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 the Filipino Information Extraction Tool for Twitter (FILIET), a system that extracts relevant information from Filipino disaster-related tweets. The system consists of several modules but the experiments outlined in this paper focuses on that of the Category Classifier module where the tweets are classified into either of the four categories – Caution and Advice (CA), Casualty and Damage (CD), Donations (D), and Others (O). The experiments are to test which is the best algorithm to be used for classifying the tweets. To improve the results of the tests, it is important to increase the instances of the corpus.

Key Words: information extraction; disaster management; Twitter

1. INTRODUCTION

According to a report 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 (Southgate et al., 2013). 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 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" (Universal McCann, 2008; Stockdale and McIntyre, 2011). 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[1] containing the official social media accounts and unified hashtags to help in disaster relief efforts.

[1] 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/

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 recommendations and future works.

2. RELATED WORKS

The works of (Imran et al., 2013) focus on the extraction of relevant information from disaster-related tweets. The approach includes text classification and information extraction. In their first paper, 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 their other paper, 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 used in the first.

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. There are machine learning-based or adaptive systems (Freitag, 2000; Turmo and Rodriguez, 2000; Tellez-Valero et al., 2005), rule-based systems (Lee and Geierhos, 2009; Pham and Pham, 2012), template-based systems (Poibeau, 2001), and ontology-based systems (Nebhi, 2012). 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 (Turmo and Catala, 2006). In (Cheng et al., 2013), they make use of an adaptive IE system that incorporates the usage of rules.

3. ARCHITECTURE

Fig. 1 shows the architectural design of the FILIET system.

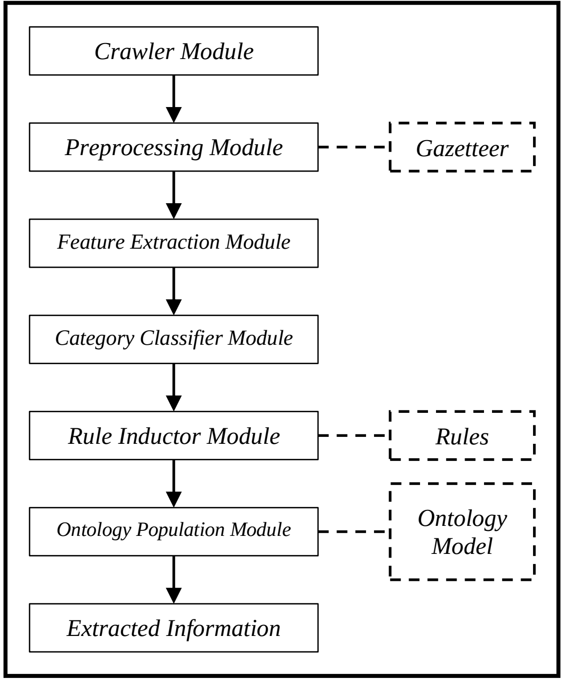

Fig. 1. Architectural Design of the System

3.1. Crawler Module

The crawler module is for retrieving and collecting tweets using Twitter's Stream API and the Twitter4j library[2]. Fig. 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>
```

Fig. 2. Sample Tweet

3.2. Preprocessing Module

The preprocessing module includes the following sub-modules:

3.2.1. Text Normalizer

[2] Twitter4J - A Java library for the Twitter API. http://twitter4j.org/en/

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. Fig. 3 shows the output of this sub-module

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

Fig. 3. Text Normalizer Output

3.2.2. Tokenizer

This sub-module splits the input into individual tokens which will be used for the subsequent sub-modules. Fig. 4 shows the output of this sub-module.

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

Fig. 4. Tokenizer Output

3.2.3. 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. Fig. 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>
```

Fig. 5. POS Tagger Output

3.2.4. 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. Fig. 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>
```

Fig. 6. Filipino NER Output

3.3. Feature Extraction Module

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

3.3.1. 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.

3.3.2. Tweet Length

This feature essentially counts the length of the input tweet.

3.3.3. 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.

3.3.4. User

This will help in determining the type of disaster. For example, @dost_pagasa will tweet about typhoons.

3.3.5. Location

This feature contains the locations mentioned in the tweet.

3.4. Category Classifier Module

With the extracted features and Weka[3]**Error! Reference source not found.** as the tool used for classification, the category classifier module classifies the tweets into one of the following categories:

3.4.1. Caution and Advice (CA)

[3] Weka 3: Data Mining Software in Java. http://www.cs.waikato.ac.nz/ml/weka/

If a tweet conveys/reports information about some warning or a piece of advice about a possible hazard of an incident.

3.4.2. *Casualty and Damage (CD)*

If a tweet reports the information about casualties or damage/s caused by an incident.

3.4.3. *Donation (D)*

If a tweet speaks about money raised, donations, goods/services offered.

3.4.4. *Call for Help (CH)*

If a tweet speaks about goods/services being asked or requesting for help.

3.4.5. *Others (O)*

If a tweet cannot be classified into one of the first three categories

Fig. 7 shows the output of this module.

```
<tweet type=”CH”>
"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>
```

Fig. 7. Category Classifier Output

3.5. *Rule Inductor Module*

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

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

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

Fig. 8. Sample Rules

3.6. *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.

4. EXPERIMENTS

4.1. *Corpus*

Disaster-related tweets during typhoon Ruby (#RubyPH and #Hagupit) last December 2014 were crawled and collected. We created four corpora. The first corpus is the combined corpus. All of the categories are present in this corpus. Because of the vast amounts of CA instances, we limited it by removing the redundant tweets so as to not overwhelm the other categories. It contains 2307 instances: 1000 CA, 202 CD, 63 CD, 4 D, and 999 O. This is used for the first experiment. The next 4 corpora are those that only contained two categories each, which is used for the second experiment. We created each corpus by getting the all instances of the selected category and then balanced it by selecting instances from the remaining categories. Take the D corpus for example. The D category has 43 instances while the other category will then have 43 instances from the CA, CD, CH, and O categories which will make a total of 86 instances in the D corpus. The CA, CD, and CH corpora have 5476, 404, and 126 instances, respectively.

For the classifier module, we tested different supervised classifier algorithms: k-Nearest Neighbors (k = 3, 5, and 7), Random Forest, and J48 (with Confidence Factor = 0.5). All of the classifiers are validated using a 10-fold cross validation. To measure the performance for each classifier, we used precision, recall, f-measure and kappa statistic.

4.2. *Experiment 1: Single Classifier*

For the single classifier, the classifier must be able to identify the tweets into the four categories (*CA*, *CD*, *CH*, and *D*).

Table 1 lists the summary of the results for this experiment. The values show that Random Forest has the highest average f-measure among all the algorithms tested, while kNN-7 ranked the lowest. 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.

Table 1. Summary of Single Classifier Results

Algorithm	Precision	Recall	F-measure	Kappa
kNN-3	0.965	0.965	0.964	0.9433
kNN-5	0.962	0.962	0.96	0.9382
kNN-7	0.95	0.952	0.949	0.9207
Random Forest	**0.972**	**0.971**	**0.97**	**0.9522**
J48	0.966	0.964	0.964	0.9417

4.3. *Experiment 2: Multiple Binary Classifiers*

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 2 lists the results for the CA binary classifier. We see that there is not much of a significant difference in the precision, recall, and f-measure of all the classifiers. The Random Forest algorithm performed the best by basing it on the kappa statistic. This means that the classification output agrees with the actual output.

Table 2. (CA) Binary Classifier Results

Algorithm	Precision	Recall	F-measure	Kappa
kNN-3	0.998	0.998	0.998	0.9961
kNN-5	0.997	0.997	0.997	0.9929
kNN-7	0.997	0.997	0.997	0.9925
Random Forest	**0.999**	**0.999**	**0.999**	**0.9976**
J48	0.998	0.998	0.998	0.9961

Table 3 shows the CD binary classifier results. Of the kNN algorithms, the one with k = 7 is the one that has the best performance with regards to precision, recall, and f-measure. However, the ones with k = 3 and k = 5 have a higher kappa than k = 7. But of all the classifiers, Random Forest performed the best while J48 performed the least.

Table 3. (CD) Binary Classifier Results

Algorithm	Precision	Recall	F-measure	Kappa
kNN-3	0.993	0.993	0.993	0.9851
kNN-5	0.993	0.993	0.993	0.9851
kNN-7	0.99	0.99	0.99	0.9802
Random Forest	**0.998**	**0.998**	**0.998**	**0.995**
J48	0.988	0.988	0.988	0.9752

Table 4 lists the CH binary classifier results. All the algorithms, except k-Nearest Neighbors where k = 7, has correctly classified all instances in the dataset. The reason as to why kNN-7 may have erred is because the size of k neighbors to compare the current instance against is large, thus, introducing noise. The high results is due to this dataset having a small number of instances.

Table 4. (CH) Binary Classifier Results

Algorithm	Precision	Recall	F-measure	Kappa
kNN-3	1	1	1	1
kNN-5	1	1	1	1
kNN-7	0.992	0.992	0.992	0.9841
Random Forest	1	1	1	1
J48	1	1	1	1

Table 5 shows the results of the D binary classifier. All algorithms have correctly classified all instances. The reason for this is due to the small number of instances in the dataset.

Table 5. (D) Binary Classifier Results

Algorithm	Precision	Recall	F-measure	Kappa
kNN-3	1	1	1	1
kNN-5	1	1	1	1
kNN-7	1	1	1	1
Random Forest	1	1	1	1
J48	1	1	1	1

5. CONCLUSION

The goal of the study is 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.

In this paper, we present the experiments we conducted for the category classification module. Based on the results, the Random Forest algorithm presents the best performance. However, there is still an issue of having a small dataset. This is attributed to the Filipinos improper use of hashtags. We have collected more garbage data than the relevant ones. It is important to increase the instances in the corpus so that there will be better results for future testing. We recommend to change the categorization of the tweets because of the difference in the way Filipinos tweet from Americans.

Future work to this study would be to extract the relevant information per category then present them in an ontology.

6. REFERENCES

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

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

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.

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.

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.

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

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.

Poibeau, T. (2001, August). An Open Architecture for Multi-Domain Information Extraction. In IAAI (pp. 81-86).

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

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/

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.

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

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.

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