

<i>Test Script Number</i>	<i>Test Description</i>
CR-001	Check if the crawler collects tweets given the specified hashtags
CR-002	Check if the tweets contained special characters/encoding
CR-003	Check if the crawler can connect to Twitter
CR-004	Check if the crawler can not connect to Twitter
<i>Test Script Number</i>	<i>Test Description</i>
FE-001	Check if the training data can be loaded into the system correctly
FE-002	Check if the features are extracted from a CSV file
FE-003	Check if the features are extracted from a single input
FE-004	Check if the features are extracted from a single input

	from a single input
FE-005	Check if the features are extracted from a single input
FE-006	Check if the training data file exists
FE-007	Check if the training data file exists
<i>Test Script Number</i>	<i>Test Description</i>
CL-001	Check if model can be read
CL-002	Check if the correct class label was specified
CL-003	Check if the model file exists
CL-004	Check if the classifier categorized the tweet correctly
CL-005	Check if the classifier categorized the tweet correctly
CL-006	Check if the classifier categorized the tweet correctly
CL-007	Check if the classifier categorized the tweet correctly

FILIET: An Information Extraction Central Testing and Validation Framework

Pack A: CRAWLER MODULE

<i>Test Input</i>	<i>Test Expected Output</i>
List of hashtags	Prints the tweet
[tweet]	This tweet cannot be stored
Access Key	The crawler is now streaming.
Access Secrets	
Consumer Key	
Consumer Secret	
Invalid Access Key	The crawler cannot connect to Twitter.
Access Secrets	
Consumer Key	
Consumer Secret	

Pack B: FEATURE EXTRACTION MODULE

<i>Test Input</i>	<i>Test Expected Output</i>
Cleaned-Categorized-JP-3.csv	Prints the values of each columns
char-ngram [file]	
wordcounts [file]	
Cleaned-Categorized-JP-3.csv	CSV file containing the extracted features
char-ngram [file]	
wordcounts [file]	
[tweet]	Prints the values of the features
char-ngram [file]	
wordcounts [file]	
[tweet]	Prints the values of the word features
Invalid char-ngram [file]	

wordcounts [file]	Features
[tweet]	Prints the values of the n-gram features
char-ngram [file]	
Invalid wordcounts [file]	
[non-existent file]	The file does not exist
Cleaned-Categorized-JP-3.csv	Load successful



Pack C: CLASSIFIER MODEL

<i>Test Input</i>	<i>Test Expected Output</i>
Model file (kNN-7.model)	Load successful
Model file (kNN-7.model)	Prints "Category"
[non-existent file]	The file does not exist
[Caution & Advice tweets]	Category: CA
kNN-7.model	
[Casualties & Damage tweets]	Category: CD
kNN-7.model	
[Donation tweets]	Category: D
kNN-7.model	
[Others tweets]	Category: O
kNN-7.model	

**n System for Disaster-Related
Validation Procedure Pack**

MODULE TEST SCRIPTS/PLANS

Test Acceptance Criteria

The tweets contained at least one of the specified hashtags

The system does not store the tweets

The system is streaming the user's timeline.

The system is not streaming the user's timeline.

IN MODULE TEST SCRIPTS/PLANS

Test Acceptance Criteria

The training data is now loaded into the system

The values of the extracted features must be correct.

The values of the extracted features must be correct.

The values of the extracted features must only be the word

features.
The values of the extracted features must only be the n-gram features.
The system displays an error message.
The system has successfully loaded the CSV file.

MODULE TEST SCRIPTS/PLANS

<i>Test Acceptance Criteria</i>
It does not produce error messages
"Category" was printed
The system displays an error message.
The classifier and the user's classification must be the same.
The classifier and the user's classification must be the same.
The classifier and the user's classification must be the same.
The classifier and the user's classification must be the same.

d Tweets

Test Steps

1. The user will provide the tweets
2. The system will listen for incoming tweets
3. The system will store the database
1. The user will provide the tweets
2. The system will listen for incoming tweets
3. The system will store the database
1. The user provides the tokens and secrets
2. The system will connect to Twitter via Twitter API
1. The user provides the tokens and secrets
2. The system will connect to Twitter via Twitter API

ANS

Test Steps

1. The user provides the path to the training data.
1. The user provides the path to the training data.
2. The system loads the model files (char-ngram and wordcounts)
3. The system extracts the word features.
4. The system extracts the n-gram features.
1. The system loads the model files (char-ngram and wordcounts).
2. The system extracts the word features.
3. The system extracts the n-gram features.
1. The system loads the model files (char-ngram and wordcounts).
2. The system extracts the word features.

3. The system extracts the n-gram features.
1. The system loads the model files (char-ngram and wordcounts).
2. The system extracts the word features.
3. The system extracts the n-gram features.
1. The user provides the path to the training data.
2. The system loads the data.
1. The user provides the path to the training data.
2. The system loads the data.
Test Steps
1. The user provides the path to the model
2. The system loads the model file
1. The user provides the path to the model
2. The system loads the model file
1. The user provides the path to the training data.
2. The system loads the data.
1. The user provides the path to the model file
2. The user provides the tweet to be classified
3. The system reads the model file.
4. The system classify the tweet
1. The user provides the path to the model file
2. The user provides the tweet to be classified
3. The system reads the model file.
4. The system classify the tweet
1. The user provides the path to the model file
2. The user provides the tweet to be classified
3. The system reads the model file.
4. The system classify the tweet
1. The user provides the path to the model file
2. The user provides the tweet to be classified
3. The system reads the model file.
4. The system classify the tweet

[illegible]

[illegible]