# Resource Gathering

Feature Words

Feature Words are words that are used for the classification module. In order to get the list of feature words, a large number of tweets is needed. The dataset contained 1028 instances. The instances is then manually classified them into the categories. Then, they were separated into files. After that, the TFIDF scores were computed for each of the word in the dataset. The top 100 scoring for each dataset is used as the feature words. If there are words that appeared in other dataset, that word is compiled into one.

Feature N-Grams

Feature N-Grams are character n-grams that are used for the classification module. The same dataset was used to collect the feature n-grams. To collect the n-grams for each dataset, the top 30 highest counts is used as the feature n-grams. If there are n-grams that appeared in other n-grams, the n-gram is compiled into one.

# Crawler Module

The crawler module is responsible for the collection of disaster-related tweets. It uses Twitter4j API, an unofficial Java library that uses Twitter API, to crawl Twitter. This module uses an authenticated account that follows users. In order to get disaster-related tweets, the official hashtags used by the government were used to filter the tweet (Appendix). The module uses Twitter’s User Stream API to continuously listen for new tweets. Each tweet is then stored into the database.

Code Listing:

// Filters

FilterQuery fq = **new** FilterQuery();

String keywords[] = {"#reliefPH","#nopower", "#nowater", "#roadalert", "#tracingPH", "#rescuePH", "#floodPH", "#queenieph"};

fq.track(keywords);

TwitterStream tweetStream = twitterStreamFactory.getInstance();

tweetStream.addListener(listener);

tweetStream.filter(fq);

Twitter4j retrieves the tweets the moment a tweet is updated in the account’s timeline. The listener is handled by the UserStreamListener, specifically the onStatus() method. The method will first receive a Status object. This contains all information regarding the tweet. Then, the following information is then processed: TweetID, User, Tweet, Latitude, Longitude, Language, IsUrl, IsHashtag, and IsRetweet to a Tweet object. After binding the Status object to the Tweet object, the tweet object handles the storing of the tweet information to the database.

Code Listing:

Tweet tweet = **new** Tweet(status);

**try** {

tweet.StoreTweet();

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

There were several issues in the crawler module. In first implementation of the crawler module was ran to collect Mario dataset, the tweets that the module was collecting were multi-lines. The collected tweet is then manually cleaned, removing line breaks. Another problem that the module faced was the amount of irrelevant tweets that the crawler is collecting. Even with using the official hashtags as a filter, the crawler is still getting numerous irrelevant tweets because people has the tendency to use the hashtags even though it is not relevant.

# Preprocessing Module

## Normalizer

## Tokenizer

This is the second step in processing the incoming tweets that will be coming from web crawler or a CSV file. The tokenizer takes a tweet and parses them into tokens. It returns a Sentence object that contains the array list of Tokens.

The Tokenizer module has two implementations: ArkNLP and OpenNLP. However, other implementations could be easily added by implementing the TokenizerInterface. The TokenizerInterface returns a Sentence object that contains the tweets.

The tokenizer from ArkNLP is called using the static method tokenizeRawTweetText(String text) from the ArkNLP Library. It will then return the tokenized tweet through an array list of string.

Code Listing

List<String> tokens = Twokenize.*tokenizeRawTweetText*(text);

For the OpenNLP tokenizer, a TokenizerModel is needed to initialize the TokenizerME.

Code Listing

FileInputStream is = **new** FileInputStream("./resources/en-token.bin");

TokenizerModel model = **new** TokenizerModel(is);

TokenizerME tokenizer = **new** TokenizerME(model);

Then, the incoming tweets will be tokenized using the *tokenize()* method. It returns an array of strings that contains the tokenized tweets.

Code Listing

tokens = tokenizer.tokenize(text);

To use the tokenizer in FILIET, a Tokenizer must be initialized first. The Tokenizer is accepting a class that implemented the TokenizerInterface. This is done so that tokenizer could be easily changed or switched in the future

Code Listing

// ArkNLP implementation

Tokenizer tokenizer = **new** Tokenizer(**new** ArkNLPTokenizerImpl());

// OpenNLP implementation

Tokenizer tokenizer = **new** Tokenizer(**new** OpenNLPTokenizerImpl());

Then, call the *executeStrategy(String)* to use the tokenizer.

Code Listing:

tokens = tokenizer.executeStrategy(normalizedTweet);

The issue of implementing the OpenNLP tokenizer is that it needed a model. There are currently no Filipino model that could be used for the tokenizer, so the group used an English model. Another issue in OpenNLP is tokenizing emoticons. It splits them into two tokens, when it should only be one. In comparison to ArkNLP tokenizer, it is very simple to use as it only needed to call a static method from the library. It is much more adept to tweets, as it can handle emoticons. Also, ArkNLP performs significantly faster than OpenNLP. Therefore, ArkNLP tokenizer was preferred.

## POS Tagger

After tokenization, each of the tokens will be tagged with its corresponding part-of-speech. The POSTagger will return the Sentence object that now contains the POS Tags. The POSTagger uses a lookup to tag each token.

The POS Tagger starts with the initialization of the POSTagger. The POSTagger needs a class that has implemented a POSTaggerInterface.

Code Listing:

POSTagger post = **new** POSTagger(**new** POSHashLookupImpl());

Then, call the executeStrategy(Sentence). The executeStrategy() will expect a Sentence object that contains the array list of tokens. The POSTagger will return a Sentence object that is now tagged with POS.

Code Listing:

tokens = post.executeStrategy(tokens);

The main issue in implementing the POS Tagger is that there are no available tools for a Filipino POS Tagger. For the POS Lookup implementation, it uses a dictionary that contains the tags for each word. However, the words found in the dictionary does not really match the tweets as the dictionary contains word from a novel, which is not suited for tweets.

## Named Entity Recognition (NER) Module

Named Entity Recognition Module accepts a Sentence objects that contains the tokenized and POS tagged tweet. Using a dictionary, it tags the words that are locations. The NER module needs an object that implements the NER Interface. This object contains the actual implementation of the NER. The SomidiaNERImpl contains the implementation of a lookup NER. It uses the dictionary from SOMIDIA.

Code Listing:

// ArkNLP implementation

NamedEntityRecognizer ner = **new** NamedEntityRecognizer(**new** SomidiaNERImpl());

After the initialization, the NamedEntityRecognizer can now be used by calling the executeStrategy(Sentence). The method returns the Sentence that is now tagged with NER.

Code Listing:

ner = ner.executeStrategy(tokens);

The main issue in this module is that it can only tagged location that contains one word. The NER module only looks one token at a time. The problems now arises from locations that contains two or more words. This leads to incomplete extraction of the location. Another problem facing the NER module is that there are locations that are considered as adjectives and vice versa. In the NER gazetteer, locations like Maginhawa and Salamat could be considered as other type. So, when the NER encountered a tweet like “Maraming salamat po”. The word Salamat will be tagged as an NER when it should not be.

## Preprocessor Manager

This module is responsible for initializing all of the modules under preprocessing module. The preprocessor manager accepts a String and then outputs a Sentence object that has been normalize, tokenized, POS tagged and NER tagged.

First, the preprocessor must be initialized. The preprocessor manager will then initialize the modules under preprocessor. For the normalizer, it will initialize a normalizer using a NormApiImpl object. The tokenizer is initialized with ArkNLPTokenizer object. The POS Tagger is initialized with POSHashLookupImpl. Lastly, the NER Tagger is initialized with SomidiaNERImpl.

Code Listing:

PreprocessorManager preprocess = new PreprocessorManager();

Then, the PreprocessorManager has a PreprocessText(String) method that accepts a String, then outputs a preprocessed Sentence object.

Code Listing:

Sentence preprocessed = preprocess.PreprocessText(String);

# Feature Extractor

The feature extractor module is to extract the features inside a tweet. The feature extractor extracts word features, and n-gram features. The feature extractor will require two files, n-gram files that contains the n-grams features and the word files that contains the words features. The feature extractor will then count how many times a word from the features appeared in the tweet. The feature extractor uses the top 30 highest count n-grams features for each category and top 100 TFIDF scores for the word features for each category, excluding stop words.

FeatureExtractor must be first initialized. It requires two parameters, the path to the n-gram files, and the path to the word files.

Code Listing:

String ngram = "./resources/model/ngram/ruby-ngram";

String word = "./resources/model/word/ruby-word";

FeatureExtractor fe = **new** FeatureExtractor(word,ngram);

After initializing, the FeatureExtractor can either do single processing or batch processing. For the single processing, the method extract(Sentence) is used. The method will be expecting a Sentence object that has been preprocessed. Then it will now output a Sentence object that now contains extracted features. For the batch processing, the method extract(String, String) is used, where the method expects the path to a CSV file that contains the tweets, and the path where the output will be saved.

Code Listing:

// Single Processing

fe.extract(sentence);

// Batch Processing

String tweets = "./resources/tweets/mario-datasets/original/ruby-dataset.csv”;

String saveModel = "./resources/tweets/test-extracted/mario-tfidf/ruby-extracted.csv

fe.extractFeatures(tweets,saveModel);

# Classifier

The classifier module categorizes the tweet into the following: Caution and Advice (CA), Casualties and Damage (CD), Donation (D), Call for Help (CH), and Other (O). The classifier uses a model in order to classify the tweets. Models are trained using Weka. The classifier module accepts a String object that has passed through the Preprocessing Module and Feature Extractor Module and returns the category of the tweet.

First, the classifier must be initialized. The classifier accepts a class that has implemented a *ClassifierInterface*. The ClassifierImpl class has two constructors. If there is no provided path, the classifier will use the default model.

Code Listing:

// Default model

Classifier classifier = **new** Classifier(**new** ClassifierImpl());

// Path to the model resource.

Classifier classifier = **new** Classifier(**new** ClassifierImpl("./resources/model/classifier/testmodel.model"));

The classifier will then initialize a ClassifierBuilder class. This class is responsible for binding the Sentence object to an Instance object that will be used for the classification. The ClassifierBuilder has two constructors. The first constructor takes no parameters. This will set the word file and the n-gram file to its default. The second constructor takes two parameters: path to the n-gram file, and to the word file.

Code Listing:

// Default model

ClassifierBuilder builder = **new** ClassifierBuilder();

// Path to the model resource.

ClassifierBuilder builder = **new** ClassifierBuilder(wordPath, ngramPath);

Then to classify the tweet, the *executeStrategy(Sentence)* method is used.

Code Listing:

String category = classifier.executeStrategy(temp);

0.015664678

# Information Extractor

Information extractor is the module responsible for extracting the relevant information from the tweets. The module accepts a preprocessed and classified tweets and outputs the sentence object with the extracted information. It uses the hand-crafted rules to extract the information.

To initialize the Rule Inductor module, the constructor accepts a single string parameter. This is the file path to the rule file.

Code Listing:

RuleInductor ruleInductor = **new** RuleInductor(rulePath);

For the rule file, the rules are categorized by into four categories. They are separated by <Category>: [category]. One rule is listed per line. Then, the <end> tag is used to end the list.

For the construction of extraction rules, each rule can consists of the following tags: string, number, pos, ner. The string tag will match to the token’s word. The pos tag will match to the token’s POS. The number tag is used to match numbers. Lastly, the ner will match to the token’s NER. To use wildcards, the key “ANY” to match any values.

Sample File:

<Category>: CA

<pos:JJ> <pos:NN> <pos:PSNS> <number:ANY>

<ner:LOCATION>[as]LOCATION

<pos:JJ> <string:#1>

<pos:JJ> <string:#2>

<pos:JJ> <string:#3>

<pos:VBZ> <string:classes> <pos:IN> <pos:JJ> <pos:VBZ>

<pos:VBP> <string:classes> <pos:IN> <pos:JJ> <pos:VBZ>

<string:#walangpasok> <pos:JJ> <pos:VBZ>

<string:signal> <pos:NN> <pos:PSNS> <number:ANY>

<string:#walangpasok> <pos:PSNS> <string:klase>

<string:#walangpasok> <string:sa> <pos:PIDP> <pos:NA> <string:antas>

<end>

The Rule Inductor module uses match(Sentence) method to apply the rules.

Code Listing:

ruleInductor.setExtractedInformation(ruleInductor.match(extractedTweet));

The implementation of the module presented numerous problems. First, the problems in the preprocessing is piling up. Incomplete and incorrect POS tags made the creation of rules difficult. Incorrect NER tagging results into incorrectly extracting information. For the classifier, incorrectly classified will result to applying the incorrect rules.

# Ontology

Appendix

Hashtags

#reliefph

#nopower

#nowater

#roadalert

#tracingPH

#rescuePH

#floodph

#marioph

#rubyph

Appendix

CA

W\_violent 20.87911787925307

W\_patayin 20.87911787925307

W\_area 20.906082434957888

W\_araw 20.94461771787911

W\_gma 20.94461771787911

W\_rains 20.94461771787911

W\_@govramil 20.94461771787911

W\_sibuyan 20.94461771787911

W\_din 20.968274407933336

W\_calauag 20.982040184373524

W\_may 21.025044612014316

W\_pero 21.041990491852765

W\_areas 21.058618433071288

W\_las 21.090468114278227

W\_erap 21.090468114278227

W\_naman 21.093616795951434

W\_tandaan 21.09600545526976

W\_lang 21.103225854742906

W\_ay 21.116261546322598

W\_hanggang 21.15061145377966

W\_jtwc 21.151794569238803

W\_navotas 21.151794569238803

W\_orange 21.151794569238803

W\_ingat 21.151794569238803

W\_kanina 21.151794569238803

W\_habang 21.151794569238803

W\_mamayang 21.151794569238803

W\_phl 21.167280610137155

W\_m 21.187935293655332

W\_hagupit 21.187935293655332

W\_bahagyang 21.19141271925574

W\_muntinlupa 21.19141271925574

W\_tuesday 21.19141271925574

W\_tubig 21.197867798817075

W\_lakas 21.202121554772337

W\_hernandez 21.207600939352925

W\_sabado 21.215616318655638

W\_ito 21.226898551578344

W\_#3 21.22848969765599

W\_km 21.22848969765599

W\_kalupaan 21.249115533666846

W\_surge 21.253475972793122

W\_capiz 21.284458796046668

W\_per 21.285431265854903

W\_@iamsumulong 21.285431265854903

W\_ticao 21.285431265854903

W\_luzon 21.29635213581166

W\_posibleng 21.29635213581166

W\_par 21.297440906044358

W\_n 21.300967842424626

W\_5am 21.300967842424626

W\_lalawigan 21.300967842424626

W\_inaasahang 21.302961558568867

W\_responsibility 21.310442160607867

W\_warning 21.310442160607867

W\_on 21.310442160607867

W\_@radyopatrol39 21.313776328320976

W\_): 21.31749655909268

W\_paalala 21.31749655909268

W\_ninyo 21.31749655909268

W\_sumusunod 21.31749655909268

W\_antique 21.31749655909268

W\_aming 21.318611563181395

W\_serbisyo 21.318611563181395

W\_numero 21.318611563181395

W\_kailanganin 21.318611563181395

W\_pag-akyat 21.318611563181395

W\_grp 21.325369990264992

W\_suspends 21.325369990264992

W\_negros 21.325369990264992

W\_tarlac 21.325369990264992

W\_@rizalgov 21.325369990264992

W\_monday 21.328737409119363

W\_umaga 21.331515403770837

W\_huling 21.331515403770837

W\_@robertmanodzmm 21.331515403770837

W\_11pm 21.331515403770837

W\_namataan 21.331515403770837

W\_@iskomoreno 21.3344466614766

W\_mamayaâ€¦ 21.504837490149264

W\_http://t.â€¦ 21.504837490149264

W\_queâ€¦ 21.504837490149264

W\_http://t.coâ€¦ 23.007381555720713

W\_http://t.co/ekfowâ€¦ 23.007381555720713

W\_piã 25.378220190135295

W\_paraã±aque 25.378220190135295

W\_â€? 26.34889093451958

W\_http://t.co/c61â€¦ 28.002653484065497

W\_camarinesâ€¦ 28.71972956893416

W\_piã±as 28.71972956893416

W\_câ€¦ 29.990112475109143

W\_http://t.co/i5ibubfnâ€¦ 30.558694882161575

W\_http://t.co/lonurxpgâ€¦ 30.558694882161575

W\_vâ€¦ 36.271042895375025

W\_http://t.co/lkvkotzxâ€¦ 39.043969867121454

W\_â€œ 39.237888841187626

W\_http://tâ€¦ 41.56067383420147

W\_http://t.câ€¦ 42.11961179761311

W\_romblâ€¦ 42.515264229775624

W\_http://t.co/â€¦ 44.03924179546624

CD

W\_galera 12.779923969230259

W\_naibalik 12.779923969230259

W\_airport 12.779923969230259

W\_#votynews 12.779923969230259

W\_dead 12.779923969230259

W\_herrera 12.779923969230259

W\_ormoc 12.779923969230259

W\_thank 13.211322591055797

W\_ayon 13.584670498681378

W\_legazpi 13.91523363537655

W\_#aksyon 14.419500657334172

W\_buong 14.557634565767035

W\_@zhandercayabyab 14.804184822851791

W\_baylon 14.988365721796793

W\_bubong 14.988365721796793

W\_ajuy 14.988365721796793

W\_utos 14.988365721796793

W\_years 14.988365721796793

W\_nasaktan 14.988365721796793

W\_old 14.988365721796793

W\_pinapakuan 14.988365721796793

W\_agbobolo 14.988365721796793

W\_@arnellozaeta 14.988365721796793

W\_causalities 14.988365721796793

W\_9,584 14.988365721796793

W\_#radyopatrol 15.388948396738515

W\_matapos 15.388948396738515

W\_odiongan 15.388948396738515

W\_press 15.388948396738515

W\_eastern 15.468282816699496

W\_katao 15.822762373529738

W\_preemptive 15.822762373529738

W\_brgy 15.822762373529738

W\_@gepelyle 15.822762373529738

W\_higit 15.822762373529738

W\_nang 16.041593592943087

W\_dilg 16.356507497847893

W\_28,000 16.356507497847893

W\_state 16.356507497847893

W\_camalig 16.356507497847893

W\_@sunstarcebu 16.356507497847893

W\_tinangay 16.356507497847893

W\_tan 16.356507497847893

W\_romblon 16.490924059619076

W\_kuryente 16.50520946752559

W\_tacloban 16.58340940659383

W\_by 16.827976478180986

W\_puno 16.827976478180986

W\_@akosijaysent 16.923884280941536

W\_northeastern 17.302356611842104

W\_samar- 17.302356611842104

W\_downed 17.302356611842104

W\_electrical 17.302356611842104

W\_@edlingao 17.302356611842104

W\_abucay 17.302356611842104

W\_nkklk 17.302356611842104

W\_topples 17.302356611842104

W\_uy-tan 17.302356611842104

W\_zhander 17.302356611842104

W\_6:00 17.302356611842104

W\_electric 17.302356611842104

W\_isinasagawa 17.302356611842104

W\_evacuation 17.48210799446251

W\_photo 17.535843167999637

W\_lumikas 17.98609544141216

W\_#cebu 18.003748032113865

W\_ann 18.003748032113865

W\_barangay 18.284295645158867

W\_nakatira 18.284295645158867

W\_pswd 18.545288004643915

W\_imprastraktura 18.545288004643915

W\_total 18.545288004643915

W\_iniwan 18.545288004643915

W\_bagsak 18.545288004643915

W\_inilikas 18.607934569160065

W\_mahigit 18.84348886290733

W\_ulat 18.93698480446941

W\_dagat 18.97307777332978

W\_taclobanon 18.97307777332978

W\_sapilitan 18.97307777332978

W\_pinalilikas 18.97307777332978

W\_komunikasyon 18.97307777332978

W\_evacuees 18.97307777332978

W\_linya 18.97307777332978

W\_pamilya 19.00882843370869

W\_catbalogan 19.195581931273235

W\_residente 19.262639720301475

W\_probinsya 19.27735831057902

W\_naitalang 19.319654207668115

W\_#aksyonsahagupit 19.348557039594567

W\_30,689 19.60921856001766

W\_post 19.60921856001766

W\_malawak 19.82734315459205

W\_casualty 20.058224314594536

W\_pananalasa 20.09273460429346

W\_video 20.25421585957266

W\_dalawa 20.628742020975263

W\_nagdulot 20.646448051992703

W\_nasawi 20.726842987965018

W\_pinsala 20.86377006268562

CH

W\_quezon 2.415553218815157

W\_via 3.25008357360453

W\_@dzbb 4.083898628001237

W\_po 4.2626798770413155

W\_ilang 4.736198448394496

W\_inilikas 6.042632833682381

W\_i 6.042632833682381

W\_@9newsph 6.295266001439646

W\_tapos 6.854354502255021

W\_pinaghahanda 7.5480289699350145

W\_maaapektuhan 7.5480289699350145

W\_pa 8.035359051319245

W\_am 8.126822771691883

W\_mauubusan 8.24117615049496

W\_pre-emptive 8.24117615049496

W\_inirekumenda 8.24117615049496

W\_near 8.24117615049496

W\_hotline 8.24117615049496

W\_patag 8.24117615049496

W\_hashtag 8.24117615049496

W\_19,000 8.24117615049496

W\_river 8.24117615049496

W\_calumpang 8.24117615049496

W\_namn 8.24117615049496

W\_peligroso 8.24117615049496

W\_evacuated 8.24117615049496

W\_bool 8.24117615049496

W\_paglilista 8.24117615049496

W\_nos 8.24117615049496

W\_marvel 8.24117615049496

W\_libanan 8.24117615049496

W\_wawa 8.24117615049496

W\_pamimigay 8.24117615049496

W\_sinimulan 8.24117615049496

W\_households 8.24117615049496

W\_water 8.24117615049496

W\_residents 8.24117615049496

W\_units 8.24117615049496

W\_taft 8.24117615049496

W\_pagtulong 8.24117615049496

W\_families 8.24117615049496

W\_#aksyonsahagupit 8.305352160984837

W\_island 8.538024889258198

W\_offices 8.715465760364589

W\_bahay 9.44335587082269

W\_@zhandercayabyab 9.60577129421655

W\_maapektuhan 9.60577129421655

W\_as 9.786630986868186

W\_lugar 10.370591514291535

W\_narito 11.226722879586054

W\_:/ 11.226722879586054

W\_bayan 11.404210347066746

W\_maaaring 11.906105926072806

W\_nakatutok 12.25724235587069

W\_http 12.460586444397796

W\_romblon 12.709269073403117

W\_tumulong 12.779923969230259

W\_nagsisiksikang 12.779923969230259

W\_supply 12.779923969230259

W\_complex 12.779923969230259

W\_evacuate 12.779923969230259

W\_@untvradio1350 12.779923969230259

W\_delpan 12.779923969230259

W\_evacuee 12.779923969230259

W\_hospital 12.779923969230259

W\_sports 12.779923969230259

W\_tumama 13.91523363537655

W\_lumikas 14.777633867935151

W\_nagsimula 14.988365721796793

W\_taga-eastern 14.988365721796793

W\_magkumpuni 14.988365721796793

W\_kanilang 14.988365721796793

W\_banta 15.023748268553774

W\_photo 15.108372282361477

W\_buong 15.217498960310474

W\_nang 15.68141664291429

W\_@akosijaysent 16.300422770030902

W\_regional 16.356507497847893

W\_@allangatus 16.503338690643297

W\_@24orasgma 16.651069376032837

W\_wala 16.67973393663493

W\_tacloban 16.813880662710172

W\_lubog 16.95589940235178

W\_evacuation 17.126091736653525

W\_kailangang 17.200608882219456

W\_highway 17.206005535451542

W\_siâ€¦ 17.295033542507273

W\_center 17.490647149520154

W\_binuksan 18.003748032113865

W\_village 18.003748032113865

W\_kapuso 18.150363179415432

W\_kuryente 18.924580994206224

W\_tablas 18.97307777332978

W\_8:30 18.97307777332978

W\_bilang 19.055679147282532

W\_kapilya 19.557409258992998

W\_ipinagagamit 19.557409258992998

W\_simbahan 19.557409258992998

W\_@\_iancruz 20.058224314594536

W\_gumaca 20.058224314594536

D

W\_tumatanggap 8.24117615049496

W\_offer 8.24117615049496

W\_bangus 8.24117615049496

W\_supermalls 8.24117615049496

W\_mgs 8.24117615049496

W\_mangudadatu 8.24117615049496

W\_victims 8.24117615049496

W\_ginawa 8.24117615049496

W\_#dswd 8.24117615049496

W\_benloi 8.24117615049496

W\_9-10 8.24117615049496

W\_australia 8.24117615049496

W\_countries 8.24117615049496

W\_court 8.24117615049496

W\_8am-6pm 8.24117615049496

W\_globe 8.24117615049496

W\_handang 8.24117615049496

W\_sagip 8.24117615049496

W\_providing 8.24117615049496

W\_donasyon 8.24117615049496

W\_sm 8.24117615049496

W\_hinihikayat 8.24117615049496

W\_maaari 8.24117615049496

W\_@enjoyglobe 8.24117615049496

W\_calls 8.24117615049496

W\_grocery 8.24117615049496

W\_naka 8.24117615049496

W\_ngiti 8.24117615049496

W\_#9newsph 8.24117615049496

W\_offers 8.24117615049496

W\_biktima 8.24117615049496

W\_bigas 8.24117615049496

W\_nailikas 8.24117615049496

W\_magbigay 8.24117615049496

W\_pbb 8.24117615049496

W\_mabibiktima 8.24117615049496

W\_united 8.24117615049496

W\_house 8.24117615049496

W\_overnight 8.24117615049496

W\_pangangailangan 8.24117615049496

W\_bgy 8.24117615049496

W\_@un 8.24117615049496

W\_rh19 8.24117615049496

W\_munang 8.24117615049496

W\_parking 8.24117615049496

W\_palo 8.24117615049496

W\_nations 8.24117615049496

W\_inihanda 8.24117615049496

W\_3/3 8.24117615049496

W\_sending 8.24117615049496

W\_ngayon 8.42634270437694

W\_offices 8.715465760364589

W\_center 8.715465760364589

W\_the 9.155137639651025

W\_#unanghirit 10.149480641391612

W\_has 10.918918895442687

W\_to 12.390056101795334

W\_ating 12.68114352067525

W\_volunteers 12.779923969230259

W\_sampaloc 12.779923969230259

W\_nroc 12.779923969230259

W\_kapamilya 12.779923969230259

W\_@anakbayan\_ph 12.779923969230259

W\_dswd 12.779923969230259

W\_point 12.779923969230259

W\_st. 12.779923969230259

W\_muling 13.071384189492413

W\_maapektuhan 13.538219321261609

W\_http://t.co/oq8rw7jwâ€¦ 13.953524163708405

W\_mahigit 14.10967901949149

W\_salamat 14.16870916513339

W\_san 14.237600391676462

W\_200,000 14.268563439747771

W\_photo 14.626378374680842

W\_free 14.988365721796793

W\_for 16.300422770030902

W\_food 17.200608882219456

W\_@dswdserves 17.302356611842104

W\_#reliefph 17.302356611842104

W\_family 17.574868470828566

W\_mateo 18.003748032113865

W\_preso 18.003748032113865

W\_@vargasmannysen 18.003748032113865

W\_ilikas 18.003748032113865

W\_packs 18.003748032113865

W\_navy 18.545288004643915

W\_municipality 18.545288004643915

W\_gym 18.545288004643915

W\_towns 18.97307777332978

W\_inihahanda 19.60921856001766

W\_relief 20.275841505975105

W\_goods 20.399035244223743

W\_transported 20.503400639927143

W\_personnel 20.503400639927143

W\_ps36 20.503400639927143

W\_agutaya 20.628742020975263

W\_magsaysay 20.628742020975263

W\_http://t.co/xckâ€¦ 23.007381555720713

W\_http://t.co/â€¦ 24.277764461895703

W\_http://t.coâ€¦ 25.378220190135295

N-Gram

CA

N\_@d 1624

N\_bu 2038

N\_er 2196

N\_ah 2417

N\_vi 2417

N\_il 2559

N\_ga 2571

N\_ro 2652

N\_ta 2744

N\_ag 3077

N\_ba 3143

N\_na 3159

N\_at 3521

N\_ru 3552

N\_by 3556

N\_:\_ 3821

N\_on 4022

N\_yp 4041

N\_ub 4165

N\_\_@ 4259

N\_.\_ 4285

N\_sa 4520

N\_ph 4625

N\_\_# 4885

N\_al 5636

N\_as 6160

N\_la 7059

N\_,\_ 7209

N\_ng 8374

N\_an 10524

CD

N\_\_b 181

N\_:/ 187

N\_tt 191

N\_ag 192

N\_at 196

N\_/t 198

N\_tp 200

N\_as 200

N\_co 201

N\_#r 206

N\_ub 209

N\_ru 210

N\_yp 211

N\_\_# 212

N\_by 217

N\_\_h 218

N\_ba 235

N\_la 238

N\_ma 243

N\_:\_ 259

N\_t\_ 260

N\_al 270

N\_na 294

N\_g\_ 312

N\_ng 373

N\_\_n 376

N\_\_s 377

N\_sa 453

N\_an 477

N\_a\_ 606

CH

N\_\_h 43

N\_vi 47

N\_al 48

N\_\_v 48

N\_\_a 49

N\_ac 50

N\_\_m 51

N\_te 52

N\_\_k 52

N\_ia 53

N\_yp 62

N\_#r 62

N\_:\_ 64

N\_by 64

N\_ub 70

N\_\_b 71

N\_on 76

N\_ru 76

N\_ba 79

N\_la 81

N\_ag 82

N\_\_n 86

N\_t\_ 86

N\_ga 93

N\_sa 96

N\_\_s 111

N\_g\_ 137

N\_ng 147

N\_an 206

N\_a\_ 252

D

N\_to 33

N\_\_h 35

N\_ha 35

N\_el 36

N\_h\_ 40

N\_yp 43

N\_t\_ 43

N\_ru 43

N\_ub 44

N\_:\_ 44

N\_ag 44

N\_by 44

N\_na 44

N\_#r 48

N\_,\_ 48

N\_\_# 52

N\_al 52

N\_li 52

N\_\_n 56

N\_ay 58

N\_\_a 61

N\_in 62

N\_ma 71

N\_g\_ 74

N\_sa 74

N\_\_m 75

N\_ph 75

N\_ng 97

N\_a\_ 114

N\_an 143