FILIET: An Information Extraction System

For Filipino Disaster-Related Tweets

***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 disaster-related tweets composed in Filipino. The system was able to classify the tweets to caution and advice (CA), casualties and damages (CD), call for help (CH), and Donations (D). After classification, it was able to extract relevant information by applying rules catered to the category the tweet was classified in. The extractor could extract information such as typhoon signals, suspension of classes, casualties, damages, and items donated with a 0.4 f-measure score.**

Keywords—information extraction; disaster management; Twitter

I. INTRODUCTION (*HEADING 1*)

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 [11]. 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" [12][16]. 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.

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 them 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.

II. RELATED WORKS

The works of [5][6] focus on the extraction of relevant information from disaster-related tweets. The approach includes text classification and information extraction. In [6], 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 [5], 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 of [6].

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 [3][13][15], rule-based systems [7][9], template-based systems [10], and ontology-based systems [8]. 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 [14]. In [1], they make use of an adaptive IE system that incorporates the usage of rules.

[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/

III. METHODOLOGY

Fig. 1 shows the architecture of the FILIET system.

A. 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. The translation of the tweet is: "`Military efforts are really needed in most part of Leyte. It is getting chaotic. ☹`"

B. Preprocessing Module

The preprocessing module includes the following sub-modules. Fig. 3 shows the output of this module.

1. **Text Normalizer**. This sub-module handles the conversion of TXTSPK words to its full-word format for the uniformity and consistency of the extracted information. The text normalizer uses NormAPI [2], a normalizer that focuses on the Filipino text speak.
2. **Tokenizer**. This sub-module splits the input into individual tokens which will be used for the subsequent sub-modules. It uses ArkNLP tokenizer [4]. The ArkNLP tokenizer focuses on Tweet texts.
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. This uses a POS Lookup to tag the word with its corresponding POS Tag.
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. This uses a lookup using SOMIDIA gazetteer [1].

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 hashtags, URL, and retweets. The value of "1" is given if the keyword is present; "0" if it is absent.
2. **Tweet** Length. This feature essentially counts the number of words of the input tweet.
3. **Word**. This is mainly responsible for generating/extracting the different relevant words for the input tweets. The word features are extracted from the two datasets, the Mario and Ruby datasets. The Ruby dataset contains 2583 instances, which is composed of 1279 (CA), 245 (CD), 9 (CH), 37 (D), and 1013 (0); while the Mario dataset contains 1365 instances and is composed of 651 (CA), 99 (CD), 58 (CH), 47 (D) and 510 (O). To create the list of word features, the instances are grouped into their respective categories per dataset. From each category per dataset, the distinct words, excluding stop words, accented characters (i.e. Ã), and links are collected. From the collected words, the TFIDF score is then computed per category and the top 30% highest scoring TFIDF words serve as the word features. To further clean the list of word features, the irrelevant words are removed. This is done by having the proponents vote whether the word is deemed relevant or not. The word features used is from the Mario dataset.

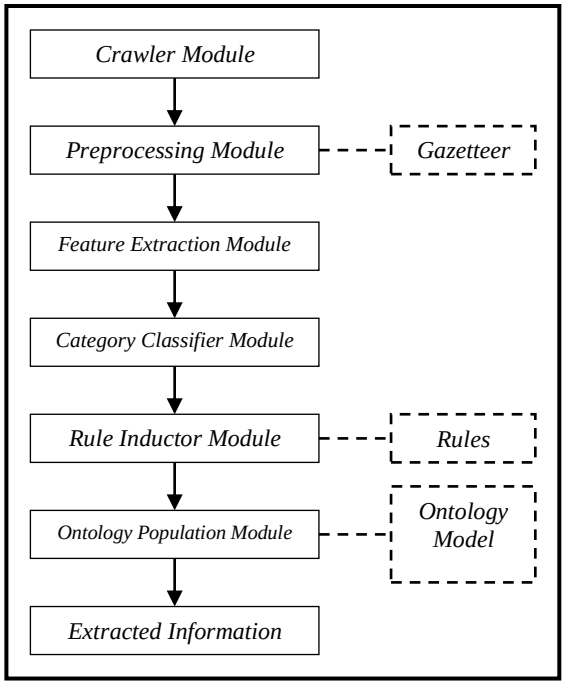

Fig. 1. FILIET Architecture

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

Fig. 2. Sample Tweet

```
<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. 3. Preprocessing Module Output

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

D. Category Classifier Module

With the extracted features and Weka[3] as the tool used for classification, the category classifier module classifies the tweets into one of the following categories. Fig. 4 shows the output of this module.

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. **Call for Help (CH)**. If a tweet speaks about goods/services being asked or requesting for help.
4. **Donation (D)**. If a tweet speaks about money raised, donations, goods/services offered.
5. **Others (O)**. If a tweet cannot be classified into one of the first three categories.

E. Rule Inductor Module

The rule inductor module applies the set of rules by looking for patterns in the text. Fig. 5 shows some of the sample rules. The Rule Inductor uses handrafted rules to extract information. The rules are classified according to the category to avoid overapplication of the rules. Most of the handcrafted rules uses POS as its marker to make sure that the rules could still be adapted to other storms. The handcrafted rules are based on the Ruby dataset.

```
<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. 4. Category Classifier Module Output

```
<string:#walangpasok>[as]ADVICE
      <pos:JJ><pos:VBZ>

      <string:lalim>[as]ADVICE
<string:ANY><string:baha>[as]ADVICE
```

Fig. 5. Sample Rules

F. Ontology Population Module

After extracting the relevant information from the tweets based on their respective categories, they are now stored to an ontology that contains object relations between the different extracted information. The actual structure of the ontology was made using an external tool called Protegé[4] that makes use of the OWL API[5]. This module takes an instance or a list of instance of categorized tweet classes. The categorized tweet classes include the following: the `CallForHelpTweet` class for containing the information that were gathered under the Call For Help category; `CasualtiesAndDamageTweet` class for containing the information that were gathered under the Casualties and Damage category; `CautionAndAdviceTweet` class for containing the information that were gathered under the Caution and Advice category; lastly, `DonationTweet` class for containing the information that were gathered under the Casualties and Damage category. This module has two sub-parts that are both responsible for storing and accessing information in the ontology. The `OntologyModule` class is responsible for storing the extracted information to the ontology and the `OntologyRetriever` class is responsible for retrieving the information that was stored in the ontology.

IV. Results

A. Corpus

The Twitter crawler was deployed during the duration of Typhoons Mario (September 9, 2014) and Ruby (December 8, 2014). From the tweets collected, two datasets were formed. The first dataset is composed of Mario tweets whereas the second dataset is composed of Ruby tweets. Both datasets were manually annotated to which of the following categories they belong to: Caution and Advise (CA), Casualty and Damage (CD), Call for Help (CH), Donation (D), and Others (O). The Mario datasets contained 655 (47.99%) retweets and 1626 (63.34%) retweets for the Ruby dataset. Less than 1% of the tweets came from official government accounts. The labels are manually annotated by the proponents. The annotation is done by having a vote. Each of the proponents classifies the instance then the majority was used as the label of the instance. To measure the annotators' agreement, kappa's statistics was used as the metrics. The Mario dataset scored 0.884 kappa, and the Ruby dataset scored 0.979 kappa. Table 1 shows the statistics of the dataset.

B. Information Extraction

The objective of the information extraction is to extract the relevant information on the tweet based on the classified category. For the CA category the following information is extracted: location mentioned in the tweet and the advice/caution. For the CD category, location in tweet, the object that is destroyed, the details of the object, and the victim's name. For the CH category, the location in tweet and the victim's name. For the D category, the location in tweet, the items to be donated and the details of the item. However, due of the low number of available instances for the CH category, no relevant information could be extracted from the instances, thus no rules were created. Table 2 and Table 3

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

[4] Protégé: A free, open-source ontology editor and framework for building intelligent systems. http://protege.stanford.edu/

[5] OWL API is a Java API and reference implmentation for creating, manipulating and serialising OWL Ontologies. http://owlapi.sourceforge.net/

shows the results of the information extraction for the Mario dataset and Ruby dataset, respectively.

TABLE I. CONTENTS OF THE DATASET

Category	Dataset	
	Mario	***Ruby***
CA	651 (47.69 %)	1279 (49.52 %)
CD	99 (7.25 %)	245 (9.49 %)
CH	58 (4.25 %)	9 (0.35 %)
D	47 (3.44 %)	37 (1.43 %)
O	510 (37.36 %)	1013 (39.22 %)
Total	1365	2583
Retweets	655 (47.99 %)	1636 (63.34 %)
Official	7 (0.5%)	10 (0.4%)
Kappa	0.884	0.979

TABLE II. RESULTS FOR THE MARIO DATASET

	Precision	Recall	F-Measure
CA-Advice	0.5593	0.3388	0.4219
CA-Location	0.6762	0.3352	0.4482
CD-Object	0.4737	0.1125	0.1818
CD-Detail	0	0	0
CD-Victim	1	0.9825	0.9912
CD-Location	0.4700	0.0803	0.1372
D-Resource	1	1	1
D-Detail	1	1	1
D-Victim	1	1	1
D-Location	1	0.2602	0.4130

TABLE III. RESULTS FOR THE RUBY DATASET

	Precision	Recall	F-Measure
CA-Advice	0.6332	0.3010	0.4080
CA-Location	0.8216	0.4454	0.5777
CD-Object	0.5693	0.3790	0.4550
CD-Detail	0.7531	0.1317	0.2247
CD-Victim	1	1	1
CD-Location	0.6274	0.5142	0.5652
D-Resource	0.9688	0.8267	0.8921
D-Detail	1	1	1
D-Victim	1	1	1
D-Location	1	0.2778	0.4348

1. **Location**. For the location, the system has a fairly low precision and recall. This is because the dataset mostly contained single word locations. For the low precision, this could be attributed to a number of reasons: (1) sometimes the location words repeat; (2) the words that were extracted were not deemed to be actual location words; and (3) there are instances in the dataset that contains multiple location words. For the low recall, this could be attributed to a number of reasons: (1) there is great diversity in how locations are named in the country. Some other location names can be mistakenly tagged as another part of speech in the tweet instance (e.g. Talisay as a noun instead of a place); and (2) the locations mentioned in the tweet instance are not actual words (e.g. Region 6, 7, 8).
2. **Advice**. The advice extracts the warnings in the tweet. It extracts information about flood levels, storm signals, class suspensions, and road blocks. The system was able to score a 0.4219 f-measure on the Mario dataset, while 0.4080 on the Ruby dataset. This is because the rules fitted on the Ruby dataset, but not on the Mario dataset. This is because the two datasets have different contents. The Ruby dataset mostly contained tweets about storm signals. However, the Mario dataset contained about road blocks and flood levels. This caused a decrease in the performance in the Mario dataset.
3. **Object**. The object extracts the objects that was destroyed. The system scored low on the extraction of the objects because of the (1) some of the details were considered as objects (2) multiple objects are presence in the tweet thus confusing the extraction. The low recall is attributed to (1) incorrectly POS tags of the word.
4. **Object Detail**. The object details extract details about the object. This extracts the status of the object. The system scored 0 in f-measure because the system extracts the detail as part of the object. Thus, no extraction for the details.
5. **Victim**. The victim extracts the name of the victim mentioned in the tweet. The system almost scored a perfect score in the Ruby dataset because there are only few instances where the names are mentioned. The Mario dataset got a perfect score because no names were mentioned. However, the system could not extract names because (1) could not be tagged by the POS tagger (2) could not be tagged by the NER tagger because of the lack of dictionary.
6. **Resource**. The resource extracts the object that was donated. The system scored 1 in f-measure because no relevant information in the tweet for the Mario dataset. For the Ruby dataset, the system extracted the common items like relief goods, food, rice and water.
7. **Resource Detail**. The resource detail extracts the details about the resource. It extracts the number of resource that was donated. The system was able to extract all the details. This is because it only extracted numbers. However, the system will have

difficulty handling spelled out numbers, which is unlikely because Twitter can only contains at most 140 characters.

V. Conclusion

FILIET was able to classify and extract relevant information from disaster-related tweets using a rule-based information extraction. The data were collected from Twitter using an external twitter mining tool called Twitter4j. After collecting the tweets, they were preprocessed through a number of techniques, specifically, normalization, tokenization, part-of-speech tagging and named entity recognition. After which, features were extracted from the tweet so that they may be used for the classification. The features that were used were the combination of features that were extracted from the Ruby and Mario dataset since they yielded the better results after numerous testing procedures. The tweets were then classified to different categories and based on the categories, specific rules are applied to extract information that was needed. Finally, the extracted information was stored to an ontology that was specifically made for this system.

However, there were numerous problems that were encountered when the system was developed. First, the lack of preprocessing tools that were built for the Filipino language posed problems for the extraction. The POS dictionary was still incomplete as there are words in the different tweet instances that could not be tagged by the system's POS tagger. Also, the NER tagger was not able to fully and correctly tag the named entities in the tweet instances because of the nature of the Filipino language, wherein, named entities can be locations and vis-à-vis. Second, the ambiguity of the tweets posed problems for the classifier as some of the tweets could fall into more than two categories. Third, a large number of tweets were collected for both typhoons. However, a large percentage of these tweets were irrelevant, i.e. not disaster-related, irresponsible use of hashtags.

VI. Recommendation

These are the recommendations that may improve the system's information extraction:

- **Improvement of the preprocessing module.** Create a more stabilize tool for Filipino (POS Tagger and NER). Add a lemmatizer submodule in order for the NER to detect multi-words location
- **Improvement of the categories.** There are ambiguous tweets that were then caused by the ambiguous nature of the Filipino language. Improvements can come from allowing multilabel classification for tweets that fall into more than one category since there have been tweet instances that really fall into more than one category when inspected manually. Furthermore, it seems that the existing labels are not enough to cover the diversity of the contents of the tweet; thus, it could be necessary to introduce more labels (inquiry, reports, and prayers) for different contents that can be made available in tweets.
- **Improvement of the rules.** Make the rules more specific. This could be done by adding rules that take advantage of specific words rather than just plainly relying on the presence of words that were tagged with certain POS markers. Another improvement that can be done is to make use of other approach in building the rules. An adaptive approach can be used to facilitate an automated process of building the rules.

Acknowledgments

We would like to acknowledge the following for being instrumental to the success and completion of this research. We would like to thank our adviser Mr. Ralph Vincent Regalado, for guiding us in every step of this research and for always stirring up our thinking process to produce ideas and insights that would largely contribute to the research; Ms. Nathalie Rose Lim-Cheng and Ms. Charibeth Cheng, for constantly providing us with opportunities to improve upon our research and for always being there to provide clarifications for the grey areas that have been encountered in this research; and Mr. Nathaniel Oco for being the best resource person for most of our needs in this research (research approaches, datasets and word gazetteers).

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] Cuevas, J., Magat, E., Nocon N., Suministrado, P. (2014). NormAPI: An API for Normalizing Filipino Shortcut Texts. Unpublished undergraduate thesis, De La Salle University, Manila, Philippines.

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

[4] Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., & Smith, N. A. (2011, June). Part-of-speech tagging for twitter: Annotation, features, and experiments. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2 (pp. 42-47). Association for Computational Linguistics.

[5] 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.

[6] 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.

[7] 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.

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

[9] 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.

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

[11] 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

[12] 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/

[13] 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.

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

[15] 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.

[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