FILIET: An Information Extraction System

for Filipino Disaster-Related Tweets

Thesis Document

Presented to

the Faculty of the College of Computer Studies

De La Salle University Manila

In Partial Fulfillment

of the Requirements for the Degree of

Bachelor of Science in Computer Science

by

DELA CRUZ, Kyle Mc Hale B.

GARCIA, John Paul F.

KALAW, Kristine Ma. Dominique F.

LU, Vilson E.

REGALADO, Ralph Vincent

Adviser

August 15, 2014

Abstract

The Philippines, being a disaster-prone country and the social media capital of the world, uses the social media to report the status of their areas, their needs, warnings and advices whenever disaster occurs. Collecting valuable information from Twitter will help organizations in making decisions. However, extracting information will difficult as natural language does not have any structures. Another problem that information extraction is facing is that some language, like Filipino, is a morphologically rich language, making it more difficult to extract information. The goal of this research is to create an information extraction system that extract the relevant information from Filipino tweets.

Keywords: Information extraction, disaster management, Twitter

Table of Contents

[1.0 Research Description 1-1](#_Toc395779197)

[1.1 Overview of the Current State of Technology 1-1](#_Toc395779198)

[1.2 Research Objectives 1-1](#_Toc395779199)

[1.2.1 General Objective 1-1](#_Toc395779200)

[1.2.2 Specific Objectives 1-1](#_Toc395779201)

[1.3 Scope and Limitations of the Research 1-1](#_Toc395779202)

[1.4 Significance of the Research 1-1](#_Toc395779203)

[1.5 Research Methodology 1-1](#_Toc395779204)

[1.5.1 Investigation and Research Analysis 1-2](#_Toc395779205)

[1.5.2 System Design 1-2](#_Toc395779206)

[1.5.3 Sprints 1-2](#_Toc395779207)

[1.5.4 Sprint Planning Meetings 1-3](#_Toc395779208)

[1.5.5 Scrum Meetings 1-3](#_Toc395779209)

[1.5.6 System Development 1-3](#_Toc395779210)

[1.5.7 System Integration and Testing 1-3](#_Toc395779211)

[1.5.8 System Evaluation 1-3](#_Toc395779212)

[1.5.9 Documentation 1-3](#_Toc395779213)

[1.5.10 Calendar of Activities 1-5](#_Toc395779214)

[2.0 Review of Related Works 2-1](#_Toc395779215)

[3.0 Theoretical Framework 3-1](#_Toc395779216)

[4.0 The System 4-1](#_Toc395779217)

[4.1 System Overview 4-1](#_Toc395779218)

[5.0 References 5-1](#_Toc395779219)

List of Tables

[Table 1‑1. Timetable of Activities (April 2014 - April 2015) 1-5](#_Toc395779147)

List of Figures

[Figure 1‑1. Research Methodology Phases 1-2](#_Toc395779220)

# Research Description

This chapter introduces the research which will be undertaken in the field of Text Classification (TC) and Information Extraction (IE) in Natural Language Processing (NLP) for disaster management. This chapter is divided into four sections. The first section will talk about the motivations and the problem that needs to be addressed. The second section will discuss the objectives of the research. The third section will discuss the scope and limitations of the study. Lastly, the fourth section will tackle the significance of the research with regards to the Philippine society.

## Overview of the Current State of Technology

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. Just between 2000 and 2012, about 2 million people died and an estimate of US$ 1.7 trillion of damage were sustained in disasters. In the same report, the UNISDR posits the use and research of new scientific and technological advancements in disaster management (Southgate et al., 2013).

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 of Universal McCann and an analysis of 24/7 Wall St., LLC about social media, the Philippines got a high rank in most of the categories. This led to the country being dubbed as the “Social Media Capital of the World” (Universal McCann, 2008; Stockdale & McIntyre, 2011).

Social media plays a vital role in disaster management. For example, after the Haiti earthquake in 2010, numerous posts and photos were published in various social media sites. 48 hours later, the Red Cross has received a donation of US$8 million. Social media has enabled the generation of community crisis maps and interagency maps, a map that works as an intermediary between the public and relief organizations (Gao, Barbier & Goolsby, 2011). Patrick Meier, a crisis mapper, makes use of social media to improve the efficiency of relief efforts. He launched the website MicroMappers[[1]](#footnote-1), that quickly sort through online data, from tweets to uploaded photos, and then display the information on satellite maps, to assist in relief efforts during the disaster of Super Typhoon Haiyan (also called Yolanda) in the Philippines (Howard, 2013).In a study commissioned by the American Red Cross[[2]](#footnote-2), it was revealed that 74% of the respondents expect response agencies to answer social media calls for help within an hour.

Twitter is a social media microblogging platform where users can post statuses in real-time. In times of disaster, Twitter is used share information regarding the disaster as well as response efforts. As part of the disaster management of the Philippines for natural calamities, the government has released an official newsletter indicating the official social media accounts and hashtags[[3]](#footnote-3). The Filipino Twitter users tend to post tweets about request for help and prayer. Other tweets pertain to traffic updates, weather updates, observations, and class suspensions. While some users have a preference to post in English, there is still a larger number of user that use their native language when tweeting during disasters (Lee et al., 2013).

## Research Objectives

This section presents the general and specific objectives of the proposed research.

### General Objective

To develop an information extraction system that extracts relevant relief effort information from Filipino disaster-related tweets.

### Specific Objectives

The following are the specific objectives of the research:

1. To review different information extraction systems;
2. To identify the different types of disaster-related tweets and the relevant information needed in relief operations;
3. To review different NLP techniques that are applicable in pre-processing Twitter data;
4. To review different approaches used in implementing an information extraction system;
5. To evaluate existing tools and resources which could be incorporated in the information extraction components of the system;
6. To determine the metrics for evaluating the information extraction system;

## Scope and Limitations of the Research

The research aims to design an information extraction system for the Filipino language. It will cover the review of various information extraction systems in order to know the different approaches on implementing them. Different existing domain-independent, domain-dependent information extraction systems will be reviewed in order to understand the architectures, implementation and components of an information extraction system. It will also review information extraction for MRL in order to understand the techniques used to extract from MRL since the Filipino language is considered to be an MRL.

In order to for the system to extract relevant information, the research must first determine which information are deemed relevant in times of disaster, especially in relief operations. Also, the research must also determine the different types of disaster-related Tweets as this will help in determining the relevant information from the given tweets. To do that, different information extraction systems that work with disaster-related domains shall be reviewed and evaluated. Also, other researches about the use of Twitter in disaster management shall be reviewed and evaluated to help the researchers in formulating the ontologies of the information extraction system to be developed.

In order for the information extraction system to perform better, the research will review different natural language processing techniques that will preprocess the data before feeding it to the information extraction system. Examples of the NLP techniques that will be reviewed are text classification and text normalization. Text classification is the process of automatically assigning a text or document into a predefined category based on their content (Özsu & Liu, 2009). Texts may need to be classified according to categories so that the system can use appropriate algorithm to extract the information. Text normalization is the transforming of ill-formed words into their canonical forms (Han & Baldwin, 2011). The information extraction system will need a text normalizer as data coming from Twitter are noisy. Most of the text has no structure, incorrectly spelled words, and invented terms.

The research will review different information extraction techniques that will be used for the information extraction systems. Some of the techniques that will be reviewed are Named Entity Recognition (NER), lexical analysis, and conference analysis. Lexical analysis involves splitting up sentences into words and performing Part-Of-Speech tagging to each word (Grishman, 1997). NER is the classification of each word into a category (Zhou & Su, 2002). Coreference analysis is the resolving of references for the pronouns (Grishman, 1997).

Existing tools that will be used in building the information extraction system will be reviewed and evaluated. Examples of NLP tools are OpenNLP and Lingpipe. OpenNLP is a machine learning based toolkit for the processing of natural language text that can support a number of common NLP tasks like tokenization, sentence segmentation, part-of-speech tagging, named entity extraction, chunking, parsing, and co-reference resolution (Apache Software Foundation, 2010). On the other hand, Lingpipe is a toolkit for processing text using computational linguistics that can perform certain tasks like finding names of people/organizations/event, classify Twitter data, and check spellings (Alias-I, 2011).

In order to evaluate the information extraction system, the research will determine the different metrics that can be used to measure the system’s performance.

## Significance of the Research

Being the social media capital of the world, the Philippines generates a lot of diversified information that cannot be easily tapped because of the limited capabilities and tools that are available in processing the language unto which these information are written in, the Filipino language. And with Twitter being one of the most commonly used social media platforms in the country, a new level of information dissemination has been established. With an information extraction system that is specifically built for the Filipino language and at the same time for supporting texts that are found in Twitter, respective stakeholders can explore more possibilities and opportunities with regards to effectively utilizing this information from the web with regards to using them for disaster management purposes.

In the disaster management standpoint, there are a number of advantages to having an information extraction system that is specifically made to work with Twitter texts that are written in the Filipino language.

First, respective stakeholders can collect disaster-related information in a way that is less strict because with an information extraction system specifically built for the Filipino language, these stakeholders can effortlessly accept and process information that are written in a much more natural and open way. With this, they can reach out to more people and to more places because they can have a system that can extract information from how Filipinos speak and communicate through the different social media platforms available, and to be specific, in Twitter.

Second, with an information extraction system, respective stakeholders can easily make use of the information that are written in the format of the different variations in the Filipino language like the ‘TXTSPK’ and ‘Code Switching’. With a custom-built information extraction algorithm, the information extraction system will be able to increase the probability of accurately and precisely extracting relevant information from the normal Filipino language and at the same time, include support for the different variations like the ‘TXTSPK’ and ‘Code Switching’.

Third, the information that can be extracted from Twitter can be further utilized to help in disaster relief efforts. With a system that can further categorize tweets automatically can help in extracting more straightforward and meaningful information about the current state of disasters. Certain types of tweets can indicate a specific set of relevant information that can be extracted. Take, for instance, Disaster Information Tweets. Information that can be extracted from this kind of tweets can include, but not limited to, the type of disaster, location of disaster and etc. Or take, for instance, Casualty Report Tweets. Information like the number of casualties or the names of missing people can be extracted from this type of tweets.

Lastly, with can information extraction system that can organize the extracted relevant information, respective stakeholders can now expedite the process of conducting relief operations since they can be presented with information that has already been processed to be easily read and understood by the normal people. With this information extraction system, the process of consolidating necessary relevant disaster-related information can be more intuitive and faster.

## Research Methodology

This section discusses the different activities that will be performed throughout the research. Scrum-based methodology, an iterative software development life cycle, will be applied in the course of this research in order to ensure that the research will be able to adapt to changes in requirements.

Figure 1‑1 shows a diagram of the phases the research will undergo. The phases are as follows: investigation and research analysis, system design, system development, system integration and testing, system evaluation, and documentation. Regular consultation with the thesis adviser will also be conducted in order to keep the research on track for the whole duration of the thesis.

Figure 1‑1. Research Methodology Phases



### Investigation and Research Analysis

This phase involves the study and understanding of the fundamental knowledge of the concepts, algorithms, techniques, and tools which can be used to implement the system as well as identifying the modules and requirements of the system to be developed. The main key activity involved in this phase is various literature reviews of related works. From those related works, the pre-processing techniques, information extraction techniques, tools, and evaluation metrics used are identified. The listed techniques, tools, and metrics are then compared and evaluated to see which ones can be adopted to the system.

### System Design

In this phase, the system will be designed according to the information gathered during the course of the Investigation and Research Analysis phase. It is in this phase where appropriate architectures, algorithms, information extraction techniques, and other necessary tools shall be identified so that they can be effectively utilized in the making of the system. Also, it is in this phase where necessary modules for the system will be identified based on the different processes and features that will be built into the system. This phase will cover the designs of the User Interfaces and the basic architecture for the databases that will store the data that will be gathered and used by the system. Finally, this phase will also cover the identification of the source of the data that will be used and processed by the system. And once the data sources have been identified, data collection will immediately commence.

### Sprints

A two-week timeframe for each sprint will be used. This is to ensure that there is progress in the research. Each member is expected to produce a working output based on the tasks assigned to him during the sprint planning meetings. The tasks may vary from developing a part of the system or to conduct further study regarding a certain concept.

### Sprint Planning Meetings

At the beginning of each sprint, a sprint-planning meeting is conducted. Tasks that must be accomplished for the current sprint will be discussed here. Included in these meeting is the assignment and division of the tasks among the members of the team. Also, the evaluation of the tasks in the previous sprint is done here. If there are any unmet tasks, these will be carried over to the next sprint.

### Scrum Meetings

Scrum meetings of 10-15 minutes in duration will be conducted daily. The purpose of this is to update each member what has or has not been accomplished yet in the assigned task. This ensures that there is daily progress and if there are issues that hinder a member from accomplishing his assigned task.

### System Development

In this phase, actual development of the system will be done. It will follow the design made during the System Design phase. Data collection will also be done in this phase. Each member of the team will be assigned to modules. The development of the system will follow a scrum-based methodology wherein the system is developed in an iterative manner. Daily and weekly meetings, as well as regular consultations with the adviser, are conducted in order to assess the progress of the thesis and to plan the succeeding tasks.

### System Integration and Testing

In this phase, all the modules that have been developed during the System Development phase will be integrated into one system. This phase will cover unit testing processes for each module to ensure that there will be no significant bugs that can be found after integration processes are completed. After finishing the integration process, the system will be subjected to another round of tests to check for any faulty integration and bugs that may have arose during the integration process.

### System Evaluation

In this phase, the system's performance will be evaluated based on the metrics that were chosen. As of the moment, the metrics that will be used in this phase will be the Precision, Recall and F-measure results of the information extracted by the system. The information that were extracted by the system will be subjected to a number of tests that will test its Precision, Recall and F-measure when compared to the information that were extracted manually and to those that are extracted from the training set. Although, the set of metrics that will be used might change during the course of the research as these metrics will be modified to fit the needs in accurately measuring the performance of the system to be developed.

### Documentation

Every activity or methodology that is performed will be fully documented so that they can be monitored when it comes to the modifications and progress that are made in accomplishing the documents and the system proposed in this research. Also, the documentation will be used for further references, in case there is a need to validate or cross-reference any future work that is in mind.

### Calendar of Activities

Table 1-1 shows a Gantt chart of the activities for the thesis period. Each bullet represents one week worth of activities

Table 1‑1. Timetable of Activities (April 2014 - April 2015)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Activities** | **Apr**  **(2014)** | **May** | **Jun** | **Jul** | **Aug** | **Sept** | **Oct** | **Nov** | **Dec** | **Jan** | **Feb** | **Mar** | **Apr**  **(2015)** |
| Investigation and Research | \_ \* \_ \* | \_ \_ \* \* | \_ \* \* \* | \* \* \* \* | \* \* \* \* |  |  |  |  |  |  |  |  |
| System Design |  |  |  |  | \_ \_ \* \* | \_ \* \* \* | \* \* \* \* | \* \* \* \* | \* \* \_ \_ | \_ \* \* \* | \* \* \_ \_ |  |  |
| System Development |  |  |  |  | \_ \_ \* \* | \_ \* \* \* | \* \* \* \* | \* \* \* \* | \* \* \_ \_ | \_ \* \* \* | \* \* \_ \_ |  |  |
| System Integration and Testing |  |  |  |  | \_ \_ \* \* | \_ \* \* \* | \* \* \* \* | \* \* \* \* | \* \* \_ \_ | \_ \* \* \* | \* \* \_ \_ |  |  |
| System Evaluation |  |  |  |  | \_ \_ \* \* | \_ \* \* \* | \* \* \* \* | \* \* \* \* | \* \* \_ \_ | \_ \* \* \* | \* \* \_ \_ | \* \* \* \* | \* \_ \_ \_ |
| Documentation | \_ \* \_ \* | \_ \_ \* \* | \_ \* \* \* | \* \* \* \* | \* \* \* \* | \* \* \* \* | \* \* \* \* | \* \* \* \* | \* \* \_ \_ | \_ \* \* \* | \* \* \* \* | \* \* \* \* | \* \_ \_ \_ |

# Review of Related Works

This chapter discusses the features capabilities, and limitations of existing research, algorithms, or software that are related or similar to the thesis.

## Machine Learning-Based Information Extraction Systems

This part discusses information extraction systems that use machine learning-based techniques.

##### Machine Learning for Information Extraction in Informal Domains (Freitag, 2000)

The researchers of the paper explored one of variation of the slot-filling problem and that is to find the best-unbroken fragment of text to fill a given slot in the answer template. There is a definite template that is given to an IE task. The template consists of fields that need to be filled with instances from the text source. The researchers set two ways of simplifying how to study the behavior of the algorithms to be developed: to isolate each field learning problem and focus on fields that is not instantiated or have a unique instance in a text source. With this, they found two primary aspects: multi-strategy learning and feature engineering. Multi-strategy learning because they believed that there is no single representation for all IE problems. Feature engineering because ML of a feature set is needed to help adapt to domains containing novel structures since they will target informal domains. The researchers used four ML components: rote learning, term-space learning, learning abstract structure with grammatical inference, and relational learning for information extraction. They did experiments to gauge the performance of the four learners.

To conclude, the researchers found out that it is possible to perform IE from informal domains found in the internet. Also, they stated that ML is a rich source of ideas for different algorithms that can be trained to perform IE. They have shown that with the right ML techniques it is possible to train effective extractors with very simple document representations.

##### TOPO - Information Extraction System for Natural Disaster Reports from Spanish Newspaper Article (Téllez-Valero, 2005)

This information extraction system extracts information related to natural disasters from newspaper articles written in Spanish. The system extracts the following information: (1) information related to the disaster itself (date, place and magnitude), (2) information related to buildings (number of destroyed buildings, affected houses), (3) information related to people (number of dead, missing or wounded), (4) information related to infrastructure (number of affected hectares, economic lost). It is able to extract information on natural disaster like hurricanes, forest fires, inundations, droughts and earthquake.

The system uses general information extraction system architecture. First, the document is turned into Boolean vectors representing the presence and absence of certain words. This is the document feature extraction stage. In order to limit the dimension, they used information gain technique. After it is turned into a Boolean vector, it will now be classified. They used Support Vector Machine (SVM), Naïve Bayes (NB), C4.5, k-Nearest Neighbors (kNN). After it has been classified, it needed to select text that might contain relevant information. This is the candidate text selection stage. They used grammar to select the text and a dictionary of names and number to treat grammar exceptions. Then the output will be candidates of relevant information. Then, the system will now select which of the information will be used. This uses the same algorithms in the text classification stage. They used different classifier for different output.

This architecture boasts its portability because it is language independent and domain adaptive. It is language independent because its training features and candidate text segments are based on simple lexical rules. It is domain adaptive because it only needed to change the training corpus.

The text filtering stage was evaluated on 134 news reports on the metrics of precision, recall and F-measure. The algorithm that produced the best result was the SVM. They got an F-measure from 72% to 88% on classification of news reports. The information extraction stage was evaluated on 1353 text segments that consist of names, dates and quantities randomly taken from 365 news reports. The best classifier for name and quantities was SVM, while kNN for dates. The overall system got an average of 72% on F-Measure.

## Rule-Based Information Extraction Systems

This part discusses information extraction systems that use rule-based techniques.

##### Vietnamese Real Estate (VRE) Information Extraction (Pham & Pham, 2012)

The Vietnamese Real Estate (VRE) Information Extraction system extracts information from Vietnamese Real Estate Advertisements. It collects information like the type of estate, category of the estate, area, zone, price, name of the author, and contact details. The system uses the GATE framework for its architecture.

For its data, it has to pass certain criteria before it is fed into the system. First, it must be news articles related to real estate advertisement. Second, only one advertisement from each input data file. Lastly, it must be strip off of all its HTML tags. After the data has met all the criteria, it will now go to data normalization first. The data normalization helps reduce ambiguity and helps the human in annotation. First, it must add the necessary punctuation at the end of the sentence. Second, it merges multiple paragraphs into one. Third, normalize the punctuations, remove redundant spaces and capitalizes the first character after each punctuation. Then lastly, normalize the telephone, price, area and zone to a common pattern. After the data is normalized, it will now be manually annotated using Callisto, an annotation software.

After it has been annotated, it is now ready to go to the information extraction system. It will go first through the tokenizer. The tokenizer will output two types of annotations, Word and Split. The Word annotations contains the part-of-speech, the word, checks if the first letter is capitalized, and other features (kind and nation). This will be used to create the Java Annotation Pattern Engine (JAPE) rules. The Split annotation contains the delimiter. After it goes through the tokenizer, it will now go through the Gazetteer. The gazetteers are dictionaries that are created during the system development. It contains dictionaries for potential named entities (person, location) or categories, phrases uses in contextual rules (name prefix or verbs that are likely to follow a person’s name), and potential ambiguous entities. The output of the gazetteer is a lookup annotation covering the specific semantics. After the gazetteer, it will now be passed to the JAPE transducer. The JAPE transducer is responsible for extracting the information. It uses JAPE rules to recognize the entities that will be needed to extract. The output is the annotated documents.

The system has been tested in a lenient and strict criterion. An entity that is recognized correctly when the type is correct but the span overlap in the annotated corpus is called the lenient criteria. On the other hand, an entity that is recognized correctly when the type and span are the same in the annotated corpus is called strict criteria. On the lenient criteria on test data, it measured 96% in F-measure. While on the strict criteria, it measured 91% in F-measure. The problem is on the data. The writing styles of the people are very diverse. The system has problem in recognizing some of the entities like the zone entity because some of the zone entity are very long and does not use capitalization.

##### Business Specific Online Information Extraction from German Websites (Lee & Geierhos, 2009)

The Business Specific Online Information Extraction System is a system that extracts information from the information pages of a German business website like its company profile, contact page, imprint and then identifies relevant business specific information. The system concentrates on the extraction of specific business information like company names, addresses, contact details, names of CEOs, etc. With regards to the way how the researchers pre-process their chosen input data, they interpret the HTML structure of documents and analyse some contextual facts to transform the unstructured web pages into structured forms. The approach applied by the researchers is quite robust in variability of the DOM (for the web pages), upgradeable and keeps data up-to-date. The evaluation metrics showed high efficiency of information access to the generated data. In their conclusion, they stated that the developed technique is also adaptive to non-German websites with slight language-specific modifications, and experimental results from real-life websites confirm the feasibility of their approach.

In their proposed system, the researchers had two main modules for processing and extracting information from the German Information Web Pages: one for establishing a relational database storing company information and the other is for providing a query module. Within these two modules are three sub process that are done to further process the input data: (A) Localization of the Information Pages on the Web; (B) Document Analysis and Information Extraction; lastly, (C) Query Processing. In sub process A (Localization of the Information Page), a web crawler is fed with the URL’s of the web pages that are stored in the specialized database and then it fetches them from the web. Afterwards, the proposed system will then retrieve the document by following the anchor tags that lead to the information pages. On the other hand, in sub process B (Document Analysis and Information Extraction), the fetched Information Pages are sent to an ‘info analyser’ module which examines the HTML content of the page and then extracts the needed information bits. Here, the system exploits the internal structure of the named entities and uses sublanguage-specific contexts or attribute classes to identify the attribute-value pairs. Lastly, in sub process C, the user of the system is given the right to query the database for the information bits that he/she needs and then add these bits to the index.

For the Information Page Analyser (info analyser) in sub process B, the input data has to further go through a number of processes to finally extract the information needed by the user. When given an Information Page, the analyser starts by pre-processing the frame structure and existing JavaScript of the page. And before creating the expressive DOM Tree, the HTML file of the page has to be validated and corrected, if needed, by using a special tool called ‘tidy’. After doing so, the system will now be able to locate the minimal data region (or the data region of the information bit searched for) surrounded by a number of HTML tags which contain the information record being searched for. By doing a depth- first traversal of the expressive DOM tree, the desired sub tree can be isolated based on the headings of the data record like the following: “Herausgeber” (publisher), “Betreiber” (operator), “Anbieter” (provider) and etc. The system was programmed to disregard domain name irrelevant information; thus, the analyser will work further on with a pruned DOM tree. After identifying the minimal data region, all information bits that are relevant to the domain name are extracted by using the Named-Entity Recognition technique and the attribute-value process (each attribute has a corresponding value that is indicated by the structure of the HTML file it is in) with respect to its external contexts and internal features. The system’s analyser module considers about 20 attribute classes and searches their corresponding values on the information page of business websites. The following are some of the attribute classes that are considered by the analyser: company name, address, phone and fax number, e-mail, CEO, management board, domain owner, contact person, register court, financial office, register number, value added tax number (VAT ID), and etc. After extracting the information bits needed from the pruned DOM trees, the information bits are then normalized to make sure that all information are consistent. The following are the classes that are affected by the normalization process: company names, legal form, register number, address (street, zip code, city), contact (phone and fax number, email), person name, and legal notification (tax number, VAT ID).

To conclude, the system performed surprisingly accurate with an average precision score of 99.1% and a recall score of 91.3% from a small test corpus that is composed of approximately 150 business web pages. The only encountered problem by the system is when value for certain attributes is erroneously represented like text in phone numbers and etc.

##### Ontology-Based Information Extraction (OBIE) System for French Newspaper Articles (Nebhi, 2012)

Since most of the information extraction systems are based on the English language, it poses a problem on other languages for there are not much tools available. In order to address this problem, the system maps the extracted entities to the ontology.

This system extracts person, location and organization on a French newspaper article. It collects data from LeMonde.fr. The system uses the GATE framework for annotation of entities in text and maps them to the ontology. It uses DBpedia databank that is based on Wikipedia projects. It contains 3,220,000 instances and is organized into a hierarchy of 320 classes and 1650 different properties. The system consists of 4 parts: pre-processing, gazetteer, rule-based semantic annotation, and the output. First, the system will pre-process the text. It will perform tokenizer, sentence splitter and POS tagger using the GATE application. After it is pre-processed, it will now go to the gazetteer. It will perform a lookup for the named entity recognition. After it passes through the gazetteer, grammar rules will be applied to create semantic annotation. The rules are written in JAPE which is part of the GATE framework. The system contains approximately 100 rules.

The system is evaluated using the Balance Distance Metrics (BDM) to take account the ontological similarity. It is also evaluated with the gold standards. They manually annotated the documents using concepts on DBpedia ontology, and then compared it with the gold standard. They only evaluated person, organization and location named entities. The system scored an average of 0.94 on the BDM and achieved a 91% F-Measure

## Other Information Extraction Systems

This part discusses information extraction systems that use other techniques.

##### SOMIDIA - Social Monitoring for Disaster Management (Cheng et al. 2011)

SOMIDIA is a crisis mapping system that focuses on plotting disaster on an interactive map in near real time. SOMIDIA collects data from different sources like news feeds, posts, SMS, blogs and microblogs. One of the main component of SOMIDIA is its information extraction module. It extracts from both Filipino and English texts.

For the information extraction module, first, documents go through a tokenizer. They used OpenNLP to tokenize the document, then it will go to a sentence splitter. The sentence splitter accepts the list of tokens and annotation list. It has a list of abbreviations so that the system can distinguish abbreviation periods from a period. The goal of the sentence splitter is to separate sentences by adding appropriate ending markers (period). The system used OpenNLP’s sentence splitter for its sentence detection. After the document has been split into sentences, it will go through a language guesser. They needed to differentiate English text from Filipino text because the language has different extraction techniques. They used frequency distribution of the words to detect the language. The output of the language guesser is the document with added metadata of the language. If the text is in English, the language guesser will passed the document to the POS tagger. Else, it would be passed to a Filipino NER.

For the English information extraction module, first it will go through the POS tagger. It uses the OpenNLP’s POS tagger function. The output is a list of token with its corresponding POS tags. After the POS tagger, it will go through a chunker. The chunker groups the tokens to their corresponding part-of-speech tag. This will be used to determine noun and verb phrases. It uses OpenNLP’s noun and verb chunker. After chunking, it will passed through the English NER. The NER only focuses on proper nouns. It uses LingPipe because of flexibility. LingPipe’s NER uses three types of approaches, dictionary-based, rule-based and statistic based approaches. After the NER, it will go through coreference resolution. The coreference resolution will find the noun counterpart of the pronouns. It uses Russian Mitkov algorithm for the resolution and WordNet for the lexicon. The normalization (standardizing data, collapsing of same sentences) will be done in this phase. The last step is now the information extraction phase. It uses JAPE rules to extract the information. The rules are paired with the two-tiered bootstrapping algorithm. The first tier bootstrapping algorithm starts with a small seed of words or rules. Then from the seed, it will try to learn the extraction pattern. The learned pattern will be used to generate new extraction pattern. The process will then repeat. The second-tiered bootstrap is responsible for keeping the most relevant extraction pattern.

For the Filipino extraction module, the document will go through the Filipino NER. They created their own NER because there are no existing Filipino NER tool. It uses a dictionary-based and rule-based approaches for their NER. After it has been tagged, it will now go through the Filipino extractor, the Filipino extractor has pre-defined rules (e.g. <event> sa <location>) that will extract the needed information.

The system is evaluated using precision, recall and F-measure. They evaluated it on Tweets and news feeds. For English tweets, it scored a 75.17% F-measure on extracting disaster and 62.83% on extracting location. For Filipino Tweets, it scored 82.13% F-measure on disaster and 56.32% on extracting location. For news feeds, it scored 45.40% F-measure on English news feeds, while 38.82% on Filipino news feeds. The tweets scored higher because it is much easier to extract patterns on shorter text. The needed information will most likely be located near the text. On longer texts, the information needed might be located far away.

Table 2‑1 shows a summary of all the reviewed information extraction system. The table indicates the system name, the language and type of data it can extract, the domain, NLP pre-processing techniques, information extraction techniques, and evaluation metrics used by the system.

Table 2‑1. Summary of Reviewed Information Extraction Systems

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **System** | **Language** | **Type of Data** | **Domain** | **Pre-processing Techniques** | **Information Extraction Techniques** | **Evaluation Metrics** |
| Machine Learning for Information Extraction in Informal Domains  (Freitag, 2000) | N/A | Documents (i.e. email) | Informal Domain | Not mentioned | Machine Learning-  Based | Precision, Recall |
| TOPO - Information Extraction System for Natural Disaster Reports From Spanish Newspaper Article (Téllez-Valero, 2005) | Spanish | Free-text | Natural Disasters | Text Classification, Document Feature Extraction | Machine Learning- Based | Precision,  Recall,  F-measure |
| VRE Information Extraction System (Pham & Pham, 2012) | Vietnamese | Free text | Real Estate Advertisement | Text Normalization | Rule-Based | Precision,  Recall,  F-measure |
| Business Specific Online Information Extraction from German Websites  (Lee & Geierhos, 2009) | German | Structured Text | Business Specific Information | Named Entity Recognition, Text Normalization, Attribute-Value Process | Rule-Based | Precision, Recall |
| Ontology-Based Information Extraction (OBIE) System  (Nebhi, 2012) | French | Free text | News article | Tokenization, POS Tagging, Sentence Splitter | Rule-Based, Ontology | Precision, Recall,  F-measure,  BDM |
| Social Monitoring for Disaster Management  (Cheng et al. 2011) | English, Filipino | Free text | News article, tweets | Tokenization, Sentence Splitter, Language Guesser | Machine-Learning Based | Precision, Recall, F-measure |

## Twitter and Disaster

This part discusses the systems that uses of Twitter in times of disaster as well as the information used.

##### Extracting Relevant Information Nuggets from Disaster-Related Messages in Social Media (Imran et al., 2013)

This paper focuses on the extraction of relevant information from disaster-related tweets. The data set the authors worked with are Twitter data during hurricane Joplin last May 22, 2011 with #joplin. Their approach includes text classification and information extraction.

First, the tweets were classified into what categories they belonged to. Table 2‑2 shows the categorization they used. After filtering the tweets, only those of the Informative category were used. The informative tweets were further categorized into what information type they contain. Their basis for the categories was from the ontology by (Vieweg et al., 2010).

Table 2‑2. Tweet Categories

|  |  |
| --- | --- |
| **Category** | **Description** |
| Personal Only | If a message is only of interest to its author and her immediate circle of family/friends and does not convey any useful information to other people who do not know the author. |
| Informative  (Direct) | If the message is of interest to other people beyond the author's immediate circle, and seems to be written by a person who is a direct eyewitness of what is taking place. |
| Informative  (Indirect) | If the message is of interest to other people beyond the author's immediate circle, and seems to be seen/heard by the person on the radio, TV, newspaper, or other source. The message must specify the source. |
| Informative  (Direct or Indirect) | If the message is of interest to other people beyond the author's immediate circle, but there is not enough information to tell if it is a direct report or a repetition of something from another source. |
| Other | If the message is not in English, or if it cannot be classified. |

Table 2‑3. Informative Tweet Categories

|  |  |
| --- | --- |
| **Category** | **Description** |
| Caution and advice | If a message conveys/reports information about some warning or a piece of advice about a possible hazard of an incident. |
| Casualties and damage | If a message reports the information about casualties or damage done by an incident. |
| Donations of money, goods or services | If a message speaks about money raised, donation offers, goods/services offered or asked by the victims of an incident |
| People missing, found, or seen | If a message reports about the missing or found person effected by an incident or seen a celebrity visit on ground zero |
| Information source | If a message conveys/contains some information sources like photo, footage, video, or mentions other sources like TV, radio related to an incident. |

To classify the tweets into the mentioned categories, Naïve Bayesian classifiers were trained and implemented using Weka. Their features include binary features (if the tweet contains the ‘@’ symbol, a hashtags, emoticons, links or URLs, and numbers), scalar features (the length of the tweet), and text features (unigrams, bigrams, POS tags, POS tag-bigrams, and VerbNet classes).

For each informative tweet category, they have extracted various types of information which they refer to as information nuggets. Table 2‑4 shows the extractable information nugget per informative tweet category as well as that category’s type subsets. The location references, time references, and number of casualties were extracted using the Stanford Named Entity Recognizer. All the Twitter Handlers (i.e. all words starting with the ‘@’ symbol and URLs) were extracted from the tweet for the sources. Caution/Advice and Damaged Object were extracted using the Stanford Part of Speech Tagger and WordNet. For the intention of the tweet, another classifier was trained to determine if the tweet is a donation effort or it requests for help. Lastly, the type information nugget pertains to the Type Subset column. For each informative tweet category, another classifier was trained to classify the category into its corresponding subset.

Table 2‑4. Extractable Information Nugget per Informative Tweet Category

|  |  |  |
| --- | --- | --- |
| **Informative**  **Tweet Category** | **Information Nugget** | **Type Subsets** |
| Caution and advice | * Location references * Time references * Caution/Advice * Source * Type | * Warning issued or lifted * Siren heard * Shelter open or available * Disaster sighting or touchdown |
| Casualties and damage | * Location references * Time references * Number of Casualties * Damaged Object * Source * Type | * Infrastructure * Death * Injury * Unspecified * No Damage * Both Infrastructure and People |
| Donations of money, goods or services | * + Location references   + Time references   + Intention of Tweet   + Source   + Type | * + Money   + Blood   + Voluntary Work   + Food   + Equipment   + Shelter   + Discounts   + Other |
| Information source | * + Location references   + Time references   + Source   + Type | * + Photo   + Video   + Website   + TV Channel   + Radio Station   + Unspecified |

##### Practical Extraction of Disaster-Relevant Information from Social Media (Imran et al., 2013)

Based on their previous paper Extracting Relevant Information Nuggets from Disaster-Related Messages in Social Media, after classifying the tweets into the informative tweet category, they extracted the information by employing a different approach. This time, they used two datasets: (1) tweets during hurricane Joplin last May 22, 2010 with #joplin and (2) tweets during hurricane Sandy last October 29, 2012 with #sandy #nyc.

To detect class-relevant information, they treated it as a sequence labeling task. For each token in the tweet, they labeled it as either part of the relevant information or not. The (+) label indicates that the token is part of the relevant information while the (-) label indicates that it is not. After labeling, they applied Conditional Random Fields (CRF) to extract the information. A tool they also used in this paper is ArkNLP, a Twitter-specific POS tagger.

# Theoretical Framework

This chapter presents a discussion on the different theoretical concepts associated to information extraction systems, and as well as common architectures, approaches, modules, and resources needed in developing such systems.

## Information Extraction

## Information Extraction Architectures

## Ontology

Ontologies are set of classes (concepts), attributes and relationships that are used to represent a domain knowledge. They are in a language (first-order logic) that can be abstracted from the data structures and implementations. Since ontologies are in the semantic level, they could be used to combine heterogenous database, thus making interoperability between systems possible (Gruber, 2009). Cimiano (2006) said as the number of applications using ontologies are growing, they must now clearly and formally defined an ontology.

Cimiano (2006) formally define ontology as:

Where,

## Tools

This section discusses the different NLP tools that could be used in implementing the information extraction system.

### Apache OpenNLP (OpenNLP, 2011)

Apache OpenNLP is a Java-based library used for commonly used modules in NLP. It has tokenization, sentence segmentation, part-of-speech (POS) tagging, named entity recognition (NER), chunking, parsing and coreference resolution. It also uses maximum entropy and perceptron based machine learning to train the model. OpenNLP is extendable so that the module could be customized to suit the requirements.

#### Sentence Detector

The OpenNLP Sentence Detector knows if a punctuation mark signifies the end of the sentence or not. They defined the sentence as the longest whitespace trimmed character sequence between two punctuation marks. Their module looks for the first non-whitespace character. This signifies the beginning of the sentence. The last non-whitespace character signifies the end of the sentence. It can also be trained be providing it a model. It also has an evaluation module in which it computes the precision, recall and f-measure. The limitation of the sentence detector is that it cannot identify the sentence boundary based on the content.

#### Tokenizer

The OpenNLP tokenizer split the character sequence into word, punctuation marks, and numbers. OpenNLP has three implementations of the tokenizer: white space tokenizer, simple tokenizer and learnable tokenizer. The whitespace tokenizer just uses the whitespace as the boundary to identify as a token. The simple tokenizer uses the sequence of character class to detect the token. The learnable tokenizer uses a machine learning technique, maximum entropy, to detect the boundary. It is based on probability models. The tokenizer can also generate the probability of each tokens. It can also be trained by providing it a model.

#### Named Entity Recognition (NER)

The OpenNLP NER is used to detect names in the sentence. It accepts a tokenized string as input. OpenNLP offers pre-trained models that are trained on freely available corpora. The limitation is that it can only detect the names based on the corpora it used to train the model. It is highly recommended that it uses a custom model. For the training data, it should contain at least 15,000 sentence in order for the model to be trained. It can also evaluate the model by measuring the precision, recall and f-measure.

#### Document Categorization

The OpenNLP document categorizer uses maximum entropy to classify the document to a pre-defined category. The module needs a model because the requirements varies for different users. So, there are no pre-defined models.

#### Part-Of-Speech (POS) Tagging

The OpenNLP POS Tagging marks a word based on the word type and the context. A word can have multiple tags. OpenNLP POS Tagger uses probability model to correctly tag the word. To limit the number of possible tags, they used a dictionary tags. This contains the list of tags that will be used by the POS tagger. The use of the dictionary has two positive effect. First, it limits the assignment of inappropriate tags to the token is now limited. Second, OpenNLP POS tagging uses beam search algorithm to search for the most appropriate tags. By using a dictionary tags, it now limits the possibilities. In result, it search faster. The POS tagger has a built in evaluator that measures the accuracy of the model.

#### Chunker

The OpenNLP chunker groups syntactically correlated words. The chunker needs a tokenized and tagged sentence in order to perform the chunker. The limitation is that it only tells what words are in the same group. It does not tell the internal structure and the context. The chunker needs to be trained as the performance of the chunker is not that good outside the predefined model (news). It has an evaluation module where it can be tested using test dataset or cross validation. It measures the performance of the model using precision, recall and f-measure.

#### Parser

The OpenNLP parser is responsible for turning the input text into a tree. OpenNLP has two implementation, chunking parser and tree insert parse. The tree insert is still on experimental stage. The parser expect a whitespace tokenized sentence as inputs.

#### Coreference Resolution

The OpenNLP Coreference resolution is not fully developed yet. It is still very much limited as its implementation is only for noun mentions.

### ANNIE (Cunningham et al., 2002)

ANNIE or A Nearly-New IE System is a system that contains different modules for NLP tasks. ANNIE is part of the GATE framework. ANNIE uses finite state transducers and JAPE rules to implement the modules. ANNIE has a tokenizer, gazetteer, sentence splitter, semantic tagger and name matcher.

#### Tokenizer

ANNIE tokenizer uses rules to split the sentence into words, numbers and punctuations. The tokenizer can also differentiate lower case and upper case words as well as special punctuations. It uses grammar rules to tokenize the strings. By using grammar rules, the tokenizer will be more flexible.

There are different types of tokens that the tokenizer can use for the rules. First is the word tokens. A word is defined as any set of contiguous upper or lower case (including and only hyphen). The word has an attribute called “orth”, which can have the following values: upperInitial (first letter is upper case, the rest are lower case), allCaps (all uppercase letters), lowercase (all lower case), mixedCaps (mixture of upper and lower case letters). The second type of tokens are numbers. This are combinations of consecutive digits. Third type is the symbol. The symbol has two categories, currency (“$”,”₱”, “₤”) and symbol (“&”, “^”). Fourth type is the punctuation. The punctuation has three categories, start punctuation (‘(‘, ‘[‘), end punctuation (‘)’, ‘]’) and other punctuations (‘:’). The tokenizer consider a punctuation as an individual token. The fifth type of token is the SpaceToken. This are the whitespaces. There are two types of whitespace, space and control. Any set of contiguous space and control characters is considered as a SpaceToken.

#### Gazetteer

The gazetteer just contains the list of names, organizations, cities, days of the weeks, and others in plain text. It uses an index files to access the lists which will be compiled in the finite state machines.

#### Sentence Splitter

The sentence splitter uses a finite state transducers to split the text into sentences. It uses gazetteer to check if punctuation is part of an abbreviations or signals the end of the sentence. The sentence are annotated with the type “Sentence”, the breaks with “Split”. The sentence splitter is domain and application independent.

#### Part-Of-Speech (POS) Tagger

ANNIE POS Tagger uses a modified version of Brill Tagger. It uses lexicons and ruleset that has been trained in the Wall Street Journal corpus. However, the lexicon and rulesets can be changed based on the requirements. There are two additional lexicons, the lexicon for all caps and the lexicon for lowercase.

#### Semantic Tagger

The semantic tagger uses JAPE rules to annotate the entities. The grammar could be designed in such a way that it would recognize the entities. The output of the semantic tagger is the annotated text, which will be needed by the orthographic coreference.

#### Orthographic Coreference (OrthoMatcher)

The orthographic coreference is responsible for finding relationship between named entities. It tries to classify those unclassified entities. To prevent an entity from being recategorized, the matching rules makes sure that the two entities have the same category. It uses a lookup table to categorized different strings but same entity (i.e. Coca-Cola – Coke, IBM – Blue). It also has table called table of spurious matches. Here, strings matches but represents different entities.

### Twitter NLP Tools (Ritter et al., 2011)

The performance of standard NLP tools is degraded when used with Twitter data. This is because the standard NLP tools are trained with a structured news corpora. Twitter corpus, on the other hand, is noisy and informal in nature. Classifying named entities in tweets is challenging because of two things. The first is that tweets may contain a lot of named entity types and the second is that due to Twitter’s 140 character limit, it is difficult to determine the context of the named entity. The tool addresses the issue by rebuilding the NLP pipeline from POS up to NER. Testing and results show that each module of the tool performs better than its standard NLP counterpart.

The system, called T-NER, is composed of five modules: POS Tagger (T-POS), Chunker (T-CHUNK), Capitalization (T-CAP), Name Entity Segmentation (T-SEG), and Named Entity Classification (T-CLASS). The first three modules were trained with a dataset of 800 randomly sampled tweets and a 4-fold cross validation. The latter two were trained with 2,400 randomly sampled tweets and a 4-fold cross validation also. T-NER doubles the F1 score of the Stanford NER.

#### Part-of-Speech Tagging (T-POS)

Standard POS taggers suffers because of the nature of the Twitter corpora’s style, vocabulary, and lexical variations. To overcome the problems pertaining to style and vocabulary, they manually annotated the 800 tweets with tags from the Penn TreeBank tag set. They also added new tags which are Twitter-related such as “retweet”. To overcome the lexical variations, they performed a hierarchal clustering using Jcluster on 52 million tweets and applied Brown clusters to capture the lexical variations. T-POS uses Conditional Random Fields (CRF). T-POS shows a 41% reduction in error against the Stanford POS tagger.

#### Shallow Parsing or Chunking (T-CHUNK)

To create the model for T-CHUNK, they annotated the 800 tweets with tags from CoNLL Shared Task and used the set of shallow parser features described by Sha and Pereira (2003). They also included the Brown clusters and the POS tag features predicted by T-POS. T-CHUNK also applies CRF and has a 22% reduction in error over the OpenNLP chunker.

#### Capitalization (T-CAP)

One of the key features in recognizing named entities is the capitalization but this is unreliable in tweets. To address this problem, they incorporated information based on the entire content of the tweet to determine if the capitalization is informative. To train the classifier, they manually tagged the 800 tweets if it has an “informative” or “uninformative” capitalization. The classifier is a Support Vector Machine with features such as the fraction of words in the tweet which are capitalized, the fraction which appear in a dictionary of frequently lowercase/capitalized words but are not lowercase/capitalized in the tweet, the number of times the word ‘I’ appears lowercase and whether or not the ﬁrst word in the tweet is capitalized. Results show that T-CAP’s precision, recall, and F1 score is 0.77, 0.98, and 0.86 versus the baseline’s 0.70, 1.00, and 0.82

#### Segmenting the Named Entities (T-SEG)

T-SEG utilizes the features of T-CAP and is trained by manually tagging the 2,400 tweets with named entities. @usernames were not considered as a named entity. To implement T-SEG, they used a sequence-labeling task using IOB encoding for representing segmentations, CRF, Freebase, and T-POS, T-CHUNK, T-CAP, Brown clusters for feature generation.

#### Classifying Named Entities (T-CLASS)

Freebase is used as a baseline source of distant supervision of entities. To model the unlabeled entities, they made use of a Distant Supervision with Topic Models that applies LabeledLDA. They annotated the 2,400 tweets with the ten types which are popular on twitter: PERSON, GEO-LOCATION, COMPANY, PRODUCT, FACILITY, TV-SHOW, MOVIE, SPORTSTEAM, BAND, and OTHER. Note that a tweet can be annotated with two or more types and that the annotation is used for evaluation purposes. For training, they ran T-SEG on 60 million tweets and kept the entities that appeared 100 or more times. For each entity, they collected the words occurring in a context window of three words. The results show that T-CLASS outperforms the baseline and achieves a 25% increase in F1 score over the co-training algorithm of Collins and Singer (1999).

### Weka (Weka 3, n.d.)

Waikato Environment Knowledge Analysis (Weka) is a Java-based open source collection of machine learning algorithms that are used in data mining tasks. It contains various tools for preprocessing, classification, regression, clustering and visualization. It provides a library that could be used. It is also flexible as users can extend the API to customize the machine learning algorithms (Weka 3, n.d.).

### TwitIE (Bontcheva et al., 2013)

Bontcheva and her team of researchers have proposed an information extraction system that is specifically targeted for extracting relevant information from texts coming from microblogs. Bontcheva et al made use of the GATE ANNIE architecture in developing the system and took advantage of some of its built-in tools to further streamline the process of information extraction. In their paper, they presented how they designed the architecture of TwitIE and how they used the existing tools from ANNIE.

ANNIE offers a wide array of information extraction tools like tokenizer, sentence splitter, POS tagger, gazetteer lists, finite state transducer (from GATE’s built-in regular expression over annotation language), orthomatcher and coreference resolver but in the case of TwitIE, Bontcheva et al only reused the sentence splitter and name gazetteer because the other components/tools have to be modified to fit the specifics of microblog texts like noisiness, brevity, idiosyncratic language and social context.

Overall, TwitIE has the following main components: Language Identifier, Tokenizer, Gazetteer, Sentence Splitter, Normalizer, POS Tagger, and the Named Entity Recognizer. For the Language Identifier, TwitIE made use of the TextCat language identification algorithm, which heavily relies on n-gram frequency models to identify languages. For the Tokenizer, TwitIE followed Ritter’s Tokenization Scheme to treat abbreviations and URL’s as one token and hashtags and mentions as two token. This scheme also features orthography and capitalization preservation. For the Gazetteer, TwitIE used the unmodified version from ANNIE, which compiles lists of entities into finite state machines that can match text tokens. For the Sentence Splitter, TwitIE still used the unmodified version from ANNIE, which is a cascade of finite-state transducers that segments text into sentences. For the Normalizer, TwitIE made use of a combination of a generic spelling-correction dictionary and a social media-specific dictionary. The list of variations is also dynamic by using heuristics to correct spellings. For the POS Tagger, TwitIE made use of the same technique used by a Stanford Tagger, which was trained on tweets that were tagged using the Penn TreeBank Tagset. The improved tagger also included tag labels to support retweets, mentions, URL’s, hashtags and user mentions. Lastly, for the Named Entity Recognizer, TwitIE made use of the existing ANNIE NER as its pattern and just added some features that would help it support recognition of entities in social media texts. The following are the main functionalities of TwitIE:

* Custom Language Identifier to support language identification for social media/Twitter data;
* Twitter Tokenizer to enable the system to properly handle smilies, usernames, hashtags, URL’s and etc.;
* Twitter POS Tagger (also available as a standalone tool) to enable the system to properly perform POS tagging in Twitter data; lastly,
* Text Normalization PR to enable the system to perform correction of slangs and misspellings in the tweet.

### OWL API (Horridge & Bechhoffer, 2011)

OWL API is a open-source Java based library to create, manipulate, and serialize ontologies. It contains different parsers: RDF/XML parser, OWL/XML parser, Turtle parser, KRSS parser, OBO Flat File format parser, and OWL Functional syntax writer and parser. It also interfaces with different reasoners: FacT, HermiT, Pellet, and Racer.

# The FILIET System

This chapter presents the proposed system. It is divided into six sections. The first section will discuss the system overview. The second section outlines the objectives the system must be able to achieve. The third section tackles the scope and limitations of the system based on the outlined objectives. The fourth section presents the architectural design. The fifth section discusses the front-end and back-end features. Lastly, the sixth section will present the resources that will be used in implementing the system.

# References

Gao, H., Barbier, G., & Goolsby, R. (2011). Harnessing the crowdsourcing power of social media for disaster relief. Intelligent Systems, IEEE, 26(3), 10-14. doi: 10.1109/MIS.2011.52

Gruber, T. (2009). Ontology. *Encyclopedia of database systems*, 1963-1965.

Horridge, M., & Bechhofer, S. (2011). The owl api: A java api for owl ontologies.*Semantic Web*, *2*(1), 11-21.

Howard, B. (2013, November 8). Scanning social media to improve typhoon haiyan relief efforts. *National Geographic Daily News*. Retrieved from http://news.nationalgeographic.com/news/2013/11/131108-typhoon-haiyan-philippines-crisis-mapping/

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, J. B., Ybañez, M., De Leon, M. M., Estuar, M., & Regina, E. (2013). Understanding the Behavior of Filipino Twitter Users during Disaster. GSTF Journal on Computing, 3(2).

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/>

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>

Vieweg, S., Hughes, A. L., Starbird, K., & Palen, L. (2010, April). Microblogging during two natural hazards events: what twitter may contribute to situational awareness. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 1079-1088). ACM.

1. MicroMappers digital disaster response system. http://micromappers.com/ [↑](#footnote-ref-1)
2. The American Red Cross, *Web Users Increasingly Rely on Social Media to Seek Help in a Disaster*, Press Release, Washington, DC, August 9, 2010. http://newsroom.redcross.org/2010/08/09/press-release-web-users-increasingly-rely-on-social-media-to-seek-help-in-a-disaster/ [↑](#footnote-ref-2)
3. 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/ [↑](#footnote-ref-3)