FILIET: An Information Extraction System

For Filipino Disaster-Related Tweets

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

{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 a system or tool that extracts relevant information from Filipino disaster-related tweets.

*Keywords:* information extraction, disaster management, Twitter

# 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. 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 and 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.

# 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.

# Methodology

Figure 1 shows the architectural design of the system being developed.

*Crawler Module*

*Preprocessing Module*

*Feature Extraction Module*

*Category Classifier Module*

*Rule Inductor Module*

*Ontology Population Module*

*Rules*

*Extracted Information*

*Ontology Model*

*Gazetteer*

Figure . Architectural Design of the System

## 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 . Sample Tweet

## 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 . Text Normalizer Output

1. *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 . Tokenizer Output

1. *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 . POS Tagger Output

1. *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 . Filipino NER Output

## 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.

## 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 . Category Classifier Output

## 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.

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

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

Figure . Sample Rules

## Ontology Population Module

The ontology population module handles the filling out of the ontology with instances. It includes the following sub-modules:

1. *Refinements*: This sub-module is responsible for checking the instance’s uniqueness. If the instance is not found in the ontology, it will be placed in container *I*. If it is found, it will see if the instance in *I* needs to be updated. If the instance needs to be updated, it will be added in *I*. Otherwise, it will be discarded.
2. *Ontology Population*: This sub-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.

# Experiments

## Corpus (Crawler Module)

Disaster-related tweets during typhoon Mario last September 2014 were crawled and collected. We manually categorized them into one of the four categories. The corpus contained 2711 instances. The instances were categorized into one of the four categories. *CA* has 462 instances; *CD* has 77 instances; *D* has 39 instances; and *O* has 2133 instances.

## Feature Extraction Module

The feature extractor was able to extract 106 features. The features are the tweet length, presence of hashtags, URL, and links, top 50-character n-gram, and top 50 words. shows the extracted top-50 character n-grams and words.

Table I  
Top 50 Character n-grams and Words

| Top 50 Character n-gram | Top 50 Words |
| --- | --- |
| N\_ee 561  N\_hi 599  N\_io 602  N\_\_u 618  N\_en 629  N\_el 640  N\_aa 658  N\_il 664  N\_"@ 665  N\_lo 671  N\_li 673  N\_co 676  N\_it 677  N\_to 687  N\_ko 740  N\_ul 793  N\_ak 799  N\_er 799  N\_un 859  N\_di 868  N\_yo 936  N\_ph 972  N\_ra 974  N\_on 1016  N\_"r 1048  N\_\_# 1049  N\_ag 1096  N\_as 1146  N\_ri 1151  N\_ta 1226  N\_ay 1277  N\_ga 1315  N\_\_@ 1393  N\_at 1395  N\_ar 1467  N\_ka 1491  N\_al 1526  N\_:\_ 1538  N\_\_h 1547  N\_pa 1751  N\_in 2252  N\_na 2275  N\_ma 2312  N\_la 2530  N\_sa 2549  N\_ba 3041  N\_ah 3326  N\_an 3975  N\_ha 3979  N\_ng 4253 | W\_labas 53  W\_up 60  W\_http://t.co/nw0wc6vulz 63  W\_marilao 65  W\_@citizenpatrol 65  W\_grabe 66  W\_infographic 69  W\_kayo 72  W\_para 73  W\_ngayon 84  W\_paalala 84  W\_flood 84  W\_sana 90  W\_:( 90  W\_#rescueph 92  W\_daw 93  W\_haha 94  W\_kung 97  W\_marikina 113  W\_di 115  W\_sayo 122  W\_may 127  W\_#floodph 128  W\_hanggang 135  W\_@theustf 136  W\_dito 143  W\_to 144  W\_rin 147  W\_feelings 148  W\_#hugotsatagulan 149  W\_hahaha 158  W\_ka 158  W\_ako 169  W\_ba 179  W\_hindi 194  W\_lang 217  W\_ulan 223  W\_mga 245  W\_at 271  W\_ang 375  W\_? 379  W\_yung 380  W\_#marioph 388  W\_ng 578  W\_na 930  W\_rt 1080  W\_sa 1357  W\_: 1563  W\_. 1592  W\_baha 1996 |

## Category Classifier Module

For the classifier module, we tested different supervised classifier algorithms: k-Nearest Neighbors (k = 3, 5, 7, and 9), Bayesian Network, and Naïve Bayes. To measure the performance for each classifier, we used the f-measure and kappa statistic.

Two experiments were conducted in this module, single classifier and the multiple binary classifier. For the single classifier, the classifier must be able to identify the tweets into the four categories (*CA*, *CD*, *D*, and *O*). 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)*.

### Single Classifier

Table II lists the initial results for this experiment. The values show that the k-NN algorithms have a better performance than the Naïve Bayes and Bayesian Network. Although kNN-3 has the best performance, there is no significant difference among the k-NN classifiers.

Table II  
Single Classifier Initial Results

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Algorithm | Precision | Recall | F-measure | Kappa |
| **kNN-3** | **0.889** | **0.896** | **0.891** | **0.6853** |
| kNN-5 | 0.897 | 0.231 | 0.888 | 0.6823 |
| kNN-7 | 0.889 | 0.898 | 0.889 | 0.6791 |
| kNN-9 | 0.889 | 0.898 | 0.888 | 0.6731 |
| Naïve Bayes | 0.896 | 0.588 | 0.694 | 0.2638 |
| Bayesian Network | 0.866 | 0.817 | 0.837 | 0.5434 |

### Multiple Binary Classifier

Table III has the initial results for the *CA* classifier. It shows that the kNN-3 algorithm has the best performance, while Naïve Bayes has the lowest performance.

Table III  
(CA) Binary Classifier Initial Results

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Algorithm | Precision | Recall | F-measure | Kappa |
| **kNN-3** | **0.819** | **0.819** | **0.819** | **0.6373** |
| kNN-5 | 0.81 | 0.809 | 0.809 | 0.6181 |
| kNN-7 | 0.797 | 0.796 | 0.796 | 0.5925 |
| kNN-9 | 0.797 | 0.796 | 0.796 | 0.5925 |
| Naïve Bayes | 0.779 | 0.777 | 0.777 | 0.5544 |
| Bayesian Network | 0.797 | 0.794 | 0.794 | 0.5586 |

Table IV lists the initial results for the *CD* classifier. It shows that the kNN-3 algorithm has the best performance, while Bayesian Network has the lowest performance.

Table IV  
(CD) Binary Classifier Initial Results

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Algorithm | Precision | Recall | F-measure | Kappa |
| **kNN-3** | **0.908** | **0.907** | **0.907** | **0.8139** |
| kNN-5 | 0.872 | 0.87 | 0.87 | 0.7389 |
| kNN-7 | 0.882 | 0.88 | 0.879 | 0.7573 |
| kNN-9 | 0.864 | 0.861 | 0.86 | 0.7199 |
| Naïve Bayes | 0.868 | 0.861 | 0.861 | 0.7234 |
| Bayesian Network | 0.865 | 0.861 | 0.861 | 0.7228 |

Table V has the initial results for the *D* classifier. The values show that the kNN-9 algorithm has the best performance, while Naïve Bayes has the lowest performance.

Table V  
(D) Binary Classifier Initial Results

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Algorithm | Precision | Recall | F-measure | Kappa |
| kNN-3 | 0.775 | 0.774 | 0.774 | 0.547 |
| kNN-5 | 0.72 | 0.8 | 0.758 | 0.5066 |
| kNN-7 | 0.785 | 0.785 | 0.785 | 0.5688 |
| **kNN-9** | **0.796** | **0.796** | **0.796** | **0.5913** |
| Naïve Bayes | 0.712 | 0.558 | 0.475 | 0.1406 |
| Bayesian Network | 0.782 | 0.774 | 0.773 | 0.5501 |

## Ontology Population Module

For the Ontology Population Module, first, we finalized the structure of the main ontology that will be used for this system. After finalizing the ontology structure, it is then translated into system-readable format, that is, an .owl format. The system will work with the classes in conjunction with the instances that will be produced by the module before this to finalize the information extraction process.

# 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.

# 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/