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

Ralph Vincent J. Regalado, Kyle Mc Hale B. Dela Cruz, John Paul F. Garcia,
Kristine Ma. Dominique F. Kalaw, and Vilson E. Lu
Center for Language Technologies
De La Salle University, Manila
ralph.regalado@delasalle.ph, {kyle_dela_cruz, john_paul_garcia, kristine_kalaw, vilson_lu}@dlsu.edu.ph

***Abstract* – The Philippines is considered the social media capital of the world, and the role of social media has become even more pronounced in the country during disasters. Twitter is among the many forms of social media. As experienced, information and data shared through Twitter have helped individuals, institutions, and organizations (government, public, and private) during emergency response, in making decisions, conducting relief efforts, and practically mobilizing people to humanitarian causes. However, extracting the most relevant information from Twitter is a challenge because natural languages do not have a particular structure immediately useful when programming. Another problem that information extraction faces is that some languages, like Filipino, is morphologically rich, making it even more difficult to extract information. Therefore, the goal of this research is to create Filipino Information Extraction Tool for Twitter (FILIET), a system that extracts relevant information from Filipino disaster-related tweets.**

***Keywords:* information extraction, disaster management, Twitter**

I. INTRODUCTION

According to a report [10] of the United Nations International Strategy for Disaster Reduction (UNISDR) Scientific and Technical Advisory Group, disasters have destroyed lives as well as livelihood across the world. Between 2000 and 2012, about 2 million people have died in disasters and related damages have reached an estimated US$ 1.7 trillion. In the same report, the UNISDR posits the use and research of new scientific and technological advancements in disaster management. This is where social media come in.

Social media are online applications, platforms, and media which aim to facilitate interaction, collaboration, and the sharing of content. Social media can be accessed by computers or by smart phones. In a study and analysis of [11] and [16][16] about social media, the Philippines has a high ranking in most of the categories, which led to the country being dubbed as the "*Social Media Capital of the World*". In addition to this, social media have also played a vital role in disaster management. Twitter, a popular microblogging platform where users can post statuses in real-time, is used to share information regarding disasters as well as response efforts. As part of the disaster management of the Philippines for natural calamities, the government has released a newsletter [7] containing the official social media accounts and unified hashtags to help in disaster relief efforts.

With many Filipino netizens sharing various types of disaster-related information in Twitter, it would be very beneficial to have a system that extracts relevant information from Twitter so they can be used to assist in disaster relief efforts. The challenge here is to create an information extraction (IE) system for disaster-related Twitter content which is written in the Filipino language (with respect to the TXTSPK and code-switching writing styles).

The rest of the paper proceeds as follows, Section II reviews existing literature related to our approaches. Section III introduces the main processes of our approach. Section IV describes our experiments and findings. In Section V, we conclude our efforts and discuss some future works.

II. RELATED WORKS

The works in [3] and [4] focus on the extraction of relevant information from disaster-related tweets. The approach includes text classification and information extraction. In [3], the authors worked with Twitter data during hurricane Joplin last May 22, 2011 with #joplin. They used Naïve Bayes classifiers to organize the tweets into meaningful or relevant categories of information for extraction. In [4], they used two datasets: (1) tweets during hurricane Joplin last May 22, 2011 with #joplin and (2) tweets during hurricane Sandy last October 29, 2012 with #sandy #nyc. They employed a new approach, known as Conditional Random Fields (CRF), to extract relevant information. Our work utilized the tweet categorization concept specified in [3].

For information extraction, we have reviewed various approaches used in morphologically rich languages such as the Filipino language. We determined the components of each IE system as well as the tools and evaluation metrics they have used. For [2], [12], and [14], they are machine learning-based (adaptive); [5] and

[8] are rule-based; [9] is template-based; and [6] is ontology-based. Our work focused on machine-learning and rule-based IE systems which will be displayed ontologically. An adaptive IE system uses machine-learning techniques in order to automatically learn rules that will extract certain information [13]. In [1], an adaptive IE system that incorporates the usage of rules is applied.

III. Methodology

Figure 1 shows the architectural design of the FILIET system.

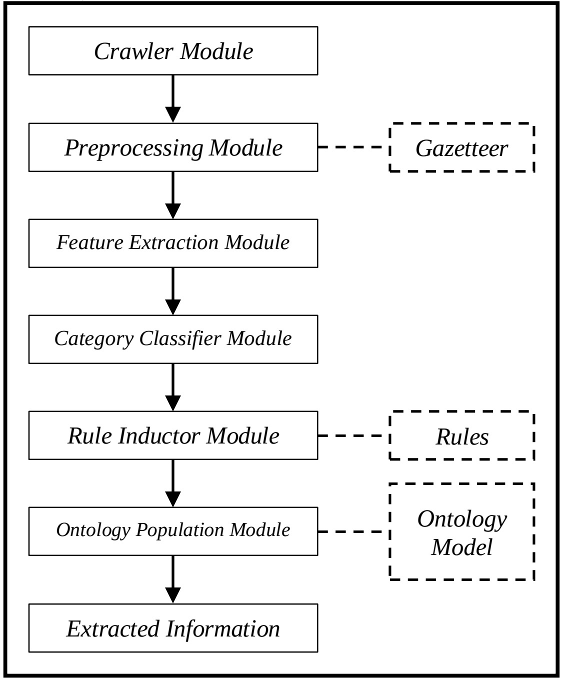

Figure 1. Architectural Design of the System

A. Crawler Module

The crawler module is for retrieving and collecting tweets using Twitter's Stream API and the Twitter4j library [15]. Figure 2 shows a sample tweet from the crawled and collected tweets of this module.

```
<tweet>
Kailangan na talaga ng military
efforts sa most part of Leyte.
Nagkakagulo na. ☹
</tweet>
```

Figure 2. Sample Tweet

B. Preprocessing Module

The preprocessing module includes the following sub-modules:

1) *Text Normalizer*: This sub-module handles the conversion of TXTSPK words to its full-word format as well as the removal of emoticons, links, and hashtags for the uniformity and consistency of the extracted information. Figure 3 shows the output of this sub-module.

```
<tweet>
Kailangan na talaga ng military
efforts sa most part of Leyte.
Nagkakagulo na.
</tweet>
```

Figure 3. Text Normalizer Output

2) *Tokenizer*: This sub-module splits the input into individual tokens which will be used for the subsequent sub-modules. Figure 4 shows the output of this sub-module.

```
<tweet>
"Kailangan", "na", "talaga", "ng",
"military","efforts", "sa", "most",
"part", "of", "Leyte", ".",
"Nagkakagulo", "na", "."
</tweet>
```

Figure 4. Tokenizer Output

3) *POS Tagger*: This sub-module tags each of the tokens with its corresponding part-of-speech. A token can be tagged as a noun, a verb, an adjective, an adverb, or other part-of-speech tags. Figure 5 shows the output of this sub-module.

```
<tweet>
"Kailangan_VOTF", "na_NA",
"talaga_IRIA", "ng_NA",
"military_NCOM", "efforts_NNS",
"sa_NCOM", "most_JJS", "part_JJ",
"of_IN", "Leyte_NPRO", "._PSNS",
"Nagkakagulo", "na_NA", "._PSNS"
</tweet>
```

Figure 5. POS Tagger Output

4) *Filipino NER*: This sub-module is responsible for identifying and tagging the proper nouns in the input. The proper nouns are identified with the use of a gazetteer. Figure 6 shows the output of this sub-module.

```
<tweet>
"Kailangan_VOTF", "na_NA",
"talaga_IRIA", "ng_NA",
"military_NCOM", "efforts_NNS",
"sa_NCOM", "most_JJS", "part_JJ",
"of_IN", "<location: Leyte/>",
"._PSNS", "Nagkakagulo", "na_NA"
"._PSNS"
</tweet>
```

Figure 6. Filipino NER Output

C. Feature Extraction Module

The feature extraction module extracts the following features from the input:

1) *Presence*: This is a binary feature that indicates the presence of keywords like disaster words, mentions, hashtags, emoticons, retweets, and also detects if code switching has occurred in the input tweet. The value of “1” is given if the keyword is present; “0” if it is absent.

2) *Tweet Length*: This feature essentially counts the length of the input tweet.

3) *N-gram*: This is mainly responsible for generating/extracting the different n-grams for the input tweets, specifically, the bi-gram and the tri-gram of the input tweets.

4) *User*: This will help in determining the type of disaster. For example, @dost_pagasa will tweet about typhoons.

5) *Location*: This feature contains the locations mentioned in the tweet.

D. *Category Classifier Module*

With the extracted features and Weka [17] as the tool, the category classifier module classifies the tweets into one of the following categories:

1) *Caution and Advice (CA)*: If a tweet conveys/reports information about some warning or a piece of advice about a possible hazard of an incident

2) *Casualty and Damage (CD)*: If a tweet reports the information about casualties or damage/s caused by an incident

3) *Donation (D)*: If a tweet speaks about money raised, donations, goods/services offered or asked by the victims of an incident

4) *Others (O)*: If a tweet cannot be classified into one of the first three categories

Figure 7 shows the output of this module.

```
<tweet type=”D”>
"Kailangan_VOTF", "na_NA",
"talaga_IRIA", "ng_NA",
"military_NCOM", "efforts_NNS",
"sa_NCOM", "most_JJS", "part_JJ",
"of_IN", "<location: Leyte/>",
"._PSNS", "Nagkakagulo", "na_NA"
"._PSNS"
</tweet>
```

Figure 7. Category Classifier Output

E. *Rule Inductor Module*

The rule inductor module applies the set of rules by looking for patterns in the text. Figure 8 shows some of the sample rules.

F. *Ontology Population Module*

The ontology population module handles the filling out of the ontology with instances that are taken from the previous model. This module receives the instances in *I*. For each instance in *I*, it will look for its matching class. If a match is found, the instance will be added to the ontology.

```
<string:
naman><disaster><string:sa> AS
Disaster
```

```
<POS: NNS><location><POS:
PSNS>AS Location
```

Figure 8. Sample Rules

IV. EXPERIMENTS

A. *Corpus*

Disaster-related tweets during typhoon Mario last September 2014 were crawled and collected. We created four corpus. The first corpus is the combined corpus. All of the categories are present in this corpus. It contains 2817 instances: 553 CA, 92 CD, 39 D, and 2133 O. This is used for the first experiment. The next 3 corpus are those that only contained two categories each, which is used for the second experiment. We created the corpus by getting the instances of the selected category and then the O category. We limited the number of instances of O based on the number of instance of the selected category. For CA corpus, it contains 1028 instances: 567 CA and 461 O. For the CD corpus, it contains 123 instances: 72 CD and 51 O. For the D corpus, it contains 199 instance: 45 D and 145 O.

For the classifier module, we tested different supervised classifier algorithms: k-Nearest Neighbors (k = 3, 5, and 7), Random Forest, J48 (Confidence Factor = 0.5), and Naïve Bayes. To measure the performance for each classifier, we used precision, recall, f-measure and kappa statistic.

B. *Experiment 1: Single Classifier*

For the single classifier, the classifier must be able to identify the tweets into the four categories (*CA*, *CD*, *D*, and *O*). The classifier is then validated using a 10-fold cross validation.

Table I lists the summary of the initial results for this experiment. The values show that Random Forest has the highest average f-measure among all the algorithms tested, while Naïve Bayes ranked the lowest. Here, we can see that the precision of each category in the random forest is higher than the rest of the algorithms. The Random Forest works best here because of the large number of attributes present in the dataset. The algorithm works by creating subsets of decision trees, then these subsets of decision trees will then classify the instance. The majority of the results of the decision trees will now be then the result. Because the trees are much smaller, they can classify more accurately, because they have less things to consider, and the results are validated by other trees. Naïve Bayes performed poorly because of the large number of attributes and instances, each of the attributes contributes to the results. There are some attributes that are not relevant to the classification, but Naïve Bayes goes through all the attributes.

Table I
Summary of Single Classifier Initial Results

Algorithm	Precision	Recall	F-measure	Kappa
kNN-3	0.888	0.894	0.889	0.7102
kNN-5	0.888	0.896	0.889	0.7099
kNN-7	0.888	0.896	0.889	0.7099
Naïve Bayes	0.87	0.831	0.846	0.6001
Random Forest	**0.898**	**0.903**	**0.895**	**0.7226**
J48	0.891	0.897	0.893	0.7209

C. *Experiment 2: Multiple Binary Classifier*

For the multiple binary classifier, each classifier will only classify two categories, either it is classified to the classifier's assigned category or it is not. If it is classified as not belonging to the category, it will cascade onto the next binary classifier until a category is chosen. If the tweet is not categorized at all, only then will it be classified as *Others (O)*. The classifier is then validated using a 10-fold cross validation.

Table II lists the initial results for the *CA* classifier. Both random forest and kNN-3 almost have equal results. We can see that kNN-3 has the higher precision on CA, but the random forest has higher recall. From the k-NN algorithm, we can see that the performance is already decreasing as the number of k increases. As for J48, although the performance is good, the kappa statistics is close to 0.5 which is low. The decision tree is almost random.

Table II
(CA) Binary Classifier Initial Results

Algorithm	Precision	Recall	F-measure	Kappa
kNN-3	0.83	0.828	0.828	0.654
kNN-5	0.829	0.827	0.827	0. 6524
kNN-7	0.814	0.812	0.813	0.6225
Naïve Bayes	0.799	0.79	0.79	0.5815
Random Forest	**0.831**	**0.831**	**0.83**	**0.6555**
J48	0.809	0.753	0.748	0.5206

Table III has the initial results for the *CD* classifier. kNN-5 is the best algorithm for classifying category CD, while J48 is the worst. Regarding the k-Nearest Neighbor algorithms, kNN-3 might not have enough neighbors that could properly classify that instance. On the other hand, when k = 7, we might be introducing noise. We can see that there is a slight decline on performance from k = 5 to k = 7. J48 is performing poorly because of the small number of instances. It could not get enough instance to build the decision tree.

Table III
(CD) Binary Classifier Initial Results

Algorithm	Precision	Recall	F-measure	Kappa
kNN-3	0.886	0.886	0.886	0.7655
kNN-5	**0.896**	**0.894**	**0.893**	**0.7791**
kNN-7	0.888	0.886	0.885	0.7614
Naïve Bayes	0.876	0.87	0.871	0.7365
Random Forest	0.837	0.837	0.836	0.6612
J48	0.779	0.78	0.779	0.5412

Table IV shows the initial result for the *D* classifier. We can see that kNN-3 has the highest performance, followed by random forest. kNN-3 has the best result because of the small number of instances in the corpus. The performance of random forest will improve as the number of instance increase.

Table IV
(D) Binary Classifier Initial Results

Algorithm	Precision	Recall	F-measure	Kappa
kNN-3	**0.91**	**0.91**	**0.91**	**0.7416**
kNN-5	0.885	0.884	0.885	0.6723
kNN-7	0.88	0.883	0.881	0.6505
Naïve Bayes	0.899	0.884	0.889	0.6961
Random Forest	0.914	0.915	0.91	0.7328
J48	0.894	0.889	0.891	0.6938

V. Discussion and Future Work

In this paper, we attempted to apply an adaptive information extraction architecture that extracts information from disaster-related Filipino tweets and displays them in an ontology. At present, the system is still being developed and we are working on the pre-processing, rule induction, and ontology modules. Only the crawler, feature extraction, and classification modules have a working output and are yet to be integrated with the rest of the modules. It is important to increase the instances in the corpus so that there will be better results for future testing.

References

[1] Cheng, H., Chua, J., Co, J., & Magpantay, A. B. (2013). Social media monitoring for disasters. Unpublished undergraduate thesis, De La Salle University, Manila, Philippines.

[2] Freitag, D. (2000). Machine learning for information extraction in informal domains. Machine Learning, 39(2-3), 169-202.

[3] Imran, M., Elbassuoni, S. M., Castillo, C., Diaz, F., & Meier, P. (2013). Extracting information nuggets from disaster-related messages in social media. *Proc. of ISCRAM, Baden-Baden, Germany*.

[4] Imran, M., Elbassuoni, S., Castillo, C., Diaz, F., & Meier, P. (2013, May). Practical extraction of disaster-relevant information from social media. In *Proceedings of the 22nd international conference on World Wide Web companion* (pp. 1021-1024). International World Wide Web Conferences Steering Committee.

[5] Lee, Y. S., & Geierhos, M. (2009). Business specific online information extraction from German websites. In Gelbukh, A. (Eds.), CICLing (pp. 369-381). Germany: Springer-Verlag Berlin Heidelberg.

[6] Nebhi, K. (2012). Ontology-based information extraction for French newspaper articles. In KI 2012: Advances in Artificial Intelligence (pp. 237-240). Springer Berlin Heidelberg.

[7] Official Gazette of the Republic of the Philippines, *Prepare for natural calamities: Information and resources from the government*, July 21, 2012.http://www.gov.ph/crisis-response/government-information-during-natural-disasters/

[8] Pham, L. V., & Pham, S. B. (2012, August). Information Extraction for Vietnamese Real Estate Advertisements. In Knowledge and Systems Engineering (KSE), 2012 Fourth International Conference on (pp. 181-186). IEEE.

[9] Poibeau, T. An Open Architecture for Multi-Domain Information Extraction. IAAI-01. Retrieved May 28, 2014, from www.aaai.org

[10] Southgate, R., Roth, C., Schneider, J., Shi, P., Onishi, T., Wengner, D., Amman, W., Ogallo, L., Beddington J., & Murray, V. (2013). Using science for disaster risk reduction. Retrieved from www.preventionweb.net/go/scitech

[11] Stockdale, C. & McIntyre, D.A. (2011, May 09). The ten nations where facebook rules the internet. Retrieved from http://247wallst.com/technology-3/2011/05/09/the-ten-nations-where-facebook-rules-the-internet/

[12] Téllez-Valero, A., Montes-y-Gómez, M., & Villaseñor-Pineda, L. (2005). A machine learning approach to information extraction. In Computational Linguistics and Intelligent Text Processing (pp. 539-547). Springer Berlin Heidelberg.

[13] Turmo, J., Ageno, A., & Català, N. (2006). Adaptive information extraction. *ACM Computing Surveys (CSUR)*, *38*(2), 4.

[14] Turmo, J., & Rodriguez, H. (2000, September). Learning IE rules for a set of related concepts. In *Proceedings of the 2nd workshop on Learning language in logic and the 4th conference on Computational natural language learning-Volume 7* (pp. 115-118). Association for Computational Linguistics.

[15] Twitter4J - A Java library for the Twitter API. (n.d.). *Twitter4J - A Java library for the Twitter API*. Retrieved July 29, 2014, from http://twitter4j.org/en/

[16] Universal McCann. (2008). Power to the people: Social media tracker wave 3. Retrieved from http://web.archive.org/web/20080921002044/http://www.universalmccann.com/Assets/wave_3_20080403093750.pdf

[17] Weka 3: Data Mining Software in Java. (n.d.). *Weka 3*. Retrieved July 15, 2014, from http://www.cs.waikato.ac.nz/ml/weka/