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

March 2, 2015

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, the information and data shared through Twitter have helped individuals, institutions, and organizations (government, public, and private) during emergency response, in making decisions, conducting relief efforts, and practically mobilizing people to humanitarian causes. However, extracting the most relevant information from Twitter is a challenge because natural languages do not have a particular structure immediately useful when programming. Another problem that information extraction faces is that some languages, like Filipino, is morphologically rich, making it even more difficult to extract information. Therefore, the goal of this research is to create a system or tool that extracts relevant information from Filipino disaster-related tweets.

Keywords: information extraction, disaster management, Twitter

Table of Contents

[1.0 Research Description 1-1](#_Toc396489504)

[1.1 Overview of the Current State of Technology 1-1](#_Toc396489505)

[1.2 Research Objectives 1-2](#_Toc396489506)

[1.2.1 General Objective 1-2](#_Toc396489507)

[1.2.2 Specific Objectives 1-2](#_Toc396489508)

[1.3 Scope and Limitations of the Research 1-2](#_Toc396489509)

[1.4 Significance of the Research 1-3](#_Toc396489510)

[1.5 Research Methodology 1-4](#_Toc396489511)

[1.5.1 Investigation and Research Analysis 1-5](#_Toc396489516)

[1.5.2 System Design 1-5](#_Toc396489517)

[1.5.3 Sprints 1-5](#_Toc396489518)

[1.5.4 Sprint Planning Meetings 1-5](#_Toc396489519)

[1.5.5 Scrum Meetings 1-5](#_Toc396489520)

[1.5.6 System Development 1-6](#_Toc396489521)

[1.5.7 System Integration and Testing 1-6](#_Toc396489525)

[1.5.8 System Evaluation 1-6](#_Toc396489526)

[1.5.9 Documentation 1-6](#_Toc396489527)

[1.5.10 Calendar of Activities 1-7](#_Toc396489528)

[2.0 Review of Related Works 2-1](#_Toc396489529)

[2.1 Machine Learning-Based Information Extraction Systems 2-1](#_Toc396489530)

[2.2 Rule-Based Information Extraction Systems 2-2](#_Toc396489531)

[2.3 Template-Based Architecture 2-4](#_Toc396489532)

[2.4 Ontology-Based Information Extraction Systems 2-5](#_Toc396489533)

[2.5 Other Information Extraction Systems 2-6](#_Toc396489534)

[2.6 Disaster Management – Relief Operations 2-9](#_Toc396489535)

[2.7 Twitter and Disaster 2-10](#_Toc396489536)

[3.0 Theoretical Framework 3-1](#_Toc396489575)

[3.1 Information Extraction 3-1](#_Toc396489576)

[3.1.1 Information Extraction Modules 3-2](#_Toc396489577)

[3.2 Information Classification 3-4](#_Toc396489578)

[3.2.1 Document Representation 3-4](#_Toc396489579)

[3.2.2 Dimensionality Reduction (Feature Selection) 3-5](#_Toc396489580)

[3.2.3 Classification 3-6](#_Toc396489581)

[3.3 Information Extraction Architecture 3-6](#_Toc396489582)

[3.3.1 Adaptive Architecture 3-6](#_Toc396489583)

[3.4 Ontology 3-12](#_Toc396489584)

[3.4.1 Ontology Design 3-13](#_Toc396489585)

[3.4.2 Ontology Population 3-14](#_Toc396489586)

[3.5 Twitter 3-15](#_Toc396489587)

[3.5.1 Use of Twitter 3-15](#_Toc396489588)

[3.5.2 Twitter and Disasters 3-15](#_Toc396489589)

[3.6 Evaluation Metrics 3-17](#_Toc396489590)

[3.6.1 F-measure 3-17](#_Toc396489591)

[3.6.2 Kappa Statistics 3-18](#_Toc396489592)

[3.7 Tools 3-18](#_Toc396489616)

[3.7.1 ANNIE (Cunningham et al., 2002) 3-18](#_Toc396489656)

[3.7.2 Weka (Weka 3, n.d.) 3-18](#_Toc396489657)

[3.7.3 JENA API (McBride, 2002) 3-19](#_Toc396489659)

[3.7.4 ArkNLP (Gimpel et al., 2011) 3-19](#_Toc396489706)

[3.7.5 NormAPI (Nocon et al., 2014) 3-20](#_Toc396489707)

[4.0 The FILIET System 4-1](#_Toc396489710)

[4.1 System Overview 4-1](#_Toc396489711)

[4.2 System Objectives 4-1](#_Toc396489712)

[4.2.1 General Objective 4-1](#_Toc396489713)

[4.2.2 Specific Objectives 4-1](#_Toc396489714)

[4.3 System Scope and Limitations 4-1](#_Toc396489715)

[4.4 Architectural Design 4-3](#_Toc396489716)

[4.4.1 Crawler Module 4-4](#_Toc396489717)

[4.4.2 Preprocessing Module 4-4](#_Toc396489724)

[4.4.3 Feature Extraction Module 4-6](#_Toc396489725)

[4.5 Category Classifier Module 4-7](#_Toc396489726)

[4.5.1 Rule Inductor 4-7](#_Toc396489757)

[4.5.2 Ontology Population Module 4-8](#_Toc396489760)

[4.5.3 Data Source 4-8](#_Toc396489762)

[4.6 System Functions 4-11](#_Toc396489763)

[4.6.1 Tweet retrieval 4-11](#_Toc396489764)

[4.6.2 Information extraction 4-12](#_Toc396489765)

[4.6.3 Ontology population 4-12](#_Toc396489766)

[4.6.4 Ontology access 4-13](#_Toc396489767)

[4.7 Physical Environment and Resources 4-13](#_Toc396489768)

[4.7.1 Minimum Software Requirements 4-13](#_Toc396489769)

[4.7.2 Minimum Hardware Requirements 4-13](#_Toc396489770)

[5.0 References 5-14](#_Toc396489854)

[6.0 Appendix 6-1](#_Toc396489855)

[6.1 Appendix A 6-1](#_Toc396489856)

[6.2 Appendix B 6-2](#_Toc396489857)

[6.3 Ontology 6-4](#_Toc396489858)

[6.4 Resource Person 6-9](#_Toc396489859)

[6.5 Personal Vitae 6-10](#_Toc396489860)

List of Tables

[Table 1‑1. Timetable of Activities (April 2014 - April 2015) 1-7](#_Toc396489861)

[Table 2‑1. Summary of Reviewed Information Extraction Systems 2-8](#_Toc396489862)

[Table 2‑2. Tweet Categories 2-11](#_Toc396489863)

[Table 2‑3. Informative Tweet Categories 2-12](#_Toc396489864)

[Table 2‑4. Extractable Information Nugget per Informative Tweet Category 2-12](#_Toc396489865)

[Table 3‑1. Examples of official government institution 3-16](#_Toc396489866)

[Table 3‑2. Examples of disaster-related tweets with extractable information 3-17](#_Toc396489867)

[Table 3‑3. Confusion Matrix (Davis and Goadrich, 2006) 3-17](#_Toc396489868)

[Table 4‑1. Sample Input/Output for Text Normalizer 4-4](#_Toc396489869)

[Table 4‑2. Sample Input/Output Tokenizer 4-5](#_Toc396489870)

[Table 4‑3. Sample Input/Output POS Tagger 4-6](#_Toc396489871)

[Table 4‑4. Sample Input/Output Gazetteer 4-6](#_Toc396489872)

[Table 4‑5. Sample Input/Output Category Classifier Module 4-7](#_Toc396489873)

[Table 4‑8. Sample Entries of Tweets in CSV File 4-8](#_Toc396489874)

[Table 4‑9. Sample Gazetteer for Storm Names (Philippines) 4-9](#_Toc396489875)

[Table 4‑10. Sample Extracted Rules 4-9](#_Toc396489876)

[Table 4‑11. Excerpts of the list of seed words 4-10](#_Toc396489877)

[Table 4‑12. Excerpts of the POS Dictionary 4-10](#_Toc396489878)

[Table 6‑1. Results of the study conducted by University McCann 6-1](#_Toc396489879)

[Table 6‑2. Example of Filipino Morphemes 6-2](#_Toc396489880)

List of Figures

[Figure 1‑1. Research Methodology Phases 1-4](#_Toc396489881)

[Figure 2‑1. Poibeau's General Architecture 2-5](#_Toc396489882)

[Figure 3‑1. Structure of an Information Extraction System 3-2](#_Toc396489883)

[Figure 3‑2. StaLe Lemmatization Process 3-4](#_Toc396489884)

[Figure 3‑3. Architecture of LearningPinocchio 3-7](#_Toc396489885)

[Figure 3‑4. Rule Induction Step 3-8](#_Toc396489886)

[Figure 3‑5. Algorithm for Choosing the Best Rules 3-9](#_Toc396489887)

[Figure 3‑6. Information Extraction Process of LearningPinocchio 3-9](#_Toc396489888)

[Figure 3‑7. Figure 3 7. Architecture of IE2 Adaptive Information Extraction System 3-11](#_Toc396489889)

[Figure 3‑8. SOMIDIA's Architecture 3-12](#_Toc396489890)

[Figure 3‑9. Process of Semi-Automatic Ontology Population 3-14](#_Toc396489891)

[Figure 4‑1. Architectural Design 4-3](#_Toc396489892)

[Figure 4‑2. FILIET Ontology 4-11](#_Toc396489893)

[Figure 4‑3. Tweet Retrieval Screenshot 4-12](#_Toc396489894)

[Figure 4‑4. Information Extraction Screenshot 4-12](#_Toc396489895)

[Figure 4‑5. Ontology Population Screenshot 4-13](#_Toc396489896)

[Figure 4‑6. Ontology Access Screenshot 4-13](#_Toc396489897)

**List of Code Listing**

[Code Listing 3‑1. Example code to create an ontology 3-19](#_Toc396489898)

[Code Listing 3‑2. Example code to create a class 3-19](#_Toc396489899)

[Code Listing 3‑3. Example code to create object properties 3-19](#_Toc396489900)

[Code Listing 3‑4. Example codes to create an instance 3-19](#_Toc396489901)

[Code Listing 3‑5. Example code for tokenizing text. 3-20](#_Toc396489902)

[Code Listing 6‑1. Representation of Ontology in OWL Format 6-8](#_Toc396489903)

# Research Description

This chapter introduces the research 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 talks about the motivations and the problem that needs to be addressed. The second section discusses the objectives of the research. The third section details the scope and limitations of the study. Lastly, the fourth section tackles the significance of the research and its benefits to Philippine society.

## Overview of the Current State of Technology Relative to Disaster Management

According to a report of the United Nations International Strategy for Disaster Reduction (UNISDR) Scientific and Technical Advisory Group, disasters have destroyed lives, properties, and livelihood across the world. Just between 2000 and 2012, about 2 million people have died during disasters and an estimated US$ 1.7 trillion in damages have been recorded. 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 and platforms that aim to facilitate interaction, collaboration, and 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 received 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 play 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. Just 48 hours later, the Red Cross has raised US$8 million. Social media have also enabled the generation of community crisis maps and interagency maps. They are maps that work as intermediaries 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-2), that quickly sorts through online data, from tweets to uploaded photos, and then displays 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). To illustrate further how social media are significantly regarded, in a study commissioned by the American Red Cross[[2]](#footnote-3), 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 to share information regarding the disaster including updates on response efforts. As part of the Philippine disaster management for natural calamities, the government has released an official newsletter detailing the official social media accounts and hashtags[[3]](#footnote-4). Filipino Twitter users tend to post tweets about requests for help and prayer. Other tweets pertain to traffic and weather updates, related observations, and class suspensions. While some users prefer to post messages in English, a large number of users also communicate with their native language when tweeting during disasters (Lee et al., 2013).

Knowing that various emergency response organizations aim to, as much as possible, attend to all requests for help, it would be very important and beneficial to have a system that is capable of extracting relevant disaster relief operation information from the contents that are posted by Filipino netizens in Twitter. Furthermore, it would be very helpful to have an information extraction system that is able to mine relevant information from the language that is dominant in the disaster-stricken areas, which, in the case of the Philippines, is the Filipino language and, at the same time, support the way how content is posted in Twitter like having certain formats (having #tags), writing style (TXTSPK and code-switched styles), etc. In general, having this system can open up opportunities for improving how disaster relief operations are planned and conducted in the Philippines, and eventually, can help save lives.

## 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 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 analyze different approaches used in implementing an information extraction system;
5. To evaluate existing tools and resources that could be incorporated in the information extraction components of the system;
6. To determine the metrics for assessing the performance or effectiveness of the information extraction system.

## Scope and Limitations of the Research

The research is about the design of an information extraction system for the Filipino language. Review of various information extraction systems to know the different approaches to implementation was covered. Different existing domain-independent, domain-dependent information extraction systems were also analyzed to understand their components, architectures, and implementation. Additionally, this study examined information extraction for MRL to grasp the techniques used to extract from MRL given that the Filipino language is considered an MRL.

For the system to extract relevant information, the research determined which information details are deemed relevant in times of disaster, especially in relief operations. The research also identified the different types of disaster-related Tweets to support the task of discerning relevant information from the given tweets. Other researches on the use of Twitter in disaster management were also evaluated to aid in the formulation of ontologies of the information extraction system developed in this study.

In terms of system performance, the research looked into different natural language processing techniques used for data preprocessing before feeding them into the information extraction system. Examples of the NLP techniques are text classification and text normalization. Text classification is the process of automatically assigning a text or document into a predefined category based on its 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.

Different information extraction techniques were also examined. Some of these 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). Co-reference analysis is the resolving of references for the pronouns (Grishman, 1997).

Existing NLP tools for building information extraction systems were also reviewed. Examples of these 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).

Lastly, metrics were determined to measure system performance, metrics were determined.

## 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. 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 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 regard to effectively utilizing such information from the web and use them for disaster management purposes.

From a 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 built for the two languages, 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 of the languages 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.

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

Scrum-based methodology, an iterative software development life cycle, was applied in the course of this research.



11

[]

### Investigation and Research Analysis

This phase involves the study and understanding of the fundamental knowledge of the concepts, algorithms, techniques, and tools that can be used to implement the system as well as identifying the modules and requirements of the system to be developed. The key activity involved in this phase is literature reviews of related works. From the review process, the techniques, tools, and metrics were then compared to verify which can be adapted to the system.

### System Design

With the information takeaways from the earlier phase, system design commences. In this phase, appropriate architectures, algorithms, information extraction techniques, and other necessary tools are identified so that they can be effectively utilized in system creation. In addition, in this phase, necessary modules for the system are also identified based on the different processes and features that will be built into the system. The designs of the User Interfaces and the basic architecture for the databases are also covered in this stage. Finally, this phase also addressed data source identification for use and processing by the system. Once the data sources have been identified, data collection immediately commences.

### Sprints

There is a two-week timeframe for each sprint 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 to conducting 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 are discussed here. Included in these meetings is the assignment and division of the tasks among the members of the team. The evaluation of tasks from the previous sprint is also 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 are conducted daily. The purpose is to update each member about what has or has not been accomplished yet in the assigned tasks. This ensures that there is daily progress and if there are issues that hinder members from accomplishing their assigned tasks so that they can be assisted.

### System Development

From the design phase, system development follows. Data collection will also be done in this phase. Each team member is assigned to modules. The development of the system follows 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 to assess the progress of the thesis and to plan the succeeding tasks.

### System Integration and Testing

All the modules that have been developed are integrated into one system. This phase is also about 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 integration, the system is then subjected to another round of tests to check again for any faulty integration and bugs.

### System Evaluation

This phase is system performance evaluation following the metrics selected and reviewed. The following metrics have been identified so far: Precision, Recall, and F-measure results of the information extracted by the system. A number of tests of information extracted manually and those from the training set are undertaken to compare and validate results. The metrics can also be modified as needed depending on additional tests and findings in the future.

### Documentation

Documentation of activities, methodologies, and of the system developed is important for monitoring and modification or improvement purposes. It will also be used for further reference, in case there is a need to validate or cross-reference any future work.

### Calendar of Activities

Table 1‑1shows 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 applications that are related or similar to this research.

## 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 variation of the slot-filling problem, specifically how to find the best unbroken fragment of text to fill a given slot in the answer template. A definite template 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, learn the problem and focus on fields that are 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 the ML of a feature set is needed to help adapt to domains containing novel structures because 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 conducted experiments to gauge the performance of the four components.

In summary, the researchers found 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, training effective extractors with very simple document representations is feasible.

##### 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 casualties, missing or wounded), (4) information related to infrastructure (number of affected hectares, economic losses. It is able to extract information on natural disasters 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 stage is the document feature extraction. To limit the dimension, they used information gain technique. After conversion to a Boolean vector, classification follows. They used Support Vector Machine (SVM), Naïve Bayes (NB), C4.5, and k-Nearest Neighbors (kNN). After classification, text that might contain relevant information is selected. This stage is the candidate text selection. This process uses grammar to select the text and a dictionary of names and number to treat grammar exceptions. Then the output becomes candidates of relevant information. The system will then select which of the information will be used. This system uses the same algorithms in the text classification stage. They used different classifiers for different outputs.

This architecture boasts of 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 needs to change the training corpus.

In this work, 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 obtained an F-measure from 72% to 88% on the 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 the name and quantities was SVM, while it was kNN for dates. The overall system obtained an average of 72% on the F-Measure.

##### EVIUS (Turmo & Rodriguez, 2000)

EVIUS is a multi-concept learning system for free text that follows a multi-strategy constructive learning (MCL) approach. The system also supports insufficient amounts of training corpora. M-TURBIO is the multilingual IE system where EVIUS is its component. The system’s input is both a partially parsed semantically tagged training corpus and a description of the desired target structure. The system’s approach to learn is by using MCL with constructive learning, closed-loop learning, and deductive restructuring (Ko, 1998). EVIUS decides which concepts to learn and updates the IE rule sets continuously. The system uses FOIL (First-Order Induction Learning) (Quinlan, 1990) to create an initial rule set from a set of positive and negative examples. Positive examples can be selected using a friendly environment either as text and ontology relations. Negative examples are automatically selected. If any uncovered positive examples remain after using FOIL, this is because there are insufficient examples. The system tries to develop recall by growing the positive examples with artificial examples (pseudo-examples). Combining the uncovered example vector and a randomly selected covered vector makes a pseudo-example. This is done as follows: For each dimension, one of both possible values is randomly selected as the value for the pseudo-example. The new set of positive examples is now executed again using FOIL, the resulting set will be combined with the first rule set.

## 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 specific criteria before it is fed into the system. First, input must be news articles related to real estate advertisements. Second, only one advertisement is allowed from each input data file. Lastly, it must be stripped off of all its HTML tags. After the data have met all the criteria, it will now go to data normalization. This process helps reduce ambiguity and assists in the annotation process. The necessary punctuation at the end of each sentence is also added. Second, it merges multiple paragraphs into one. Third, punctuations are normalized, redundant spaces are removed, and the first character after each punctuation is capitalized. Lastly, the telephone, price, area, and zone datails are normalized to a common pattern. Upon completion, the data will now be manually annotated using Callisto, an annotation software.

After annotation, data are 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 annotation contains the part-of-speech, the word; it also checks if the first letter is capitalized, and has other features (kind and nation). This will be used to create the Java Annotation Pattern Engine (JAPE) rules. The Split annotation contains the delimiter. The next process is through the Gazetteer. Gazetteers are dictionaries that are created during system development and they include potential named entities (person, location) or categories, phrases used 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 this process, the text or data will now be passed on to the JAPE transducer. The JAPE transducer is responsible for extracting the information. It uses JAPE rules to recognize the entities that will need to be extracted. The annotated documents are the output.

The system has been tested using a lenient criterion and a strict criterion. An entity that is recognized correctly when the type is correct but the span overlaps in the annotated corpus is called the lenient criterion. 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 criterion. On the lenient criterion on test data, it registered 96% on the F-measure. While on the strict criterion, it registered 91% on the F-measure. The problem is on the data. The writing styles of the people are very diverse. The system has a problem in recognizing some of the entities like the zone entity because some of the zone entities are very long and do not observe 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, and 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 regard to how the researchers pre-process their chosen input data, they interpret the HTML structure of documents and analyze some contextual facts to transform the unstructured web pages into structured forms. The approach applied by the researchers is quite robust in the variability of the DOM (for the web pages); it is also upgradeable and keeps data up-to-date. The evaluation metrics showed high efficiency of information access to the generated data. In 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 subprocesses 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 subprocess A (Localization of the Information Page), a web crawler is fed with the URLs 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 subprocess B (Document Analysis and Information Extraction), the fetched Information Pages are sent to an ‘info analyzer’ 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 subprocess C, the user of the system is given the right to query the database for 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 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. 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. By doing a depth- first traversal of the expressive DOM tree, the desired subtree can be isolated based on the headings of the data record like the following: “Herausgeber” (publisher), “Betreiber” (operator), “Anbieter” (provider), etc. The system was programmed to disregard domain-name irrelevant information; thus, the analyser will work further 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 was composed of approximately 150 business web pages. The only encountered problem by the system was when value for certain attributes were erroneously represented like text in phone numbers, among others.

## Template-Based Architecture

A template-based information extraction system uses templates to extract information. A template-based information extraction will only be able to extract information that is deemed important by the user. Its performance is based on how the user created the templates (Corney et al., 2008).

##### An Open Architecture for Multi-Domain Information Extraction (Poibeau, 2001)

Thierry Poibeau has provided a general architecture for developing information extraction systems regardless of its domain (Poibeau, 2001). In his paper, he proposed an information extraction architecture that takes advantage of the capabilities of machine learning to help researchers define new templates (this is where the extracted information is being filled in) with respect to the IE system’s domain.

Poibeau’s architecture is divided into 5 main modules: (1) the module for extracting information from the structure of the text; (2) the module for named entity recognition which is responsible for recognizing places/dates/etc.; (3) the module for the semantic filters; (4) the module for the extraction of specific domain-dependent information; and lastly, (5) the module for filling in a result template.

In module 1, a number of information are extracted from the structure of the input text. It is in this module where information that is embedded in the structure of the text is extracted such as those that are written in HTML or XML formats. On the other hand, in module 2, relevant information are extracted/recognized through linguistic analysis. This module is responsible for recognizing the different named entities present in the input text like names, places, and dates. Poibeau made use of the finite-state tool *Intex* to develop this module. Furthermore, in module 3, text categorization is performed on the set of so-called “semantic signatures” that were produced from a semantic analysis of the input text. Poibeau made use of the French system Intuition™ to develop this module. In addition, in module 4, specific information like the specific relationships between named entities are extracted by applying a grammar of transducers or extraction patterns on the input text. Lastly, in module 5, all the information extracted from the input text are linked together to fill in a specific result template(s) that present(s) a summarized view of the extracted information. Figure 2‑1 illustrates the general architecture proposed by Poibeau.

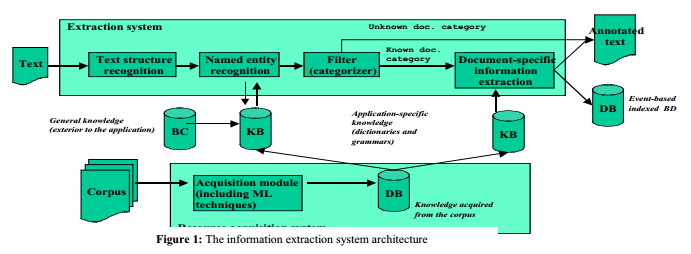


Figure 2‑1. Poibeau's General Architecture

## Ontology-Based Information Extraction Systems

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

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

As most information extraction systems are based on the English language, it poses a problem for other languages in terms of limited tools available. To address this problem, the system maps the extracted entities to the ontology.

This system extracts names of persons, locations, and organizations from French newspaper articles. 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 consider ontological similarity. 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., 2013)

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 components 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 use OpenNLP to tokenize the document, then it will go to a sentence splitter. The sentence splitter accepts a list of tokens and an 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 uses 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 pass the document to the POS tagger. Otherwise, it would be passed on 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 tokens with its corresponding POS tags. After the POS tagger, it will go through a ‘chunker’. The chunker groups the tokens into 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 pass through the English NER. The NER only focuses on proper nouns. It uses LingPipe because of its flexibility. LingPipe’s NER uses three types of approaches, dictionary-based, rule-based, and statistic-based approaches. After the NER, it will go through co-reference resolution. The co-reference resolution will find the noun counterpart of the pronouns. It uses the 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 the information extraction phase. It uses JAPE rules to extract the information, and 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 a new extraction pattern. The process will then be repeated. 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 is no existing Filipino NER tool. It uses dictionary-based and rule-based approaches for their NER. After tagging, 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.

Table2‑1. Summary of Reviewed Information Extraction Systems

shows a summary of all the reviewed information extraction systems. The table lists 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 |

## 

## Disaster Management – Relief Operations

This part discusses more about Relief Operations and the different information that are essential to this aspect of Disaster Management.

##### Humanitarian Knowledge Management (King, 2005)

This paper discusses the complexities and numerous challenges that many humanitarian organizations face whenever complex international humanitarian emergencies occur and how certain critical information in relation to disaster management activities, such as humanitarian assistance or relief operations can be utilized to help facilitate needed actions. King mentioned that the problem lies on the management of the data needed about these emergencies. In his paper, King stated that data management includes identifying, presenting, and disseminating critical information about the situation although such critical information, in itself, present a serious problem that could greatly affect data management. The problem lies in how this critical information is gathered: what information should be gathered and where should these be taken from? Upon efficiently identifying this in the early stages of these kinds of activities, as King mentioned, humanitarian organizations can more effectively make contingency plans and respond to natural disasters and complex emergencies and at the same time, potentially save a significant number of lives.

In the paper, a specific section was made to discuss what information are essential and crucial to different humanitarian organizations whenever they would conduct relief operations as a response to international complex emergencies like natural disasters and etc. According to King, humanitarian organizations like NGOs, UN agencies, local and national government, etc. need two specific types of information: (1) background and (2) situational information. Furthermore, information that is not within these types is more pertinent, relevant and critical to various specific personnel that are also within the said organizations. To support this claim, King gave an example through a scenario. He mentioned, “*policy makers want “big picture snapshot” analysis in order to understand the issues, to make decisions on providing assistance, and to be alerted to problems and obstacles*…*field personnel and project and desk officers in aid organizations, on the other hand, need more detailed operational and programmatic information in order to plan and implement humanitarian assistance and reconstruction programs*” (King, 2005).

With all of these, King listed down four main categories for the different vital information that is needed by organizations whenever they would conduct relief operations. The four categories are as follows: (1) **Situational awareness**–*information about the latest situation on the ground and information about the conditions, needs, and locations of affected populations*; (2) **Operational/Programmatic**–*information necessary to plan and implement humanitarian assistance programs*; (3) **Background** –*information about the unique history, geography, population, political and economic structure, infrastructure and culture of the country to be able to compare the emergency situation and conditions to previous normal conditions*; and lastly (4) **Analysis** – *humanitarian information needs to be interpreted in context and related to other thematic information. Analysis can include evaluations of issues and responses, projections about the future, and recommendations for policies and actions* (King, 2005).

To be able to streamline the process of determining the information that can fall within each of the categories, King has given some ‘guide questions’ for each of the categories (King, 2005).

* **Situational Awareness**
* *What is the latest/current humanitarian situation in the country?*
* *What are the most recent severity indicators? (Death tolls, mortality rates, malnutrition rates, economic impact, infrastructure damage, etc.)*
* *Who are the affected populations (refugees, IDPs, children and other vulnerable groups, resident populations, etc.); how many are there, and where are they located?*
* *What are the conditions and humanitarian needs of the affected populations?*
* *What is the assessment of damage to infrastructure? (Transport, buildings, housing, communications, etc.)*
* *What is the latest/current security situation in the affected areas of the country?*
* **Operational/Programmatic**
* *Where are and what are the conditions of the logistical access routes for delivering humanitarian assistance?*
* *Who’s doing what where? What humanitarian organizations are working in the country, what are their programs, what are their capacities, and where are they working?*
* *How is the host country/government responding and can it provide more?*
* *What are the programmatic/financial needs of the humanitarian organizations?*
* *What and how much are being provided to the humanitarian response organizations and who are the donors?*
* **Background**
* *What are the country’s population (national, province/state, city/town) and composition (ethnicity, religion, age cohorts, urban/rural, political, etc.)?*
* *What is the geography of the country?*
* *What are the country’s past disasters and natural hazards?*
* *What are the most recent annual baseline health indicators for the population? (crude mortality rate, infant/child mortality rates, HIV adult prevalence, malnutrition, etc.)*
* *What are the annual economic indicators? (GDP, GNP, agricultural/food production, staple food prices, etc.)*
* **Analysis**
* *What are the causes and contributing factors of the emergency?*
* *What are the constraints to providing humanitarian assistance? (Insecurity, inaccessibility, government, interference, etc.)*
* *How effective are humanitarian assistance programs and responses?*
* *What are the future impacts of the emergency?*
* *What are the options and recommendations for action?*

## Twitter and Disaster

This part discusses the uses of Twitter in times of disaster, the information that are useful during disasters, the information that can be extracted from disaster-related tweets and lastly, systems that make use of Twitter for disaster management procedures.

##### 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 on May 22, 2011 with #joplin. Their approach includes text classification and information extraction.

First, the tweets were classified into their respective categories (See Table 2‑2. Tweet Categories). After filtering the tweets, only those of the Informative category were used. The informative tweets were further categorized into the information types they contained. The basis for the categories was from the ontology by Vieweg et al. (2010).

|  |  |
| --- | --- |
| **Category** | **Description** |
| Personal Only | If a message is only of interest to its author and his/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 have been 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. |
| Others | If the message is not in English, or if it cannot be classified. |

Table 2‑2. Tweet Categories

categories

,red, were extracteda

|  |  |
| --- | --- |
| **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.  Example: “Alerto sa Mayon Volcano, itinaas ng Phivolcs sa level 2” |
| Casualties and damage | If a message reports the information about casualties or damage done by an incident.  Example: “Bush fires destroy 50 hectares in Baler, Aurora – NDRRMC http://t.co/Oc70OMeung49” |
| 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  Example: “Repacking of Mineral waters! (@ Dano Residenza) http://t.co/iHUn4XA7jb” |
| People missing, found, or seen | If a message reports about the missing or found person affected by an incident or a celebrity seen visiting ground zero  Example: “@philredcross missing joahnna nicole juliana ortiz sn isidro sulat eastern samar maytigbao church evacuation http://t.co/PGLnSEOtmY” |
| 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.  Example: “VIDEO: Alert level 2, itinaas sa Mayon Volcano http://t.co/g6U5AziDFt” |

Table 2‑3. Informative Tweet Categories

|  |  |  |
| --- | --- | --- |
| **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 |

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

##### 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 on May 22, 2010 with #joplin and (2) tweets during hurricane Sandy on 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.

##### Safety Information Mining - What can NLP do in a disaster (Neubig et al., 2011)

In the article presented by Neubig and his team of researchers, they described the efforts of researchers in the field of Natural Language Processing in creating an information extraction system that aided in the relief operations during the 2011 East Japan Earthquake. The system that was described was primarily built to ease the mining of information regarding the safety of those affected by the earthquake from one of the most prevalent information source during that time, that is, Twitter. The system included subsystems that work for the following NLP and IE techniques like word segmentation, named entity recognition, and tweet classification.

The development cycle of the IE system has two phases: (1) resource-building phase and the (2) actual IE system development phase. To begin the development of the information extraction system, the researchers first started out by making the prerequisite resources for the system (or the resource-building phase). The researchers first focused on developing the different Language Resources and Tweet Corpus of the system. These language resources included dictionaries (used to improve the performance of the different text analyzers and classifiers in the system) and a labeled corpus of tweets (this contains safety information about the disaster and was used for the extraction from unlabeled tweets).

For the creation of the dictionaries, the researchers made use of the “Balanced Corpus of Contemporary Written Japanese” and the “UniDic dictionary” for general domain languages while the “Mozc Japanese Input Method Dictionary” and other publicly available resources like the last names specific to northeast Japan and the database of postal code were used for the domain-specific language. An additional list containing station names and locations, landmarks, etc. were made to aid in the extraction process.

For the creation of the Tweet corpus, the researchers collected tweets that contain the word ‘earthquake’, and those that contains the following hashtags: #anpi (safety information), #hinan (evacuation), #j\_j\_helpme (help request) and #save\_<location>. To complete the corpus, the researchers tried to recognize the topic of the tweet (tweet classification) and the people mentioned in the tweet (named-entity recognition). To do so, the researchers defined nine classifications for the labels/topic of the tweets and are (1) I - Himself/Herself is alive; (2) L - Alive; (3) P - Passed away; (4) M - Missing; (5) H - Help request; (6) S - Information request; (7) O - Not safety information; (8) R - External link; and lastly, (9) U - Unknown.

After developing the prerequisite resources, the researchers proceeded with the actual development of the information extraction system. According to Neubig et al., the first step in IE for the Japanese language is Morphological Analysis. The MA is responsible for the tokenization and POS tagging of the tweets and for this, they made use of an open-source tool called KyTea. To accommodate the proper named-entity recognition in the Japanese language, the researchers trained the POS tagging model and replaced all proper nouns with subcategory tags (e.g. “first name”, “last name”, “place name”, and etc.) together with the introduction of a Conversational & News Text Corpus (containing a large list of Japanese first and last names). However, even though the POS tagging has been polished, the NER still failed to detect named entities that are grouped (NER still works on a word-by-word basis) that’s why the researchers made a simple rule-based system to accommodate the grouping of the Japanese named entities.

With all these, the researchers finally combined the two developed systems (the language resources and the MA system) to make the final information extraction system. The combination of the language resources with the MA system tends to increase the performance (accuracy) of the developed information extraction system by being able to accommodate the variations in the different styles in the different datasets that were used in this research.

# 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

There is already huge amount information freely available in the internet. The problem is that people could not process these information easily because of the huge volume. It becomes more difficult as the information are written in natural language, which can be ambiguous. However, using an information extraction system, it can now automatically collect information from different sources like news, papers, and journals. Information extraction is the identification of the class of events or relationship and the extraction of relevant arguments of the event or relationship inside a natural language. It involves the creation of a structured representation of the facts that will be extracted. An information extraction system can only extract those facts that are represented (Grisham, 1997).

An information extraction system is divided into two parts, local text analysis and discourse analysis. Local text analysis is responsible for extracting the information from a text document. It consists of lexical analysis, name recognition, partial syntactic analysis, and scenario pattern analysis. Lexical analysis is responsible for splitting up the text into tokens. After splitting the text, it looks up a dictionary to fill out the part of speech and features of each token. After lexical analysis, it goes through name recognition. Name recognition is responsible for identifying proper nouns, aliases, and other special forms (dates and currency). It uses regular expressions that are stated in the POS, syntactic features, and orthogonal features to identify names. It also uses a dictionary that contains the list of proper nouns such as company names to easier identification. After going through name recognition, it passes through a partial syntactic analysis to identify some of the syntax of the text. It is responsible for identifying some of the like noun groups and verb groups. However, some systems do not implement a syntactic analysis. After syntactic analysis, it goes through scenario pattern matching. Scenario pattern matching is the extraction of related events or relationship relevant to the scenario. The outputs of the scenario pattern matching are two clauses. The first clause is a reference to an event structure while the second clause is a reference to a created entity (Grisham, 1997).

After going through the phases of local text analysis, it can now pass through the discourse analysis. Discourse analysis is the combination of all the information extracted during the local text analysis, and the formatting of the information. Under the discourse analysis are co-reference analysis and inference. Co-reference analysis attempts to resolve anaphoric references (pronouns and definite noun phrases). To determine which entity is referenced, the most recent previous mention of the entity is the anaphoric reference. After the co-reference analysis, it will undergo inference and event merging. Inference is responsible for making implicit information explicit. It uses system production rules to implement the inference module. After the inference, it can now be place in the data representation. **Error! Reference source not found.** shows the general flow of an information extraction system (Grisham, 1997).



Figure 3‑1. Structure of an Information Extraction System

### Information Extraction Modules

This section explains the different modules that are commonly used in information extraction systems.

#### Tokenizer

Tokenizer is the module that segments a given text into tokens for further use in the natural language process. Tokens are usually the elements between spaces in the given input string. This module of natural language processing encounters several difficulties that need to be addressed such as tokenizing, email addresses, and uniform resource locators (URLs). Tokenizers today can identify that "15MB" is interpreted as "15 megabytes" if even there is no space between '15' and 'MB', and words with punctuation marks are also read correctly if tokenized. However, these tokenizers face two major problems, first is that the tokenizer performs its task independent of any knowledge, contained in the system. Another problem is that tokenizers are hard coded in the system. Thus, systems using these tokenizers end up tokenizing the input text without even caring whether the output of the tokenization made sense.

The researchers invented a tokenizer that validates the proposed output of the tokenization in a linguistic knowledge component, and this proposal validation repeats until there is no more possible segmentation or the text is validated. Lastly, the invented tokenizer also includes a language-specific data that contain a precedence hierarchy for punctuation (Bradlee et. al., 2001).

#### Sentence Splitter

The sentence splitter is a cascade of finite-state transducers that segments the text into sentences, and this module is used for the POS tagger (Cunningham et al., 2002). This module uses the set of regular expression-based rules that define sentence breaks like using periods, exclamation marks, and question marks (Zeng et al., 2006).

#### Normalizer

The presence of text speaks, slangs, and lingos is very high in SMS, social networks, and microblog sites. This presence makes it difficult for information extraction. In Aw and colleague’s work (2006), they viewed text normalization as a specialized machine translation problem, called SMS Normalization. They see that text speaks, slangs, and lingos are just a variant of the English language. However, applying general machine translation will not work against SMS Machine Translation. General machine translation is based on non-standard words that have been well studied. However, with SMS, most of the lingos, for example “b4” (before) and “bf” (boyfriend) are not formally defined by linguistics yet. These words can still evolve as time passes by and more new text speaks, slangs, and lingos might be created by the younger generation.

There are two types of approach used in Aw and colleague’s paper (2006): basic word-based model and phrase-based model. In basic word model, an SMS word will be mapped to exactly one word. In phrase-based model, the SMS text will be split into k-phrases and the English words will also be split into k-phrases. Then, it will map the SMS phrase to an English phrase.

#### POS Tagger

The tagger produces a part-of-speech tag as an annotation on every word or symbol. These annotations produced can be used by a grammar checking tool to increase its power and coverage (Cunningham et al., 2002).

#### Gazetteer

The gazetteer contains lists of cities, organizations, days of the week, etc. It does not only contain entities, but also names of useful indicators, such as typical company designators (e.g. ‘Ltd.’), titles, etc. The gazetteer lists are collected into finite state machines, which can match tokens (Cunningham et al., 2002).

#### Lemmatizer

Lemmatization is the reduction of inflectional forms and sometimes derivationally related forms of a word to a common base form. It uses vocabulary and morphological analysis to remove inflectional ending and return the root word (Manning et al., 2008). The traditional method of lemmatizing is to use morphological rules and dictionaries. However, with the presence of new words, it will be very difficult for the lemmatizer. Statistical method needs a large training corpus. StaLe is a lightweight statistical lemmatizer. In StaLe, the system produces result tokens based on the rules. **Error! Reference source not found.** shows StaLe’s lemmatization process. Each token will be ranked according to its confidence factor and then pruned according to its candidate check-up phase. Those who pass will be the lemma of that word. However, if no token passed the candidate check-up phase, the input word will be the lemma. The problem with StaLe is that it sometimes produces a nonsense word resulting to a poorer outcome than a traditional dictionary-based lemmatizer.



Figure 3‑2. StaLe Lemmatization Process

#### Co-reference Resolution

This module consists of a main module and a set of submodules. The main module is responsible for initializing the submodules, and executes them in a particular order, then combines the results generated from the submodules, and eventually performs some post-processing over the result. There are three submodules in the main module: quoted-text module, pleonastic-it module, and pronoun-resolution module. The quoted text submodule recognizes the quoted fragments inside the text. The identified fragments are used by the pronoun-resolution submodule. The next module is the pleonastic-it submodule; it is responsible for finding pleonastic occurrences of “it”. The last and the main function of the co-reference resolution module is in the pronoun-resolution submodule. This submodule uses the results of the other two submodules after execution. The module works following an algorithm; first, it inspects the appropriate context for all candidate antecedents for this kind of pronoun and then chooses the best antecedent, if there is any. Then it creates the co-reference chains from the individual anaphor/antecedent (this step is performed from the main co-reference module) (Dimitrov, 2005).

#### Named-Entity Recognition

Named-entity recognition (NER) involves the automatic or semi-automatic processing of a series of words and then extracting or recognizing tokens in the text that refer to named entities (Lim et al., 2007). Named entities are phrases that contain the names of persons, organizations, and locations.

## Information Classification

Text classification or information classification is the automatic classification of text into different categories based on their content. It consists of several important components: document representation, dimensionality reduction, classification algorithm, and performance evaluations (Shen, 2010). This will be useful as different types of text may need different types of extraction techniques.

### Document Representation

Classification algorithms cannot understand texts directly. The text must be converted into some form that can be easily understood by the algorithm. There are different methods that could be used to represent documents. The traditional representation of documents is the Bag-of-Words (BOW) representation, which is based on the Vector Space Model. The use of BOW may vary as it can have different representations (Shafiei et al., 2007), one of which is

word representation. In word representation, each word in the document is considered as a feature. The problem with word representation is the ‘curse’ of dimensionality because text documents have a lot of unique words (Shafiei et al., 2007).

Another representation is term representation. Here, it uses multi-words or phrases as its feature. This drastically reduces the number of features. However, there has been mixed results on experimental results (Shafiei et al., 2007).

Character N-gram is another feature representation that could be used. Character N-gram takes n characters as a feature. Instead of focusing on the word, the character n-gram uses the characters. This makes model language independent. It is less susceptible to typographical errors and grammatical errors. It also does not require any linguistic preprocessing (Shafiei et al., 2007).

### Dimensionality Reduction (Feature Selection)

The problem with text classification is the huge number of features present in the vector space. This huge number of features could drastically reduce the performance of the algorithm. It is important that when a number of features are reduced, accuracy is not sacrificed. The reduction of feature is called feature selection. There are different methods that could be used in feature selection.

Document Thresholding (DF) counts all the occurrences of each word in the document, then all the words that did not reach the specified threshold will be removed. The rationale behind this is that those words that have few occurrences are irrelevant (Wei et al., 2010).

Information Gain (IG) measures bits of information that could be gained in a document. The information gain of a word (w) is defined as:

where is the set of all possible categories and is the probability of a document classified into a category. This will be computed for all the words in the documents. Then, the words that did not reach the specified threshold are removed (Wei et al., 2010).

Mutual Information (MI) is the modeling of the word to a category. The mutual information criterion between term *t* and category *c* is defined as:

and is estimated using

where,

A = number of times *t* and *c* co-occurs

B = number of times *t* occurs without *c*

C = number of time *c* occurs without *t*

N = number of documents

### Classification

There are different classification algorithms that could be used in classifying text. One of which is the Bag-of-Word technique. In the work of Sriram et al., (2010), they classified short-text messages (Tweets) into news (N), events (E), opinions (O), deals (D), and private messages (PM). They used Bag-Of-Words to classify the tweets. First, they were able to extract 8 features: author, presence of shortening of words, slangs, time-event phrases, opinion words, emphasis on words, currency, and percentages. They used the author feature to determine the type of user. Corporate tweeters composed their message in a professional way. It uses less slangs, emotions, and shortening because they wanted to convey their message clearly. On the other hand, personal tweets contain usage of slangs, emotions, and shortening. These features can be used to distinguish corporate tweeters from personal tweeters. They collected 5407 English tweets, broken down into N = 2107, O = 625, , D = 1100, E = 1057, , and PM = 518. They also contained 6747 unique words. For the classification, they tried different setups: BOW, BOW and author feature (BOW-A), BOW and the seven features (BOW-7F), the 8 features (8F), and BOW and the 8 features (BOW-8F).

Another type of classification that could be used is the k-nearest neighbor (k-NN). k-NN is an instance-based lazy learner. It means it only trains when a new instance comes in. k-NN computes for the k nearest instances (neighbors). Then, k-NN will use the neighbors’ categories to determine the class of the unknown instance. There are several ways to compute for the distance between the neighbors and the instances, Euclidean distance and Manhattan distance are some examples (Wajeed & Adilakshmi, 2011).

## Information Extraction Architecture

This section discusses the different architectures that can be applied in an information extraction system.

### Adaptive Architecture

The problem with some information extraction systems (knowledge-based systems) is that they are not portable and are highly dependent to the domain. With sources rapidly growing and becoming more diverse, it will be very hard for an information extraction system to extract as these text are unstructured, especially given the natural language used. Another problem is that an error may propagate as it goes through each module, as the modules in information extraction architecture are cascaded. The use of machine-learning techniques tries to solve these problems. Adaptive Information Extraction systems use machine-learning techniques to automatically learn rules that will extract certain information (Turmo et al., 2006).

-,inginformation proceed ,**Error! Reference source not found.**

,.ing,a the sthe , as wellhandles ing

Figure 3‑3. Architecture of LearningPinocchio

e**Error! Reference source not found.**

Figure 3‑4. Rule-Induction Step

sa - ve a specialistsa they

they are with

---**Error! Reference source not found.**

Figure 3‑5. Algorithm for Choosing the Best Rules

during The l**Error! Reference source not found.**

Figure 3‑6. Information Extraction Process of LearningPinocchio

Twas tested

were performedA was not done the ’sFsextracting the ’sFsa of

ID,,Twith was doneHowever, a of

#### IE2 (Aone et al., 1998)

Aone and his team of researchers (1998) have presented an adaptive Information Extraction system that can be used to extract information from different types of texts like unstructured, structured, and semi-structured texts. In their article, they presented the architecture they used in building the system. Aone’s IE system has six main modules in its architecture. Module 1 is responsible for the named-entity recognition part of the IE system. For this module, they used a commercial tool called NetOwl Extractor 3.0 to recognize general named-entity types. It is in this module where time/numerical expressions, names (persons/places/organizations), acronyms (organization names/locations), and semantic subtypes (country/city) are being recognized and extracted. Module 2 or the Custom NameTag module is responsible for the recognition of restricted-domain named-entities by using pattern matching. The output phrases for this module are SGML-tagged (Standardized Generalized Markup Language) into the same input document. On the other hand, Modules 3 and 4 are responsible for SGML-tagging the phrases in the sentences that are considered to be values for the slots defined in the templates and they work hand-in-hand. Module 3 or the PhraseTag module works by applying syntactico-semantic rules to identify the noun phrases in the previously recognized/extracted named-entities. Module 4 or the EventTag module works by applying a set of custom-built syntactico-semantic multi-slot rules to recognize/extract events from the input sentence. Module 5 or the Discourse Analysis Module is responsible for co-reference resolution or the merging of the previously extracted noun phrases. This module is implemented using three different strategies so that it can be modified to reach optimal performance regardless of the extraction scenario. Strategy A or the Rule-Based Strategy uses a set of custom-built rules to resolve definite noun phrases and singular personal pronoun co-reference. Strategy B or the Machine Learning-Based Strategy uses a decision tree that has been formed from learning a corpus tagged with co-references. Strategy C or the Hybrid Strategy uses Strategy A to filter false antecedents and then uses Strategy B to rank the remaining antecedents. In general, Module 5 is just merging the partial templates formed by the previous module. Lastly, Module 6 or the TempGen Module is responsible for the completion of the templates generated from the previous module by considering the consistency of the values in the slots of the event templates after resolving the noun phrase co-references and the generation of the output in the desired format. **Error! Reference source not found.** illustrates the architecture of the system proposed by Aone et al.



Figure 3‑7. Figure 3 7. Architecture of IE2 Adaptive Information Extraction System

#### SOMIDIA (Cheng et al., 2013)

SOMIDIA uses an adaptive information extraction system that extracts relevant information (English and Filipino) from different sources (i.e. blogs, social media sites, news articles). After crawling the internet for documents, the documents are fed to the information extraction system. First, it performs a tokenizer. They used OpenNLP to do the tokenization (OpenNLP, 2013). Then, it goes through the sentence splitter. It accepts a tokenized document. The system will now split the document into sentences. They use OpenNLP for the sentence detection (OpenNLP, 2013). After the sentence splitter, the document will be classified into English documents or Filipino documents. This is done because different information extraction modules will be applied for English and Filipino. For English, they used POS Tagger, Chunker, English NER, Co-reference Resolution and English Extractor. For Filipino, they used Filipino NER and Filipino Extractor. The English information extraction process has POS Tagger, Chunker, English NER, Co-reference Resolution, and English Extractor. The Filipino information extraction process has Filipino NER and Filipino Extractor. For the Filpino NER, they build their own gazetteer for there is no existing gazetteer for Filipino. They used dictionary-based and rule-based approach in implementing the NER. **Error! Reference source not found.** describes the architecture of SOMIDIA.

For SOMIDIA to adapt to new instances, the rules must be adaptable. SOMIDIA has a pattern extractor module that is mainly responsible for extracting different patterns from a set of documents and seed words so that they can be later used for the extraction process. SOMIDIA defines a document as any text that is related to the domain of the extraction system. This module of the system works in this manner: For each document, it will identify first the seed words present in the document. Seed words are words that will be extracted. For each seed word identified, the module will try to generate possible rules by using Windowing, a term to describe the section of the document that is considered for computation. The module experiments with all possible combinations of tokens and window setups to produce as much rules by considering a number of windowing concepts like the minimum window size, maximum left window size, and maximum right window size. The minimum window size is the minimum number of tokens that is included in the window. In addition, the maximum left window size is the maximum number of tokens included in the window that is found to the left of the seed word. On the other hand, the maximum right window size is the maximum number of tokens included in the window that is found to the right of the seed word. After generating all possible rules from the combination of tokens and various window setups, it then stores the generated rules for that specific seed word in a HashMap together with the number of times the rules were generated. This process is done continuously until rules are generated for all the seed words in the document and until all of the documents are completely processed.

After the process of generating rules, the module will do some optimization of the rules generated to further improve the efficiency of the extraction module. The module will minimize rules by removing rules that fall into these two scenarios: (1) rules that occur only once because they are too specific and they would only work with a very small percentage of the documents and (2) rules that are able to extract more than its corresponding occurrence because these rules are too general and may have the tendency to extract irrelevant data.



Figure 3‑8. SOMIDIA's Architecture

## Ontology

Ontologies are sets 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. Because ontologies are in the semantic level, they could be used to combine heterogeneous database, thus, making interoperability between systems possible (Gruber, 2009). Cimiano (2006) said that as the number of applications using ontologies is growing, then every such application must now be clearly and formally defined into an ontology.

Cimiano (2006) formally defines ontology as

Where,

.

In Vangelis et al. (2011), they presented four levels of classification on how an IE system exploited the ontology. The first level is the use of domain entities (including the variations), and word classes. For the first level, they can be represented by a gazetteer (flat) or ontologies (structured). By using ontologies, it can identify the text based on some constraints posed by the conceptual properties. An example system that uses the first level ontology is LearningPinocchio (Ciravegna & Lavelli, 2004). The second level uses concept hierarchies. In the second level, they focus more on taxonomic relations (consists of super/sub-ordination, is-a and part-of relationships). They could be used to generalize or specify extraction rules or check constraints. An example system is NAMIC (Basili et al., 2003). The third level uses the concepts’ properties and relationships between concepts. These properties and relationships could then be used as guides for the information extraction process. An example system would be OBIE (Wang et al., 2005). The fourth level is the domain model. It combines the first three levels to be able to semantically interpret information. Domain models can merge with different structures, check consistency, make valid assumptions (for missing values), and discover implicit information. An example is BOEMIE (Maedche, 2002). BOEMIE uses bootstrap or layered extraction process for its information extraction process. First, it extracts the entities, and then the relations. BOEMIE populates and enriches the ontology. It adds new individual entities and at the same time add snew concepts and relations.

### Ontology Design

In creating a domain-specific ontology, the following tasks must be done: selection of domain and scope, consideration of reusability, finding important terms, defining classes and class hierarchy, defining properties of classes and constraints and creation of instances of classes (Saloun & Klimanek, 2011).

There are different approaches to creating ontology: hand-making by expert, automatic, and semi-automatic. In hand-made by expert, the ontology is manually done by the experts. Its advantage is that the result will be in high quality. However, they are very expensive and time consuming. In an automatic approach, the creation of the model is done by a machine. It is fast and low cost, but the problem is that implementing it will be very difficult and will result to inaccurate models. In a semi-automatic approach, the concepts and relationship will be generated by a machine, and the expert will complete and validate it. It produces relatively good results at a short amount of time. The disadvantage is that the machine-generated concepts and relations might be inaccurate and might also cause an inconvenience (Saloun & Klimanek, 2011).

### Ontology Population

Ontology Population is the extraction and classification instances of classes and relationships of an ontology. There are three approaches for ontology population: manual, semi-automatic and automatic approaches. The manual population of ontology should be done by experts and a knowledgeable engineer. This could be costly and time consuming and the automatic approach might be inaccurate. For automatic and semi-automatic approaches, they have a common approach. They do entity name recognition, NLP techniques, and information extraction.

In Faria & Girardi (2011), the techniques they used are NLP and IE. The process has two phases: Extraction and Classification of Instances and Instance Representation. For the Extraction and Classification of Instances, all the possible relationships and class instances are generated. They consist of Corpus Analysis (Morpho-lexical analysis, Named-Entity Recognition and Co-reference), Specification of Extraction and Classification Rules, and Extraction and Classification of Instances. Then, they will manually generate a set of extraction rules based on the last task. After generating the rules, it will now use the extraction rules to look for text matching the patterns. This will now produce the instances. After the first phase, it will now go to Instance Representation. Instance Representation has two tasks: Refinement of Instances and Ontology Population. For Refinement of Instances, it will try first to see if the instance already exists in the ontology. If it does not, then it will go to . If it already exists in the ontology, it will look in to see if the instance needs to be updated. If it is, then it will be part of If not, it will be discarded. After refinement, the instance is now ready to fill the ontology. Given , it will now look in the ontology to find the class. Then if a class is found, the instance will now be instantiated. **Error! Reference source not found.** shows the process of Faria & Girardi’s (2011) semi-automatic ontology population.

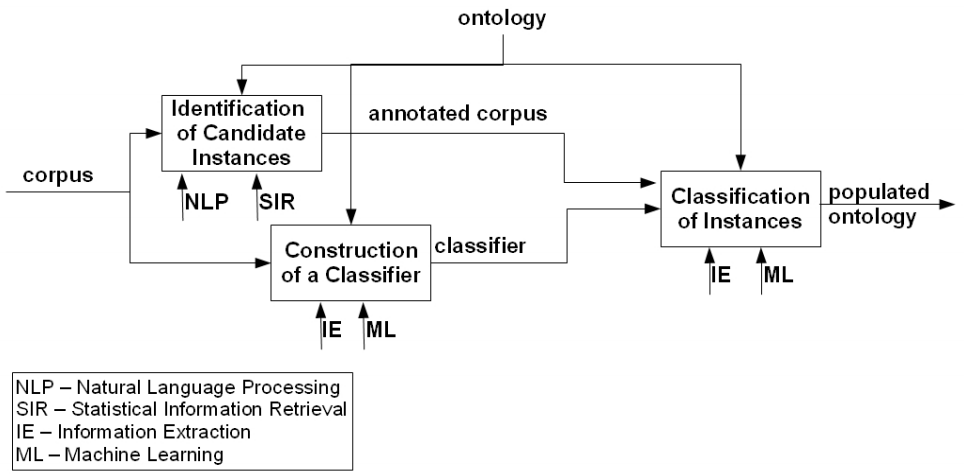


Figure 3‑9. Process of Semi-Automatic Ontology Population

## Twitter[[4]](#footnote-5)

Twitter is a microblogging social media platform wherein users may post messages of up to 140 characters long. Each of these posts is known as a "tweet". Mainly, a tweet is an expression of a moment or idea. Tweets may contain text, photos, and videos. Millions of tweets are shared in real time, every day.

A tweet may be replied to, retweeted, ‘favorited’, and may contain hashtags. A "reply" to a tweet is when another user comments or joins in the conversation of a tweet. A "retweet" is where you share the tweet of another user. A "favorite" indicates that a user likes the tweet. "Hashtags" assign a topic to the tweet. Thus, if one searches for #WorldYouthDay, the search results will contain all tweets with related topics about World Youth Day. When a Twitter user "follows" another user, this means that they subscribe to the tweets posted by that user (Twitter, n.d.).

### Use of Twitter

Aside from Twitter’s social media aspect, Twitter has been used as a source of data for various fields, one of which is in disaster management (Imran et al., 2013). Other fields that Twitter data have contributed to are linguistics (Mocanu et al., 2013), prediction (Tumasjan et al., 2010; Choy et al., 2012), real-time event detection (Sakaki et al., 2010), marketing (Jansen et al., 2009; Bollen et al., 2011), sentiment analysis, and opinion mining (Pak et al., 2010), education (Grosseck et al., 2008; Junco et al., 2011), newscasting (Phelan et al., 2009), medicine (Hawn, 2009; Chew & Eysenbach, 2010), and business processes (Culnan et al., 2010).

### Twitter and Disasters

During disasters, Filipino Twitter users tend to retweet 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 users that use their native language when tweeting during disasters (Lee et al., 2013).

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 (Official Gazette of the Republic of the Philippines, 2012). It shows some of the official twitter accounts of government institutions as well as the official hashtags being used during disasters. It also shows the extractable information from the tweets per disaster.

|  |  |  |
| --- | --- | --- |
| **Category** | **Official Government Institution**  **Twitter Account** | **Unified Hashtag** |
| Typhoon | @dost\_pagasa | #(storm name)PH  (i.e. #YolandaPH, #GlendaPH) |
| Flood | @PAGASAFFWS, @MMDA | #FloodPH |
| Volcanic activities, earthquakes, and tsunamis | @phivolcs\_dost | #EarthquakePH |
| Relief and rescue efforts | @PIAalerts, @PIANewsDesk, @NDRRMC\_Open, @pcdspo, @DSWDserves | #ReliefPH  #RescuePH |
| Suspension of classes | @DepEd\_PH | #walangpasok |

Table 3‑1. Examples of official government institution

|  |  |  |
| --- | --- | --- |
| **Type of Disaster** | **Tweet** | **Extractable Information** |
| Typhoon | @ANCALERTS:  NDRRMC says 77 dead, 220 injured, 5 missing due to Typhoon Glenda #GlendaPH | * 77 dead * 220 injured * 5 missing * Typhoon Glenda |
| Typhoon | @ABSCBNChannel2:  Bagyong Glenda patuloy na nagbabanta sa Luzon. #GlendaPH pic.twitter.com/2ygRWj6Z3D | * Typhoon Glenda * Luzon |
| Typhoon | @rapplerdotcom:  #GlendaPH: Marikina River now at alert level 1 [rplr.co/1mSTdnR](http://t.co/rqpfnzcLza" \t "_blank)[pic.twitter.com/mECHfZfiyK](http://t.co/mECHfZfiyK" \t "_blank) | * Marikina River * Alert level 1 |
| Typhoon | [@ABSCBNNews](https://twitter.com/ABSCBNNews/" \t "_blank):  200 families in Laguna lose homes due to 'Glenda' [bit.ly/UfEDeO](http://t.co/6Kn0frqBsJ" \t "_blank)[#southAlerts](https://twitter.com/search?q=%23southAlerts" \t "_blank)[#GlendaPH](https://twitter.com/search?q=%23GlendaPH" \t "_blank) | * 200 families * Laguna * Glenda |
| Earthquake | @dswdserves:  DSWD Region 11 prepositioned 12,170 food packs&55,206 assorted food for victims of recent quake in Davao Occ. [#EarthquakePH](https://twitter.com/search?q=%23EarthquakePH" \t "_blank)[@dinkysunflower](https://twitter.com/dinkysunflower/" \t "_blank) | * DSWD Region 11 * 12,170 food packs * 55,206 assorted food * Davao Occ |
| Earthquake | [@phivolcs\_dost](https://twitter.com/phivolcs_dost/" \t "_blank):  No expected damage from 6.1-magnitude [#earthquakePH](https://twitter.com/search?q=%23earthquakePH" \t "_blank) off Davao Occidental; aftershocks expected: [bit.ly/1ra30ZZa](http://t.co/IRX5SMSr3h" \t "_blank) | * 6.1 magnitude * Davao Occidental |
| Earthquake | @manila\_bulletin:  BREAKING: 6.1 magnitude quake felt, east of Davao at 3:59PM. #EarthquakePH | * 6.1 magnitude * Davao * 3:59pm |
| Earthquake | @seanbofill:  Magnitude 6.1 earthquake recorded in Davao earlier today. #EarthquakePH | * Magnitude 6.1 * Davao |
| Flood | [@saabmagalona](https://twitter.com/saabmagalona/" \t "_blank):  Ortigas st across La Salle GH ankle-deep [#floodph](https://twitter.com/search?q=%23floodph" \t "_blank) | * Ortigas st * La Salle GH * Ankle-deep |
| Flood | [@MMDA](https://twitter.com/MMDA/" \t "_blank):  [#FloodPH](https://twitter.com/search?q=%23FloodPH" \t "_blank): As of 11:12 am, Orense to Estrella Southbound, leg deep, not passable to light vehicles | * 11:12am * Orense * Estrella Southbound * Leg deep * Not passable to light vehicles |
| Flood | @rqskye:  @MovePH MT @PIAalerts 5m: #FLOODPH ALERT: Greenhills, La Salle Street, San Juan, Metro Manila: Knee-high. #TrafficPH | * Greenhills * La Salle Street * San Juan * Metro Manila * Knee-high |
| Flood | @rqskye:  @MovePH MT @MakatiTraffic 11:27am: Flooded area in Brgy. Pio del Pilar: Medina St. corner... tl.gd/n\_1s2geia |#FloodPH #TrafficPH | * 11:27am * Brgy. Pio del Pilar * Medina St. corner |

Table 3‑2. Examples of disaster-related tweets with extractable information

## Evaluation Metrics

This section discusses the different metrics that will evaluate the performance of the information extraction system.

### F-measure

Precision and recall are the two primary metrics. Given a subject and a gold standard, precision is the percentage of cases that the subject is correctly classified as positive or true in the gold standard.

Recall is the percentage of cases in the gold standard that is correctly classified as positive or true by the subject.

The two metrics are often combined as their harmonic mean known as the F-measure (Hripcsak and Rothschild, 2005).

The True positive category means a positive instance is correctly predicted as positive while the False positive category denotes a negative instance is predicted as positive. Then, the True negative category signifies a negative instance predicted correctly as negative while the False negative means a positive instance is predicted as negative (Davis and Goadrich, 2006). Table 3‑3 shows its confusion matrix.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

Table 3‑3. Confusion Matrix (Davis and Goadrich, 2006)

### Kappa Statistics

The common way of summarizing interrater agreement among observers is the kappa statistics. It allows measurement not only by chance and the observed agreement beyond chance is divided by the maximum agreement (beyond chance) that is possible for the dataset. The general kappa formula is

where and are the observed and expected proportions of agreement, respectively (Malpica et al., 2005).

## Tools

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

### 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. This will be used for the POS tagger and the JAPE.

#### Gazetteer

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

#### Sentence Splitter

The sentence splitter uses finite state transducers to split the text into sentences. It uses the gazetteer to check if punctuation is part of abbreviations or signals the end of the sentence. The sentence is 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 rule sets that have been trained in the Wall Street Journal corpus. However, the lexicon and rule sets 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 co-reference.

### 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 and it is also flexible as users can extend the API to customize the machine-learning algorithms (Weka 3, n.d.).

### JENA API (McBride, 2002)

JENA is a semantic web application that helps in building ontologies. It is a Java-based API that handles OWL and SPARQL. It also includes inference engines based on OWL and RDF. This will be used to create and manage the ontology.

Code Listing 3‑1 shows how to create an ontology.

|  |
| --- |
| **Code Listing:** |
| OntModel ontModel = ModelFactory.createOntologyModel(<model spec>); |

Code Listing 3‑1. Example code to create an ontology

Code Listing 3‑2 shows how to create a class.

|  |
| --- |
| **Code Listing:** |
| Resource r = m.getResource(NS+”Paper”);  OntClass paper = r.as(OntClass.class); |

Code Listing 3‑2. Example code to create a class

Code Listing 3‑3 shows how to create object properties.

|  |
| --- |
| **Code Listing:** |
| ObjectProperty hasProgramme = m.createObjectProperty( NS + "hasProgramme" );  hasProgramme.addDomain( orgEvent );  body.addRange( programme );  body.addLabel( "has programme", "en" ); |

Code Listing 3‑3. Example code to create object properties

Code Listing 3‑4 shows how to create instance/individuals.

|  |
| --- |
| **Code Listing:** |
| OntClass c = m.createClass( NS + "SomeClass" );  Individual ind0 = m.createIndividual( NS + "ind0", c );  // second way: use a call on OntClass  Individual ind1 = c.createIndividual( NS + "ind1" ); |

Code Listing 3‑4. Example codes to create an instance

### ArkNLP (Gimpel et al., 2011)

ArkNLP developed by Carnegie Mellon is a Java-based Tokenizer and POS tagger that was specifically made for Twitter. For the tokenizer, it now identifies the emoticon tokens. For the POS tagger, it can also tag slangs and emoticons. This will be used for tokenizing the tweets.

|  |
| --- |
| **Code Listing:** |
| List<String> tokens = Twokenize.tokenizeRawTweetText(text); |

Code Listing 3‑5. Example code for tokenizing text.

### NormAPI (Nocon et al., 2014)

NormAPI is a text normalization API that is specifically built for the Filipino language. It currently has implementations for Dictionary Substitution Approach (DSA) and Statistical Machine Translation (SMT). The user can choose if the normalization will perform: (1) DSA only, (2) SMT only, (3) SMT after DSA, or (4) SMT before DSA. NormAPI accepts file or text as inputs. It also allows setting configuration files and training a new model. This will be used for the text normalization.

# 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 of the system. 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.

## System Overview

Filipino Information Extraction for Twitter (FILIET) is a hybrid information extraction system that incorporates the architectures of an adaptive IE system and a rule-based IE system for Filipino disaster related tweet. The FILIET system will work with extracting information from tweets that were written in Filipino and English, along with their variations such as TXTSPK and code-switch. The system will follow the methodology described below. The disaster-related tweets will be loaded into the system. The system will then classify according to the following categories: (1) caution and advice, (2) casualties and damage, (3) donations, and (4) others. The tweets will now proceed to the information extraction engine of the system wherein the system will extract trelevant information from the tweets with regard to its given type of disaster. Extracted information from the given tweets will vary based on the type of information the tweet contains.

## System Objectives

This section will discuss the objectives of the system.

### General Objective

To develop an information extraction system that extracts relevant information from disaster-related tweets and considers the different available variations of the Filipino language.

### Specific Objectives

The following are the specific objectives of the system:

1. To preprocess the tweets;
2. To extract relevant features from the tweets;
3. To classify the tweets into according to their content (i.e. caution and advice, casualties and damages, donations, and others);
4. To extract relevant information according to the type of tweet.

## System Scope and Limitations

The system to be developed in this research is expected to be able to do a number of tasks that are within the scope of extracting information from Filipino disaster-related tweets. These tasks include the following: Text Preprocessing, Feature Extraction, Disaster Classification, and actual Information Extraction.

The system must be able to perform some preprocessing techniques onto the input tweet. These preprocessing tasks shall be limited to the following: (1) text normalization to include support for input tweets that were written in the TXTSPK format; (2) text tokenization, to enable word level analysis of the input tweet; (3) part-of-speech tagging, to enable semantic level analysis of the input tweet; (4) named-entity recognition, to enable proper identification of named-entities; and lastly, (5) disaster keyword tagging, to enable proper recognition of disaster words in the input tweet. Lastly, by looking at the initial data and from the study of (Lee et al., 2013), it was observed that a high probability that Filipinos will post tweets in the Filipino language and that TXTSPK and code-switching were the variations being used.

Moving on, the system must be able to extract features from the input tweet. The features that will be extracted from the input tweet are categorized into two: (1) binary features, those that have discrete values 0 and 1; and (2) nominal features, those that have continuous values. For the binary features, they will be limited to the following: Presence features (presence of keywords like disaster words, mentions, hashtags, emoticons, retweets, and Code Switching).On the other hand, the nominal features will be limited to the following: (1) Tweet length; (2) User; and lastly, (3) Location.

Using the extracted features, the system must be able to classify the input tweet based on the type of tweets. The tweet must be classified into the following: caution and advice (CA), casualties and damage (CD), donations (D), and others (O). This is important because each type of tweet will have different extracted information. The categories are based on Extracting Relevant Information Nuggets from Disaster-Related Messages in Social Media by (Imran et al., 2013).

The system must be able to extract two types of information from the given input tweet. The two main types of information are (1) General Information and the (2) Type-Specific Information. For the General Information, only the location references, time references, and source shall be extracted from the input tweet. On the other hand, for the Type-Specific Information, the following shall be extracted from the input tweet: (a) for caution and advice tweets: the caution and/or advice part of the tweet; (b) for casualties and damage tweets: the number of casualties and the damaged objects; and (c) for donation tweets: if the tweet is a donation effort or a request for help and what are the objects being donated or requested. The information to be extracted are also based on the study by (Imran et al., 2013).

The data that will be used in the development of the system will come from the Twitter Web Crawler developed by the De La Salle University - College of Computer Studies as well as from the crawler to be developed by the group. The system will only be processing data that are written in the Filipino language.

## Architectural Design

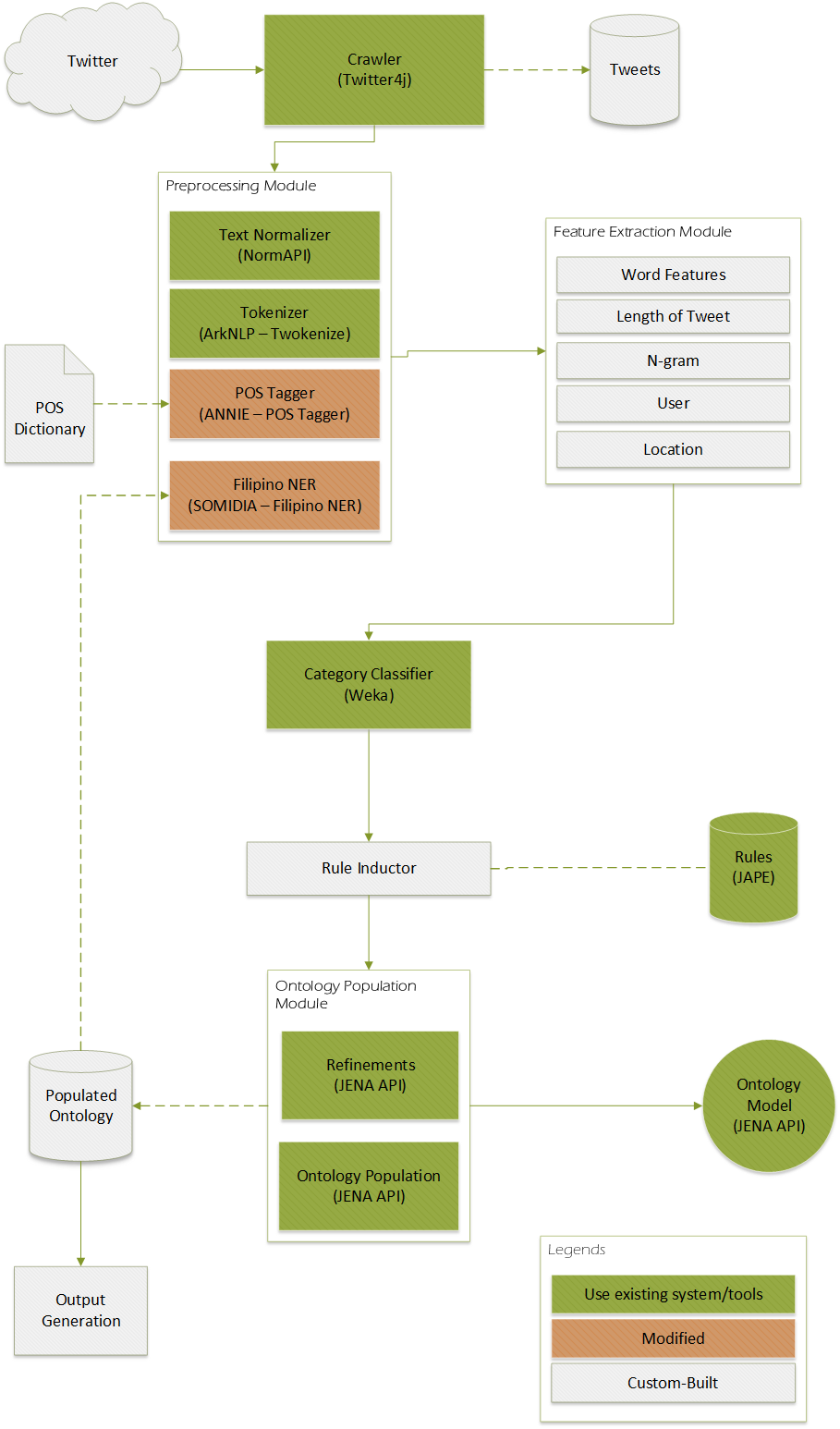


Figure 4‑1. Architectural Design

### Crawler Module

This module will be crawling Twitter to retrieve tweets. The system will continuously collect the tweets using Twitter’s Stream API through the Twitter4j library.

### Preprocessing Module

This module will be responsible for preprocessing the input tweets before they are passed on to the information extraction module. This module will include the following text processing techniques: text normalizer, tokenizer, and POS Tagger. After going through this module, the preprocess tweets will then be passed on to the Information extraction module.

#### Text Normalizer

The first step in preprocessing the input tweets is text normalization. The main responsibilities of the text normalizer are (1) to convert the TXTSPK format of the tweets into full-word format so that the information when extracted will be consistent and(2) remove emoticons, links, and hashtags. The text normalizer will accept a text as input. The output of this module is the normalized tweets where the TXTSPK is converted to its full form, and links and emoticons are removed. For this module, the researchers will use NormAPI (Nocon et al., 2014). Table 4‑1 shows a sample input and its corresponding output.

|  |  |
| --- | --- |
| **Input** | **Output** |
| <tweet>  Dear Adnu sana po damit naman ang idonate natin para sa mga binagyo in case na may donation na ganapin. Plus canned goods na rin. Haha. :)  </tweet> | <tweet>  Dear Adnu sana po damit naman ang idonate natin para sa mga binagyo in case na may donation na ganapin. Plus canned goods na rin. Haha.  </tweet> |
| <tweet>  Kailangan na talaga ng military efforts sa most part of Leyte. Nagkakagulo na.</tweet> | <tweet>  Kailangan na talaga ng military efforts sa most part of Leyte. Nagkakagulo na.</</tweet> |

Table 4‑1. Sample Input/Output for Text Normalizer

#### Tokenizer

After normalizing the tweets, the tokenizer will now split the input tweets into tokens like numbers, punctuations, words, abbreviations and other special characters like emoticons, hashtags, mentions, and the like. The tokenizer will take as an input the normalized tweet from the Text Normalizer. The tokenizer will output an array containing the tokenized tweet in a form that is similar to this. Tokenized = {“@<username>”, “<punctuations>”, “#<hashtag>”...} or an array that would contain all the tokens in a given tweet. For this module, the researchers will use ArkNLP’s Twokenize (Gimpel et al., 2011).  Table 4‑2

shows a sample input and its corresponding output.

|  |  |
| --- | --- |
| **Input** | **Output** |
| <tweet>  Dear Adnu sana po damit naman ang idonate natin para sa mga binagyo in case na may donation na ganapin. Plus canned goods na rin. Haha.  </tweet> | <tweet>  "Dear", "Adnu", "sana", "po", "damit", "naman", "ang", "idonate", "natin", "para", "sa", "mga", "binagyo", "in", "case", "na", "may", "donation", "na", "ganapin", ".", "Plus", "canned", "goods", "na", "rin", ".", "Haha", "."</tweet> |
| <tweet>  Kailangan na talaga ng military efforts sa most part of Leyte. Nagkakagulo na.</</tweet> | <tweet>  "Kailangan", "na", "talaga", "ng", "military", "efforts", "sa", "most", "part", "of", "Leyte", ".", "Nagkakagulo", "na", "."</tweet> |

Table 4‑2. Sample Input/Output Tokenizer

#### POS Tagger

After tokenizing the tweets, the POS tagger will accept the tokenized Filipino tweet as an input and then, it will tag each token with its corresponding part-of-speech. Each of the tokens can be tagged as a noun, a verb, an adjective, an adverb or others. After tagging the tokens, the POS tagger will then output the tokens with their corresponding POS tag in the form of a text. For the module, the researchers are considering modifying ANNIE’s POS Tagger (Cunningham et al, 2002) for Filipino, or use Filipino Tagger Dictionary (Oco & Borra, 2011). Table 4‑3 shows the sample input and output of POS tagger.

|  |  |
| --- | --- |
| **Input** | **Output** |
| <tweet>  "Dear", "Adnu", "sana", "po", "damit", "naman", "ang", "idonate", "natin", "para", "sa", "mga", "binagyo", "in", "case", "na", "may", "donation", "na", "ganapin", ".", "Plus", "canned", "goods", "na", "rin", ".", "Haha", "."</tweet> | <tweet>  "Dear\_UH", "Adnu", "sana\_VOTF", "po\_MAHM", "damit\_NCOM", "naman\_ENCL", "ang\_NA", "idonate", "natin\_PNGP", "para\_PRTA", "sa\_NCOM", "mga\_NA", "binagyo", "in\_IN", "case\_VBP", "na\_NA", "may\_MAEM", "donation\_NN:UN", "na\_NA", "ganapin", ".\_PSNS", "Plus\_JJ", "canned\_JJ", "goods\_NNS", "na\_NA", "rin\_ENCL", ".\_PSNS", "Haha\_NN", ".\_PSNS"</  tweet> |
| <tweet>  "Kailangan", "na", "talaga", "ng", "military", "efforts", "sa", "most", "part", "of", "Leyte", ".", "Nagkakagulo", "na", "."</tweet> | <tweet>  "Kailangan\_VOTF", "na\_NA", "talaga\_IRIA", "ng\_NA", "military\_NCOM", "efforts\_NNS", "sa\_NCOM", "most\_JJS", "part\_JJ", "of\_IN", "Leyte\_NPRO", ".\_PSNS", "Nagkakagulo", "na\_NA", ".\_PSNS" </tweet> |

Table 4‑3. Sample Input/Output POS Tagger

#### Filipino NER

The Filipino NER will identify the proper nouns in the tweets. The module will accept the tweets that have passed through the preprocessing module. The outputs of the NER are tagged as proper nouns in the tweet. For the gazetteer, the plan is to use the SOMIDIA gazetteer and update the gazetteer. Table 4‑4shows a sample input and its corresponding output.

|  |  |
| --- | --- |
| **Input** | **Output** |
| <tweet>  "Dear\_UH", "Adnu", "sana\_VOTF", "po\_MAHM", "damit\_NCOM", "naman\_ENCL", "ang\_NA", "idonate", "natin\_PNGP", "para\_PRTA", "sa\_NCOM", "mga\_NA", "binagyo", "in\_IN", "case\_VBP", "na\_NA", "may", "donation\_NN:UN", "na\_NA", "ganapin", ".\_PSNS", "Plus\_JJ", "canned\_JJ", "goods\_NNS", "na\_NA", "rin\_ENCL", ".\_PSNS", "Haha\_NN", ".\_PSNS"</tweet> | <tweet>  "Dear\_UH", "Adnu", "sana\_VOTF", "po\_MAHM", "damit\_NCOM", "naman\_ENCL", "ang\_NA", "idonate", "natin\_PNGP", "para\_PRTA", "sa\_NCOM", "mga\_NA", "binagyo", "in\_IN", "case\_VBP", "na\_NA", "may", "donation\_NN:UN", "na\_NA", "ganapin", ".\_PSNS", "Plus\_JJ", "canned\_JJ", "goods\_NNS", "na\_NA", "rin\_ENCL", ".\_PSNS", "Haha\_NN", ".\_PSNS"</tweet> |
| <tweet>  "Kailangan\_VOTF", "na\_NA", "talaga\_IRIA", "ng\_NA", "military\_NCOM", "efforts\_NNS", "sa\_NCOM", "most\_JJS", "part\_JJ", "of\_IN", "Leyte\_NPRO", ".\_PSNS", "Nagkakagulo", "na\_NA", ".\_PSNS" </tweet> | <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> |

Table 4‑4. Sample Input/Output Gazetteer

### Feature Extraction Module

This module is responsible for extracting the feature from the tweet. The module will extract the presence of disaster words, tweet length, character n-gram, user, location, and trusted accounts. The Feature Extraction Module will take the preprocessed tweets as inputs, then output the tweet with the features. Table 4-13 shows a sample of the features and their respective values.

#### Presence

The Presence feature is a binary feature that indicates the presence of keywords like disaster words, mentions, hashtags, emoticons, retweets, and Code Switching in the input tweet. The value of “1” is given if the keyword is present; otherwise it is given “0”.

#### Tweet Length

The Tweet Length feature essentially counts the length of the input tweet.

#### N-gram

The N-gram feature is mainly responsible for generating/extracting the different n-grams for the input tweets, specifically, the bi-gram and the tri-gram of the input tweets. To accomplish the n-gram generation/extraction tasks, the module will make use of the SRILM tool, which is specifically built for generating/extracting n-gram models.

#### User

The User feature will help in determining the type of disaster. For example, @dost\_pagasa will tweet about typhoons.

#### Location

The location feature is where the disaster occurred. There are instances which are specific to certain disasters, for example, the disaster is flood, and the location given is usually a street. It can be also be a region, city or province for typhoon- or earthquake- related tweets.

## Category Classifier Module

Using the extracted features, the Category Classifier Module will classify the tweets into the following categories: (1) caution and advice (CA), (2) donation (D), (3) casualties and damage (CD), and (4) others (O). The module will use Weka (Weka, n.d.) and will try out different classifiers. Table 4‑5 shows a sample input/output of the Category Classifier Module.

|  |  |
| --- | --- |
| **Input** | **Output** |
| <tweet>  "Dear\_UH", "Adnu", "sana\_VOTF", "po\_MAHM", "damit\_NCOM", "naman\_ENCL", "ang\_NA", "idonate", "natin\_PNGP", "para\_PRTA", "sa\_NCOM", "mga\_NA", "binagyo", "in\_IN", "case\_VBP", "na\_NA", "may", "donation\_NN:UN", "na\_NA", "ganapin", ".\_PSNS", "Plus\_JJ", "canned\_JJ", "goods\_NNS", "na\_NA", "rin\_ENCL", ".\_PSNS", "Haha\_NN", ".\_PSNS"</tweet> | <tweet type=”D”>  "Dear\_UH", "Adnu", "sana\_VOTF", "po\_MAHM", "damit\_NCOM", "naman\_ENCL", "ang\_NA", "idonate", "natin\_PNGP", "para\_PRTA", "sa\_NCOM", "mga\_NA", "binagyo", "in\_IN", "case\_VBP", "na\_NA", "may", "donation\_NN:UN", "na\_NA", "ganapin", ".\_PSNS", "Plus\_JJ", "canned\_JJ", "goods\_NNS", "na\_NA", "rin\_ENCL", ".\_PSNS", "Haha\_NN", ".\_PSNS"</tweet> |
| <tweet>  "Kailangan\_VOTF", "na\_NA", "talaga\_IRIA", "ng\_NA", "military\_NCOM", "efforts\_NNS", "sa\_NCOM", "most\_JJS", "part\_JJ", "of\_IN", "Leyte\_NPRO", ".\_PSNS", "Nagkakagulo", "na\_NA", ".\_PSNS" </tweet> | <tweet type=”D”>  "Kailangan\_VOTF", "na\_NA", "talaga\_IRIA", "ng\_NA", "military\_NCOM", "efforts\_NNS", "sa\_NCOM", "most\_JJS", "part\_JJ", "of\_IN", "<location: Leyte/>", ".\_PSNS", "Nagkakagulo", "na\_NA" ".\_PSNS"</tweet> |

Table 4‑5. Sample Input/Output Category Classifier Module



### Rule Inductor

The Rule Inductor module will accept tokenized and tagged tweets. It will now apply the rules coming from the database. It will look for patterns in the text and apply the classification. It will generate the instances that will be used to populate the ontology.

### Ontology Population Module

The ontology population module is responsible for filling up the ontology with instances. It has two modules: Refinements and Ontology Population.

#### Refinements

The Refinements module will be responsible for checking the instance’s uniqueness. If the instance is not found in the ontology, it will be placed in a container *I*. If it is found, it will see if the instance in *I* needs to be updated. If the instance needs to be updated, it will be added in *I*. Otherwise, it will be discarded.

#### Ontology Population

After the Refinements module, the Ontology Population module will receive the instances in *I*. For each instance in *I*, it will look for its matching class. If a match is found, the instance will be added to the ontology.

### Data Source

The data that will be collected will come from the filtered tweets. Some of these will be provided by the Twitter Web Crawler developed by the De La Salle – College of Computer Studied, while the rest will come from the Crawler module to be discussed in the next section. The list of trusted Twitter accounts is based on the list provided by SOMIDIA.

To be able to crawl the tweets that are strictly related to disaster relief operations, the researchers will make use of certain national official hashtags that are used by a number of relief organizations in the country. Examples of the unified hashtags are #ReliefPH, #RescuePH, #PHalert

The output of the crawler will be saved in a CSV file. Each entry in the CSV file will have the following content: <tweet ID>,<username>,”<tweet>”,”<date and time it was tweeted>”,<longitude>,<latitude>. shows a sample of what can be seen in the CSV file.

|  |  |
| --- | --- |
| **#** | **Sample Output** |
| 1 | 5280d16567833c59e17ebb66, SandyCervas, Dear Adnu sana po damit naman ang idonate natin para sa mga binagyo in case na may donation na ganapin. Plus canned goods na rin. Haha. :) , 11/11/2013 8:45, 13.7053384, 123.1980436 |
| 2 | 414017377517326337,Ehmai123,"""@ANCALERTS: Magnitude 4.3 quake jolts Antique, Boracay http://t.co/c2BczJEa6Y"" Lindol everywhere :3","Fri Dec 20 21:00:09 CST 2013",14.527157,121.0033549 |

Table 4‑6. Sample Entries of Tweets in CSV File

#### Gazetteer

The gazetteer is a text file that contains the list of names and locations to identify the proper nouns in the tweets. This will be used for the Filipino NER module. The plan is to update and use SOMIDIA’s gazetteer. Table 4‑7 shows a sample gazetteer for the storm names in the Philippines.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Agaton  Amang  Ambo  Auring  Basyang  Bebeng  Bising  Butchoy  Caloy  Chedeng  Cosme  Crising  Dante  Dindo  Dodong  Domeng  Egay  Emong  Enteng  Ester | Falcon  Feria  Florita  Frank  Gener  Gloria  Goring  Gorio  Hanna  Helen  Henry  Huaning  Igme  Inday  Ineng  Isang  Jolina  Juan  Juaning  Julian | Kabayan  Karen  Katring  Kiko  Labuyo  Lando  Lawin  Luis  Marce  Maring  Milenyo  Mina  Nando  Neneng  Nina  Nonoy  Ofel  Ompong  Ondoy  Onyok | Pablo  Paeng  Pedning  Pepeng  Quedan  Queenie  Quiel  Quinta  Ramil  Ramon  Reming  Rolly  Santi  Seiang  Sendong  Siony  Tino  Tisoy  Tomas  Tonyo | Udang  Unding  Ursula  Usman  Venus  Vinta  Violeta  Viring  Waldo  Weng  Wilma  Winnie  Yayang  Yolanda  Yoyong  Yoyoy  Zeny  Zigzag  Zoraida  Zosimo |

Table 4‑7. Sample Gazetteer for Storm Names (Philippines)

#### Rules

Based on the tweets, the rules will be handcrafted using JAPE. Then, the rules will now be stored in the database which will be used for extracting the information. Table 4‑8 shows a sample of the rules.

|  |
| --- |
| **Rules** |
| <string: naman><disaster><string:sa> AS Disaster |
| <string: magnitude><number>AS Intensity |
| <POS: NNS><location><POS: PSNS>AS Location |

Table 4‑8. Sample Extracted Rules

#### Seed Words

The seed words will be used for generating the rules. The list of seed words will be stored in a text file. It will SOMIDIA’s seed word and will update it. Table 4‑9 shows the excerpts of the list of seed words.

|  |  |  |
| --- | --- | --- |
| tubig  kuryente  pagkain  tulong  donation  damit  gutom  water  clothes  food  help  bahay | rice  kanin  bigas  inumin  sardinas  sardines  canned goods  instant noodles  damit  pera  gamot  medicine | health kit  medical kit  relief goods  kasuotan |

Table 4‑9. Excerpts of the List of Seed Words

#### POS Dictionary

The POS Dictionary is a dictionary that contains a list of words with its POS tag. This will be used in the POS Lookup. The dictionary is stored in a file. It contains a list of English and Filipino words. Table 4‑10 shows a sample of the excerpts of the POS dictionary

|  |
| --- |
| storms storm ENG NNS  storms storm ENG VBZ  storm storm ENG NN  storm storm ENG VB  bukid bukid TAG NCOM 2  bukirin bukirin TAG NCOM 2  buko buko TAG NCOM 2  bula bula TAG NCOM 2  bulag bulag TAG NCOM 2  bulak bulak TAG NCOM 2  bulalas bulalas TAG NCOM 2 |

Table 4‑10. Excerpts of the POS Dictionary

#### Ontology

For the ontology, this will be created manually. The domain of the ontology will be disaster, specifically for relief operations. The next step would be identification of the terms. After identifying the terms, the concept, properties, and constraints will be defined. Class instantiations then follow. The format of the ontology will be in OWL (Code Listing 6‑1). Figure 4‑2 shows the ontology of the system.

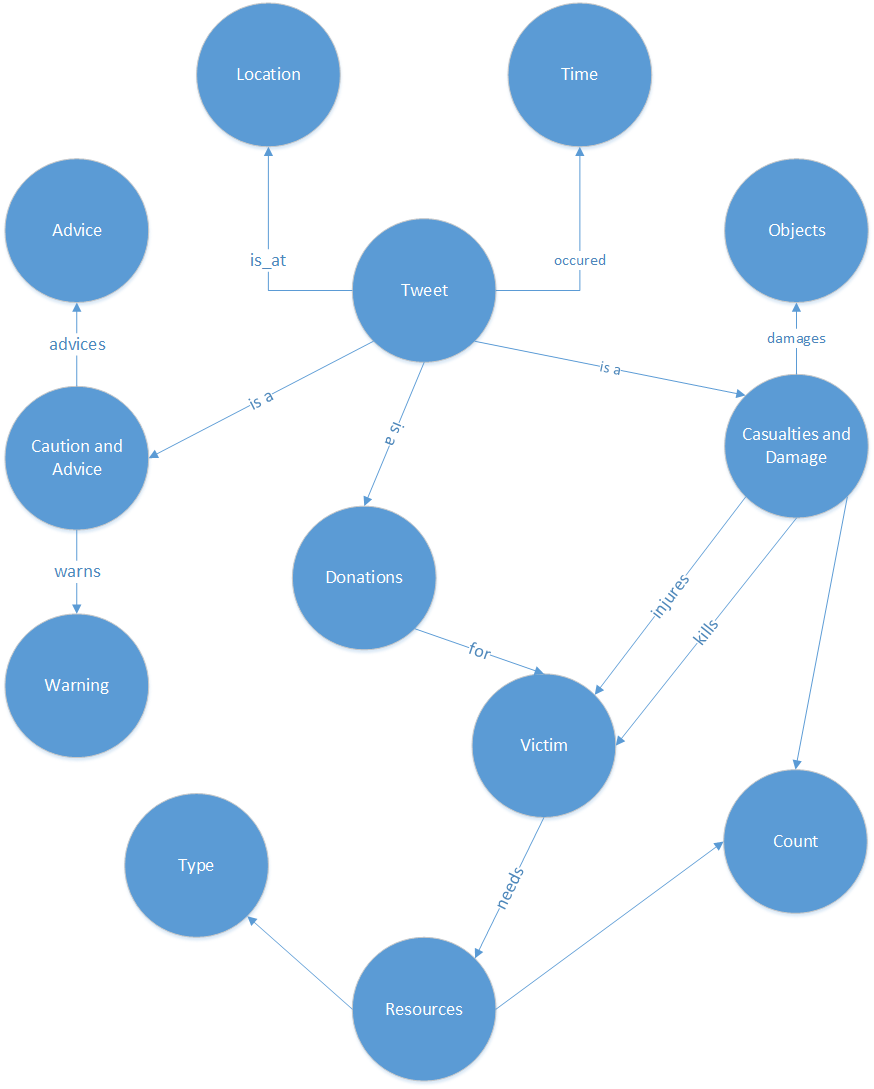


Figure 4‑2. FILIET Ontology

## System Functions

This section discusses the functions of the proposed systems.

### Tweet Retrieval

In this function, the system will access the tweets that were stored in the database by the Twitter crawler. The user can opt to filter the tweets for retrieval. Figure 4‑3 shows the screenshot of this function.

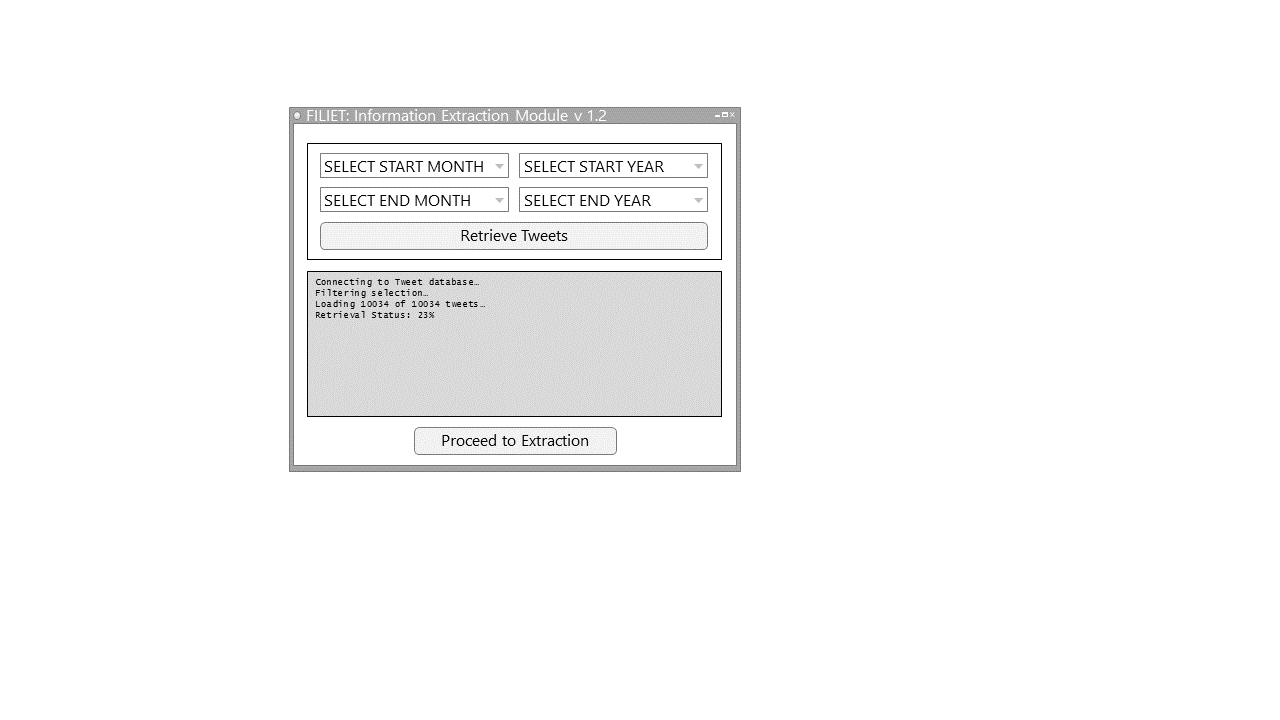


Figure 4‑3. Tweet Retrieval Screenshot

### Information Extraction

In this system function, the information extraction process starts with feature extraction which shall then be used for the classification of the tweets based on the categories defined in the system. After classification, the tweets shall then be examined for possible rules. The rules to be generated will then be applied to the tweets. Extracted information will be fed into the next function. Figure 4‑4 shows a screenshot of this function.

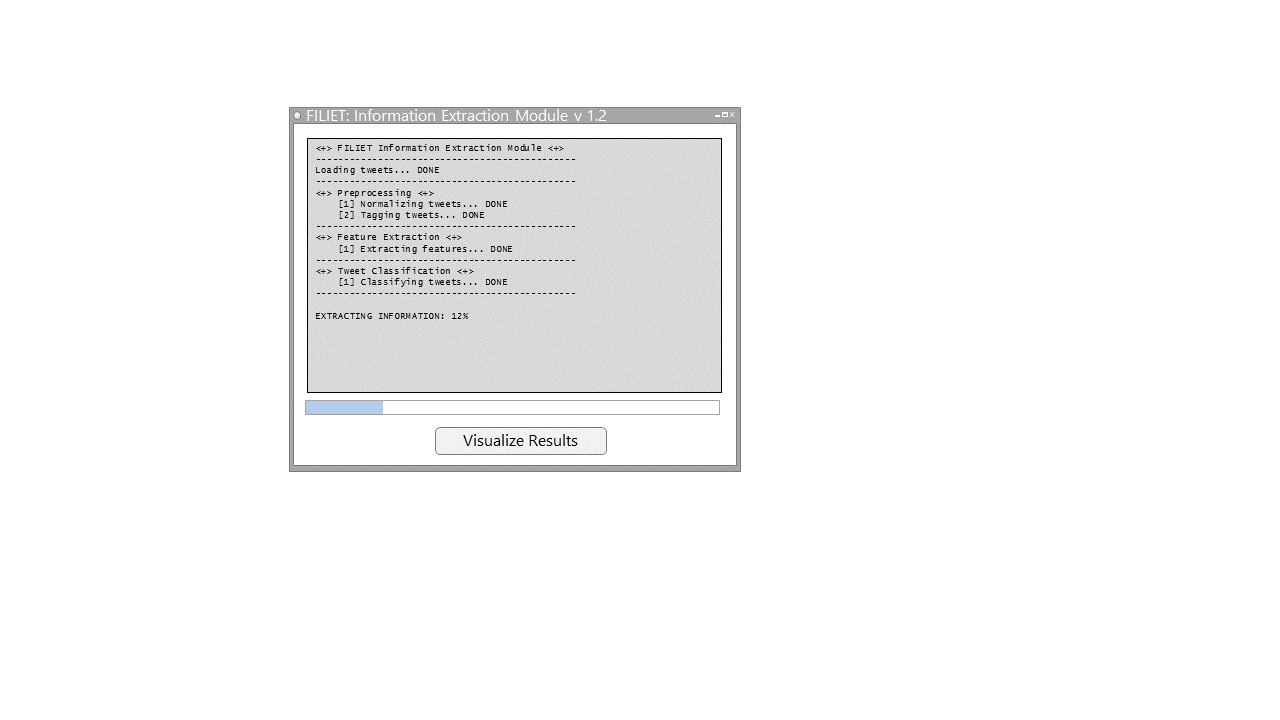


Figure 4‑4. Information Extraction Screenshot

### Ontology Population

In this system function, the extracted information will be initialized as entity instances for population of the ontology. The system, by default, automatically validates each of the extracted information before being introduced to the ontology. During validation, the system will check if the entity instances exist and if they do, the system will match the instances to their corresponding entity class/es. If they do not, the instances will immediately be discarded. Figure 4‑5 shows the screenshot of this function.

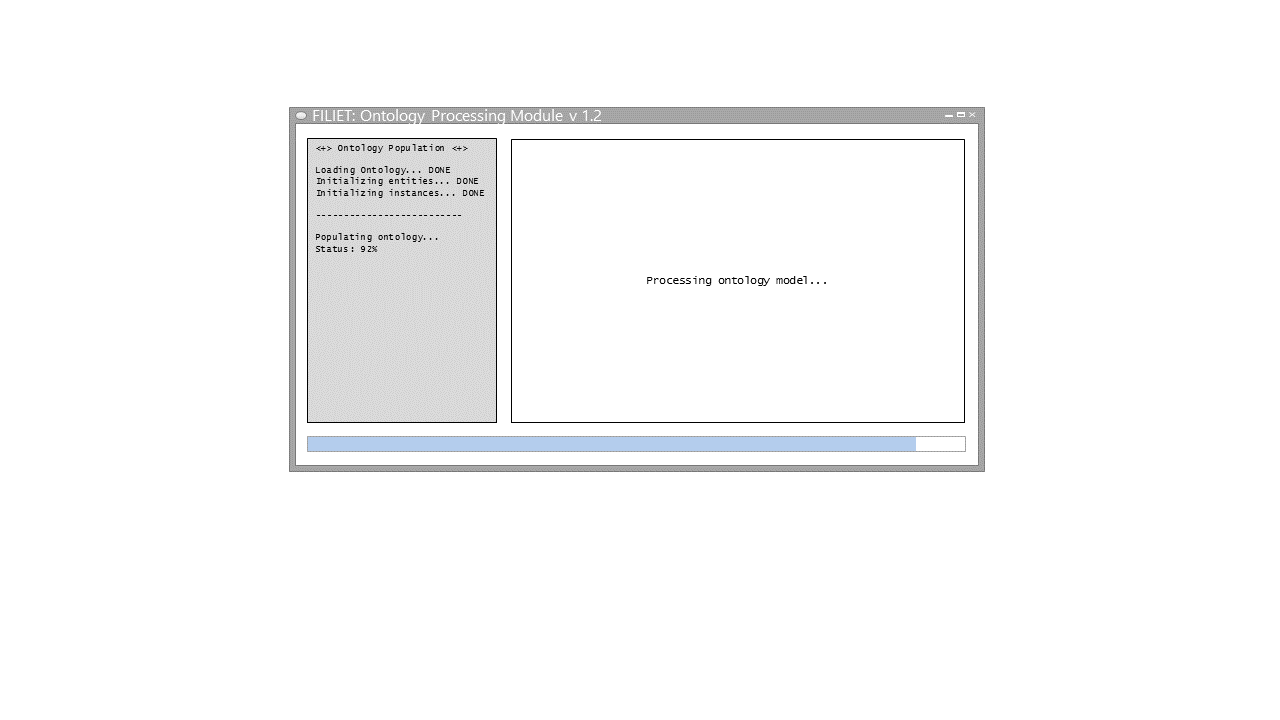


Figure 4‑5. Ontology Population Screenshot

### Ontology Access

In this system function, the populated ontology and the details of the instances per entity class can also be viewed. Relationships within entities can be seen or searched given a selected instance from the ontology model. Figure 4‑6 shows a screenshot of this function.

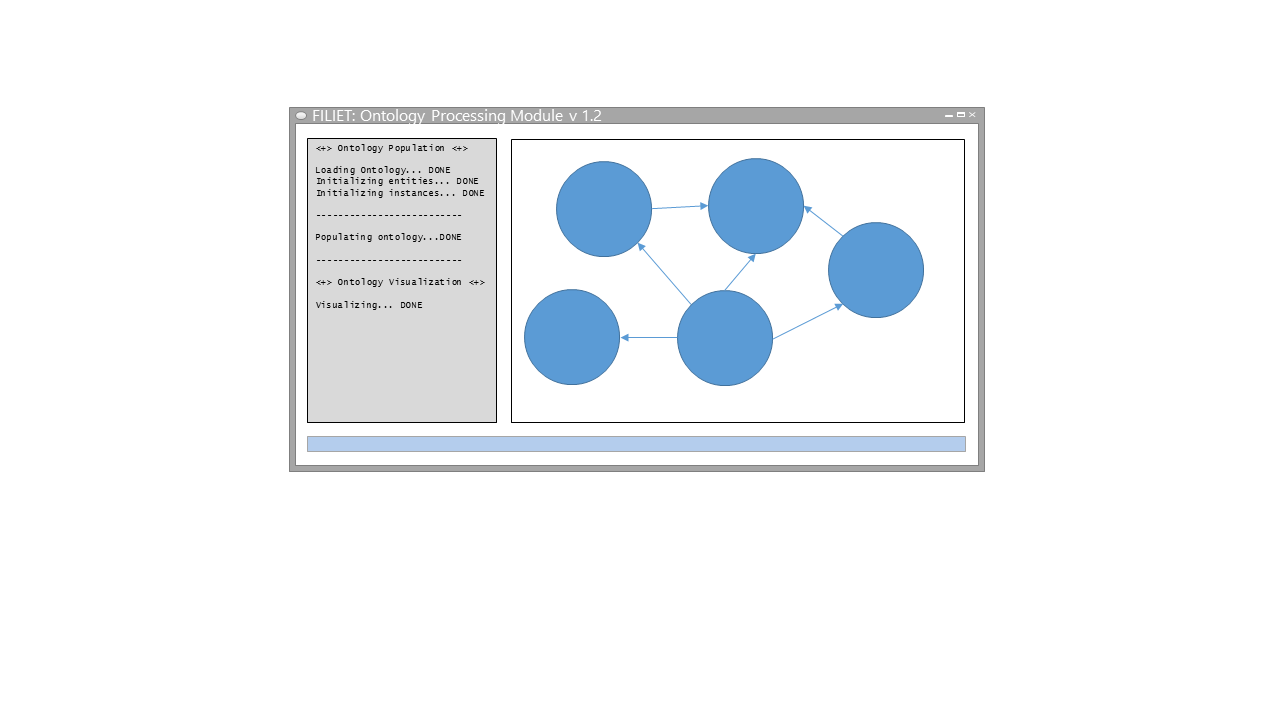


Figure 4‑6. Ontology Access Screenshot

## Physical Environment and Resources

This section outlines the minimum software and hardware requirements of the system.

### Minimum Software Requirements

* Windows 7
* MySQL
* Java 1.7.0

### Minimum Hardware Requirements

* 2 GB RAM
* Server

# References

alias-i. (2011). What is lingpipe?. Retrieved from http://alias-i.com/lingpipe/

Aone, C., Halverson, L., Hampton, T., & Ramos-Santacruz, M. (1998, April). SRA: Description of the IE2 system used for MUC-7. In *Proceedings of the seventh message understanding conference (MUC-7)*.

Apache Software Foundation. (2010). Welcome to apache opennlp. Retrieved from https://opennlp.apache.org/

Asahara, M., & Matsumoto, Y. (2003, May). Japanese named entity extraction with redundant morphological analysis. In Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1 (pp. 8-15). Association for Computational Linguistics.

Aw, A., Zhang, M., Xiao, J., & Su, J. (2006, July). A phrase-based statistical model for SMS text normalization. In Proceedings of the COLING/ACL on Main conference poster sessions (pp. 33-40). Association for Computational Linguistics.

Basili, R., Moschitti, A., Pazienza, M. T., & Zanzotto, F. M. (2003). Personalizing web publishing via information extraction. IEEE Intelligent Systems, 18(1), 62-70.

Bollen, J., Mao, H., & Zeng, X. (2011). Twitter mood predicts the stock market. Journal of Computational Science, 2(1), 1-8.

Bontcheva, K., Derczynski, L., Funk, A., Greenwood, M.A., Maynard, D., Aswani, N. TwitIE: An Open-Source Information Extraction Pipeline for Microblog Text. Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2013).

Bradlee, D., Knoll, S., & Pentheroudakis, J. (2001). Tokenizer for a natural language processing system.

Califf, Mary Elaine. 1998. Relational Learning Techniques for Natural Language Information Extraction. Ph.D. thesis, Univ. of Texas at Austin, http://www.cs.utexas.edu/users/mecaliff

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

Cheng, T. T., Cua, J. L., Tan, M. D., Yao, K. G., & Roxas, R. E. (2009, October). Information extraction from legal documents. In Natural Language Processing, 2009. NLP'09. Eighth International Symposium on (pp. 157-162). IEEE.

Chew, C., & Eysenbach, G. (2010). Pandemics in the age of Twitter: content analysis of Tweets during the 2009 H1N1 outbreak. PloS one, 5(11), e14118.

Choy, M., Cheong, M., Laik, M. N., & Shung, K. P. (2012). US Presidential Election 2012 Prediction using Census Corrected Twitter Model. *arXiv preprint arXiv:1211.0938*.

Ciravegna, F., & Lavelli, A. (2004). LearningPinocchio: Adaptive information extraction for real world applications. Natural Language Engineering, 10(02), 145-165.

Corney, D., Byrne, E., Buxton, B., & Jones, D. (2008). A logical framework for template creation and information extraction. In *Data Mining: Foundations and Practice* (pp. 79-108). Springer Berlin Heidelberg.

Culnan, M. J., McHugh, P. J., & Zubillaga, J. I. (2010). How large US companies can use Twitter and other social media to gain business value. MIS Quarterly Executive, 9(4), 243-259.

Cunningham, H. (2002). GATE, a general architecture for text engineering. Computers and the Humanities, 36(2), 223-254.

Cunningham, H., Maynard, D., Bontcheva, K., &Tablan, V. (2002, July). A framework and graphical development environment for robust NLP tools and applications. In ACL (pp. 168-175).

Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., Ursu, C., Dimitrov, M. & Aswani, N. (2002). Developing language processing components with GATE (a user guide). University of Sheffield, Sheffield UK, 5.

Davis, J., & Goadrich, M. (2006, June). The relationship between Precision-Recall and ROC curves. In Proceedings of the 23rd international conference on Machine learning (pp. 233-240). ACM.

Dimitrov, M. (2005). A Lightweight Approach to Conference Resolution for Named Entities in Text Marin Dimitrov, Kalina Bontcheva, Hamish Cunningham and Diana Maynard. Anaphora Processing: Linguistic, cognitive and computational modelling, 263, 97.

Dung, T. Q., & Kameyama, W. (2007, March). A proposal of ontology-based health care information extraction system: Vnhies. In Research, Innovation and Vision for the Future, 2007 IEEE International Conference on (pp. 1-7). IEEE.

Faria, Carla, and Rosario Girardi. "An Information Extraction Process for Semi-automatic Ontology Population." *Soft Computing Models in Industrial and Environmental Applications, 6th International Conference SOCO 2011*. Springer Berlin Heidelberg, 2011.

Farkas, R. (2009). Machine learning techniques for applied information extraction (Doctoral dissertation, University of Szeged).

Feilmayr, C. (2011, August). Text Mining-Supported Information Extraction: An Extended Methodology for Developing Information Extraction Systems. In Database and Expert Systems Applications (DEXA), 2011 22nd International Workshop on (pp. 217-221). IEEE.

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

Freitag, D., & Kushmerick, N. 2000. Boosted wrapper induction. In: Basili, R., Ciravegna, F., & Gaizauskas, R. (eds), ECAI2000 Workshop on Machine Learning for Information Extraction. http://www.dcs.shef.ac.uk/ fabio/ecai-workshop.html

Freitag, D., & McCallum, A. 1999. Information Extraction with HMMs and Shrinkage. In: AAAI-99 Workshop on Machine Learning for Information Extraction.

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

Ghedin, G. (2011, November 16). A social media lesson. from the Philippines. Retrieved from <http://www.youngdigitallab.com/en/social-media/a-social-media-lesson-from-the-philippines>

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.

Grishman, R. (1997). Information extraction: Techniques and challenges. In Information Extraction A Multidisciplinary Approach to an Emerging Information Technology (pp. 10-27). Springer Berlin Heidelberg.

Grosseck, G., & Holotescu, C. (2008, April). Can we use Twitter for educational activities. In 4th international scientific conference, eLearning and software for education, Bucharest, Romania.

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

Han, B., & Baldwin, T. (2011, June). Lexical normalisation of short text messages: Makn sens a# twitter. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1 (pp. 368-378). Association for Computational Linguistics.

Hawn, C. (2009). Take two aspirin and tweet me in the morning: how Twitter, Facebook, and other social media are reshaping health care. Health Affairs, 28(2), 361-368.

Howard, B. (2013) . 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/

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

Hripcsak, G., & Rothschild, A. S. (2005). Agreement, the f-measure, and reliability in information retrieval. Journal of the American Medical Informatics Association, 12(3), 296-298.

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.

Intarapaiboon, P., Nantajeewarawat, E., & Theeramunkong, T. (2009). Information extraction from Thai text with unknown phrase boundaries. In Advances in Knowledge Discovery and Data Mining (pp. 525-532). Springer Berlin Heidelberg.

Jansen, B. J., Zhang, M., Sobel, K., & Chowdury, A. (2009). Twitter power: Tweets as electronic word of mouth. Journal of the American Society for Information Science and Technology, 60(11), 2169-2188.

Junco, R., Heiberger, G., & Loken, E. (2011). The effect of Twitter on college student engagement and grades. Journal of Computer Assisted Learning, 27(2), 119-132.

King, D. (2005, April). Humanitarian knowledge management. In *Proceedings of the Second International ISCRAM Conference* (Vol. 1, pp. 1-6). Brussels.

Ko, H. (1998). Empirical assembly sequence planning: A multistrategy constructive learning approach. Machine Learning and Data Mining. John Wiley & Sons LTD.

Krishnamurthy, R., Li, Y., Raghavan, S., & Reiss, F. SystemT: A system for declarative information extraction. SIGMOD Record, 37. Retrieved May 28, 2014, from http://www.almaden.ibm.com/cs/projects/avatar/

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).

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.

Lim, N. R., New, J. C., Ngo, M. A., Sy, M., & Lim, N. R. (2007). A Named-Entity Recognizer for Filipino Texts. Proceedings of the 4th NNLPRS.

Loponen, A., & Järvelin, K. (2010). A dictionary-and corpus-independent statistical lemmatizer for information retrieval in low resource languages. In *Multilingual and Multimodal Information Access Evaluation* (pp. 3-14). Springer Berlin Heidelberg

Maedche, A., Neumann, G., & Staab, S. (2003). Bootstrapping an ontology-based information extraction system. In *Intelligent exploration of the web* (pp. 345-359). Physica-Verlag HD.

Malpica, A., Matisic, J. P., Niekirk, D. V., Crum, C. P., Staerkel, G. A., Yamal, J. M., ... & Follen, M. (2005). Kappa statistics to measure interrater and intrarater agreement for 1790 cervical biopsy specimens among twelve pathologists: qualitative histopathologic analysis and methodologic issues. Gynecologic Oncology, 99(3), S38-S52.

Manguilimotan, E., & Matsumoto, Y. (2009). Factors affecting part-of-speech tagging for Tagalog. In PACLIC (pp. 763-770).

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval* (Vol. 1, p. 6). Cambridge: Cambridge University Press.

Mocanu, D., Baronchelli, A., Perra, N., Gonçalves, B., Zhang, Q., & Vespignani, A. (2013). The Twitter of Babel: Mapping world languages through microblogging platforms. *PloS one*, *8*(4), e61981.

Maynard, D., Bontcheva, K., & Rout, D. (2012). Challenges in developing opinion mining tools for social media. Proceedings of@ NLP can u tag# user\_generated\_content.

Maynard, D., Peters, W., & Li, Y. (2006, May). Metrics for evaluation of ontology-based information extraction. In International world wide web conference.

McBride, B. (2002). Jena: A semantic web toolkit. IEEE Internet Computing, 6(6), 55-59.

McCallum, A. (2005). Information extraction: Distilling structured data from unstructured text. Queue, 3(9), 48-57.

Meier, P. (2013, September 18). [Web log message]. Retrieved from http://irevolution.net/2013/09/18/micromappers/

N´edellec C., Nazarenko A., & Bossy R. (2009). Information extraction. In S. Staab & R. Studer (Eds), Handbook on ontologies (pp 683-685). Dordrecht: Springer.

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

Neubig, G., Matsubayashi, Y., Hagiwara, M., & Murakami, K. (2011, November). Safety Information Mining-What can NLP do in a disaster-. In *IJCNLP* (pp. 965-973).

Official Gazette of the Republic of the Philippines. (2012, July 21). Prepare for natural calamities: Information and resources from the government. Retrieved July 15, 2014, from http://www.gov.ph/crisis-response/government-information-during-natural-disasters/

OpenNLP, A. (2011). Apache Software Foundation. URL http://opennlp. apache. org.

[Özsu, M. T., & Liu, L. (2009). Text Categorization. Encyclopedia of database systems (p. 3044). New York: Springer.](http://www.bibme.org/)

Pak, A., & Paroubek, P. (2010, May). Twitter as a Corpus for Sentiment Analysis and Opinion Mining. In LREC.

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.

Phelan, O., McCarthy, K., & Smyth, B. (2009, October). Using twitter to recommend real-time topical news. In Proceedings of the third ACM conference on Recommender systems (pp. 385-388). ACM.

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

Quinlan, J. R. (1990). Learning logical definitions from relations. Machine Learning, 5(3), 239-266.

Ritter, A., Clark, S., & Etzioni, O. (2011, July). Named entity recognition in tweets: an experimental study. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (pp. 1524-1534). Association for Computational Linguistics.

Sakaki, T., Okazaki, M., & Matsuo, Y. (2010, April). Earthquake shakes Twitter users: real-time event detection by social sensors. In Proceedings of the 19th international conference on World wide web (pp. 851-860). ACM.

Saloun, P., Velart, Z., & Klimanek, P. (2011, December). Semiautomatic domain model building from text-data. In *Semantic Media Adaptation and Personalization (SMAP), 2011 Sixth International Workshop on* (pp. 15-20). IEEE.

Shafiei, M., Wang, S., Zhang, R., Milios, E., Tang, B., Tougas, J., & Spiteri, R. (2007, April). Document representation and dimension reduction for text clustering. In Data Engineering Workshop, 2007 IEEE 23rd International Conference on (pp. 770-779). IEEE.

Shen, D. (2009). Text Categorization. Encyclopedia of Database Systems, 3041-3044.

Soderland, S. 1999. Learning Information Extraction Rules for Semi-structured and Free Text. Machine Learning, **34**(1), 233–272.

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

Sriram, B., Fuhry, D., Demir, E., Ferhatosmanoglu, H., & Demirbas, M. (2010, July). Short text classification in twitter to improve information filtering. InProceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval (pp. 841-842). ACM.

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/>

Stone, R. (2004). Natural language processing challenges and advantages for Philippine languages. Proceedings from 1st Natural Language Processing Research Symposium (pp.81-86).

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.

Tumasjan, A., Sprenger, T. O., Sandner, P. G., & Welpe, I. M. (2010). Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment. ICWSM, 10, 178-185.

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

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.

Wang, T., Bontcheva, K., Li, Y., & Cunningham, H. (2005). D2. 1.2/Ontology-Based Information Extraction (OBIE) v. 2. *EU-IST Project IST-2003-506826 SEKT SEKT: Semantically Enabled Knowledge Technologies*.

Wajeed, M. A., & Adilakshmi, T. (2011). Using KNN Algorithm for Text Categorization. In *Computational Intelligence and Information Technology* (pp. 796-801). Springer Berlin Heidelberg.

Wei, G., Gao, X., & Wu, S. (2010, July). Study of text classification methods for data sets with huge features. In Industrial and Information Systems (IIS), 2010 2nd International Conference on (Vol. 1, pp. 433-436). IEEE.

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

Xu, H., Stenner, S. P., Doan, S., Johnson, K. B., Waitman, L. R., & Denny, J. C. (2010). MedEx: a medication information extraction system for clinical narratives. Journal of the American Medical Informatics Association, 17(1), 19-24.

Zeng, Q. T., Goryachev, S., Weiss, S., Sordo, M., Murphy, S. N., & Lazarus, R. (2006). Extracting principal diagnosis, co-morbidity and smoking status for asthma research: evaluation of a natural language processing system. BMC Medical Informatics and Decision Making, 6(1), 30.

Zhou, G., & Su, J. (2002, July). Named entity recognition using an HMM-based chunk tagger. In proceedings of the 40th Annual Meeting on Association for Computational Linguistics (pp. 473-480). Association for Computational Linguistics.

Zhou, S., Ling, T. W., Guan, J., Hu, J., & Zhou, A. (2003, March). Fast text classification: a training-corpus pruning based approach. In Database Systems for Advanced Applications, 2003.(DASFAA 2003). Proceedings. Eighth International Conference on (pp. 127-136). IEEE.

# Appendix

## Appendix A

Table 6‑1. Results of the Study Conducted by University McCann

|  |  |  |
| --- | --- | --- |
| **Category** | **Rank** | **Margin from Rank 1** |
| Blog Readership | 2 (90.3%) | 1.8% (South Korea) |
| Starting a Blog | 4 (65.8%) | 5.9% (South Korea) |
| Social Networks | 1 (83.1%) | -- |
| Photo Sharing | 1 (86.4%) | -- |
| Uploading Videos | 2 (60.5%) | 7.8% (China) |
| Watching Videos | 1 (98.6%) | -- |
| Podcasts | 5 (61.8%) | 12.5% (China) |
| RSS | 6 (45.2%) | 11.4% (Russia) |

## Appendix B

Table 6‑2. Examples of Filipino Morphemes

|  |  |  |  |
| --- | --- | --- | --- |
| **Morpheme Element** | **Root Word** | **Suffix** | **Filipino Word** |
| Elision | bigay | na- ; -an | nabigyan |
| Epenthesis | patay | -an | patayan |
| Metathesis | peteh (cebuano) | -en | pehten |
| Replacement | utos | -an | utusan |
| Nasal Assimilation | bigay | paN- | pamigay |
| Infixation | kain | -um- | kumain |
| Reduplication | matamis | - | matamistamis |

## Ontology

|  |
| --- |
| Code Listing: |
| <?xml version="1.0"?>  <!DOCTYPE Ontology [  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >  <!ENTITY xml "http://www.w3.org/XML/1998/namespace" >  <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >  ]>  <Ontology xmlns="http://www.w3.org/2002/07/owl#"  xml:base="http://www.semanticweb.org/vilson/ontologies/2014/7/disaster-relief-ontology"  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"  xmlns:xml="http://www.w3.org/XML/1998/namespace"  ontologyIRI="http://www.semanticweb.org/vilson/ontologies/2014/7/disaster-relief-ontology">  <Prefix name="" IRI="http://www.w3.org/2002/07/owl#"/>  <Prefix name="owl" IRI="http://www.w3.org/2002/07/owl#"/>  <Prefix name="rdf" IRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#"/>  <Prefix name="xsd" IRI="http://www.w3.org/2001/XMLSchema#"/>  <Prefix name="rdfs" IRI="http://www.w3.org/2000/01/rdf-schema#"/>  <Declaration>  <Class IRI="#Clothes"/>  </Declaration>  <Declaration>  <Class IRI="#Disaster"/>  </Declaration>  <Declaration>  <Class IRI="#Electricity"/>  </Declaration>  <Declaration>  <Class IRI="#Food"/>  </Declaration>  <Declaration>  <Class IRI="#Location"/>  </Declaration>  <Declaration>  <Class IRI="#Money"/>  </Declaration>  <Declaration>  <Class IRI="#Rescue"/>  </Declaration>  <Declaration>  <Class IRI="#Shelter"/>  </Declaration>  <Declaration>  <Class IRI="#Time"/>  </Declaration>  <Declaration>  <Class IRI="#Victim"/>  </Declaration>  <Declaration>  <Class IRI="#Volunteer"/>  </Declaration>  <Declaration>  <Class IRI="#Water"/>  </Declaration>  <Declaration>  <ObjectProperty IRI="#donate"/>  </Declaration>  <Declaration>  <ObjectProperty IRI="#has"/>  </Declaration>  <Declaration>  <ObjectProperty IRI="#is\_at"/>  </Declaration>  <Declaration>  <ObjectProperty IRI="#need"/>  </Declaration>  <Declaration>  <ObjectProperty IRI="#occured"/>  </Declaration>  <Declaration>  <ObjectProperty IRI="#offer"/>  </Declaration>  <Declaration>  <ObjectProperty IRI="#volunteer"/>  </Declaration>  <Declaration>  <DataProperty IRI="#donate"/>  </Declaration>  <Declaration>  <DataProperty IRI="#is\_located"/>  </Declaration>  <Declaration>  <DataProperty IRI="#need"/>  </Declaration>  <Declaration>  <DataProperty IRI="#occured"/>  </Declaration>  <Declaration>  <DataProperty IRI="#volunteer"/>  </Declaration>  <SubObjectPropertyOf>  <ObjectProperty IRI="#donate"/>  <ObjectProperty abbreviatedIRI="owl:topObjectProperty"/>  </SubObjectPropertyOf>  <SubObjectPropertyOf>  <ObjectProperty IRI="#has"/>  <ObjectProperty abbreviatedIRI="owl:topObjectProperty"/>  </SubObjectPropertyOf>  <SubObjectPropertyOf>  <ObjectProperty IRI="#is\_at"/>  <ObjectProperty abbreviatedIRI="owl:topObjectProperty"/>  </SubObjectPropertyOf>  <SubObjectPropertyOf>  <ObjectProperty IRI="#occured"/>  <ObjectProperty abbreviatedIRI="owl:topObjectProperty"/>  </SubObjectPropertyOf>  <ObjectPropertyDomain>  <ObjectProperty IRI="#donate"/>  <Class IRI="#Volunteer"/>  </ObjectPropertyDomain>  <ObjectPropertyDomain>  <ObjectProperty IRI="#has"/>  <Class IRI="#Disaster"/>  </ObjectPropertyDomain>  <ObjectPropertyDomain>  <ObjectProperty IRI="#is\_at"/>  <Class IRI="#Disaster"/>  </ObjectPropertyDomain>  <ObjectPropertyDomain>  <ObjectProperty IRI="#need"/>  <Class IRI="#Victim"/>  </ObjectPropertyDomain>  <ObjectPropertyDomain>  <ObjectProperty IRI="#occured"/>  <Class IRI="#Disaster"/>  </ObjectPropertyDomain>  <ObjectPropertyDomain>  <ObjectProperty IRI="#offer"/>  <Class IRI="#Volunteer"/>  </ObjectPropertyDomain>  <ObjectPropertyDomain>  <ObjectProperty IRI="#volunteer"/>  <Class IRI="#Volunteer"/>  </ObjectPropertyDomain>  <ObjectPropertyDomain>  <ObjectProperty abbreviatedIRI="owl:topObjectProperty"/>  <Class IRI="#Volunteer"/>  </ObjectPropertyDomain>  <ObjectPropertyRange>  <ObjectProperty IRI="#donate"/>  <Class IRI="#Clothes"/>  </ObjectPropertyRange>  <ObjectPropertyRange>  <ObjectProperty IRI="#donate"/>  <Class IRI="#Food"/>  </ObjectPropertyRange>  <ObjectPropertyRange>  <ObjectProperty IRI="#donate"/>  <Class IRI="#Money"/>  </ObjectPropertyRange>  <ObjectPropertyRange>  <ObjectProperty IRI="#donate"/>  <Class IRI="#Water"/>  </ObjectPropertyRange>  <ObjectPropertyRange>  <ObjectProperty IRI="#has"/>  <Class IRI="#Victim"/>  </ObjectPropertyRange>  <ObjectPropertyRange>  <ObjectProperty IRI="#is\_at"/>  <Class IRI="#Location"/>  </ObjectPropertyRange>  <ObjectPropertyRange>  <ObjectProperty IRI="#need"/>  <Class IRI="#Clothes"/>  </ObjectPropertyRange>  <ObjectPropertyRange>  <ObjectProperty IRI="#need"/>  <Class IRI="#Electricity"/>  </ObjectPropertyRange>  <ObjectPropertyRange>  <ObjectProperty IRI="#need"/>  <Class IRI="#Food"/>  </ObjectPropertyRange>  <ObjectPropertyRange>  <ObjectProperty IRI="#need"/>  <Class IRI="#Money"/>  </ObjectPropertyRange>  <ObjectPropertyRange>  <ObjectProperty IRI="#need"/>  <Class IRI="#Rescue"/>  </ObjectPropertyRange>  <ObjectPropertyRange>  <ObjectProperty IRI="#need"/>  <Class IRI="#Shelter"/>  </ObjectPropertyRange>  <ObjectPropertyRange>  <ObjectProperty IRI="#need"/>  <Class IRI="#Water"/>  </ObjectPropertyRange>  <ObjectPropertyRange>  <ObjectProperty IRI="#occured"/>  <Class IRI="#Time"/>  </ObjectPropertyRange>  <ObjectPropertyRange>  <ObjectProperty IRI="#offer"/>  <Class IRI="#Shelter"/>  </ObjectPropertyRange>  <ObjectPropertyRange>  <ObjectProperty IRI="#volunteer"/>  <Class IRI="#Rescue"/>  </ObjectPropertyRange>  <ObjectPropertyRange>  <ObjectProperty abbreviatedIRI="owl:topObjectProperty"/>  <Class IRI="#Rescue"/>  </ObjectPropertyRange>  <SubDataPropertyOf>  <DataProperty IRI="#donate"/>  <DataProperty abbreviatedIRI="owl:topDataProperty"/>  </SubDataPropertyOf>  <SubDataPropertyOf>  <DataProperty IRI="#occured"/>  <DataProperty abbreviatedIRI="owl:topDataProperty"/>  </SubDataPropertyOf>  <SubDataPropertyOf>  <DataProperty IRI="#volunteer"/>  <DataProperty abbreviatedIRI="owl:topDataProperty"/>  </SubDataPropertyOf>  </Ontology>  <!-- Generated by the OWL API (version 3.5.0) http://owlapi.sourceforge.net --> |

Code Listing 6‑1. Representation of Ontology in OWL Format

## Resource Person

**Mr. Ralph Vincent J. Regalado**

**Thesis Adviser, Faculty Member**

**College of Computer Studies**

**De La Salle University**

**rv.regalado@gmail.com**

## ****Personal Vitae****

**Mr. Kyle Mc Hale B. Dela Cruz**

Blk 5, Lot 2A, Martires St., Brgy. Martires del 96, Pateros, Metro Manila

(0917) 880-5019

kylemchale\_delacruz@yahoo.com

**Mr. John Paul F. Garcia**

36 Bohol St., Ayala Alabang Village, Muntinlupa City

(0927) 886-6999

johnpaulgarcia1208@gmail.com

**Ms. Kristine Ma. Dominique F. Kalaw**

28 New Years Avenue, GSIS Holiday Hills Village, San Pedro, Laguna

(0927) 854-4201

tintin.kalaw@gmail.com

**Mr. Vilson E. Lu**

739 – D A. Bonifacio St. Balintawak, Quezon City

(0917) 631-1374

vilson.espayos.lu@gmail.com

1. MicroMappers digital disaster response system. http://micromappers.com/ [↑](#footnote-ref-2)
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-3)
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-4)
4. Twitter, a microblogging social media platform. http://www.twitter.com/ [↑](#footnote-ref-5)