. Even though clustering can belong to many cluster. We decide to only apply one cluster to gain the most top term cluster. Figure 9 show top term cluster



```
*Python 3.7.0 Shell*
File Edit Shell Debug Options Window Help
Top terms per cluster:
Cluster 0:
interactive
experience
columns
level
fogbugz
cards
30
current
lot
adding
delivery
feature
context
functions
adjusting
dashboard
column
kanban
complex
competitor
Ln: 44 Col: 18
```

Figure 9: K-Means Cluster

### Word Cloud

Figure 10 shows the word cloud that occurs from the K-Means clustering that show the frequent term of the user review which indicates the most word that request from the user.



Copyright@FTSM