



De La Salle University

**FILIET: An Information Extraction System
For Filipino Disaster-Related Tweets**

A User Manual
Presented to
the Faculty of the College of Computer Studies
De La Salle University – Manila

In Partial Fulfillment
of the Requirements for the Degree of
Bachelor of Science in Computer Science

by
DELA CRUZ, Kyle Mc Hale B.
GARCIA, John Paul F.
KALAW, Kristine Ma. Dominique F.
LU, Vilson E.

REGALADO, Ralph Vincent
Adviser

April 27, 2015



Table of Contents

1.0	Introduction.....	1-1
1.1	System Requirements.....	1-1
1.2	Convention	1-1
1.3	Installation	1-2
1.3.1	MySQL	1-2
1.3.2	NormAPI.....	1-5
2.0	Getting Started.....	2-1
2.1	FILIET Crawler Module	2-1
2.2	FILIET.....	2-5
2.2.1	Section 1 & Section 2.....	2-7
2.2.2	Section 3 & Section 4.....	2-7
3.0	Messages	3-1



1.0 Introduction

FILIET (Filipino Information Extraction for Twitter) is an information extraction system that makes use of handcrafted rules in order to extract the information from tweets composed in the Filipino language. The system is composed of six modules: the crawler, preprocessor, feature extraction, classification, rule inductor, and ontology module. The crawler module can be run as a standalone submodule of the system whereas the rest are integrated. Through the crawler module, tweets are collected and stored in the database which is then exported to a CSV file. The remainder of the FILIET system makes use of the exported CSV file for extraction.

1.1 System Requirements

Table 1-1 lists the minimum hardware and software requirements needed to use the system.

Table 1-1. System Requirements

Machine Specification	
Operating System	Windows 8 (64-bit), Mac OSX 10.10 Yosemite
Memory	4GB
Processor	Core i5
Hard Disk	1 Gb
Software Specification	
MySQL	MySQL 5.6 or higher
Java Runtime Environment	JRE7 or higher

1.2 Convention

Table 1-2 describes the convention used in the manual so as to guide the reader in identifying the important or emphasized concepts.

Table 1-2. Manual Convention

Concept	Convention	Example
Default	Font Type: Arial Font Size: 10px	This is the default convention.
Table or Figure Caption	Font Type: Arial Font Size: 10px Font Style: Italicized, Boldface	Figure 1-1. Figure Caption Table 1-1. Table Caption
Filepath or File	Font Type: Courier New Font Size: 10px Font Style: Italicized	<i>C:/FILIET/Source Code</i> <i>Filietv3.jar</i>
Website Link	Font Type: Courier New Font Size: 10px Font Style: Italicized, Underline	<u><i>http://www.website.com</i></u>
Function or Module Names or Code Snippets	Font Type: Courier New Font Size: 10px	FunctionName ModuleName java -jar codesnippet



Button	Font Type: Arial Font Size: 10px Remarks: Enclosed in single quotes	'ButtonName'
Other concepts that needs highlighting	Font Type: Arial Font Size: 10px Font Style: Boldface	This is an example .

1.3 Installation

This subsection contains the instructions on how to install the system and the list of necessary pre-requisite tools needed. The pre-requisite tools are located in the */TOOLS/* folder.

1.3.1 MySQL

1. Run *mysql-installer-community-5.6.23.0.msi* (refer to Figure 1-1).

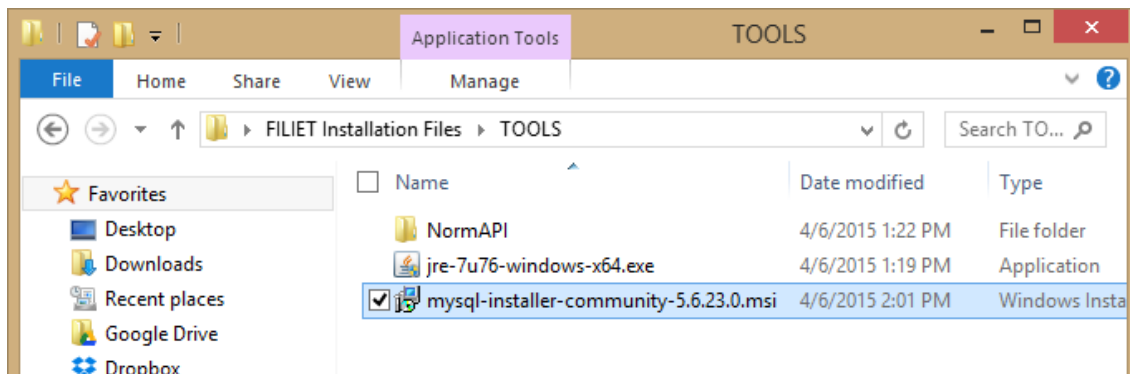


Figure 1-1. MySQL Installer

2. Follow the instructions indicated in the installer. For further information with regards to the installation of MySQL, please refer to their official documentation located at <https://dev.mysql.com/doc/refman/5.6/en/windows-installation.html> (refer to Figure 1-2).

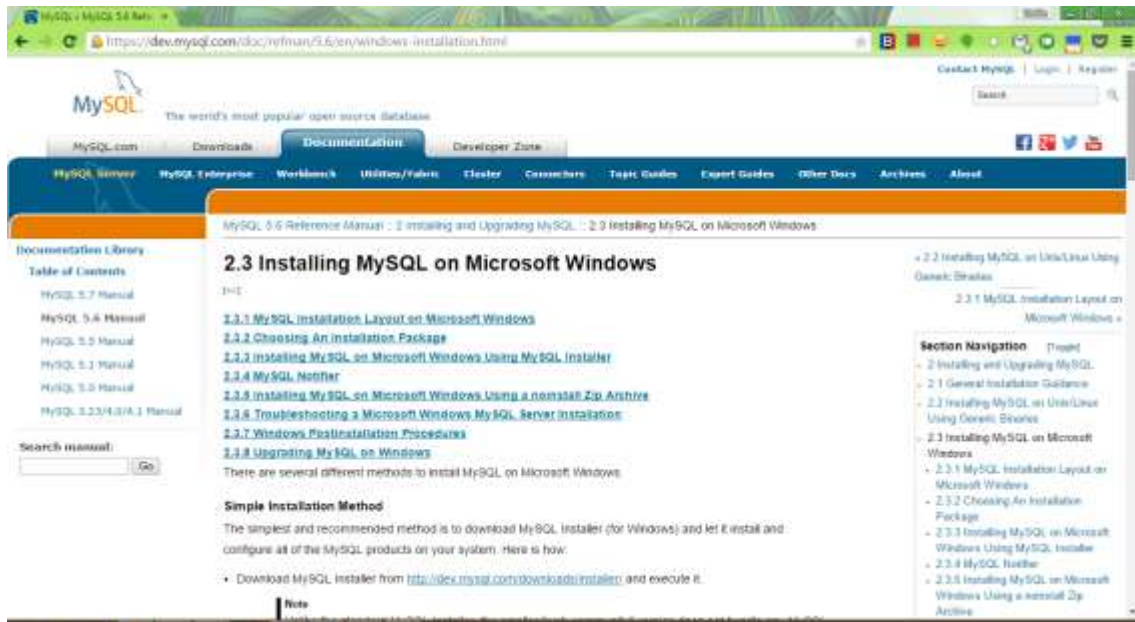


Figure 1-2. MySQL Official Documentation Regarding Installation

3. To create the schema, open the *MySQL Workbench* and establish a connection by double-clicking one of the available connections. In our example in Figure 1-3, our available MySQL Connection was **localhost**.

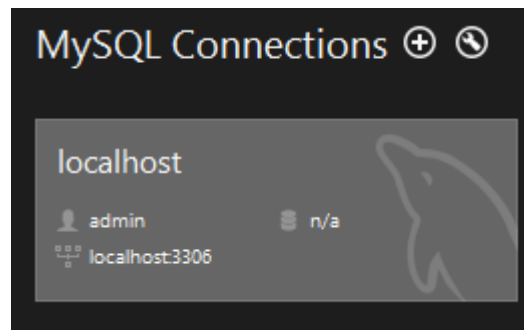


Figure 1-3. Establish a Connection in MySQL

4. Upon establishing a connection, this opens the **MySQL Editor**. In the MySQL Editor, open the *tweets.sql* located in the */SOURCE CODE/* folder (refer to Figure 1-4).

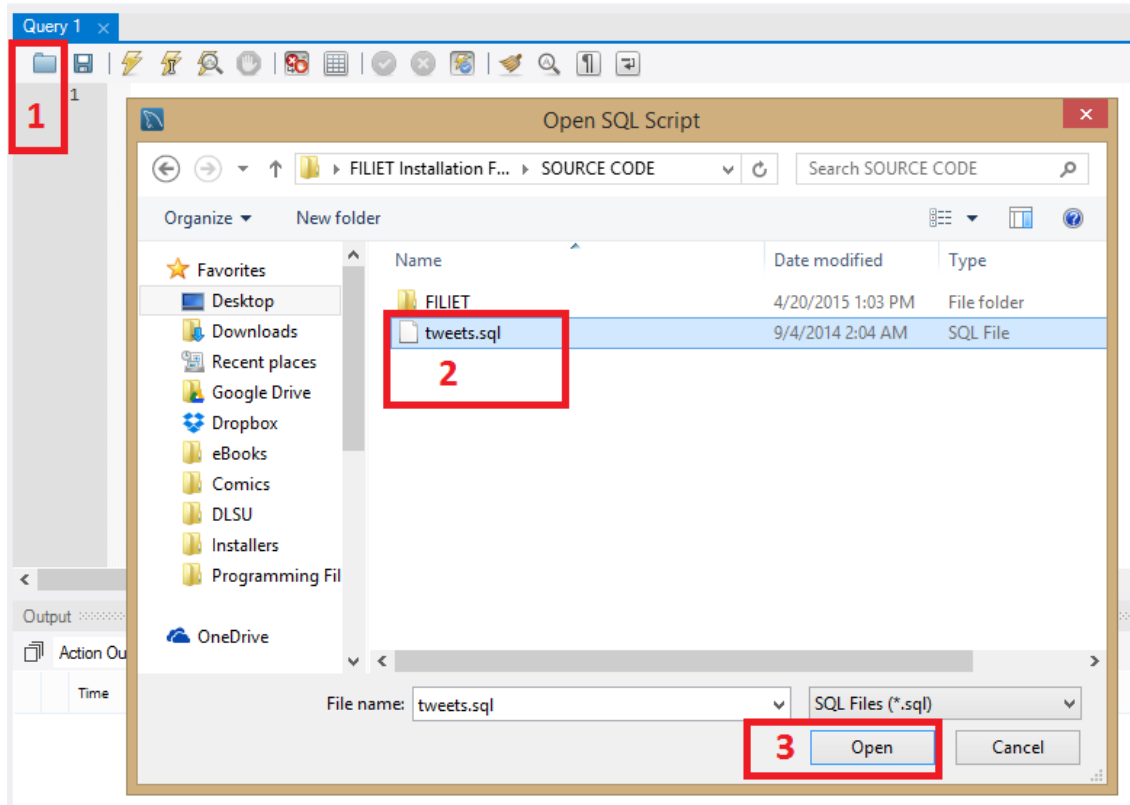


Figure 1-4. Open tweets.sql

5. The script for creating the schema for the `Crawler` Module is then loaded to the editor. To create the schema, click 'Execute' (the one with the lightning symbol) [1]. To check if the schema was successfully created, it will be shown in the **Output window pane** [2-a1] as well as in the **Navigator window pane** [2-b2] after you **refresh** it [2-b1] (refer to Figure 1-5).

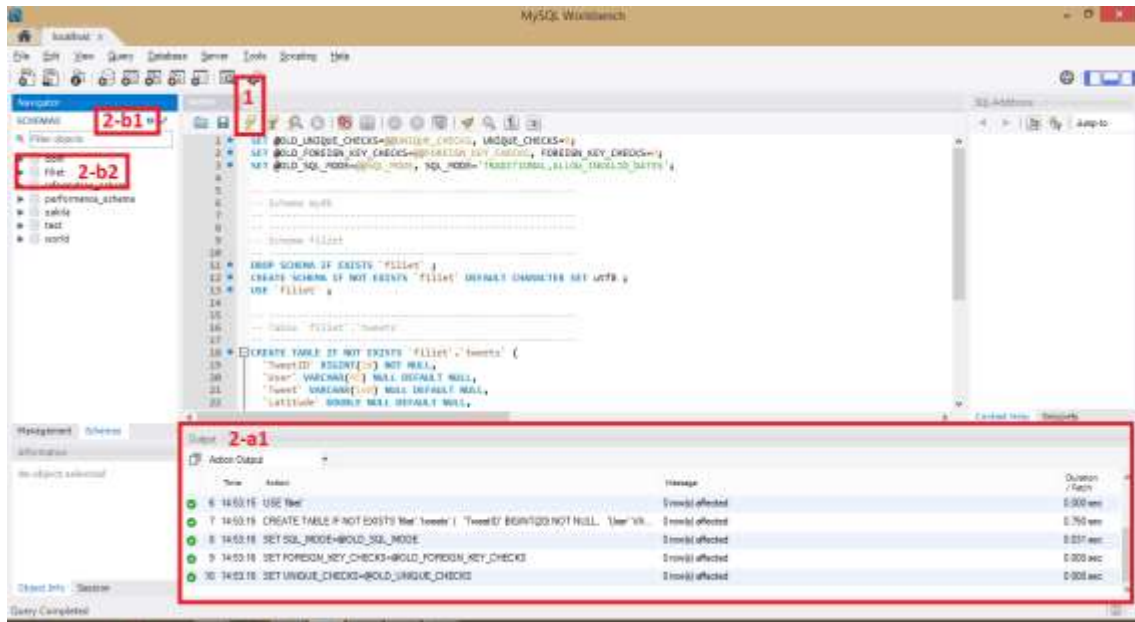


Figure 1-5. Create the schema by executing the tweets.sql script

1.3.2 NormAPI

To install NormAPI, please refer to *NormAPI User Manual.pdf*, their official documentation of NormAPI located in */TOOLS/NormAPI* (refer to Figure 1-6).

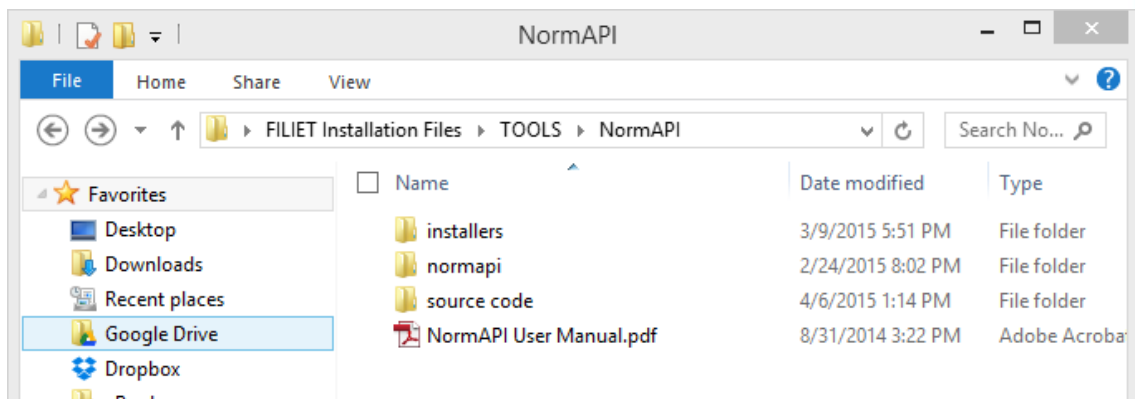


Figure 1-6. NormAPI Installation

2.0 Getting Started

2.1 FILIET Crawler Module

First, make sure that the MySQL server is running (refer to Figure 2-1).

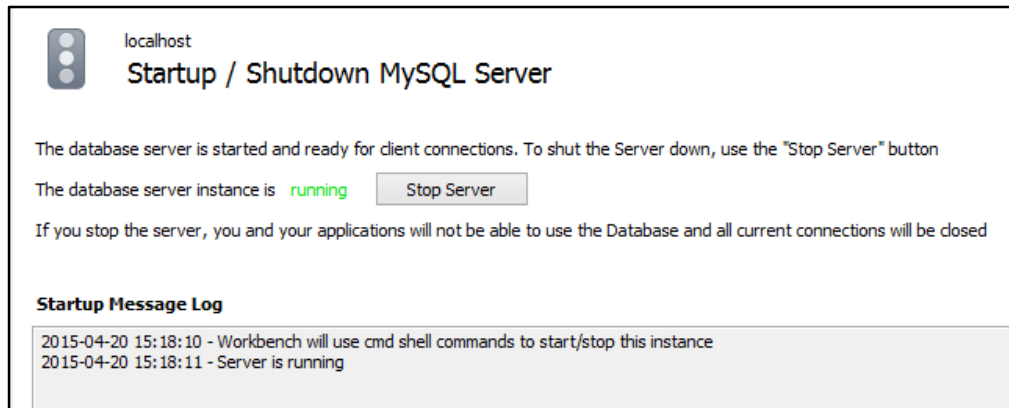


Figure 2-1. Check if MySQL server is running

List down the necessary keywords in `keywords.txt` located in `/SOURCE CODE/FILIET/resources/` folder. This will serve as the basis for the crawler in filtering the tweets it collects (refer to Figure 2-2).

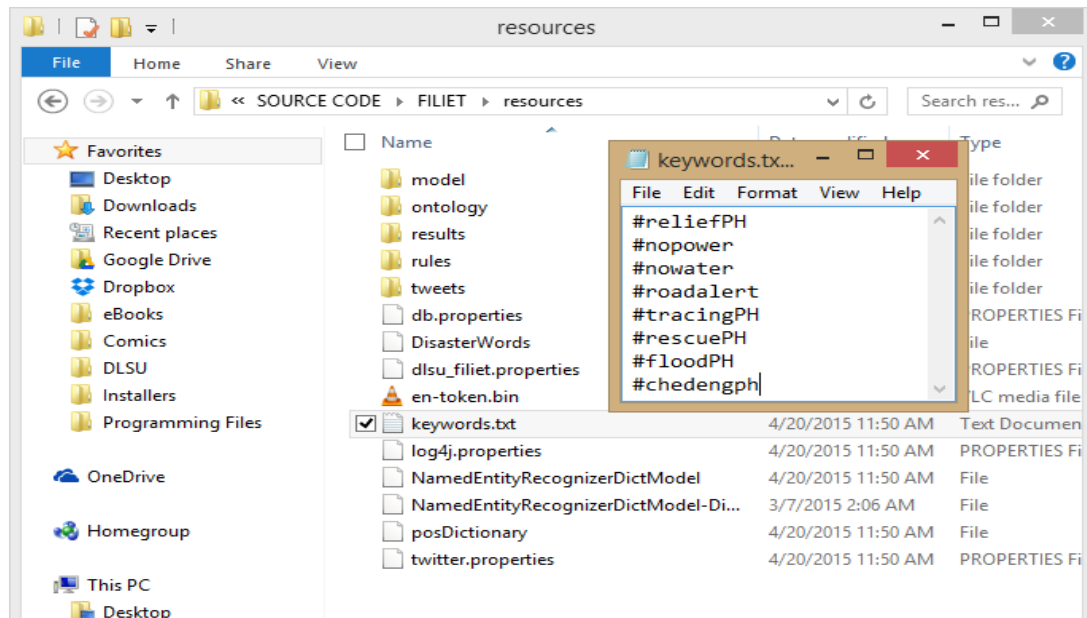


Figure 2-2. Input the keywords

Also make sure that you have properly configured the `db.properties`, also located in `/SOURCE CODE/FILIET/resources/`, so that the crawler will have access to the database (refer to Figure 2-3).

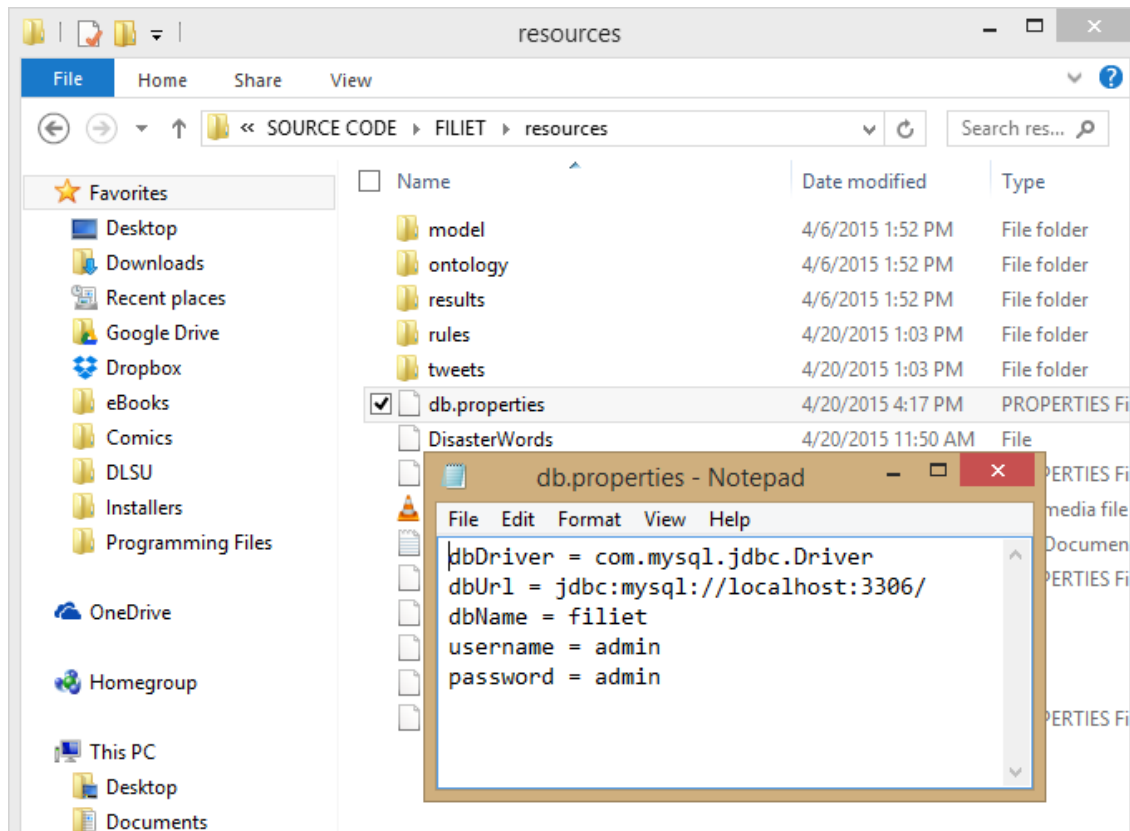


Figure 2-3. Configure `db.properties`

Open the **command prompt** and change the directory to `/SOURCE CODE/FILIET/`, where the `FILIETCrawler.jar` is located (refer to Figure 2-4).

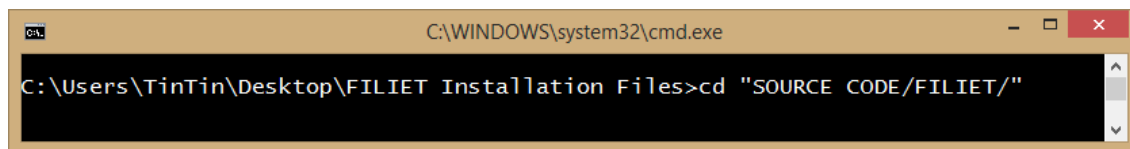
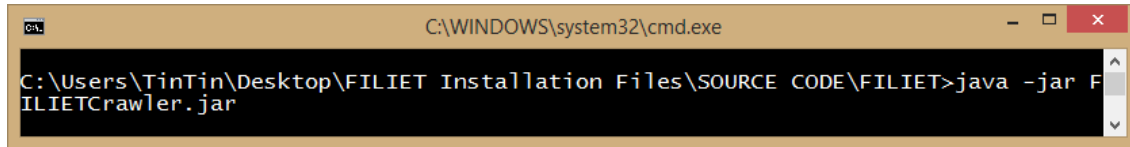


Figure 2-4. Point the directory to where the `FILIETCrawler.jar` is located

To run the crawler, enter `java -jar FILIETCrawler.jar` (refer to Figure 2-5).



```
C:\WINDOWS\system32\cmd.exe
C:\Users\TinTin\Desktop\FILIET Installation Files\SOURCE CODE\FILIET>java -jar FILIETCrawler.jar
```

Figure 2-5. Run the crawler

The following console printed in the console indicates that the crawler is functioning. The line 38551 [Twitter Stream consumer-1[Receiving stream]] DEBUG twitter4j.TwitterStreamImpl - Twitter Stream consumer-1[Receiving stream] indicates that the crawler is now waiting for tweets containing the keywords you have specified in `keywords.txt` (refer to Figure 2-6).

```
0 [Twitter Stream consumer-1[initializing]] INFO twitter4j.TwitterStreamImpl - Establishing connection.
.
.
.
2053 [Twitter Stream consumer-1[Establishing connection]] DEBUG
twitter4j.HttpClientImpl - Authorization:
*****
2058 [Twitter Stream consumer-1[Establishing connection]] DEBUG
twitter4j.HttpClientImpl - X-Twitter-Client-Version: 4.0.2
.
.
.
18939 [Twitter Stream consumer-1[Establishing connection]] DEBUG
twitter4j.auth.OAuthAuthorization - OAuth signature:
kOpqAWtoCXyb+QXvJDHUm40FIM=
18949 [Twitter Stream consumer-1[Establishing connection]] DEBUG
twitter4j.HttpClientImpl - Authorization:
*****
18956 [Twitter Stream consumer-1[Establishing connection]] DEBUG
twitter4j.HttpClientImpl - X-Twitter-Client-Version: 4.0.2
.
.
.
38551 [Twitter Stream consumer-1[Receiving stream]] DEBUG
twitter4j.TwitterStreamImpl - Twitter Stream consumer-1[Receiving stream]
```

Figure 2-6. Indicator that the crawler is running

Figure 2-7 shows two sample tweets which the crawler has detected and stored. It starts with [Twitter4J Async Dispatcher[0]] DEBUG twitter4j.StatusStreamImpl, followed by details or properties of the tweet as well as the user, followed by `onStatus @username` - The tweet itself and the tweet count for the current session of crawling. The first tweet shows a full example of what is printed in the console.

```
40210 [Twitter4J Async Dispatcher[0]] DEBUG twitter4j.StatusStreamImpl -
Received:{"in_reply_to_status_id_str":null,"in_reply_to_status_id":null,"creat
ed_at":"Mon Apr 20 08:23:03 +0000
```



```

2015","in_reply_to_user_id_str":null,"source":"<a
href=\"https://about.twitter.com/products/tweetdeck\"
rel=\"nofollow\">TweetDeck</a>","retweet_count":0,"retweeted":false,"geo":nul
l,"filter_level":"low","in_reply_to_screen_name":null,"id_str":"59006814386889
1136","in_reply_to_user_id":null,"favorite_count":0,"id":"590068143868891136","t
ext":"This is a sample tweet
#ChedengPH","place":null,"lang":"en","favorited":false,"possibly_sensitive":fa
lse,"coordinates":null,"truncated":false,"timestamp_ms":"1429518183091","entit
ies":{"urls":[],"hashtags":[{"indices":[23,33],"text":"ChedengPH"}],"user_ment
ions":[],"trends":[],"symbols":[]},"contributors":null,"user":{"utc_offset":28
800,"friends_count":589,"profile_image_url_https":"https://pbs.twimg.com/profi
le_images/2071274859/529026_3795634689062_1223046140_33798455_450424448_n_norm
al.jpg","listed_count":3,"profile_background_image_url":"http://pbs.twimg.com/
profile_background_images/258433558/165296_1892773078711_1223046140_32314952_8
340611_n.jpg","default_profile_image":false,"favourites_count":943,"descriptio
n":"Full-time DLSU student and Goodreader :D","created_at":"Tue Feb 16
10:48:12 +0000
2010","is_translator":false,"profile_background_image_url_https":"https://pbs.
twimg.com/profile_background_images/258433558/165296_1892773078711_1223046140_
32314952_8340611_n.jpg","protected":false,"screen_name":"addicteduser","id_str
":"114711441","profile_link_color":"0084B4","id":"114711441","geo_enabled":true,
"profile_background_color":"C0DEED","lang":"en","profile_sidebar_border_color"
:"C0DEED","profile_text_color":"333333","verified":false,"profile_image_url":"
http://pbs.twimg.com/profile_images/2071274859/529026_3795634689062_1223046140_
33798455_450424448_n_normal.jpg","time_zone":"Hong
Kong","url":"http://www.goodreads.com/addicteduser","contributors_enabled":fal
se,"profile_background_tile":true,"profile_banner_url":"https://pbs.twimg.com/
profile_banners/114711441/1350054705","statuses_count":14383,"follow_request_s
ent":null,"followers_count":232,"profile_use_background_image":true
,"default_profile":false,"following":null,"name":"TinTin
Kalaw","location":"Philippines","profile_sidebar_fill_color":"DDEEF6","notific
ations":null}}
onStatus @addicteduser - This is a sample tweet #ChedengPH
0

75669 [Twitter4J Async Dispatcher[0]] DEBUG twitter4j.StatusStreamImpl -
Received:{<tweet and user properties>}
onStatus @addicteduser - This is another sample tweet #ReliefPH
1

```

Figure 2-7. Sample indicator that a tweet has been stored

The tweets being collected are automatically stored in the database. To **exit the crawler**, just **exit the command prompt** or type in **CTRL+C**. To use the tweets in the FILIET system, the tweets must be exported in a CSV file and the CSV file has the following fields in the following order: TweetID, User, Tweet, Latitude, Longitude, IsURL, IsHashtag, IsRetweet, Language, and Category. The CSV file delimiter must be a semicolon (;) and enclosed with double quotation marks (" ").

2.2 FILIET

To run the FILIET System with a user interface, double click the *FILIETGUI.jar* located in */SOURCE CODE/FILIET/* (refer to Figure 2-8) or enter `java -jar FILIETCrawler.jar` (refer to Figure 2-9) in the command prompt.

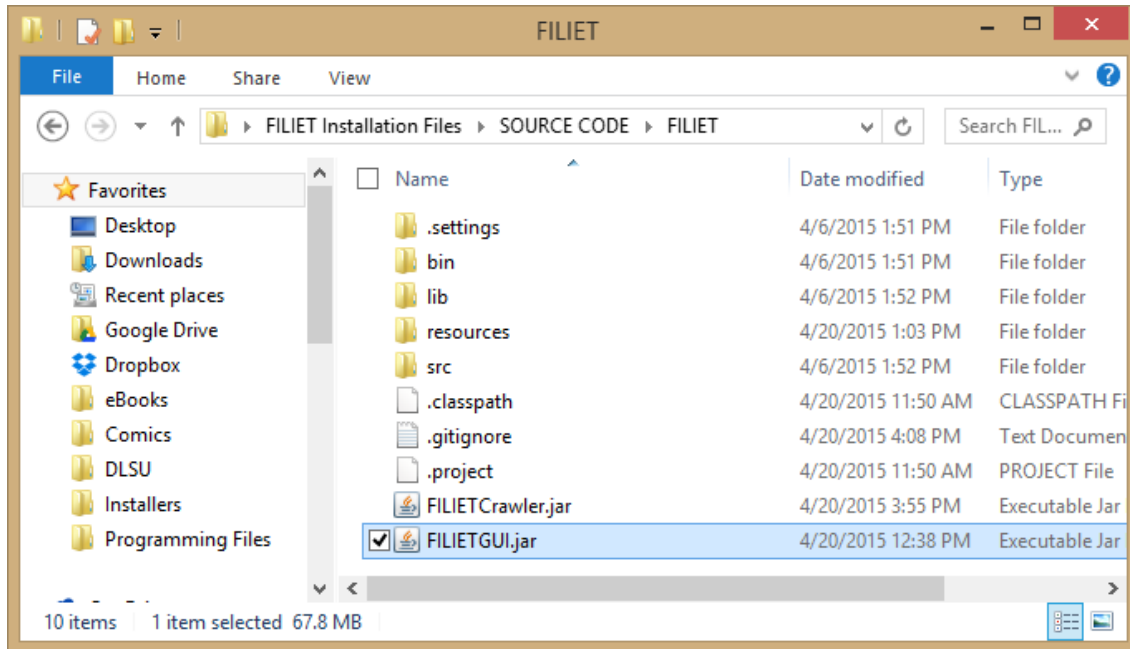


Figure 2-8. Run FILIET via Runnable JAR

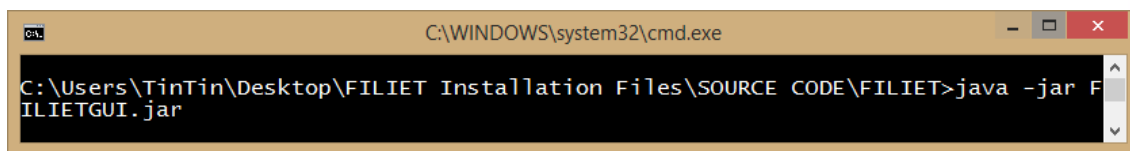


Figure 2-9. Run FILIET via Command Prompt

Figure 2-10 shows what the user interface of FILIET looks like. The Preprocessing Module, Feature Extraction Module, Classification Module, Rule Induction Module, and Ontology Module are integrated into this interface. Figure 2-11 shows the division of the user interface for further explanation.

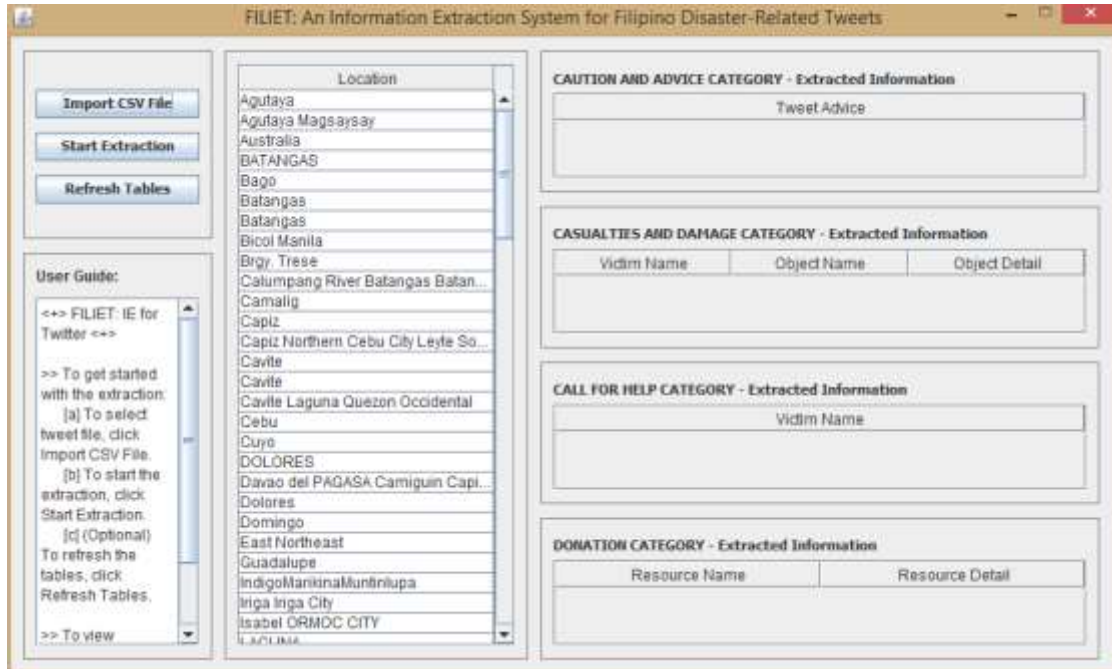


Figure 2-10. FILIET User Interface

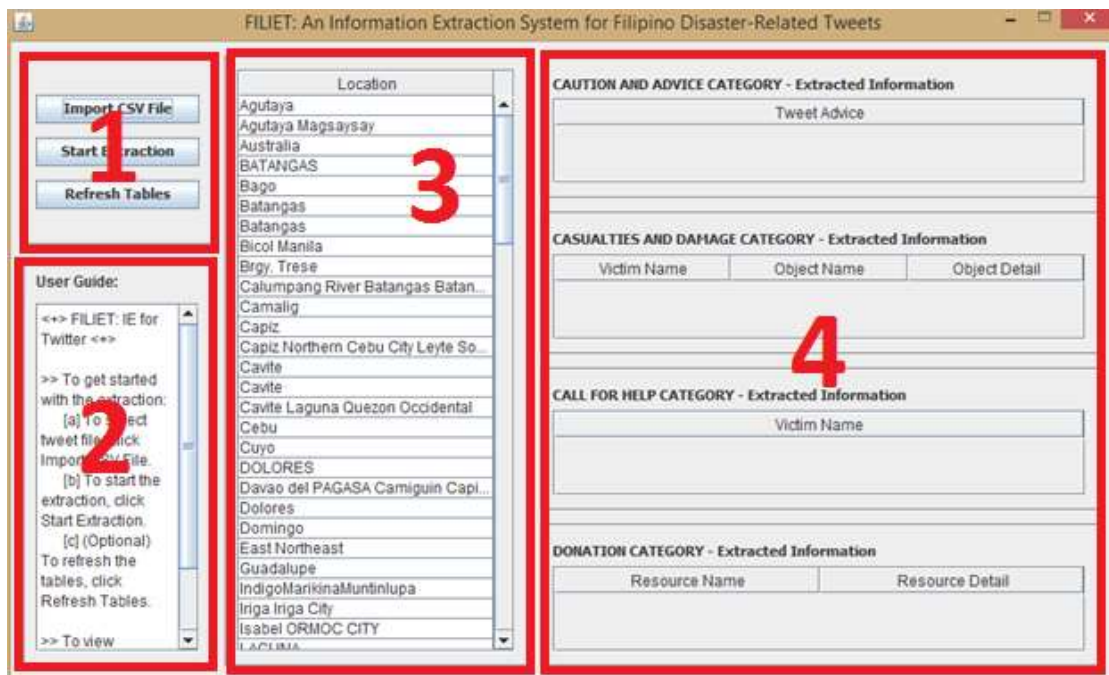


Figure 2-11. FILIET User Interface divided into sections

2.2.1 Section 1 & Section 2

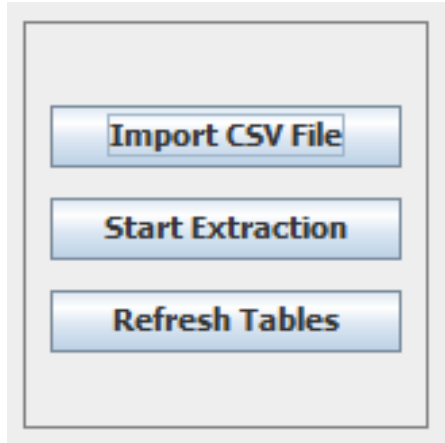


Figure 2-12. Section 1

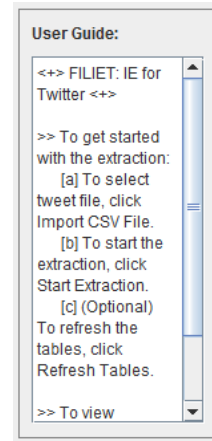


Figure 2-13. Section 2

Section 1 contains three buttons: 'Import CSV File,' 'Start Extraction,' and 'Refresh Tables' (refer to Figure 2-12). The system is basically a three-step process of import, extract, and an optional refresh. A simple user guide is stated in Section 2 (refer to Figure 2-13).

Import a CSV file by clicking 'Import CSV File'. Then select the file to open, then click 'Open'. Then start extracting by clicking 'Start Extraction', if there is an existing location list, you can click the 'Refresh Tables' to refresh the list.

2.2.2 Section 3 & Section 4

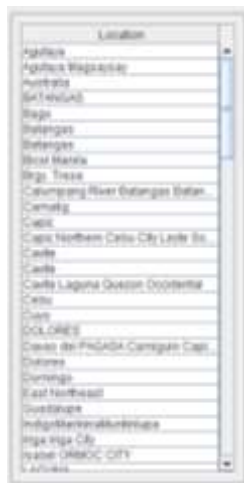


Figure 2-14. Section 3



Figure 2-15. Section 4



Section 3 (refer to Figure 2-14) contains the list of all the locations extracted in the tweets whereas Section 4 (refer to Figure 2-15) is where the extracted information will be displayed.

To view extracted information of a location, select a location from the Location list (Section 3) and then the category tables (Section 4) will display the extracted information (refer to Figure 2-16).

The screenshot shows the FILIET software interface. On the left, there are buttons for 'Import CSV File', 'Start Extraction', and 'Refresh Tables'. Below these is a 'User Guide' section with instructions. The main area is divided into two panes. The left pane, titled 'Location', contains a list of locations with 'Cavite' selected. The right pane displays extracted information for the selected location, organized into four categories: 'CAUTION AND ADVICE CATEGORY', 'CASUALTIES AND DAMAGE CATEGORY', 'CALL FOR HELP CATEGORY', and 'DONATION CATEGORY'. Each category has a table with extracted data.

CAUTION AND ADVICE CATEGORY - Extracted Information		
Tweet Advice		
Signal 1		
SIGNAL 3		
SIGNAL 3		

CASUALTIES AND DAMAGE CATEGORY - Extracted Information		
Victim Name	Object Name	Object Detail

CALL FOR HELP CATEGORY - Extracted Information	
Victim Name	

DONATION CATEGORY - Extracted Information	
Resource Name	Resource Detail

Figure 2-16. Viewing Extracted Information



3.0 Messages

Message	java.sql.SQLException: Access denied for user 'root'@'localhost' (using password: YES)
Description	Wrong username and/or password for the database access.
Action	Modify the <i>db.properties</i> located in <i>/SOURCE CODE/FILIET/resources/</i>

Message	This tweet cannot be stored.
Description	There are certain characters in Twitter such as emojis that the database could not handle.
Action	No action