APPLICATION OF LINK GRAMMAR IN SEMI-SUPERVISED NAMED ENTITY RECOGNITION FOR ACCIDENT DOMAIN

YUNITA SARI

MASTER OF SCIENCE
COMPUTER AND INFORMATION SCIENCES DEPARTMENT

UNIVERSITI TEKNOLOGI PETRONAS

APRIL 2011

STATUS OF THESIS

Title of thesis | APPLICATION OF LINK GRAMMAR IN SEMI-SUPERVISED NAMED ENTITY RECOGNITION FOR ACCIDENT DOMAIN

I YUNITA SARI

hereby allow my thesis to be placed at the Information Resource Center (IRC) of Universiti Teknologi PETRONAS (UTP) with the following conditions:

1. The thesis becomes the property of UTP
2. The IRC of UTP may make copies of the thesis for academic purposes only.
3. This thesis is classified as

☐ Confidential

☐ Non-confidential

If this thesis is confidential, please state the reason:

_____

_____

The contents of the thesis will remain confidential for _____ years.

Remarks on disclosure:

_____

_____

Endorsed by

_____ _____

Signature of Author | Signature of Supervisor

Permanent address:
Jln. Krida Mulya No.8
Purbalingga Jawa Tengah
53319 Indonesia

Name of Supervisor
DR. MOHD FADZIL HASSAN

Date : _____ Date : _____

UNIVERSITI TEKNOLOGI PETRONAS

APPLICATION OF LINK GRAMMAR IN SEMI-SUPERVISED NAMED ENTITY RECOGNITION FOR ACCIDENT DOMAIN

by

YUNITA SARI

The undersigned certify that they have read, and recommend to the Postgraduate Studies Programme for acceptance this thesis for the fulfilment of the requirements for the degree stated.

Signature: _____

Main Supervisor: DR. MOHD FADZIL HASSAN

Signature: _____

Co-Supervisor: NORSHUHANI ZAMIN

Signature: _____

Head of Department: DR. MOHD FADZIL HASSAN

Date: _____

APPLICATION OF LINK GRAMMAR IN SEMI-SUPERVISED NAMED ENTITY RECOGNITION FOR ACCIDENT DOMAIN

by

YUNITA SARI

A Thesis

Submitted to the Postgraduate Studies Programme

as a Requirement for the Degree of

MASTER OF SCIENCE

COMPUTER AND INFORMATION SCIENCES DEPARTMENT

UNIVERSITI TEKNOLOGI PETRONAS

BANDAR SERI ISKANDAR,

PERAK

APRIL 2011

DECLARATION OF THESIS

Title of thesis | APPLICATION OF LINK GRAMMAR IN SEMI-SUPERVISED NAMED ENTITY RECOGNITION FOR ACCIDENT DOMAIN

I YUNITA SARI

hereby declare that the thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at UTP or other institutions.

Witnessed by

_____ _____

Signature of Author | Signature of Supervisor

Permanent address:
Jln. Krida Mulya No.8
Purbalingga Jawa Tengah
53319 Indonesia

DR. MOHD FADZIL HASSAN

Date : _____ Date : _____

ACKNOWLEDGEMENT

This work would not have been possible without the guidance and the help of several individuals who in one way or another contributed and extended their valuable assistance in the preparation and completion of this study.

First and foremost, I would like to thank Allah the Almighty for the blessings, guidance and answering my prayers. My special gratitude is due to my lovely parents for their encouragement and understanding.

I am heartily thankful to my supervisor, Dr. Mohd Fadzil Hassan and my co-supervisor, Ms. Norshuhani Zamin, whose encouragement, supervision and support from the preliminary to the concluding level enabled me to develop an understanding of the subject. One simply could not wish for better or friendlier supervisors.

My sincere thanks are due to Assoc. Prof. Dr. Nursuriati Jamil from Universiti Teknologi MARA (UiTM), Dr. Rohiza Ahmad and Assoc. Prof. Dr. Abas M Said for the detailed and constructive comments that have been very helpful for this study.

I would also like to thank En. Wan Tarmizi B Wan Ismail and En. Mohd Rizal B Abd Wahab from Health, Safety and Environment (HSE) Department of UTP for helping me on the data collection. And last but not least, I offer my sincere thanks to my colleagues, especially postgraduate fellows in Computer and Information Sciences Department and all Indonesian students in UTP for providing a stimulating and fun environment in which to learn and grow.

The financial support of the Universiti Teknologi PETRONAS is gratefully acknowledged.

ABSTRACT

Accident document typically contains some crucial information that might be useful for analysis process for future accident investigation i.e. date and time when the accident happened, location where the accident occurred and also the person involved in the accident. This document is largely available in free text; it can be in the form of news wire articles or accident reports. Although it is possible to identify the information manually, due to the high volumes of data involved, this task can be time consuming and prone to error. Information Extraction (IE) has been identified as a potential solution to this problem. IE has the ability to extract crucial information from unstructured texts and convert them into a more structured representation. This research is attempted to explore Name Entity Recognition (NER), one of the important tasks in IE research aimed to identify and classify entities in the text documents into some predefined categories. Numerous related research works on IE and NER have been published and commercialized. However, to the best of our knowledge, there exists only a handful of IE research works that are really focused on accident domain. In addition, none of these works have attempted to either explore or focus on NER, which becomes the main motivation for this research. The work presented in this thesis proposed an NER approach for accident documents that applies syntactical and word features in combination with Self-Training algorithm. In order to satisfy the research objectives, this thesis comes with three main contributions.

The first contribution is the identification of the entity boundary. Entity segmentation or identification of entity boundary is required since named entity may consist of one or more words. We adopted Stanford Part-of-Speech (POS) tagger for the word POS tag and connectors from the Link Grammar (LG) parser to determine the starting and stopping word. The second contribution is the extraction pattern construction. Each named entity candidate will be assigned with an extraction pattern constructed from a set of word and syntactical feature. Current NER system used

restricted syntactical features which are associated with a number of limitations. It is therefore a great challenge to propose a new NER approach using syntactical features that could capture all syntactical structure in a sentence. For the third contribution, we have applied the Self-Training algorithm which is one of the semi-supervised machines learning technique. The algorithm is utilized for predicting a huge set of unlabeled data, given a small number of labelled data. In our research, extraction pattern from the first module will be fed to this algorithm and is used to make the prediction of named entity candidate category. The Self-Training algorithm greatly benefits semi-supervised learning which allows classification of entities given only a small-size of labelled data. The algorithm reduces the training efforts and generates almost similar result as compared to the conventional supervised learning technique.

The proposed system was tested on 100 accident news from Reuters to recognize three different named entities: date, person and location which are universally accepted categories in most NER applications. Exact Match evaluation method which consists of three evaluation metrics; precision, recall and F-measure is used to measure the proposed system performance against three existing NER systems. The proposed system has successfully outperforms one of those systems with an overall F-measure of approximately 9% but in the other hand it shows a slight decrease as compared to other two systems identified in our benchmarking. However, we believe that this difference is due to the different nature and techniques used in the three systems. We consider our semi-supervised approach as a promising method even though only two features are utilized: syntactical and word features. Further manual inspection during the experiments suggested that by using complete word and syntactical features or combination of these features with other features such as the semantic feature, would yield an improved result.

ABSTRAK

Pada kebiasaannya, dokumen kemalangan mengandungi beberapa maklumat penting yang mungkin berguna bagi proses analisis untuk siasatan lanjut seperti tarikh dan waktu ketika kemalangan itu berlaku, lokasi kemalangan dan juga orang yang terlibat dalam kemalangan tersebut. Dokumen ini sebahagian besarnya terdapat dalam bentuk teks bebas; sama ada dalam bentuk laporan akhbar atau laporan kemalangan. Walaupun maklumat dalam dokumen tersebut boleh dikenalpasti secara manual, namun kandungan maklumat yang terlalu banyak akan memakan masa untuk diteliti dan berkemungkinan terdedah kepada kesilapan. *Information Extraction* (IE) telah dikenalpasti sebagai langkah yang berpotensi untuk menyelesaikan masalah ini. IE mempunyai kemampuan untuk mengekstrak maklumat penting dari teks tidak berstruktur dan mengubahnya kepada bentuk yang lebih berstruktur. Penyelidikan ini berusaha untuk mendalami *Named Entity Recognition* (NER) yang merupakan peranan penting bagi IE. NER berfungsi untuk mengenal pasti dan mengklasifikasikan entiti dalam dokumen teks ke dalam beberapa kategori yang telah ditetapkan. Terdapat banyak penyelidikan yang berkaitan dengan IE dan NER telah diterbitkan dan dikomersilkan. Walau bagaimanapun, dalam pengetahuan kami, hanya terdapat beberapa penyelidikan dalam konteks IE yang benar-benar bertumpu pada domain kemalangan. Selain itu, masih belum ada penyelidikan lain yang mendalami atau fokus pada NER. Inilah yang menjadi motivasi utama untuk penyelidikan ini. Hasil penyelidikan yang dibentangkan dalam tesis ini mencadangkan pendekatan NER untuk dokumen-dokumen kemalangan yang mengaplikasikan ciri-ciri sintaksis dan kata dikombinasikan dengan algoritma *Self-Training*. Dalam rangka memenuhi matlamat kajian, tesis ini dilengkapi dengan tiga sumbangan utama.

Sumbangan pertama adalah pengenalan batas entiti. Segmentasi entiti atau pengenalan batas entiti diperlukan kerana *named entity* boleh terdiri daripada satu atau lebih kata. Kami telah menggunakan *Stanford Part-of-Speech (POS) Tagger* untuk kata tag POS dan penyambung dari parser *Link Grammar* (LG) bagi menentukan kata mula dan kata berhenti. Sumbangan kedua adalah pembangunan *pattern extraction*. Setiap calon *named entity* akan disesuaikan dengan *pattern extraction* yang dibina dari satu set kata dan ciri-ciri sintaksis. Sistem NER terkini menggunakan ciri-ciri sintaksis terhad

yang dikaitkan dengan beberapa keterbatasan. Adalah menjadi satu cabaran besar untuk mencadangkan pendekatan NER baru menggunakan ciri-ciri sintaksis yang dapat mengekstrak semua struktur sintaksis dalam satu ayat. Untuk sumbangan ketiga, kami mengaplikasikan algoritma *Self-Training* yang merupakan salah satu teknik *semi-supervised machines learning*. Algoritma ini digunakan untuk meramal satu set data tidak berlabel dalam kuantiti yang banyak dengan diberikan sejumlah kecil data berlabel. Dalam penyelidikan kami, *extraction pattern* dari modul pertama akan diberikan kepada algoritma ini dan digunakan untuk membuat ramalan kategori bagi calon *named entity*. Algoritma *Self-Training* sangat bermanfaat kepada *semi-supervised learning* yang membolehkan klasifikasi entiti dengan diberikan hanya data berlabel dalam skala kecil. Algoritma tersebut mengurangkan usaha latihan dan menghasilkan keputusan yang hampir sama dengan teknik *supervised learning* konvensional.

Sistem yang dicadangkan telah diuji pada 100 berita kemalangan dari *Reuters* untuk mengenalpasti tiga entiti nama yang berbeza: tarikh, orang dan lokasi yang diterima secara universal dalam kebanyakan aplikasi NER. Kaedah penilaian *Exact Match* yang terdiri daripada tiga metrik penilaian; *precision*, *recall* dan *F-measure* digunakan untuk mengukur prestasi sistem yang dicadangkan terhadap tiga sistem *NER*. Keputusan eksperimen menunjukkan nilai *F-measure* keseluruhan adalah lebih kurang 9% melebihi prestasi salah satu sistem tetapi terdapat sedikit penurunan jika dibandingkan dengan dua sistem lain yang telah dikenalpasti sebagai penanda aras kami. Namun, kami percaya bahawa perbezaan ini disebabkan oleh sifat dan teknik yang berbeza digunakan dalam ketiga-tiga sistem ini. Kami menganggap pendekatan *semi-supervised* sebagai kaedah yang menjanjikan meskipun hanya dua ciri sahaja yang digunakan: ciri-ciri sintaksis dan kata. Pemeriksaan lanjut secara manual sepanjang eksperimen menyarankan bahawa dengan menggunakan kata yang lengkap dan ciri-ciri sintaksis atau kombinasi ciri-ciri ini dengan ciri-ciri lain seperti semantik, akan menghasilkan keputusan yang lebih baik.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Detailed investigation of accident occurrences requires comprehensive capturing of essential information from accident news or reports. The process usually begins with the identification of crucial facts related to the incident itself i.e. the date and time when the accident happened, the location where the accident occurred and also the person involved in the accident. Although the process can be done manually, but the huge number of accidents and documents could turn this into a painstaking task. Data from National Transportation Safety Board (NTSB) an independent U.S. Federal agency that focuses on transportation accident investigation shows from January 2008 to January 2010 there were 3703 occurrences of aviation accident [1]. Thus, an automatic facts extraction is needed. Information Extraction (IE) is a perfect solution to carry out this task. IE offers an ability to extract important facts from unstructured text document which could be used to populate database entries for further analysis purpose.

Basically, there are two main processes in IE [2]. The first process is local text analysis, in which all individual facts are extracted from text document. The second process is discourse analysis in which all facts will be integrated and translated into a standard template. The increased importance of IE has led it to become the main topic of interest during the sixth and seventh Message Understanding Conference (MUC-6 and MUC-7). Both of these conferences were funded by the US Defense Advance Research Project Agency (DARPA) whose main intention is to evaluate IE system. MUC-7 which is the last series of MUCs, split IE into 6 tasks : Named Entity Task (NE) or Named Entity Recognition (NER), Multi-lingual Entity Task (MET), Template Element Task (TE), Template Relation Task (TR), Scenario Template Task (ST) and Co-reference Task (CO) [3].

This thesis focuses on one of IE subtask, namely Named Entity Recognition (NER). NER is a phase in IE which has a part to identify and classify entities in the

text document into predefined categories. A short introduction to NER system including techniques, features and extended applications of NER are presented in the next subsection. This is followed by a discussion on the problem statement, objective of this research, contributions and scope of this work. An outline of the thesis is provided at the end of this chapter.

1.1 Named Entity Recognition (NER)

R. Grishman in [2] defined the structure of IE as illustrated in Figure 1.1:

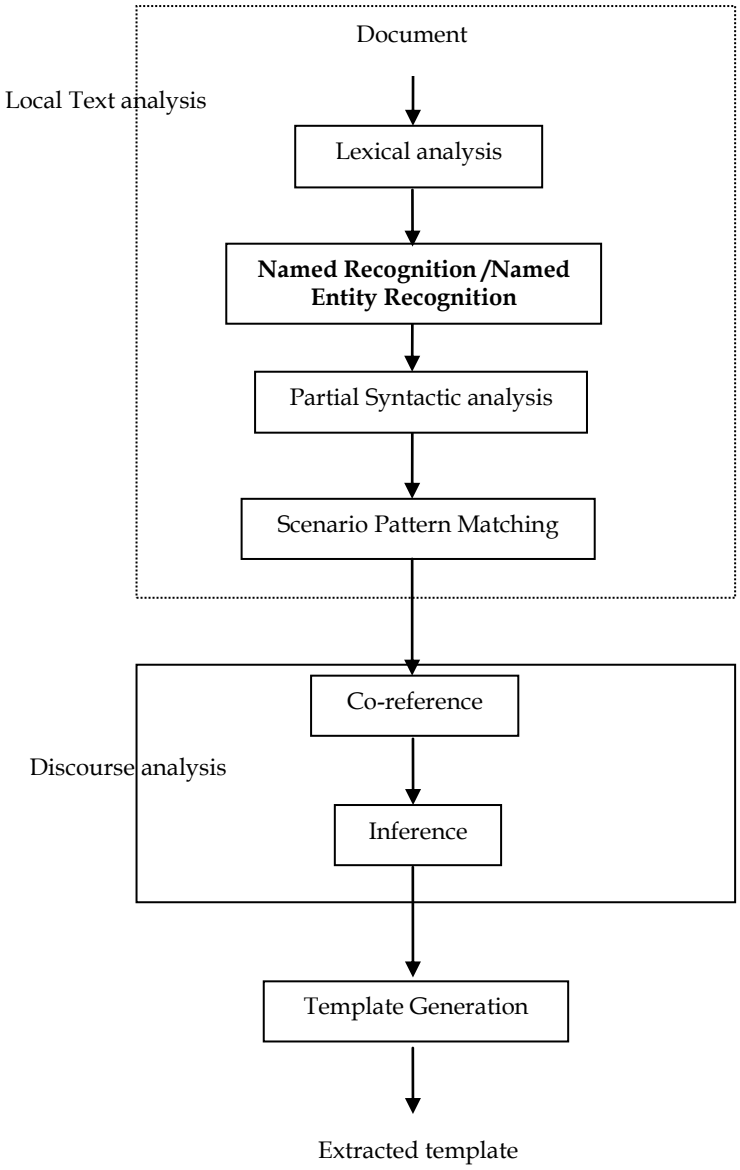

Figure 1.1 Structure of IE System

One of the most important subtasks in IE is the Named Entity Recognition (NER). NER is defined as a process of identifying proper names and other special forms. A set of features including part-of-speech, syntactic features and orthographic features are used in the identification process. MUC-6 has divided named entity task into three different parts; 1) Entity names (ENAMEX tag element) including name of ORGANIZATION, PERSON and LOCATION, 2) Temporal expression (TIMEX tag element) consists of DATE and TIME, 3) Number expression (NUMEX tag element) which covers MONETARY EXPRESSION and PERCENTAGE [4].

However, these 7 predefined categories were not enough to address new IE domain. As such, 150 categories of Extended NE were later proposed to cover those needs [5]. Rapid development in the Natural Language Processing (NLP) and text mining areas has made NER not only used in IE but it has also become a very essential component in many NLP and text mining tasks. NER is also applied in the following areas:

- Question Answering (QA)

  NER is often combined with Information Retrieval (IR) in QA application. Based on the question, IR will find the relevant document and then NER try to recognize named entity and provide it as the answer.
- Document Clustering

  Most of traditional document clustering techniques relied on word based or term based document representation. Recently, document clustering research has also included named entity as one of the features [6].
- Event Extraction

  A relation of a set of named entity in the text document can be used to identify an event. It was explained in [7] that to find relations and events entities, it needs to find the participants and modifiers (i.e. date, time, location, etc.).

Basically, there are three most important factors that influence the performance of NER system: 1) technique, 2) feature and 3) domain. IE and NER technique can be divided into two general categories: 1) knowledge engineering approach and 2) automatic training approach [8]. As the name implies, knowledge engineering approach used manually created pattern by the knowledge experts while automatic training approach tries to replace the manual process by utilizing some statistical methods. However, both approaches have their strengths and weaknesses on different conditions, thus it is important to know the most appropriate technique to use.

In order to be able to perform the recognition process, a set of features are required to be fed into a technique or algorithm. A thesis in [9] has classified NER features into three different types: 1) Word-Level-Features 2) List Look-Up Features 3) Document and Corpus Features. Word case, punctuation, digit, part-of-speech, morphology are considered as Word-Level-Features. Those features are related to the

attribute of a word itself. The second feature-List Look-Up Features, is also known as gazetteer or dictionary. It can be a general list, a list of entities or a list of entity cues.

The third feature-Document and Corpus Features describes document content and structure. Multiple occurrences, local syntax, meta-information and corpus frequency are examples of Document and Corpus Features. In [10] which captures the result of all participants on Conference on Computational Natural Language Learning-2003 (CoNLL-2003) mentioned that the choice of feature is as important as the choice of technique. However, the result showed that the usage of large number of features could not guarantee an improvement in the system performance.

With regards to the domain, different domain will have different patterns of textual structures. In this case, a set of specific pattern rules constructed from a set of features are needed based on the nature of the domain itself. For instance, date identification used in accident report and date used in e-mail text. Though both of them have the same goal, but the analysis to create an extraction pattern is different.

Accident documents (i.e. accident news and accident reports) consist of different structure with those documents. The nature of accident document is that it always describes the chronology either using direct or indirect sentences. Typically it is started with the date and time when the accident happened followed by the location of the accident. Moreover, some additional information like the person involved, the number of victim, the cause and effect of the accident and the past accident happened are also described in the document.

1.2 Problem Statement

Accident documents including accident news and accident reports contain crucial information that is useful for future investigation and analysis. For instance, NTSB gives very detail and comprehensive information in its accident reports. For instance, given an aircraft accident report, all information starting from the basic information e.g. location of the accident, date and time of the accident, phase of operation to more specific information like weather information at the accident site, pilot, flight crew and passenger information and narrative history of flight are provided [11].

Occupational Safety & Health Administration (OSHA) is another U.S. agency which focuses on safety and health in workplace. This agency also has its own standard on creating fatalities report. For example, to report fatalities which involve multiple hospitalizations, at least seven types of information must be provided. The information includes the establishment name; the location and time of the incident; the number of fatalities or hospitalized employees; the names of any injured employees; contact person; and a brief description of the incident [12].

Accident reports also contain almost similar information, with a slight difference in the document format. In addition, the report usually appears in a more structured way for official use. Typically, each company or agency has its own reporting format. On the other side, accident news is usually represented more informally and often highlight on specific information to attract the readers. However, both types of the accident documents are normally represented using similar patterns of sentences.

The task to simplify the analysis process requires those unstructured documents to be converted into more structured and comprehensive representational means. Though, the process can be done manually but certainly this will be a cumbersome task when it involves huge volume of documents. IE offers a solution for this problem by automatically extracting only the crucial information and represents it into a more structured form. NER as one of IE tasks is a perfect tool to acquire the information by identifying important named entities on those documents. Additionally, those identified named entities can be further applied in other text mining applications as discussed previously. There are a lot of research works found on IE and NER. However, to our best knowledge, there are only a few IE research works focused on this domain [13-15]. In addition, none of the work attempted to either explore or focus on NER which creates the motivation for this research.

In the introduction, it was explained that there are three different types of NER features; word features, list look-up features and document features. Given the fact that a complete dictionary is difficult to obtain has made list look-up features not a feasible option. Moreover, a complete and comprehensive lists are difficult to be constructed [16].

Most of NER research works relied on other two types of features. Results from sixteen systems that have participated in the CoNLL-2003 [10] have shown that word features are the most popular and suitable features to be used. Most of the systems used features from word e.g. affix information, chunk tags, lexical features, orthographic information, part-of-speech, etc.

Syntactical feature is one of document features that is typically used together with word features. Research works in [17, 18] provided an evidence that adding a syntactical feature can effectively improve the NER system performance. This feature is suitable for accident documents which are typically represented on a sequence of sentences. It helps the recognition process by providing a syntactical structure of the sentence. Current NER system used restricted syntactical features. For instance, research work in [19] used two contextual clues which are appositive modifier and preposition together with a set of word features. Another research used 2 types of syntactical rules; 1) constituency parse rules (e.g. appositive modifier and preposition) and 2) dependency parse rules (e.g. subject, object) to recognize named entities [18]. However, those restricted syntactic rules may not be applicable to every example since a structure of a sentence might be very complicated. Consider an example on the following sentence "*Egypt's transport minister, Mohammed Mansour, resigned in Octobe*r". There are three types of named entities that can be recognized; "Egypt" as location, "Mohammed Mansour" as person and "October" as date entity. The person and date entity can be recognized using appositive modifier and preposition. But "Egypt" is left untagged since there isn't a rule that captures this entity. It is therefore a great challenge to develop a new NER approach using syntactical features that could capture all syntactical structure in a sentence.

1.3 Objectives

In relation to the problems, there are two primary objectives of this thesis:

Objective (1) Evaluating NER performance by applying syntactical structure from Link Grammar as the NER feature. We come with this objective in order to resolve

the limitation on the current NER works that is caused by the use of restricted syntactical feature.

Objective (2) Applying the proposed NER approach into accident domain. To the best of our knowledge, there is no existing NER work that focused in the accident domain.

To support the main objectives, there are three sub-objectives as follows:

1. To apply part-of-speech and a set of connector from a syntactical parser known as Link Grammar (LG) parser [20] to determine the boundary of named entity.
2. To create extraction patterns using syntactical features from LG parser and a set of word features. The nature of accident document that consists of sentences ranging from simple to complicated make syntactical feature the most appropriate feature to be utilized.
3. To apply a particular semi-supervised machine learning technique namely Self-Training algorithm to perform the classification process based on the generated extraction patterns.

1.4 Contributions

To satisfy the research objective, this thesis highlights three main contributions:

1. The first contribution is identification of entity boundary which is the first task of named entity identification module. Named entity identification module has two main tasks which are identify all named entities and define the boundary of each named entity. Named entity may consist of one or more words; hence identification of entity boundary is required. The idea is to use part-of-speech and LG connector to determine the starting and stopping word. To the best of our knowledge, within the limited literatures available on NER, combination between part-of-speech and LG connector have never been attempted to identify entity boundary. A list of named entity candidates will be produced from this identification step. The second task of named identification module-extraction pattern construction will become the second contribution which is explained in the next point.

2. The second contribution is the construction of extraction pattern set from LG connector and word features. As mentioned in [21] a set of extraction pattern is a key component of IE system. As such, relevant information can be extracted from text document using extraction pattern. Similar to IE, NER also uses extraction pattern which is constructed from a set of NER features. In this thesis, we used a set of word features including part-of-speech, capitalization, punctuation, digit and common ending, and also syntactical features produced by a grammatical parser known as LG parser. Word feature is one of NER features that often used and is proven to produce a considerable result. In addition, syntactical feature is also used with the word feature. LG parser is able to produce syntactical structure of sentences. LG parser is one of grammar formalism which not only produces a "constituent" representation of a sentence (e.g. showing noun phrase, adjective, verb phrase, etc.) but also produces a set of labelled links connecting pairs of words [22]. LG parser has a set of link-types which have different grammatical usage. For instance, `TW` connector is used to connect days of the week to month names, `ON` connector is used to connect the preposition "on" to certain time expression, `G` connects proper noun together in series, etc. Thus we see a possibility to use those connectors as a part of extraction pattern. LG has been used in several NLP applications such as Machine Translation (MT), Grammar Checking, IE, etc. However, to the best our knowledge, LG never been used as the extraction pattern in NER application. An example of a sentence parsed by LG parser can be seen in the Figure 1.2

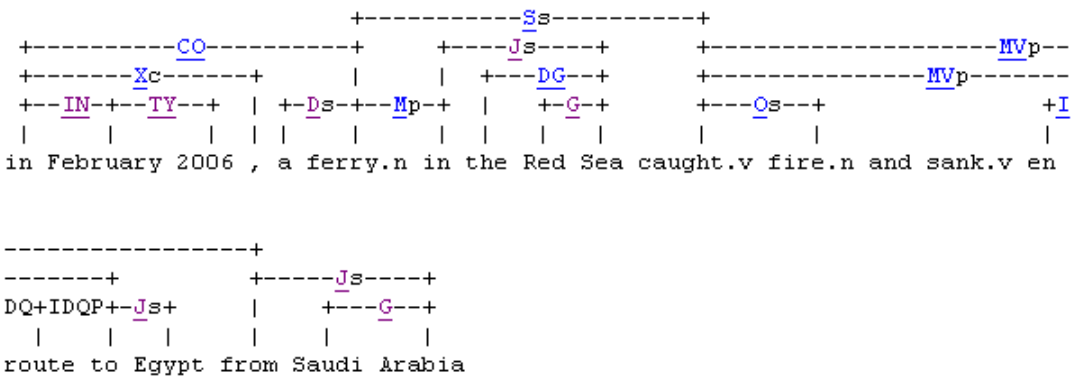

Figure 1.2 Example of Sentence Parsed by LG

3. The development of semi-supervised NER module using the generated extraction pattern becomes the third contribution. This module is the implementation of named entity categorization process. Semi-supervised learning is one of machine learning technique which falls between supervised and unsupervised learning [23]. The main goal of this learning is to minimize the usage of labelled data without decreasing the system performance. This technique is chosen as labelled data on accident domain is difficult to obtain. More over, constructing labelled data for training purpose will be very time consuming. Semi-supervised learning utilizes both labelled data and unlabeled data. A "seed" which can be a small number of labelled data or a classifier is used to initiate the learning process, while unlabeled data are used to assist the classification process.

1.5 Scope of Study

The research effort presented in this thesis focuses on proposing an improved NER approach. Scope and limitations of this work are mentioned as follows:

1. The proposed NER approach is tested on accident news.

2. The proposed NER approach is tested to recognize only three important named entities which are date, location and person. Those three entities are examples of the popularly recognized entities in common texts.

3. The performance of proposed NER approach is measured based on three evaluation metrics: 1) precision, 2) recall and 3) F-measure.

4. The aim of this research is to propose NER approach, not to use or integrate the approach into an extended application i.e. IE or QA application.

1.6 Thesis Organization

This thesis is organized as follows:

- In Chapter 2, the thesis presents theoretical background and literature review of NER, techniques and approaches in NER, and evaluation of the NER system performance. This chapter also explains LG connector, part-of-speech and a set of word features which are used in the recognition process. Some previous works on accident domain will also be included.
- Chapter 3 provides detail explanation of pattern construction followed by description on two main modules that have been built, i.e. named entity identification module and named entity recognition module.
- In Chapter 4, discussion on experiment set-up and result is provided. It is explained how to prepare the testing data, the evaluation method used, and the result of the system which used three different extraction patterns. In addition, comparison with three existing NER system is also provided.
- Conclusion and future works are drawn in Chapter 5.

CHAPTER 2

LITERATURE REVIEW

This chapter reviews several important topics related to this research work. Most research works in Named Entity Recognition (NER) focus on three important factors that determine the final performance of the system: 1) Technique 2) Features and 3) Domain. Those works are trying to find out what are the most effective technique and suitable feature for specific domain. Thus, we split this chapter into four sections that described the background of our methodology. In Section 2.1, we briefly review a few essential topics in NER including existing competition project on NER, application of NER and problem domains of NER to provide a sufficient background for understanding. Section 2.2 focuses on NER technique. In this section we provide a review on two basic methods of NER which are Knowledge Engineering approach and Automatic Training approach. A detail review on Automatic Training approach or well known as Statistical machine learning is also provided in this section. In addition, we also present a semi-supervised learning as the technique that we want to adopt in our research work. In Section 2.3, NER features and extraction pattern is explained. Moreover, the section also gives detail explanation on Link Grammar (LG) parser which has been utilized to produce syntactical features. Lastly, we draw conclusion of this chapter in the chapter summary. A summary diagram to describe the content of Chapter 2 is provided in Figure 2.1

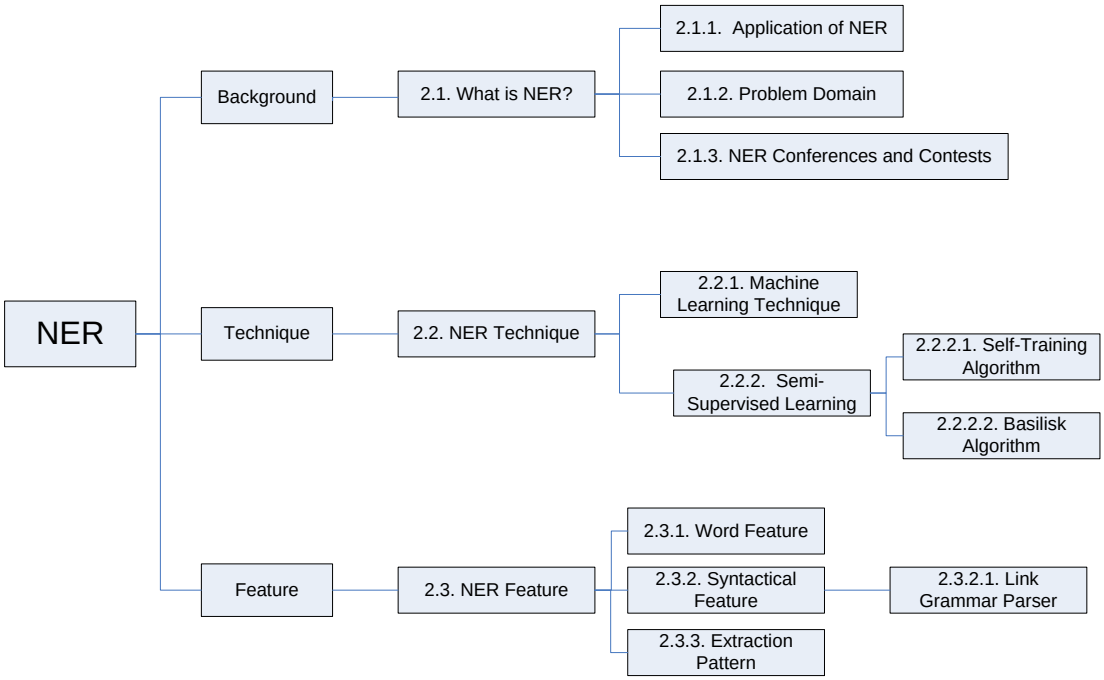

Figure 2.1 Chapter 2 Summary Diagram

2.1 What is NER?

The tremendous growth of digital text documents on and off the internet has motivated the development of Information Extraction (IE) research. IE is among one of the crucial fields in Natural Language Processing (NLP) that deals with unstructured texts. Basically, IE involves the process of structuring the text, extracting patterns in the structured data and finally performing data analysis and interpretation. NER is an ongoing research that has been supporting the IE research since 1990s [9]. It can be said that NER was introduced for the first time at the Message Understanding Conference Sixth (MUC-6) and become one of IE's important subtask

Identification and classification are the aims of NER. NER plays a significant role to recognize entities in a text and classify them into some predefined categories. Among the examples of the popularly recognized entities in common texts are the person's name, date, time, location, company's name and currency. However, these categories are varying depending on the nature of the text domain. For example, terrorism domain may require type of event as one of the important entity.

This may take word such as bombing, attack, hijack, arson and murder. Accident domain may require different types of entity such as date, time, number of victim, location, etc. An example of NER is given in the following paragraph [24]:

> In March, Prime Minister Morgan Tsvangirai was injured and his wife killed in an accident on the same Harare-Masvingo highway, one of the many roads neglected during the country's economic collapse.

Based on MUC-7 Named Entity Task Definition [25], three named entities could be identified in the paragraph, date, person and location.

> In<**DATE**>March</**DATE**>, Prime Minister <**PERSON**>Morgan Tsvangirai</**PERSON**> was injured and his wife killed in an accident on the same <**LOCATION**>Harare-Masvingo highway</**LOCATION**>, one of the many roads neglected during the country's economic collapse.

From the above example, we can manually identify those named entities easily. However, it won't be as simple as manual identification when we want to recognize those names automatically. A clearer example can be seen in the following sentences:

Sentence 1: *Washington is the first president of United States.*

Sentence 2: *Washington D.C is the capital of U.S.*

Without any doubt, manually we can identify "Washington" in Sentence 1 as a person name and "Washington" in the second sentence as a location name. However, identifying those named entities using NER is not easy. The word "Washington" might lead to an ambiguity. Thus, the context evidences of both entities must be collected before the NER system can decide which "Washington" is person or location name. In addition, the system needs to identify the word "United States" that has subsequent mention of "U.S.".

In the next subsection, a detailed background of NER is provided. We started with some explanation of several NER applications in section 2.1.1. After that, in section 2.1.2 we present a discussion on some popular domains that have been used in recent

NER research works. Section 2.1.3 will provides information about the past NER conferences and contests that gave major contribution to NER research.

2.1.1 Application of NER

NER has been proven as an essential component in IE area; therefore it has also influenced other NLP and text mining tasks to use the same approach. Many applications used named entity to improve their performance. There are many applications which take the advantages of NER; some of them are reviewed below:

Question Answering (QA)

QA application frequently uses a combination of Information Retrieval (IR) or passage retrieval with NLP technique like NER [26]. A research work in [27] used combination passage retrieval and rule based NER for Spanish QA application. It also used dictionary and Wordnet in their NER system. The result showed that by embedding NER it could reduce input data up to 26% and increase system efficiency to 9%. A study in [28] reported of an effort for finding optimal characteristic of NER that fits with QA application. The research compared single and multiple labelling for named entity. The experimental result proved that increasing NER recall by allowing multiple labelling can benefit QA task.

Web Content Filtering

A research in [29] tried to improve the effectiveness of current web content filtering software that mainly used Uniform Resource Locator (URL) blocking or keyword matching by investigating shallow linguistic processing in particular NER. Some features including binary orthographic feature, list of frequent words, punctuation symbols and predicted class for the previous words were used in the NER module. This research mentioned that the experiment showed an encouraging result. Another research in [30] used appearance of some named entities like geographical location, organization, date, time, money, etc to classify web documents for tenders domain. An improvement of 2.6% is shown in the result.

Machine Translation (MT)

One of the important problems being faced in MT research is how to identify named entities correctly. Global syntactic, lexical structure, local and immediate context of the translation may be affected due to inaccurate identification of named entities [31]. In order to improve MT quality, B. Bogdan et al. in [31] associated GATE NER module of Sheffield University to their MT system. Result from experiments on two English-France and one English-Russian MT systems indicated there is an improvement on the output quality.

Novelty Detection

N. Kok Wah et al. in [32] used NER in their novelty detection system for text document. Features from part-of-speech tagger and Wordnet were incorporated to extract some entity types including person, place, time and organization. Two different metrics, UniqueComparison and ImportanceValue are used to calculate the novelty score of each document. Benchmarked against the scores for the Text Retrieval Conference's (TREC) Novelty Track 2004 the experimental result shown was promising.

From the examples above, NER has successfully brought an improvement to the performance of some application systems. The usage of NER on several applications has demonstrated by evidence that the existence of NER not only gives a big contribution on IE but also to the whole body of NLP area. In the next subsection, a review of problems domain of NER is provided. It described how NER has been utilized in many domains such as bio-medical, terrorism, business and other domain.

2.1.2 Problems Domain

Problems domain becomes one important factor that receives a lot of attentions in NER research works. Most of NER systems are domain dependent, which means it can only be used in specific domain [9]. For example, NER system built for bio-medical domain can not be used for identifying company name on the business domain. It might be possible to port a specific NER system into a new domain; however the result, the system performance might be degrading. On this section, we provide a review on several popular domains on recent NER research works.

Newswire domain

Newswire domain became the first domain that has been used as the main focused for NER research. Researchers are trying to recognize a common entity names such as location, organization, person name, date, time etc. Since 1995 when Message Understanding Conference Sixth (MUC-6) was held, a lot of research works have been conducted. MUC-6 is regarded as the conference where NER was introduced for the first time. This conference also provided a corpus that mostly focusing on business article.

Biomedical domain

An overwhelming amount of biomedical text that contains important information has attracted researcher on NER to take biomedical domain as the main focus. Research works on this domain are mainly trying to recognize biomedical entities such as gene, protein, virus and DNA names. In recent years, many research works on this domain has been done [33-35]. As this domain start to be popular, there are several biomedical corpuses provided for research purpose such as GENIA [36] and BioInfer [37]. Moreover, several conferences and contests in this domain were also held.

Accident domain

To the best of our knowledge, there exists no work on accident domain that specifically focuses on NER. However, there are a number of works on accident domain which are closely related with IE task. One of them is a work reported in [13]. This work tried to identify incident causes on Air Investigation Reports corpus available from the Transportation Safety Board of Canada. A binary classification was adopted to identify whether a sentence contains causal or factual information. Instead of using Bag of Words (BOW) representation, Structured Sentence Representation (SSR) was used. Sentence is mapped into 4 attributes including subject, verb, object and modifiers. After that, LG parser is utilized to decompose each sentence into their constituent parts. This work also utilized Wordnet to generalize each SSR vector of a sentence. Sentences which contain different word but semantically similar word will be considered as similar. As a result, the approach reached 84% accuracy level.

Another work on [14] described the usage of LG parser and regular expression pattern matching to identify collisions in National Transportation Safety Board (NTSB) Accident Summary Reports. This work is part of CarSim [15], a system used to visualize written road accident report to 3D scenes animation. First, to detect sentence candidates, accident reports are tokenized into sentences and each sentence is matched against regular expression. A list of collision verb from Wordnet is used in this step. Then, each candidate sentence is passed to the LG parser. LG parser is applied to extract subject and object in the sentence and to help handling co-reference. Tested on 30 NTSB accident reports which contains a total of 43 collisions, this approach could reach 60.5% of hit ratio and correctly detect 26 complete collisions and 12 incomplete collisions.

However, those works only focused on IE with no specific work focused on NER. Thus we see it as a challenge, since in accident domain there is a lot of crucial information that need to be extracted. Accident domain itself can be accident report or accident news. We found that the nature of both of them is quite different. Here, in this research we intend to focus on accident news, which is somehow also different in nature with the other news document. In accident news, we often found an indirect sentence that might not be found in business news.

From those IE works on accident domain, it also can be concluded that LG parser has been successfully used in the process of extracting information from raw text document with considerable result. Syntactical structure from LG parser has been found capable of providing sufficient evidence to identify a sort of information inside a sentence. Other than that, linkage pattern from LG parser can also be used to resolve co-reference problem.

2.1.3 NER Conferences and Contests

As one of the important tasks in IE, NER becomes one of research focus on NLP area. Several research works have been conducted to find the best method in recognizing named entity. English was the first language that receives a lot of attention with regards to this aspect. Later, other languages like Spanish, German, Dutch, Arabic,

Chinese and Japanese have also been explored. In this sub-section, several NER competition projects that have provided major contributions on NER area will be discussed.

Named Entity Recognition or NER is a part of IE which has been introduced for the first time at MUC-6 in 1995 [4]. MUC-6 focused on extracting information from unstructured text. It used a set of data from Wall Street Journal newswire articles related to company and defense activities. NER was reported as one of four tasks (NER, Co-reference task, Template Element task and Scenario Template task) that have been evaluated, and compared to the other tasks, its performance could be considered has exceeded expectation. Most of the NER systems which were participated in MUC-6 could reach precision and recall over 90%. In 1998, the last series of MUCs [3] was held. It used airline crashes domain as the training data and for the testing data, launch events domain was used. It was reported that the domain change has affected the system performance. The performance of system evaluated during MUC-7 has slightly decline as compared to the performance of system evaluated during MUC-6 [38]. However, during MUC-7, more international sites were participating and for the first time, the Multilingual Entity Task (MET) evaluation was run on the same domain for all involved languages which include Chinese and Japanese. NE task in both MUC-6 and MUC-7 was focused on recognizing 4 different entities including entity names (for people and organizations), place names, temporal expressions, and numerical expressions.

Information Retrieval and Extraction Exercise (IREX) [39] is a competition-based project which has the aim to provide the same standard for all IE and IR researchers working on Japanese language. It was started in 1998 and finished in 1999 involving 45 participants from Japan and US. Named entity task is one of the subtasks in IREX. Eight types of named entities, including organization, person, location, artifact, date, time, money and percent were defined. MUC/MET definition was used to define those types. Three types of system had participated in this event namely hand created pattern based, automatically created pattern based and fully automatic system. Interestingly the top three systems identified in this event were from each of the mentioned types.

The best system was the hand created pattern based system, followed by the automatically created pattern based system and fully automatic system. From the result it could be concluded that time and numeric expressions were easier to be recognized as it could achieve 80% of average F-measure. On the other hand, the results have also shown that the accuracy of other NE types were not that good.

Conferences on CoNLL-2002 [40] and CoNLL-2003 [10] were another prominent NER evaluation event. The shared task of these evaluations involved 4 languages namely Spanish, Dutch, English and German. Four types of named entities: persons, locations, organizations, and names of miscellaneous entities that did not belong to the previous three groups were explored in these evaluations. Twelve systems participated in CoNLL-2002 and sixteen systems participated in CoNLL-2003. The data for these evaluations was taken from newswire articles. Most of the participants in those events used Machine-Learning techniques like Hidden Markov Model (HMM) [41], Decision Tree (DT) [42], Conditional Random Field (CRF) [43], and Support Vector Machine (SVM) [44]. Some features like lexical features, part-of-speech tags, and orthographic features had also been used. On the performance level, English language obtained the best result followed by Spanish, Dutch and German.

Entity detection and tracking (EDT) was one of primary tasks that had been explored in the Automatic Content Extraction (ACE) [45] project for the period of 2000-2001. Seven types of entities including person, organization, location, facility, weapon, vehicle and geo-political entity (GPEs) were identified in this project. This project did not only explore English language but also Chinese and Arabic respectively. Unlike two others NE evaluation, the result of this evaluation project was not publicly available and restricted only to participants.

Those four conferences can be regarded as important events that gave major contribution in the development of NER area. Through those events, researchers have gathered to propose best approaches in identifying named entity. In addition, the result can be used as the reference for the other researchers in order to develop NER system with a better performance. The most obvious contribution of these conferences was the establishment of standard data set and evaluation method for NER generally accepted by those researchers and practitioners working in this field. For example,

MUC employed an evaluation method where an NER system is evaluated on two axes: its ability to find the exact named entity and its ability to give the correct type. Using this evaluation, a partial credit will still be awarded when errors occur on only one axis. This method is different with Exact Match evaluation produced by CoNLL. In this evaluation, the named entity is considered correct if it exactly matches with the corresponding entity in the key test [9]. Each conference also has promoted a standard data set that has been used in most of NER research works.

After providing the background of NER, in the next section we start to discuss on the NER technique. We provide literature reviews that support our decision on the technique that will be utilized in our methodology.

2.2 NER Technique

It has been widely known that there exist two basic approaches on designing IE system including its subtask-NER system [8] : 1)Knowledge Engineering Approach and 2)Automatic Training Approach.

Knowledge Engineering Approach

Knowledge Engineering Approach or famously known as Hand-Made Rule-based technique focuses on manual rules creation by human experts or "knowledge engineer". The knowledge engineer constructs a pattern or rule by analyzing the features appears in the text. A set of features including grammatical, syntactical, orthographical features are usually used to identify the named entity aspect. The performance of the system is heavily relied on the skill of the knowledge engineer. Among rule examples highlighted in the research is "If a proper noun follows a person's title, then the proper noun is a person's name". A comparative study in [46] shows that this technique creates a better result for a specific domain. However, manual creation of rules is very labour intensive and costly. Early studies of NER mostly used Rule-based approach, based on the evidence that five of eight systems that participate in MUC-7 used this technique [9].

Automatic Training Approach

Unlike the first technique, the Automatic Training Approach or known as Machine Learning (ML) approach doesn't need human experts to manually construct the rule. Rule is constructed automatically by a trained system. The trained system may learn rules either from annotated document or from interaction from user. In addition, this approach also used statistical methods to help the classification process [8]. Trainable system is developed in order to replace the function of knowledge engineer. There are three types of ML: Supervised, Semi-supervised and Unsupervised learning. Each learning technique is differentiated based on size of training data used in the training process. Recent research works on NER started to use this technique, as reflected on all 16 participants of CoNLL 2003 that have applied this approach in their proposed work [9].

Both approaches have their own advantages and disadvantages. A system using Rule-based approach is easy to deploy when there exist skilled and experienced linguist expertises. In addition, because the system relied on a set of grammatical rules and dictionary list, it has an ability to identify complex entity that is not possible using the trained approach. On top of that, the performance of this approach still outperforms ML technique. However, the performance of this system heavily relied on the skill of knowledge engineer and difficult to be port into new domain. It can be said that Automatic Training approach is intended to address the weaknesses of Knowledge Engineering approach; it does not required any linguist expert and relatively easy to be ported into new domain. However, in term of performance, the Knowledge Engineering approaches still outperforms ML technique [8].

Apart from their strengths and limitations, the choice of using either Rule-based or ML approach is supposed to be determined by considering the availability of resource and the expected system performance. For example, ML is more suitable when linguist experts and resources to create dictionary and grammatical rules are not available. ML is also the right choice when training data is cheap and easy to be obtained. In addition, if only a reasonable performance of NER is required, then ML is preferred. On the other hand, if all resource to construct dictionary and grammatical rules is available, training data is expensive, rule writers can be found easily and an

outstanding system performance is needed, then Rule-based approach is the best choice [8]. The comparison summary between Rule-based and ML approach is provided in Table 2.1.

Using this literature review, we can easily decide which technique is the most suitable to be used. In our case, since resource and linguist expert to construct dictionary and grammatical rules is unavailable, then ML approach is more appropriate technique to be chosen. As described in subsection 2.1.2, there are only a few NER research works that focus in accident domain, thus resources like data set, list of accident term and also linguist expert experienced in this domain are still scarce. In the next subsection, more detailed explanation of ML technique that will be used in our proposed work is provided.

Table 2.1 Comparison on Two Basic NER Approach

Comparison	**Knowledge Engineering Approach**	**Automatic Training Approach**
Method description	– Also known as Hand-made Rule-based Approach. – The system relied on a set of grammatical rules and dictionary list. – A patterns or grammatical rules are constructed by knowledge engineer by analyzing the features appear in the text. – Early studies in NER mostly used this method, refer to the fact that five of eight NER systems that have participated in MUC-7.	– Also known as Machine Learning (ML) Approach. – Trained system may learn rules either from annotated document or from user interaction. – There are 3 types of ML: Supervised, Semi-Supervised, Unsupervised Learning. Each learning technique is differentiated based on the size of training data. – Recent studies in NER mostly used this method, refer to the fact that 16 participants in CoNLL 2003 are used this approach.
Performance	Perform best among other techniques.	– The performance is good enough but still can't outperform rule based technique.
Strength	– A rule-based system with good performance is easy to develop when skillful and experienced knowledge engineers are available. – The performance of this approach is best among other techniques, especially when it is used in specific domain. – Has the capability to detect complex entities that trained approaches may have difficulty to deal with.	– Linguist expertise is not required – The system is easy to be ported into new domains.

Limitation	– The performance of the system is relied on the skill of knowledge engineer. – Linguist expertise may not be available. – A manual pattern construction is tedious and time consuming. – A complete and comprehensive dictionary is difficult to obtain. – The grammatical rules and dictionary list need to be updated and maintained regularly to accommodate any changes which may be costly. – Lack of ability to be ported into new domains.	– The system may require a large number of training data. – Training data may be expensive or difficult to be obtained – Changes to named entity specification, may require re-annotation on the training data that may be time consuming.
When to use?	– There are available resources to create grammatical rules and construct dictionary list. – Linguist expertise is available. – Training data is difficult to obtain. – There is a tendency that the extraction pattern specification will slightly change over time. – Highest possible result performance is very important.	– No available resources to create grammatical rules and construct dictionary list. – No skilled and experienced knowledge engineer is available – Training data is easy and cheap to be obtained. – Extraction pattern specification is stable. – Good result performance is sufficient.

2.2.1 Machine Learning Technique

In this section we present a review on ML techniques literatures. We compare three types of ML in order to know which one is the best technique for our proposed approach. A work on [47] compiled a summary of the usage of ML for IE. It is mentioned that the weaknesses of rule-based techniques have motivated the establishment of several works on IE which used ML approach. Statistical ML approach is the right choice to be applied when human experts is unavailable, training data is easy to get, extraction specifications are stable and the system performance is not critical [8].

A work in early nineties by E. Riloff [48] was one of the first work which used automatic training approach to construct a dictionary for IE task. An automatic system called AutoSlog used *conceptual anchor point* and 13 heuristic patterns to construct terrorism-dictionary for extracting information from text document. Evaluated on two blind test sets of 100 texts, AutoSlog dictionary achieved 98% performance level as compared to the hand-crafted dictionary.

Statistical ML is divided into three different parts [9]: 1) Supervised Learning 2) Unsupervised Learning and 3) Semi-supervised Learning. Each technique is distinguished by how much supervision level is provided. X. Zhu and A. B. Goldberg in [49] gave a detail description on each technique. It is explained that in statistical machine learning, an *instance* **x** represents a specific object. A *D*-dimensional *feature vector* $x = (x_1, \ldots, x_D) \in R^D$ represents each *instance.* The representation of the feature is an abstraction of the objects. In NER, a feature can be syntactical features, grammatical features, orthographical features, etc. A training *sample* is collection of instances $(x_i)_{i=1}^{n} = (x_1, \ldots, x_n)$. This *training sample* becomes an input for the learning process. These instances are sampled independently from an underlying distribution $P(x)$, and denoted as $(x_i)_{i=1}^{n} \sim P(x)$.

Training sample in supervised learning consists of pairs of an instance x and a label y: $(x_i, y_i)_{i=1}^{n}$. A *labelled data* is defined as a pairs of (instance, label) while *unlabeled data* is defined as an instance alone without label. Supervised learning

trained a function $f : X \to Y$ on given training sample $\{(x_i, y_i)\}_{i=1}^{n} \sim P(x, y)$ where X is domain instances, Y is domain labels and $P(x, y)$ is joint probability distribution on instances and labels as $X \times Y$. Later, when a future data **x** is given**,** function $f(x)$ should predicts the right label y. A supervised learning with discrete classes y is called as classification and function $f(x)$ is called as *classifier*. A number of algorithms like Hidden Markov Model (HMM) [50-52], Decision Trees (DT) [53, 54], and Conditional Random Field (CRF) [43] have been applied as classifiers in the learning process of several NER works.

Most NER works which used supervised learning reported that their systems could yield a better performance. A learning name-finder called Nymble [50] which used HMM and word features has successfully reached 90% on the F-measure score. Another work by D. Shen et.al [52] has also used HMM for their NE recognizer on biomedical domain. The experiment result showed that their system (62.5% F-measure) outperforms the best reported NE recognizer (54.4% F-measure) in GENIA corpus Version 1.1. In addition, a NE recognizer with CRF, feature induction and web-enhanced lexicon [43] is reported to reach 84.04% on F1. However, in order to reach that performance, those systems need a large amount of training data. For instance, Nymble used almost 100,000 words of training data. In addition, it was reported that reducing the training set size have decreased the performance of the system. More over, training data can be very expensive since the process to produce it needs manual effort [23].

Unlike supervised learning where the labelled data is provided, unsupervised learning is only given unlabeled data $\{\mathbf{x}_i\}_{i=1}^{n}$ without any supervision to handle it. Clustering is one of unsupervised learning task aimed to split *instances* into *k* cluster. One of simple clustering algorithm, *hierarchical agglomerative clustering* use distance function $d(x)$ to determine whether two instances $\mathbf{x}_i$ and $\mathbf{x}_j$ is in the similar cluster.

Y. Shinyama and S. Sekine [55] used comparable news articles to discover named entity. The evaluation showed, by taking words with similarity score of ≥ 0.6; it could discover rare named entities with 90% accuracy.

However for 966 single words which have been taken as testing data, the system could only discover 462 named entities or less than 50%. Unsupervised NER in [17] is proposed to improve an existing NE recognizer using syntactic and semantic contextual evidence. From three different experiments with three different corpuses, it has been shown that unsupervised NER could improve its performance up to 18%. Yet, this unsupervised NER has only been used as a complementary to the existing NER and, it hasn't been tested independently on its own. Unsupervised learning is attempted to address the limitation of supervised learning by omitting labelled data in the learning process; however it has affected the system performance. Evaluated using the same benchmark, unsupervised learning rarely performs as well as supervised learning [23].

Semi-supervised learning falls between supervised learning and unsupervised learning. S. P. Abney in [23] mentioned that semi-supervised learning is generalization of classification and clustering. In classification, the entire training data is labelled while in clustering none of the data is labelled. Semi-supervised learning use both labelled and unlabeled data in the learning process. Semi-supervised learning is an extension of supervised and unsupervised learning [49]. It is *classification* with labelled and unlabeled data, with one assumption that the amount of unlabeled data is greater than labelled data. Semi-supervised also known as *constrained clustering* where there are *must-link constraints* and *cannot-link constraints* to separate unlabeled data into different clusters.

Semi-supervised learning is divided into two types: Inductive and Transductive semi-supervised learning. Inductive learning is described as follows: if there is a training sample $\{(x_i, y_i)\}_{i=1}^{l}$, and unlabelled data $\{x_j\}_{j=l+1}^{l+u}$, inductive learning learns a function $f : X \to Y$ that expected to be a good predictor over the unlabelled data. While in transductive learning, assume a training data $\{(x_i, y_i)\}_{i=1}^{l}$ is given then transductive learning trains a function $f : X \to Y$ to be a good predictor over the unlabelled data $\{x_j\}_{j=l+1}^{l+u}$. Minimizing the usage of labelled data without decreasing the system performance is the main goal of semi-supervised learning. A "seed" which can be a small number of labelled data or a classifier is used in the first learning process.

Thereafter in the next step, unlabeled data is used to help the classification process. Semi-supervised learning may achieve the same level of performance with supervised learning and at the same time reduced the manual effort on producing training data [23]. We believe, in choosing the most appropriate learning technique to be used we have to consider not only the past performance records of each technique but also looking at other factors like the availability of training data and the ease of algorithm implementation.

Table 2.2 provides a comparison summary between three ML techniques. In term of the system performance, supervised learning shows a superior performance as compared to the other techniques. However, in return it needs a huge number of training data. It might trigger a problem since constructing training data takes time and can be very expensive. There is a tendency that more recent research works on NER are starting to explore semi-supervised and unsupervised learning that need less training data. Though the performance of both learning still can't surpass supervised learning but several NER research works that used semi-supervised learning was reported could reach comparable [56-58] and even outperform [59] the performance of supervised learning. Looking to the unavailability of training data in accident domain and the past performance records of semi-supervised learning, we reached a conclusion that this learning is best suited to be applied in our work.

There are many algorithms in semi-supervised learning approach [49]. A number of common methods include: Self Training, Co-Training, Support Vector Machine (SVM), and Graph-based Algorithm. The discussion and literature review on the semi-supervised learning algorithms will be presented in the next subsection.

Table 2.2 Comparison Summary between Three ML Techniques

Parameter	Supervised Learning	Semi-Supervised Learning	Unsupervised Learning
Description	Supervised learning trained a function $f : X \to Y$ on given training sample $\{x_i, y_i\}_{i=1}^{n} \sim P(x, y)$ where X is domain instances, Y is domain labels and $P(x, y)$ is joint probability distribution on instances and labels as $X \times Y$. Later, when a future data **x** is given**,** function $f(x)$ should predicts the right label y.	Divide into two types: – Inductive learning is described as follows: if there is a training sample $\{x_i, y_i\}_{i=1}^{l}$, and unlabeled data $\{x_j\}_{j=l+1}^{l+u}$, inductive learning learns a function $f : X \to Y$ that expected to be a good predictor over the unlabeled data. – Transductive learning, assume a training data $\{x_i, y_i\}_{i=1}^{l}$ is given then transductive learning trains a function $f : X \to Y$ to be a good predictor over the unlabeled data $\{x_j\}_{j=l+1}^{l+u}$.	Unsupervised learning is only given unlabeled data $\{x_i\}_{i=1}^{n}$ without any supervision to handle it. Clustering is one of unsupervised learning task aimed to split *instances* into *k* cluster. One of simple clustering algorithm, *hierarchical agglomerative clustering* use distance function $d(x)$ to determine whether two instances $\mathbf{x}_i$ and $\mathbf{x}_j$ is in the similar cluster.
Example of Algorithm	Hidden Markov Model, Decision Tree, Conditional Random Field.	Self-Training, Generative models, S3VMs, Graph-based Algorithm, Multi-view Algorithm	Clustering

Training Data	Use large number of training data	Use small number of training data	Use no training data
Performance	Among other ML technique, this technique performs best.	May achieve the same level of performance with supervised learning.	Rarely performs as well as supervised learning.
Strength	The performance is best among others	– Reduced the manual effort on producing training data – Using a small sized of labelled data, the system may achieve the same level performance with supervised learning.	Reduced the manual effort on producing training data
Limitation	– Training data can be very expensive – Creating training data can be very time consuming	The system still can't perform as good as supervised learning.	Rarely performs as well as supervised learning

2.2.2 Semi-Supervised Learning

In this subsection, we will discuss on semi-supervised algorithm that will be used in our work. There are many algorithms in semi-supervised learning. Xiaojin Zhu in [60] mentioned that there is no direct answer for "which is the best method" question. The decision of choosing the method should be taken based on the problem structure. The best method is the algorithm that fits the problem structure. On this work, a simple checklist is provided to help us find the most suitable algorithm, i.e. Expectation-Maximization (EM) with generative mixture model is the best choice when the classes produce well clustered data. Co-Training may be appropriate if the features used in the learning naturally split into two sets. Graph-based method can be used if two points with similar features tend to be in the same category. And Self-Training algorithm is the right choice when we have difficulties on modifying supervised classifier.

2.2.2.1 Self-Training Algorithm

According to [49] in many real world tasks like NLP, when applying ML technique, the learners can be regarded as black boxes. It can be a simple algorithm like K-Nearest Neighbour (K-NN) algorithm or very complicated classifier. It is important to highlight that the learners may not be amenable to changes, thus a simple semi-supervised algorithm is needed. Self-Training algorithm known as practical wrapper method is the best technique when simplicity is of major concern. The algorithm procedures only "wraps" around the learner without makes any change. The detail of the algorithm is provided in Figure 2.2:

1. Train f from (X_l, Y_l)
2. Predict on $x \in X_u$
3. Add $(x, f(x))$ to labelled data
4. Repeat

Where:

Input instance x, label y

Learner $f : x \rightarrow y$

Labelled data X_l, Y_l

Unlabeled data X_u

Figure 2.2 Self-Training Algorithm

The first step of this algorithm can be considered as the supervised learning part, which is learner function $f : x \rightarrow y$ is trained on labelled data (X_l, Y_l). As mentioned earlier, in semi-supervised learning, the amount of labelled data is very few, thus it is called as *seed*. The learner function $f : x \rightarrow y$ is then used to predict a label y for an instance x where $x \in X_u$. Thereafter, prediction result $(x, f(x))$ is added to labelled data, and the process is repeated again. In Self-Training algorithm, prediction result with the highest confidence is considered as correct. Unsupervised learning can be found on the second iteration onwards, where unlabeled data is utilized in the learning process.

From the second iteration forward, learner function $f : x \rightarrow y$ is retrained on the larger labelled data. There are three possible methods to determine the stopping criterion as explained in [23]. In the first method, the algorithm is run in predetermined fixed number of times. In the second method, the algorithm is run until a convergence state is reached or in other words until all unlabeled data is processed. The third method is by using cross-validation in order to estimate the optimal number of iteration. Usually, only prediction result $(x, f(x))$ with the most confident prediction is added to the labelled data; however it is also possible for the whole prediction result to be added to the labelled data.

We can see here, the procedure of Self-Training algorithm enable user to choose the learner function $f : x \rightarrow y$ and treat the learner function as a black box. However, this algorithm still have a limitation which is when learner function $f : x \rightarrow y$ made wrong prediction and generating incorrect labelled data. Thus it is very important to determine appropriate algorithm for learner function $f : x \rightarrow y$.

Self-Training algorithm has been used in a number of NLP works. The most frequent Self-Training paper that had been cited is a work by Yarowsky [61]. In that work, Yarowsky used an iterative bootstrapping procedure for word sense disambiguation. Two powerful properties of human language; one sense per collocation and one sense per discourse are used together with decision list algorithm as the classifier. Given identical training context, the algorithm achieved 95.5% of performance almost the same as supervised learning algorithm of 96.1%. In addition, the algorithm outperformed Schutze's unsupervised algorithm [62], a pioneered work in the word sense clustering, for up to 4.5%.

A work in [63] also applied Self Training and Co-Training algorithms for Spanish NER. Self-Training algorithm is used to detect the named entities while Co-training algorithm is implemented for classification task. Four feature sets including lexical and orthographical features, trigger word and gazetteer word are incorporated. 20 hand-labelled instances are used as a seed and K-NN algorithm is utilized as the classifier. For each iteration, a pool of *P* unlabeled examples is created and only the most confident prediction results *G* (growing size) are added into labelled data. The algorithm is repeated up to 40 times. A set of parameter is applied to this algorithm with 1620 labelled data and G = 200 and as a result, a best performance achievement of 84.41% is obtained.

Another work in [64] conducted an experiment to compare performance of single-view semi-supervised algorithm including self-training and EM algorithm and multi-view semi-supervised algorithm which is Co-Training algorithm. While single-view algorithm uses only one classifier to teach itself; multi-view algorithm trains two classifiers that provide most confident prediction result for each other.

Two strong assumptions that determine the success of Co-training algorithm are; first, each classifier is sufficient to make good classification and second, both of the classifiers must be conditionally independent [49]. The experiment result has shown that Self Training algorithm achieved a better performance as compared to the others.

From the literature review it can be seen that incorporating Self-Training algorithm in NLP works shows a good performance result. Self-Training algorithm successfully outperforms the other two semi-supervised algorithm in this case namely Co-Training and EM algorithm. Moreover, it also has shown a superior performance over one of a pioneer unsupervised algorithm and reached almost the same performance as supervised algorithm that used larger training data. Considering the simplicity and also its performance, we believe that Self-Training algorithm is the appropriate method to be applied. In subsection 2.2.2.2, we provide a review on one of Self-Training algorithm that will be adopted in our proposed approach.

2.2.2.2 Basilisk Algorithm

As mentioned before, in Self-Training algorithm the choice of learner function $f : x \rightarrow y$ is completely open [49]. This learner function will give a prediction to the unlabeled data based on given seed. It plays an important role on the performance result. When learner function $f : x \rightarrow y$ made a wrong prediction and generating incorrect labelled data, then the system performance become worse in each iteration. Thus, it is important to choose which learner function $f : x \rightarrow y$ that will be used. In this subsection, we present one of Self-Training algorithm called Basilisk (Bootstrapping Approach to Semantic Lexicon Induction using Semantic Knowledge) [65]. Basilisk used collective evidence from extraction pattern to generate semantic lexicons of terrorism term.

Basilisk takes an un-annotated text document and a small number of *seed word* as the input. Before the bootstrapping process begins, AutoSlog generates an extraction pattern for every single noun phrase found in the text document. Then, Basilisk utilized the extraction pattern to determine a semantic class for every noun phrase. The Basilisk algorithm is explained in Figure 2.3 as follows:

lexicon = seed word

i = 0;

1. Score all extraction patterns
2. *pattern_pool* = top ranked *n* + *i* patterns
3. *candidate_word_pool* = extraction of patterns in *pattern_pool*
4. Score candidate word in *candidate_word_pool*
5. Add the candidate with the *p* highest score to the *lexicon*
6. Repeat

Where:
n and p is an integer value

Figure 2.3 Basilisk Algorithm

Here, we intend to highlight two scoring metrics which are used in Basilisk. The first metric is *RlogF* metric which is utilized to rank each extraction pattern. It was first introduced in [66] and successfully demonstrated in other information extraction and text processing research works [67-73]. The extraction pattern is scored using the following formula:

$$R\log F(pattern_i) = \frac{F_i}{N_i} * \log_2(F_i) \quad \textbf{(2.1)}$$

Where:

F_i is the number of seed word extracted by $pattern_i$

N_i is the total number of words extracted by $pattern_i$.

RlogF metric was originally created for IE task. *RlogF* metric is attempted to maintain a balance between reliability and frequency. When the extraction pattern is strongly correlated with a semantic class, R value ($\frac{F_i}{N_i}$) is expected to be high. Moreover, when a pattern could extract a large number of word or entity that belong to a particular semantic class, then the *F* value will be high too [71].

On Basilisk algorithm, the *n* highest *RlogF* patterns will be stored in *pattern_pool*, and the value of *n* is increased for each iteration in order to get more extraction patterns.

The second metric is *AvgLog* metric. Each word that has been extracted by the extraction pattern that has already been stored in the *pattern_pool* will be scored by this metric. If a word is extracted by patterns that have a tendency to extract members of a semantic class then it will get a high score of *AvgLog*.

$$AvgLog(word_i) = \frac{\sum_{j=1}^{P_i} \log_2(F_j + 1)}{P_i} \quad \textbf{(2.2)}$$

Where:

P_i is the number of extraction patterns that extract $word_i$

F_j is the number of distinct seed word extracted by pattern *j*.

Basilisk added *p* words with the highest AvgLog score to the lexicon, the *pattern_pool* and *candidate_word_pool* will be emptied and the algorithm is repeated again.

In [65] Basilisk algorithm was used together with AutoSlog's [74] extraction pattern. The experiment used MUC-4 corpus which contains 1700 texts on terrorism domain has shown that Basilisk outperformed meta-bootstrapping algorithm [71] which also used extraction pattern to construct dictionary for IE task.

We can see here, that the learner function $f : x \rightarrow y$ of Basilisk is built on two components: RlogF and AvgLog metric. The strong point of RlogF metric is that it gives a score to an extraction pattern based on two axes: first, the correlation of extraction pattern with semantic class and second, the ability of the pattern to extract large number of entity that belongs to a semantic class. RlogF value defines how an extraction pattern could extract a large number of named entities with high precision. The second metric, AvgLog assigns a value to each named entity candidate based on the performance of its extraction pattern. If a named entity is extracted by a pattern

that have tendency to extract members of a semantic class then that entity will gain a high score of AvgLog. It is interesting because a named entity candidate have to obtain a certain value of RlogF and AvgLog before it can be categorized into a

particular semantic class. More over, two layers of scoring metric are used to avoid the wrong prediction on unlabeled data. After giving a detail explanation on NER technique, in the next section we provide a review on another important NER factor which is NER features.

2.3 NER Features

A thesis in [9] summarized that NER features can be grouped into three different types. The first type is *Word-Level Feature*. This feature type describes all contextual evidence found in a word. Word case feature (e.g. a word starts with capital letter), punctuation (e.g. word with internal period), digit (e.g. cardinal number or word with digit), character, morphology (e.g. a word with "er" ending indicates a profession), part-of-speech (e.g. noun, verb, cardinal number) and function (e.g. token length) are considered as word-level feature.

The second type is *List Look-Up Feature*. This feature type used a list of words to help the classification process. List look-up feature is divided into three types; *general list* which may contain common words, capitalized nouns, stop words or abbreviations. The second types is *list of entities* which cover all entity names such as name of organization, first name, name of countries etc. Another type is *list of entities* cues which consist of all word that typically found with named entity e.g. "Mr." that usually precede person name, or "Corp." which used along with company name.

The last type of NER feature is *Document and Corpus Feature*. While word-level feature explores more on contextual evidence of a word, document and corpus feature focus more on content and structure of the document itself. Multiple word occurrences, local syntax (e.g. position in sentence), meta-information (e.g. url or email header) and corpus frequency are examples of features in the category of document and corpus feature type.

In [16], Y. Roman et al. mentioned weaknesses of list look-up feature highlighting that, complete and comprehensive lists are difficult to be constructed. For instance, a comprehensive list of person's first name is hard to be created, since there are a lot of variances of first name. In addition, constructing a complete dictionary is time consuming, as it needs to periodically update the dictionary when a new name is found. More over, using a larger dictionary on NER doesn't always produce a better result [8, 75]. A work in [76] shows that by increasing lexicon size from 9000 to 110000 will only add less than 3% to the improvement of system performance. Typically, list look up feature is used as a complement to the other features [77, 78]. The comparison summary between three NER features is provided in Table 2.3. The next two sections will explain more on word feature and one of document feature which is syntactical feature. Both will be used in this reported work to construct extraction pattern.

Table 2.3 Comparison Summary between Three Types of NER Features

Word-Level Feature	**List-Look Up Feature**	**Document and Corpus Feature**
– Most common features used in NER – Lexical, orthographical, morphological feature, part-of-speech – Has been proven could give considerable performance result. (CoNLL 2003-F. T. K.S. Erik and M. Fien De , 2003)	– Used a list of words to help the classification process. – Complete and comprehensive lists are difficult to be constructed. – Constructing a complete dictionary is time consuming (Y. Roman, et.al., 2002)	– One of them is local syntax feature of a sentence – Enumeration, apposition and word position – Has been proven that incorporated syntactical structure as the NER features could generate a considerable result. (M. Behrang and H. Rebecca, 2005) (M. Collins and Y. Singer, 1999)

2.3.1 Word Feature

Word feature can be said as the most common feature used in the NER approach. Lexical, orthographical, morphological feature and part-of-speech are examples of most frequently used word feature. Lexical feature utilizes each token in the named entity itself as a feature. Capitalization, punctuation, digit are included as orthographical feature. Morphological feature will use common ending, prefix and suffix as the attributes to identify named entity [9, 79].

Another word feature is part of-speech. In English, verb, noun, adverb, adjective are examples of part-of-speech. NLP software that is used to assign part-of-speech to the word is named as part-of-speech tagger. Similar word can be assigned with different part-of-speech. An instance is shown by the used of word *general* in the following sentence, "*General Electric had an extensive line of general purpose and special purpose computers*"[1]. The first *general* is a proper noun which is part of a company name, while the second *general* is an adjective that give information about a purpose of computer. From this example, it can be concluded that one of the advantages of using part-of-speech as NER feature is that it can reduce word ambiguity [8].

In this thesis we intend to use particular part-of-speech tagger software called Stanford tagger. A publicly available part-of-speech tagger from Stanford NLP group will be used to produce part-of-speech tagset of each sentence. Stanford tagger is Maximum-Entropy POS tagger that is implemented using Java. Maximum entropy technique is one of the top performing methods on part-of-speech works besides Hidden Markov Models (HMM) [80] and transformation-based learning [81]. A research in [82] reported that the performance of this tagger is better than the other taggers that used maximum entropy approach. It achieved 96.86% accuracy on the Penn Treebank[2] Wall Street Journal (WSJ) and 86.91% on previously unseen words. An improvement of this tagger was reported in [83]. In this work, K. Tautanova et al. tried to improve the system performance by providing efficient bidirectional inference using dependency network. Moreover, they also incorporated lexical and unknown

[1] This sentence is taken from http://en.wikipedia.org/wiki/General_Electric

[2] http://www.ims.uni-stuttgart.de/projekte/CorpusWorkbench/CQPHTMLDemo/PennTreebankTS.html

word features in their tagger. As the result, the tagger achieved 97.24% accuracy on Penn Treebank WSJ. It was claimed that the tagger performed better than any previous single tagger. In addition, the accuracy of this tagger is slightly better than the best known combination tagger which achieved 97.16% accuracy as reported in [84]

The usage of word feature on NER system has been proven to give considerably better performance result. Sixteen systems participated in the CoNLL 2003 provided strong evidences on this. Affix information, bag of words, global case information, lexical feature, orthographic information, orthographic pattern, part-of-speech, and trigger word are some word features used by those systems. Evaluated using English test set, the best performance was obtained by a system described in [85] which used eight features including lexical feature, part-of-speech, affix information, orthographic feature, gazetteer, chunk tags and case information in combination with three learning algorithms, namely MEM, transformation-based learning and HMM. The system reached performance level of 88.99% on precision, 88.54% on recall and 88.76% on F-measure. In addition, the average F-measure obtained by the sixteen systems that have participated reached approximately 82%. These systems had the combination of those features mentioned and several other different algorithms.

From the above review, we can obviously see that word feature is one of the basic features, most of NER works utilize in constructing extraction pattern. System that used more word features has a tendency to reach a better performance. In addition, part-of-speech as one of word feature also has been proven could resolve the ambiguity problem. However, the performance of the system is not solely relying on the feature used but also on the applied technique. In our proposed approach, we intend to use a set of word features including capitalization, punctuation, digit, part-of-speech and also the previous identified named entity. Capitalization, punctuation and digit will help to define the word structure, while part-of-speech is useful to avoid ambiguity problem. The previous identified named entity will help to directly recognize entity with the same name. Beside word features, we also utilize syntactical feature. In this case, we are trying to explore Link Grammar (LG) as the tool to produce syntactical structure of a sentence. Next subsection will discuss more on this.

2.3.2 Syntactical Feature

Most NER works explore the local syntax feature of a sentence. Enumeration, apposition and word position [9] are examples of this feature. M. Collins and Y. Singer in [19] used spelling feature together with apposition to collect evidence that a word belongs to a specific category. An instance is given in this sentence: *"., says Mr. Cooper, a vice president of..."*. Mr. Cooper in the example is categorized as a person type, since it contains *Mr.* as indication of person name. In addition, *president* which is in apposition with *Mr. Cooper* gives another "hint" that it belongs to person type.

Word position is another local syntax feature that is often used. Syntactical parser is NLP software that gives information about the position or function of a word in particular sentence. Given a sentence, this software will assign the sentence with a syntactic structure. A simple example is illustrated in the following figure:

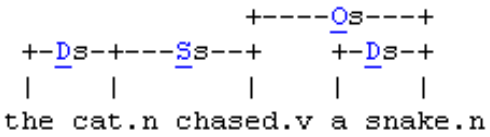

Figure 2.4 Syntactic Structure

Parser gives each word specific properties. Figure 2.4 shows a sentence with its syntactic structure. It can be explained as follow, `Ds` represents that word-*the* is determiner for a noun-*cat* and the same thing can also be said about the determiner *a* and noun-*snake*. *Cat*, which is the subject to the verb-*chased*, is depicted by `Ss`. Verb-*chased* has a noun-*snake* as the object and it is represented with `Os`.

It has been proven that by incorporating syntactical structure as one of the NER features, it could generate a considerably improved result. A work in [19] used pairs of spelling and contextual features in association with semi-supervised algorithm. Word sequences from parsed sentences were extracted as named entity examples if it satisfies the following criteria: 1) First, word sequences must be a noun phrase which consists of a sequence of consecutive proper noun. 2) Second, the noun phrase must appeared as the complement to a preposition or, the noun phrase has an appositive

modifier. 3) Third, other than those two context features, 5 spelling features including full-string, contains, allcap1 (single word-all capitals), allcap2 (single word-all capitals and contains at least one period) and non-alpha (contains characters other than upper/lower case letters) are utilized. Applied using three different algorithms, the system could achieve 91% clean accuracy on average. The limitation on this system is that two restricted contextual features might not be applicable for every named entity examples, since the example might be found in other contexts.

The result in [19] has motivated M. Behrang and H. Rebecca to propose dependency features in addition for appositive and prepositional features as described in [18]. Dependency features was proposed as proper noun which may act as subject or object in the sentence. Used together with several spelling features, semi-supervised EM algorithm and tested on three data sets, the obtained results suggested that the accuracy rates of system which used dependency features are comparable to the system that depends solely on appositive and prepositional features. In addition, the combination between dependency feature with appositive and prepositional feature has resulted for a better accuracy to be obtained. In the next subsection, LG parser will be regarded as one of the syntactical parser used to produce syntactical structure will be described.

2.3.2.1 Link Grammar Parser

Link Grammar is one of the grammar formalism developed by D. Sleator and D. Temperley from Carnegie Mellon University [20] for English parsing. Grammar formalism can be defined as a formal mechanism for capturing grammatical knowledge of natural language [86]. The principal things of LG that must be highlighted are each word in LG has a linking requirement and it can define a sequence of words as a sentence if three following conditions are gratified: There are no crossed links (*Planarity*); sequence of words can be connected together (*Connectivity*); linking requirement of each word is fulfilled (*Satisfaction*). LG has a list of linking requirement as specified in the dictionary. Each word in a sentence has one or more attributes called *connector* e.g. `Ds, Ss, Mvp`, etc. Figure 2.5 shows simple linking requirement for each word in a particular sentence.

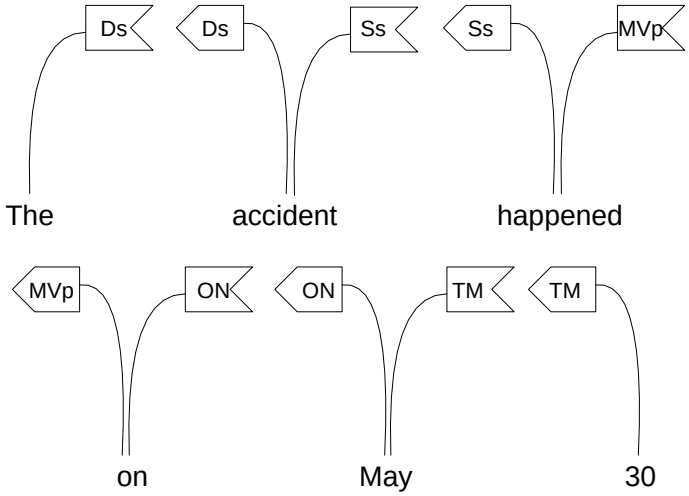

Figure 2.5 Linking requirement of each word in a sentence

Word *The* has one right `Ds` connector, thus it requires a `Ds` connector to its right. In the other hands, *accident* needs a `Ds` connector to its left. In order to be able to draw a link, those connectors must be plugged into compatible connector. Thus, *The* and *accident* can be linked since they fulfilled linking requirements of each other. Figure 2.6 illustrates a sentence that met the specified linking requirement.

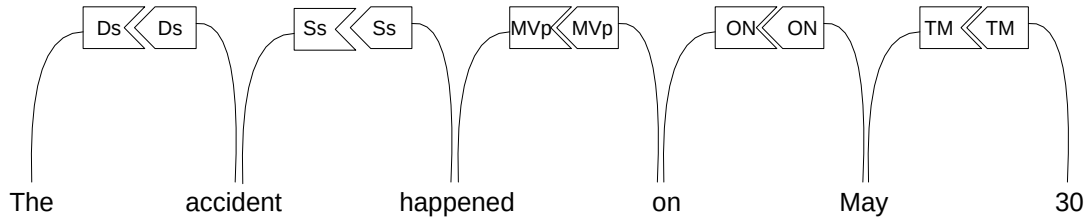

Figure 2.6 A sentence that met linking requirement

A *linkage* is a set of links that verify that a sequence of words is in the language of link grammar. For each sentence, link grammar may provide more than one linkage. An illustration of linkage is drawn in Figure 2.7.

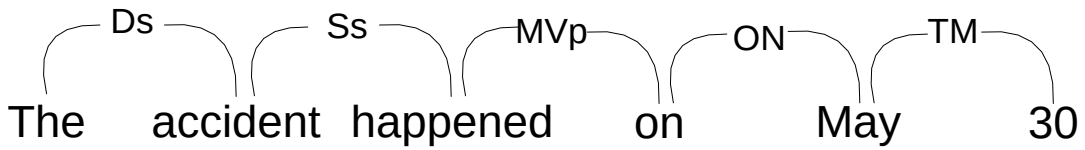

Figure 2.7 An example of linkage

LG parser is capable to capture numerous phenomena of English grammar with approximately seven hundred definitions ranging from noun-verb agreement, question, imperatives, complex to irregular verbs and many more. LG parser has a set of link-types which have different grammatical usage. For instance, in Figure 2.7, it has been shown that there are 5 different link- types, which are `Ds`, `Ss`, `MVp`, `ON`, and `TM`. `D` is used to connect determiners to nouns, because the noun is singular, then LG add `-s` after `D`. Similar thing is applicable to the on `S` connector which is used to link subject-nouns to finite verbs. `MVp` is derived from `MV` link which is used to connect verbs and also adjectives to modify phrases i.e. adverbs, prepositional phrases, time expression, etc. `MVp` is only used to connect prepositions to verbs. To connect the preposition "on" to certain time expression, LG uses `ON` connector and `TM` is used to connect month names to day numbers.

Beside LG, there are two others grammar formalisms which are Dependency Grammar (DG) and Constituency Grammar (CG). In term of performance, LG has an equivalent expressive power as compared to Context Free Grammar (CFG)-most commonly used mathematical model for CG, but the unique element of LG is that it conforms better to human linguistic intuition [87]. LG has been used in many NLP applications e.g. Machine Translation (MT) [88, 89], Extraction of Biochemical and protein interactions [90], Grammar Checking [91], QA application [92-94]. LG has also been applied in several numbers of IE works.

A work in [95] introduced a learning architecture for IE called Stochastic Real Value Units Algorithm (SRV) that offers maximum generality and flexibility. SRV used features from two general-purpose NLP systems, which are LG and Wordnet. Tested on 600 "acquisition" articles in the Reuters corpus to identify nine fields including official names of parties (acquired, purchaser, seller) and also its abbreviations, location of the company, price paid and progress of the negotiations,

SRV is able to reach a very good coverage (up to 99% for several fields) but failed to maintain the accuracy (on some fields it was dropped until 14%-15%). However, this approach surpassed two others algorithm which are Rote and Bayes [96].

Other works in [97, 98] proposed an alternative approach to generate candidate extraction rules from raw document. Work in [98] divides extraction rule into three components which are *conceptual anchor point* or triggering word, *linguistic pattern* which is grammar structure and *enabling condition* to activate the extraction rule. LG is used to identify all of the noun phrases in the sentence and give prediction whether a noun is subject, object or noun phrase in prepositional phrase. Then the prediction is used to generate candidate extraction rules. In this work, 13 predefined linguistic patterns are utilized. Candidate extraction rules is filtered using conditional probability formula called *relevance rate*. A top *x* single extraction rules will be refined into multi slot candidate rules and clustered by two dimensional models which uses combination between linguistic pattern clustering and synonym clustering. In this step, a large scale online dictionary Wordnet is utilized. This approach seems promising; however there isn't any published paper that shows the real implementation of this proposed model.

We believe that the capability of LG which could capture more than seven hundred phenomena in English grammar, made it as the appropriate parser to be utilized in our approach. More over, LG not only gives constituent part like the other parser does but also assigns each word with a specific connector. We found that the given connector has a great potencial to resolve the previous problem remains in several research works that utilized only limited syntactical feature. After giving detail review on NER feature, in the next subsection we will provide an explanation about extraction pattern and also several works that have different step on the extraction pattern construction.

2.3.3 Extraction Pattern

A set of NER features is used to construct an extraction pattern. This extraction pattern will play a significant role on identification and classification process.

Generating extraction pattern is not new in text mining and IE research. I. Muslea in [21] reviewed several types of extraction pattern either in free text documents or in more structured type documents like web pages. Most research works used machine learning algorithm to generate the extraction pattern. One of the first research in extraction pattern was AutoSlog [48]. Using heuristic rules, AutoSlog built dictionary of extraction pattern. In order to extract information from the document, AutoSlog used triggering word and 13 predefined set of linguistic patterns. An extension of AutoSlog was developed to avoid the complexity of annotation task in the system. AutoSlog-TS [66] only required pre-classified training corpus and did not need any annotation task. The performances of both systems were evaluated by applying them on the MUC-4 documents, which consist of terrorism terms. As the result, AutoSlog-TS achieved higher precision but lower recall than AutoSlog.

A system called LIEP [99] learned extraction pattern from the text. LIEP tried to generate pattern that could recognize syntactic relationships between key constituents. LIEP have been applied to extract corporate management changes and corporate acquisitions from newswire text. LIEP achieved an average F-measure of 85.2% (recall 81.6%; precision 89.4%).

Extraction pattern for semi-structured documents will be different with the system for free text documents. One of the research works in this area is RAPIER system [100], which combined syntactic information and semantic class information to generate extraction pattern. The extraction pattern consists of three different parts, Pre-Filler pattern, Post Filler pattern and the Filler pattern itself. Pre- and Post-Filler pattern give constraint to the information extracted in the left and right sides. Each part can be pattern items (word) or pattern list (tag set produced by the syntactic information).

2.4 Chapter Summary

In this chapter, we presented a detail review on NER, including the basic knowledge of NER, NER techniques and NER features. First, in section 2.1 we provided a detail background of NER. Several IE-NER competition projects and applications of NER

are described. From this section, we can see how NER has been given a lot of attention indicating the importance of NER. In section 2.2, we discussed more on the NER technique. We compare two basic NER approaches: Knowledge Engineering Approach and Automatic Training Approach (ML approach) and come with a conclusion that Automatic Training Approach is the most suitable method to be used. After that, we also compared three types of ML approach: supervised, semi-supervised and unsupervised learning. By considering the unavailability of resources to create training data and also the system performance, we come with a decision that semi-supervised suits best to be applied. In this section we also presented a review on Self-Training algorithm and one of its variant, Basilisk algorithm. The simplicity of Self-Training algorithm makes it very easy to be applied without reducing the performance of the learner function. Lastly, in section 2.3, explanation of NER features is presented. We reviewed on word features and syntactical features that will be used in our approach. In addition, we highlighted on LG parser, the English grammar parser that has very complete dictionary to capture a huge number of English grammar phenomena. With its capability, the usage of LG parser in this approach is expected on the result improvements.

CHAPTER 3

PROPOSED APPROACH

This chapter provides discussions on methodology and approach which is used in this thesis. First, in section 3.1, system architecture which depicts the whole NER system in general is presented. Afterwards, two modules in the system which are the implementation of two processes; Named Entity Identification and Named Entity Categorization are described in more detail in section 3.2 and 3.3 respectively. The summary of this chapter is provided at the end. The summary diagram of this chapter is provided in Figure 3.1.

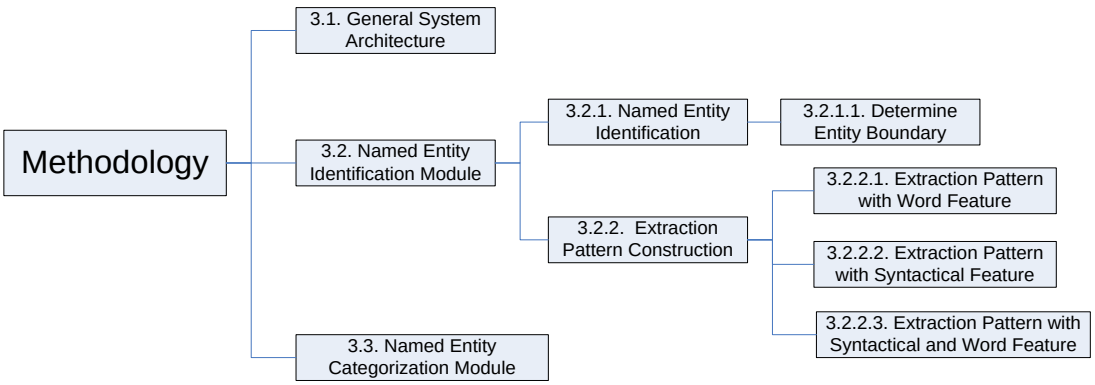

Figure 3.1 Chapter 3 Summary Diagram

3.1 System Architecture

H. Cunningham and K. Bontcheva in [101] explained that commonly, there are two processes involved in NER system that uses Machine Learning technique: 1) Named Entity Identification and 2) Named Entity Categorization. In this thesis, two modules have been developed for the NER system. Figure 3.2 shows simple system architecture of proposed NER system. In this figure, each module is highlighted and labelled as different dashed boxes.

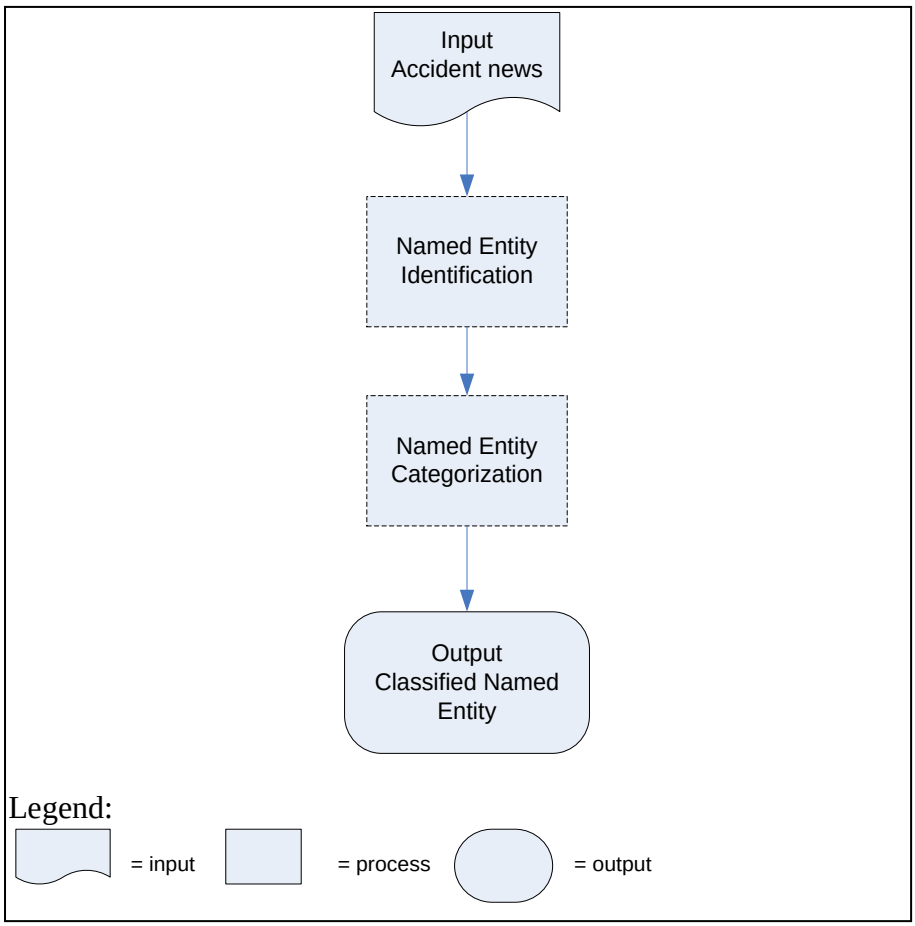

Figure 3.2 General Architecture of Proposed NER System

Generally, the named entity identification module consist of two main sub-modules to be reflected in the diagram: 1) Named entity candidates boundary identification and 2) Extraction pattern construction for the respective entity candidates. This module processed input file per sentence. Each sentence is fed to both part-of-speech tagger and syntactical parser. The features from both types of NLP software which are part-of-speech tagger and syntactical parser are utilized to perform the first task. After the identification process, an extraction pattern is created for every single identified named entity candidate. We used three different types of extraction pattern. First is extraction pattern which is constructed from several types of word features i.e. punctuation, digit, capitalization and part-of-speech. Second, we only used part-of-speech and syntactical feature from LG parser. Third, we combined both word feature and syntactical feature to see whether there is an improvement on

the performance. As the output, a list of named entity with its extraction pattern is obtained. Detail explanation process in this module is given in section 3.2.

The named entity categorization module has a task to break down the list of named entity candidates into three different categories: date, location and person. These three entities are chosen because they always appear in most of accident report or accident news article. The example of an accident news article with entities highlighted in style (person: bold-underline; location: bold; date: bold-italic) is shown in the appendix. Self-Training Algorithm as one of semi-supervised machine learning algorithm in combination with two scoring formula from Basilisk Algorithm is applied to give prediction on which category each of the named entity is belong to. A seed entity which contains the most frequent entities in the testing data is used to initiate the classification process. The final output is a list of classified named entities. Section 3.3 will provide detail description on this module.

3.2 Named Entity Identification Module

The aim of an automated NER is to simulate human NER task and to reach at least near human NER result. Although the human NER is perfect but a simple task like identifying named entity is cumbersome and arduous especially when dealing with thousands of documents. Hence, the process of named entity identification and construction of extraction pattern is automated with the help of features from two NLP software, part-of-speech tagger and syntactical parser. Here, we used Stanford part-of-speech tagger to assign tag set to each word in the sentence and LG parser to produce syntactical structure of the sentence. In addition, on the extraction pattern construction process, a set of word feature will be used too. Figure 3.3 shows the detail process in the NE identification.

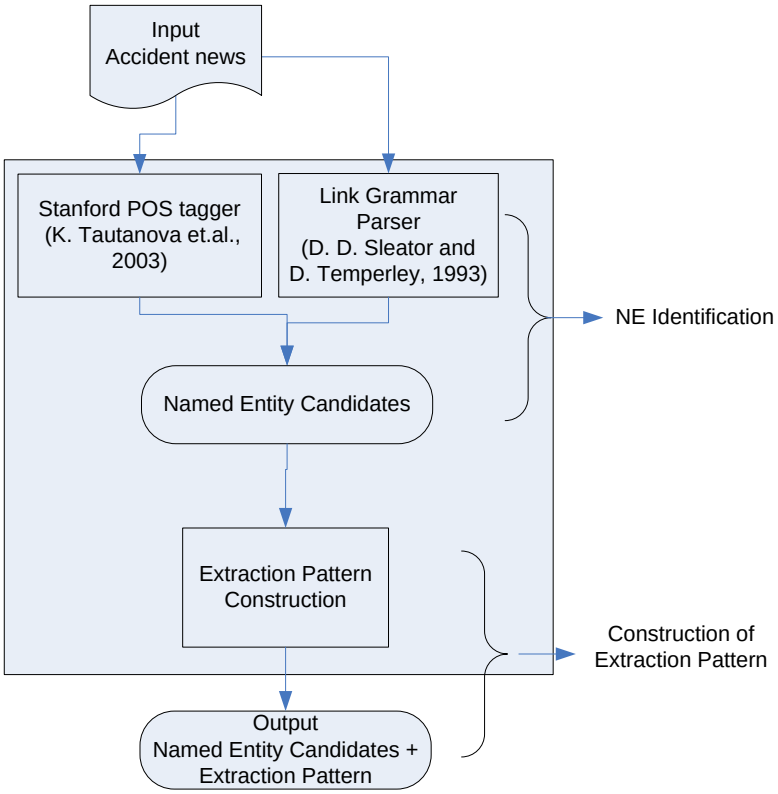

Figure 3.3 Architecture of Named Entity Identification Module

3.2.1 Named Entity Identification

N. Chincor provides a guideline in MUC-7 Named Entity Task Definition [25] for human annotator to identify several named entities in the text including 1) Entity names (ENAMEX tag element) which is limited to proper names, acronyms and other unique identifier. Three types of named entities which are included in this subtask are ORGANIZATION, PERSON and LOCATION. 2) Temporal expression (TIMEX tag element) consists of DATE and TIME, is limited for “absolute” and “relative” temporal expression 3) Number expression (NUMEX tag element) which covers MONETARY EXPRESSION and PERCENTAGE is for numeric expression, monetary expression and percentage. Based on the guideline; a named entity typically is a noun (it can be single noun, plural noun or proper noun), cardinal number or combination of the two. An example of a sentence that will be processed is shown in Figure 3.4.

In February 2006, a ferry in the Red Sea caught fire and sank en route to Egypt from Saudi Arabia.

Figure 3.4 An Example of Raw Sentence

In order to get part-of-speech tag of each word, the sentence is fed to Stanford part-of-speech tagger. Figure 3.5 illustrated the output from the tagger.

In/IN February/NNP 2006/CD ,/, a/DT ferry/NN in/IN the/DT Red/NNP Sea/NNP caught/VBD fire/NN and/CC sank/VBD en/IN route/NN to/TO Egypt/NNP from/IN Saudi/NNP Arabia/NNP ./.

Figure 3.5 Sentence tagged by Stanford part-of-speech tagger

An example of a named entity which is a noun can be seen from three LOCATION entities which are “Red Sea”, “Egypt” and “Saudi Arabia”. A DATE entity “February 2006” shows an example of a named entity which is combination of noun “February” and cardinal number “2006”. According to this standard, Stanford part-of-speech tagger was employed to identify all noun and cardinal number in the sentence. For each type of entity, different criteria is applied e.g. for LOCATION entity, the tagger needs to identify all proper noun since almost all locations consist of at least one proper noun. Besides proper noun, we also add a singular noun to the LOCATION entity as we found that sometimes location not only contains proper noun but also contains an extra noun i.e. “Calayan island”, “Ural mountain”. The words “island” and “mountain” are supposed to begin with an upper-cases letter. However, in some cases it appears with lower-cases. Our criterion for each entity is provided in Table 3.1.

Table 3.1 Tag set for each entity

Named Entity	Part-of-speech
DATE	All variant of Noun , Cardinal Number (CD)
LOCATION	Proper Noun either in single or plural form. (NNP, NNPS, NN)
PERSON	Proper Noun either in single or plural form (NNP, NNPS)

Moreover, a named entity may consist of one e.g. “Egypt” or more words e.g. “February 2006”, “Saudi Arabia”. One of the rules in MUC-7 Named Entity Task Definition mentioned that “A single-name expression containing conjoined modifiers with no elision should be marked up as a single expression”. An instance of a named entity is given, “U.S. Fish and Wildlife Service” should be marked up as one named entity not two.

3.2.1.1 Determine Entity Boundary

The boundary of named entity candidates is determined by feeding sentences into LG parser. LG parser will assign different types of connector to each word in the sentence as can be seen in

Figure 3.6.

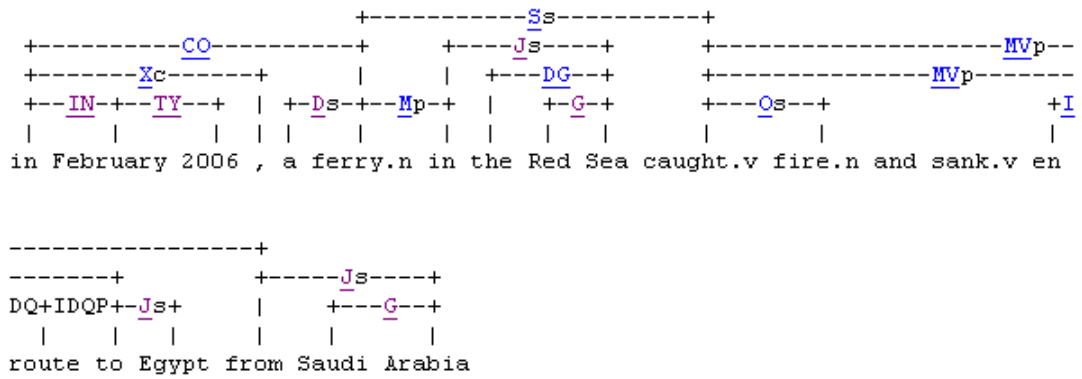

Figure 3.6 Parsed sentence by LG Parser

From this connector, a couple of words can be determined whether they must be marked up as one named entity candidate or not. A list of possible connectors returned by LG parser is created based on MUC-7 Named Entity Definition [25]. The guidelines and explanations are given as follows:

1. *Date Entity*

   a. Absolute Temporal Expression

      Either date or time expression must indicate a specific segment of time. As an instance, a particular day must be indicated by a specific name, such as

"Monday" or "Friday" but not "first day of the week" or "fifth day of the week".

b. Relative Temporal Expression

Relative temporal expressions which are followed by date or time unit as in "last month" or "next year" are tagged as part of date or time entity.

c. Miscellaneous Temporal Non-Entities

Date expression which does not specify starting or stopping dates i.e. "now", "recently" or "for the past few years" are not considered as date entity.

d. Temporal Expression Containing Adjacent Absolute and Relative Strings

If date expression contains both relative and absolute elements, then they are considered as one date entity. Examples are "July last year" and "late Tuesday".

e. Holidays

Holidays that are referenced by name are tagged as date entity i.e. "All Saints' Day".

f. Locative Entity-Strings Embedded in Temporal Expression

Location entity which modifies a contiguous time expression such as in "1:30 p.m. Chicago time" is tagged as a part of time entity.

g. Temporal Expression Based on Alternate Calendar

Fiscal year, Hebrew calendar which are categorized as temporal expression based on alternate calendar are tagged based on the above guidelines as date entity.

2. *Person Entity*

a. Titles and Generational Designator

Titles and roles name i.e. “Mr.”, “President”, “Health Minister”, etc are not part of named entity, however generational designator such as “Jr” ,”Sr”, “III” are included as part of named entity.

b. Family Entity-Expression

Family names such as “Kennedy” in “The Kennedy family” are tagged as person name.

c. Miscellaneous Personal Non-Entities

Several types of proper names like disease/prizes named after people, laws named after people are not tagged as person name.

3. *Location Entity*

a. Embedded Locative Entity-Strings and Conjoined Locative Entity Expression

A location named which is following an organization name will not be consider as location entity if there is a corporate designator i.e. Inc, Corp. An example is given like in “Hyundai of Korea, Inc.”. However if there is a corporate designator like in “Hyundai, Inc. of Korea” then “Korea” will be tagged as location entity.

b. Locative Entity-Expression Tagged in Succession

` Two or more place names in compound expression which is separated by comma are tagged as different location name. As an instance, “Washington, D.C.” will be tagged as two location entities.

c. Miscellaneous Locative Non-Entities

Location related string like adjectival form of location is not considered as location entity, e.g. American, Australian, and Japanese.

d. Locative Designator and Specifiers

Designator of place name is tagged as part of location entity. For example, word “River” that follows the name of a river, “Airport” or in the name of airport.

e. Trans-national and Sub-national Region Names

A location names including continents i.e. “Asia”, multi-country sub-continental region i.e. “Sub-Saharan Africa” and multi-country trans-continental region i.e. “the Middle East” are tagged as location entity. However sub-national region which is referenced only by compass-point modifiers is not tagged as part of location entity as it may be different for each region.

f. Times and Space Modifiers of Locative Entity Expression

Directional modifiers i.e. “north”, “south”, “east”, “west” and historic time modifiers i.e. “former” which are not intrinsic parts of location name are not considered as part of location name.

An example of this mechanism is explained as follow: according to the guideline, *a month which is followed by a year is tagged as one entity*. As displayed in

Figure 3.6, LG parser assigned a `TY` connector to link a month of “February” and a year of “2006”, hence `TY` connector is listed in the table and when two words are connected by this connector, it will be tagged as one entity. Another example can be seen from “Red Sea” and “Saudi Arabia”, all proper nouns are linked with specific connector which is `G` connector. The created list for each entity can be seen from Table 3.2 for date entity and Table 3.3 for person and location entity.

Table 3.2 provides some possible LG connectors that might be found in date entity. Most of the connectors are used to link nouns or cardinal number with article or determiner. LG parser provides specific connectors for date entity like `TA`, `TM`, `TW`, and `TY`. Those connectors can be used to indicate that a word is a month, a year or combination of them as a date entity.

Table 3.2 List of LG connectors for DATE entity

LG Connector	Explanation
Dmcn	This connector is used to connect plural and countable nouns with articles or numerical determiner Example: There have been 70 midair collisions involving 140 aircraft in the United States over the last **10 years**
Dmc	This connector is used to connect plural and countable nouns with articles or numerical determiner word Example: There have been 70 midair collisions involving 140 aircraft in the United States over the last **ten years**
Dsu	This connector is used to connect singular and uncountable nouns or articles with determiner
DT,DTi, DTie, DTn	This connector is used to connect determiners with nouns in certain time expression like “next week”, “last Tuesday” (DT, DTi, DTie) and “this week”, “every week” (DTn)
NS	This connector is used to connect numbers with certain expression which require numerical determiner but only for singular word like week. Example: Eleven miners were killed in the last explosion in the Donbass coalfield **a week** ago
NSa	Has similar function to NS, but it is used for idiomatic dictionary entries for “day” and similar words.
ND	This connector is used to connects numbers with certain expression which require numerical determiner Example: Eleven miners were killed in the last explosion in the Donbass coalfield **two weeks** ago
NF	This connector is used together with NJ in idiomatic number expression involving “of”. Example: The expansion is scheduled to begin operation in the fourth **quarter of** this year.
NJ	This connector is used together with NF in idiomatic number expression involving “of”. Example: The expansion is scheduled to begin operation in the fourth quarter **of** this **year.**

NN	This connector is used to connect number words together in series
TA	This connector is used to connect adjective like “late” to month names. Example: The accident happened in **early December**
TM	This connector is used to connects month names to day numbers Example: Sixteen women were among the dead in the **April 2** accident on Meitaung Mountain in Rakhine State.
TW	This connector is used to connects days of the week to month names Example: The accident happened on **Monday**, **May** 31.
TY	This connector is used for certain idiomatic usages of year numbers. Example: A train carrying gas is derailed, in Lawang, East Java province, Indonesia **September** 23, **2009**.
Y, Yt	This connector is used in certain idiomatic time and place expression Example: Eleven miners were killed in the last explosion in the Donbass coalfield two **weeks ago**.

Location and person entities mostly contain proper noun or noun. LG parser assigns G connector to link proper noun together in series. For person entity, we only include this connector since a persons name is always constructs from proper noun. On the named entity definition guidelines it was mentioned that *titles such as “Mr” and role names such as “President” are not considered part of person name*. LG parser has an ability to differentiate a role name i.e. “President”, “Prime Minister” from the person name and give GN connector which linked the title to the last name of the persons name.

Unlike person entity, location entity in some cases not only contains proper noun but also a noun. An example of a noun in location entity is often found in location entity which is followed by locative designator i.e. “river”, “mountains”, “airport”, etc. Therefore, beside G connector, we also included AN connector which is used to connect noun modifiers to nouns.

Table 3.3 List of LG connectors for LOCATION and PERSON entity

LG Connector	Explanation
G (location & person)	This connector is used to connects proper nouns together in series Example: President **Dmitry Medvedev** has been informed about the accident, Russian news agencies reported
AN (location only)	This connector is used to connect noun-modifiers to nouns Example: The accident occurred near the **Matura** *city* ***station*****.**

3.2.2 Extraction Pattern Construction

The second task of Named Entity Identification module is to construct an extraction pattern for each identified named entity. In this part, an extraction pattern is created; afterwards the pattern will be fed to one of semi-supervised machine learning algorithm. In this thesis, three types of extraction patterns are constructed. First, we used a set of word features including capitalization, punctuation, digit and mapped every single word into small patterns based on the character type. In addition, for particular entities we also used previous recognized entities to identify other entities.

Second, we utilized combination between part-of-speech and connector returned by the LG parser to construct extraction patterns. Syntactical feature from the LG parser is explored. The mostly explored syntactical features in previous research are the appositive modifier, prepositional phrase, subject and object which remain on several limitations. Hence in this thesis, we do not put limitation on that syntactical features but try to explore all syntactical features and let the Self-Training algorithm to identify which features are mostly found in the document. Thirdly, extraction pattern is constructed from combination between small patterns based on a set of word features and syntactical features. Either word feature or syntactical features have

their own strengths and limitations. Therefore the idea to combine between the two is expected to return a better performance result.

3.2.2.1 Extraction Pattern with Word Features

The first extraction pattern was established by utilizing a set of word features including capitalization, punctuation, digit, part-of-speech and also the previous identified named entity. We adopted a work from [102] which mapped each character in the word using a function of *type(x). Type(x)* of *x* is defined as "A" if *x* is upper-case letter. If *x* is lower-case letter, then it will be defined as "a", "0" if the character is digit and "-" if the character is punctuation. As an instance, word "Chicago" will be mapped to "Aaaaaaa" and "15 September" will be mapped to "00 Aaaaaaaaa ". If all words have been mapped to its type, then all repeated consecutive character types will be removed and there are not repeated strings in the mapped strings. For instance from the previous example, "Chicago" will be mapped as "Aa" instead of "Aaaaaaa" because there are repeated character types.

Apart from the small pattern of character types, we also used part-of-speech as part of extraction pattern. From this step, each word in the sentence will be mapped into word pattern and also given information about part-of-speech tagset. Consider the sentence in Figure 3.4., there are four examples of named entity "February 2006", "Red Sea", "Egypt" and "Saudi Arabia". Table 3.4 shows the word pattern and also part-of-speech of each entity that have been constructed from the extraction pattern.

Table 3.4 Extraction Pattern of named entity candidates

Named Entity	Word Pattern	Part-of-Speech
February 2006	Aa 00	NNP, CD
Red Sea	Aa Aa	NNP, NNP
Egypt	Aa	NNP
Saudi Arabia	Aa Aa	NNP, NNP

For recognizing date entity, we employ a small sized dictionary containing names of the day and month.

3.2.2.2 Extraction Pattern with Syntactical Features

Previous research as reported in [103] was focusied on creating extraction pattern by exploring prepositional phrase in the sentence. However, the study discovered some limitations when some entities are not part of prepositional phrase. Thus, in this thesis, we try to construct extraction patterns by applying syntactical structure from LG parser. The idea of extraction pattern representation was derived from RAPIER (Robust Automated Production of Information Extraction Rules) system [104] which divided its extraction rules into three different parts: 1) pre-filler pattern that matches with text preceding the actual slot, 2) actual slot filler and 3) post-filler pattern that matches with text following the actual slot.

Our extraction pattern is generated by considering features of the identified named entity. The extraction pattern is defined into four different parts: 1) The left connector which contains all the left LG connector 2) The right connector which contains all the right LG connectors 3) The middle connector which contains all LG connectors that connects the entities' words and 4) The entity tag which contains all the tag sets returned by the Stanford part-of-speech tagger.

The first and second parts, which are the connector that linked the entity with other words, describe the position or function of the named entity in the sentence. Consider an example in the Table 3.5. An entity "February 2006" is linked to the word "In" in the left by `IN` connector. LG parser used this connector to link preposition "IN" with certain idiomatic time expression. Similar with the first named entity, "Red Sea", "Egypt" and "Saudi Arabia" also have left connector and none of them have right connector. `Js` connect certain preposition like "in", "to", "from", etc to their object. In this example, all the named entities have a function as prepositional phrase. In other sentences named entity may be found in different position. It can be placed as a subject, object or other position.

Table 3.5 Syntactical feature of the named entity candidates

Named Entity	**Left Connector**	**Left Word**	**Middle Connector**	**Part-of-speech**	**Right Connector**	**Right Word**
February 2006	IN	In	TY	NNP,CD	-	-
Red Sea	Js	In	G	NNP,NNP	-	-
Egypt	Js	To	-	NNP	-	-
Saudi Arabia	Js	From	G	NNP,NNP	-	-

Note:

IN : Connect preposition “IN” with certain idiomatic time expression

Js : Connect preposition to their object

TY : Connects month names to year numbers

G : Connects proper noun together in series

The third and fourth parts of the extraction pattern describe the structure of the named entity. The third part which is the middle connector consists of all connector that linked words in the responding named entity. Only named entity with two or more words has middle connector. In Table 3.5, it is shown that all named entities that consist of two words have middle connector. The last part consists of part-of-speech of the named entity. This part also indicates the length of the named entity. Complete extraction pattern is illustrated in Figure 3.7.

February 2006	
Left Connector	IN
Middle Connector	TY
Part-of-Speech	NNP, CD
Right Connector	-

Red Sea	
Left Connector	Js
Middle Connector	G
Part-of-Speech	NNP, NNP
Right Connector	-

Egypt	
Left Connector	Js
Middle Connector	-
Part-of-Speech	NNP
Right Connector	-

Saudi Arabia	
Left Connector	Js
Middle Connector	G
Part-of-Speech	NNP, NNP
Right Connector	-

Figure 3.7 Example of extraction pattern

3.2.2.3 Extraction Pattern with Syntactical and Word Features

The third extraction pattern used combination between syntactical and word feature. Extraction pattern that only utilizes word feature typically only explore the word itself without considering the function or position of the word in the sentence. However, syntactical feature digs more on the contextual part of the sentence but does not explore on feature of the word itself. Hence, the idea to create extraction pattern that use combination between the two is expected to return a better performance result. Referring to the same sentence in Figure 3.4, each identified named entity candidate will be fed to the Stanford part-of-speech tagger, LG syntactical parser and also *type(x)* function. Generated extraction pattern contains three different parts, 1) syntactical pattern, 2) word pattern and 3) part-of-speech. Figure 3.8 shows the example of the extraction pattern for each entity.

February 2006	
Left Connector	IN
Middle Connector	TY
Part-of-Speech	NNP, CD
Word Pattern	Aa 00
Right Connector	-

Red Sea	
Left Connector	Js
Middle Connector	G
Part-of-Speech	NNP, NNP
Word Pattern	Aa Aa
Right Connector	-

Egypt	
Left Connector	Js
Middle Connector	-
Part-of-Speech	NNP
Word Pattern	Aa
Right Connector	-

Saudi Arabia	
Left Connector	Js
Middle Connector	G
Part-of-Speech	NNP, NNP
Word Pattern	Aa Aa
Right Connector	-

Figure 3.8 Extraction Pattern with Syntactical and Word Features

3.3 Named Entity Categorization Module

Named Entity Categorization module is the second module that is developed to perform the second task which is categorizing named entity candidates into three different predefined categories: date, person and location. This module is the implementation of one of semi-supervised machine learning algorithm known as Self-Training algorithm. Figure 3.9 shows the architecture of named entity categorization module.

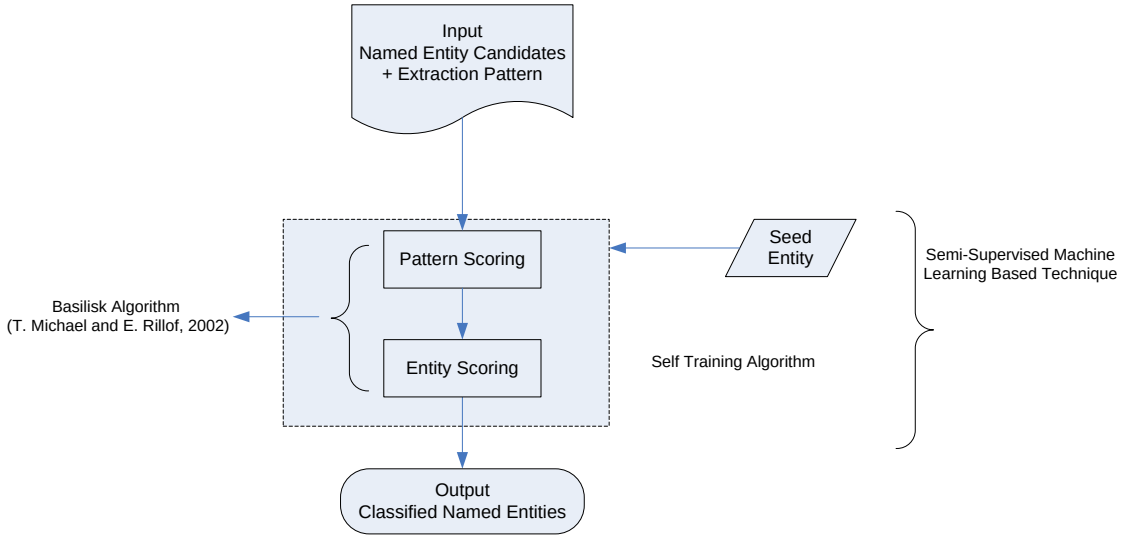

Figure 3.9 Architecture of Named Entity Categorization

This module takes named entity candidates and their extraction pattern as the input and a list of classified named entities is produced as the output. A small amount of labelled data called as *seed* is used to train a learner function *f* of the Self-Training algorithm [49]. Two scoring metrics from Basilisk algorithm [65] are used as the learner function *f*. The steps in Self-Training algorithm are explained as follow:

1. Train *f* from (X_l, Y_l)

   (X_l, Y_l) is the *seed* which contains a small number of labelled data. To the best of our knowledge, there exists no generally accepted value on the number of *seed* amount. We refer to the work on [79] which used 6% data from total number of the total identified named entity for each category as *seed*. A work in [65] which also applied Self-Training algorithm to build semantic lexicon on terrorism document used the most frequent word found as the *seed*. Considering the reported research works as a baseline, we collected 6% of the total identified entities from the most frequent named entity found as the *seed.*

2. Predict on $x \in X_u$

   The labelled data or seed then will be used to predict a label of $x \in X_u$ where X_u is unlabeled data. Two scoring metrics that have been utilized in Basilisk algorithm are used as the learner function *f*. The first metric is *RlogF* metric which is used for pattern scoring. Each extraction pattern will be assigned with *RlogF* score as calculated with equation 2.1. A pattern will gain a high score if it is strongly correlated with a specific category and could extract a large number of named entities that belong to that category. In our experiment, instead of taking *n* highest score pattern, a threshold value is used. We used value of *RlogF* as the standard of the pattern that could be stored in pattern pool. In order to avoid zero value of *RlogF*, we set a minimum value of F_i to 2. For minimum F_i we set N_i to 12, means that for each pattern, the maximum number of entity that doesn't exist in the *seed* is 10.

   All named entity candidates that were extracted by pattern in the pattern pool will then be assigned with another scoring metric called *AvgLog* as described in equation 2.2. If a named entity is extracted by patterns that have a tendency to extract members of a category then it will get a high score of *AvgLog*. For each iteration, we take 10% of highest *AvgLog* named entity identified as the growing size. Growing size is a number of predicted named entities $\langle x, f(x) \rangle$ in each iteration.

   As explained in Chapter 2, there are 3 possible methods to determine stopping criterion. In our work, we repeat the algorithm until the iterations reach convergence. However, to avoid too many iterations we set growing size to 10%. Meaning there will be more or less 10 iterations for each named entity.

3. Add $\langle x, f(x) \rangle$ to labelled data (seed)

   The next step is adding prediction result $\langle x, f(x) \rangle$ to labelled data. We take 50% best of $\langle x, f(x) \rangle$ and add it to the labelled data.

To the best of our knowledge, there exists no generally accepted value on the number of growing size and prediction result that should be added to labelled data. However, Steven P. Abney in [23] mentioned that there are variations on this. One of them is by putting a limit *k* on the number of prediction result added to labelled data.

4. Repeat

   The process is repeated again until no more named entities are found or the iterations reach convergence. From the second iteration forward, learner function $f : x \rightarrow y$ is re-trained on the larger labelled data.

3.4 Chapter Summary

In this chapter, we have described a very detail explanation on the methodology and approach used in this thesis. Two modules, Named Entity Identification and Named Entity Categorization are presented, including the detailed architectures of these two modules. In Section 3.2, explanation of the task of Named Entity Identification module is presented including determining entity boundary and constructing extraction pattern. There are three different extraction patterns presented: 1) extraction pattern with word feature, 2) extraction pattern with syntactical feature and 3) extraction pattern which used combination of word and syntactical feature. The second module, Named Entity Categorization is presented in Section 3.3. This section described how Self-Training algorithm and two scoring metrics from Basilisk algorithm are applied to separate the named entity candidates into three different categories.

CHAPTER 4

RESULT AND DISCUSSION

As stated in Chapter 1, this research aimed at identifying the performance of a semi-supervised machine learning technique taking into consideration only word and syntactical features against other types of machine learning techniques used in Named Entity Recognition (NER). This chapter starts with a detailed explanation on data collection, preparation and the evaluation metrics. This discussion is followed by a detailed presentation of results relating to each of the three existing NER systems used in the comparison study-which are LT-TTT2 from Language Technology Group (LTG) University of Edinburgh, NER system from NLP group of Stanford University and LingPipe NER system from Alias-i. The summary of this chapter is provided in the last section.

Our system runs on the Java Development Kit (JDK) 1.6. Stanford part-of-speech tagger in which a free java version of part-of speech tagger available in [105] is used to produce part-of-speech tagset. To parse the sentence and obtain the connector, we used JLinkGrammar, a java version of Link Grammar (LG) available in [106]. Normally, LG parser may generate more than one linkage for a sentence. In this case, JLinkGrammar only generates the best parse linkage; first linkage with the highest cost vector. In addition, we also utilized database MySQL 5.1. This database is used to save all the key tests, seeds and also final prediction made by the system.

4.1 Data Preparation

We conducted an experiment on accident news taken from the Reuters[3]. Our experimental data consists of 100 accident news documents that approximately contain 800 sentences or 19,000 words. The example of our data can be seen in the appendix. In order to create the key test and labelled data, the three types of entities in

[3] http://www.reuters.com

those documents were manually identified and annotated using standard guidelines from the MUC-7 Named Entity Task Definition [25]. A total of 246 date entities, 148 person entities and 595 location entities were identified. As the *seed* to initiate the training process, we collected 6% of the total identified entities from the most frequent named entity found in accordance to the specification set in [65],[79]. An example on how seed is chosen is explained as follows: Assume that there are 246 date entities in the text document. The first step is we sorted each named entity found and ranked them based on their frequency. And then we took 6% of the total identified entities (6% *246 = 15) from the most frequent named entity found. Table 4.1 shows the data set that will be used in our experiment.

Table 4.1 Data Set

Entity	Total entities in the corpus (A)	Number of seed (B=6% of A)	Number of entities that must be identified (C=A-B)
DATE	246	15	231
LOCATION	595	36	559
PERSON	148	9	139

4.2 Evaluation Metrics

There are several competition-based evaluations on NER area. Each evaluation has its own scoring method to measure the performance of NER system. Basically, in order to evaluate the performance of NER system, scoring method compares the output of the system (response) to the corresponding human generated answer key. Information Retrieval and Extraction Exercise (IREX) and Conference on Computational Natural Language Learning (CoNLL) share a simple evaluation method called *Exact Match Evaluation*. On *Exact Match Evaluation*, a named entity is considered correct only if it is exactly similar with the corresponding entity in the key test. There are three metrics used in this evaluation: precision, recall and F-Measure [6]. Those evaluation metrics are often used to measure the Information Retrieval (IR) system's

performance. As shown in the equation 4.1, recall is the percentage of named entity in the key test found by the system; precision is the percentage of correct named entity identified by the system which is shown in equation 4.2, while the F-Measure in the equation 4.3 is used to balance between recall and precision value. In NER, usually β is set to 1, which means recall is as important as precision [107].

$$recall = \frac{number\_of\_correct\_responses}{number\_of\_key\_test} \quad \textbf{(4.1)}$$

$$precision = \frac{number\_of\_correct\_responses}{number\_of\_responses} \quad \textbf{(4.2)}$$

$$F - Measure = \frac{(\beta^2 + 1) * recall * precision}{\beta^2 * recall + precision} \quad \textbf{(4.3)}$$

An example of how those three metrics are calculated is shown as follows: Assume that there are 100 named entities in the text document that need to be recoqnized (*number_of_key_test*). The NER system successfully recoqnized 80 named entities correctly (*number_of_correct_response*) and mistakenly recoqnized 40 named entities (wrong labelling i.e. give a person name label to the location entity or give named entity label to the non-named entity word). Then the precision, recall and F-measure can be calculated as follows:

$$recall = \frac{number\_of\_correct\_responses}{number\_of\_key\_test} = \frac{80}{100} = 0.8$$

$$precision = \frac{number\_of\_correct\_responses}{number\_of\_responses} = \frac{80}{80+40} = 0.67$$

$$F - Measure = \frac{(\beta^2 + 1) * recall * precision}{\beta^2 * recall + precision} = \frac{(1^2 + 1) * 0.8 * 0.67}{1^2 * 0.67 + 0.8} = 0.73$$

The value of recall, precision and F-measure can be expressed either in decimal or in percentage. In our thesis, we used percentage instead of decimal expression.

4.3 Proposed NER System Performance

In this subsection, the detailed performance results of the proposed NER system are presented. The aim of these experiments is to exploit the effectiveness of using syntactical features in classifying named entity. We conducted three different experiments with different extraction patterns to compare the performance. The scope of the comparison is focused on the word feature, syntactical feature and combination of the two. Each of the result is presented using tables and graphical charts.

4.3.1 NER with Word Feature

Our first experiment was conducted using extraction pattern constructed from word feature. A set of word feature including capitalization, punctuation, digit, part-of-speech and also the previous identified named entity are used to construct the extraction pattern. Table 4.2 shows the performance of our proposed NER system. From the table, it is found that the performance of the system is considered low. The system achieved considerable score for recall; it achieved 75% for the date entity, 65% for the location entity, and 62% for the person entity. However, it failed to maintain the precision score. As can be seen, the highest precision score is achieved for the date entity with 50% and as the result, the system only reached 47% on the average F-measure; except for the date entity that achieved higher score with 60% on F-measure. It is because date entity is easier to be recognized among others.

Table 4.2 Proposed NER result using extraction pattern constructed from word feature

Entity	Precision	Recall	F-Measure	Number of Iteration	Number of Pattern
DATE	50%	75%	60%	8	13
LOCATION	39%	65%	49%	13	45
PERSON	35%	62%	45%	13	5

Besides a set of word feature that was previously mentioned in Chapter 3, in this experiment we also utilized *seed* or labelled data to induce unidentified named entity. For example, if *Friday* is one of the *seed* members then all *Friday* in the document will be identified as the date entity. This step may reduce the number of iteration. However, it may also lead to a reduction performance rate when the labelled data contains incorrect prediction. Table 4.3 shows the fluctuation of precision, recall and F-measure score in each iteration for date entity. The graphical representation is shown in Figure 4.1. In the first iteration, the precision score is quite high, however as the process is repeated, this score decreased gradually from 75% to 50%. On the other hand, the recall score is slightly increased, totally only 14% from the first iteration. Extraction patterns that are too generic may contribute to this lower performance. This can be seen from the number of pattern obtained after the 8^{th} iteration. For instance, the system mistakenly predicted a series of cardinal number because it has similar pattern with the year number. The example is provided in Table 4.4.

Table 4.3 Date score using word feature

Iteration	Pattern	Precision	Recall	F-Measure
1	12	75.40%	61.03%	67.46%
2	12	75%	68.83%	71.78%
3	12	67.21%	70.99%	69.05%
4	12	59.35%	71.42%	64.83%
5	13	53.18%	72.29%	61.28%
6	13	50.29%	74.02%	59.89%
7	13	49.71%	74.89%	59.75%
8	13	49.71%	74.89%	59.75%

Table 4.4 Extraction pattern using word features

Example Statement	South Africa recorded 221 mine deaths last year, up from **200** in **2006**.	Remark
Extraction pattern with word features	00-[CD]	For both 200 and 2006

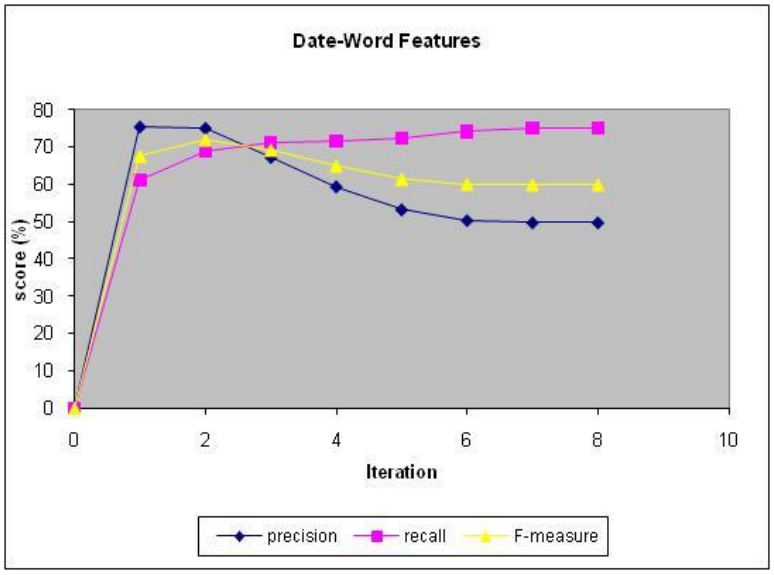

Figure 4.1 Date score using word feature

Unlike date entities, person entity showed a different graphical trend. Figure 4.3 compares the score between precision, recall and F-measure for the person entity. As explained before, for the date entity, the recall score started from around 61%. However, for the person entity, its recall on the 1st iteration is relatively low. It's only reached 14% with 53% on the precision. On the 13th iteration when the iteration reach convergence, recall increased sharply to 62% while the precision keep decreasing to 35%. For the person entity, we also utilized labelled data to induce unidentified named entity. The difference is that we didn't use full name to identify another person names, but only used the last name of person entity. This is based on our observation that in a news text if a person's name appeared more than once, on the second appearance forward, it will not be written in a full name but only the last name. An example of accident news from Reuters is shown in Figure 4.2. The person's name is shown in bold.

Texas refinery was trying to restart a giant industrial boiler when a catastrophic failure killed one worker & injured two others late on Friday, a company spokesman said on Saturday. Valero Energy Corp (VLO.N) spokesman **Bill Day** also said that the 245,000 barrel per day (bpd) refinery in Texas City, 50 miles (80 km) southeast of Houston, was currently operating at planned production levels. The boiler that failed was one of several providing power & steam at the refinery.

Tommy Manis, 40, of Alvin, Texas, died instantly when the boiler failed, **Day** told Reuters. Manis was part of a crew working on the boiler. Local media reports on Friday night said the boiler exploded, but **Day** said investigators were attempting to determine exactly what occurred. "There was definitely a loud noise" when the boiler failed, he said. "Our sympathies are with Mr. Manis' family," **Day** said. "It's a very sad event. For a company with 22,000 employees it's surprisingly tight-knit. These things reverberate throughout the Valero community."

Investigators from the U.S. Occupational Safety & Health Administration arrived at the refinery on Saturday morning to begin probing the accident. Of the two workers injured in the failure, one suffered cuts & another fell. Both men spent the night in a local hospital. One of the men is a Valero employee & the other works for an outside contractor doing work at the refinery.

The failed boiler was being restarted after it had shut down earlier on Friday, **Day** said. Boilers like the one that failed Friday night generate steam for use in the petroleum refining process. Friday's accident was the second fatality at Valero's Texas City refinery since the company bought the plant in 1998**.** The previous death was in 1998. There was no widespread release of hazardous chemicals in the accident, **Day** said.

Figure 4.2 Example of Accident News

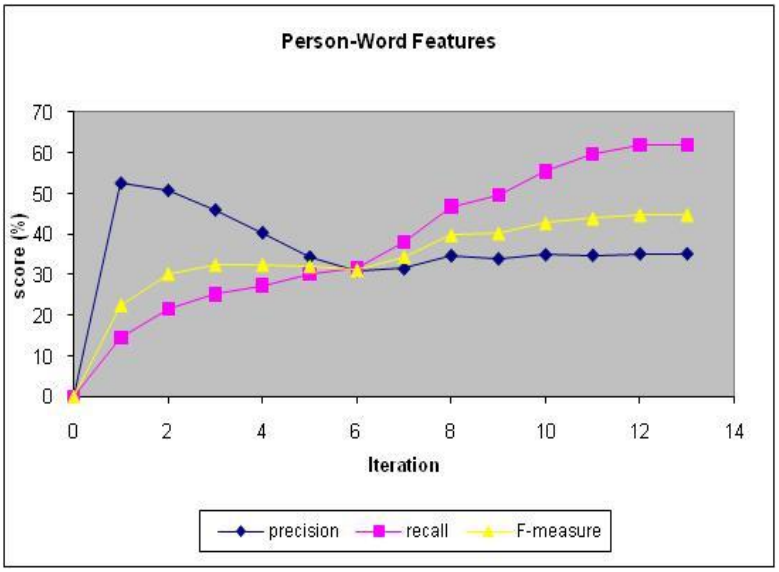

Figure 4.3 Person score using word feature

The result for location entity is not much different with the result of person entity. Figure 4.4 shows the score fluctuation of precision, recall and F-measure for location entity. During the 1st iteration, precision starts at 52% while 11% on recall. Precision then increased slightly on the 2nd iteration to 56% but decreased again on the 3rd iteration to 53%. Starting from the 4th iteration, precision keep fluctuated and dropped to 39% in the last iteration with 65% on the recall score. Again, the main factor contributed to this lower precision is the too generic extraction patterns. NER system mistakenly identified name of the day as the location entity, as it has similar pattern with a lot of location name. Consider this two following sentences:

Sentence 1: *"Ferry accidents kill hundreds of people in **Bangladesh** every year."*

Sentence 2: *"The failed boiler was being restarted after it had shut down earlier on **Friday**, Day said."*

"Bangladesh" and "Friday" have similar extraction pattern Aa-[NNP].

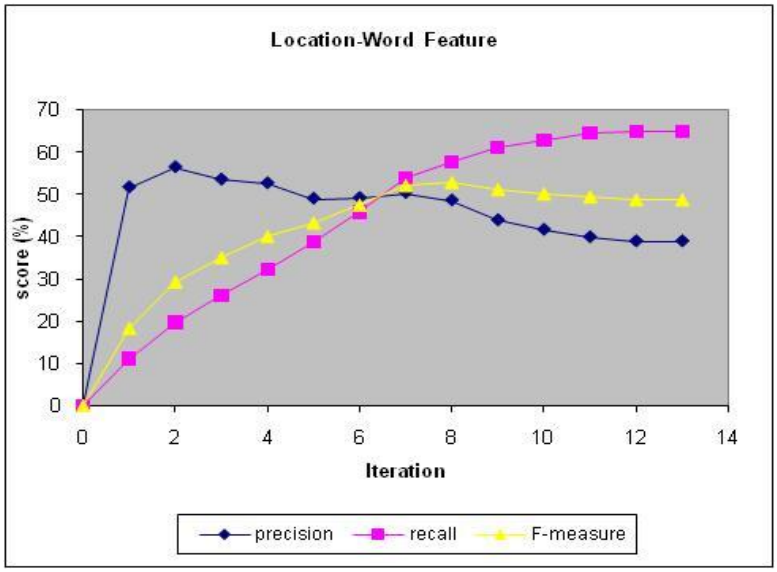

Figure 4.4 Location score using word feature

Based from the result, it can be concluded that extraction pattern using word feature is possible to recognize a considerable number of named entities. Although higher recall score were recorded for all the three tested entities-date, person and location, the precision result was lower due to the generic extraction pattern. In addition, the usage of labelled data to directly recognize another entity names is not effective enough. Early mistake made by the classifier due to the small number of seed tends to worsen the performance because in the next iteration, incorrect labelled data are used for the training.

4.3.2 NER with Syntactical Features

In the second experiment, we only utilized two features, which are part-of-speech and syntactical feature. This experiment aimed to evaluate how syntactical feature affects the system performance. In addition, through this experiment we would like to test connectors produced by the LG parser to be used as NER feature. As mentioned in the problem statement in Chapter 1, existing NER systems [18, 19] used restricted syntactical features but this method is found not to be applicable to every example of named entity. Therefore, in this experiment, rather than using specific syntactical feature like appositive modifier or preposition, we explored all types of syntactical

structures and let the Self-Training algorithm learnt from it. The following table shows the performance of proposed system using syntactical feature and part-of-speech as the extraction pattern.

Table 4.5 Proposed NER result using extraction pattern constructed from syntactical feature

Entity	Precision	Recall	F-Measure	Number of Iteration	Number of Pattern
DATE	83%	66%	74%	9	18
LOCATION	57%	59%	58%	12	55
PERSON	77%	48%	59%	9	11

Table 4.5 provide strong evidence that the usage of syntactical feature has influenced the precision and recall score. Precision is significantly increased; however on the other side recall has slightly dropped. Precision for the date and person entity successfully reached 83% and 77% but failed to reach the same recall score as in the first result in the Table 4.2. Unlike the date and person entity, precision and recall for the location entity increased only 18% from the 1st iteration while recall dropped slightly to 59%. The number of pattern obtained also shows significant difference between the first and second experiment. The number of pattern obtained has a strong relation with the rise of precision score. Syntactical feature is proven to be a more specific extraction patterns. LG provides a specific connector to each word in the sentence, thus it could differentiate two words that might have a similar word structure. This parameter certainly would help to rectify the problems in the previous experiment. Consider the example in Table 4.6, in which for the first experiment, the number "200" and "2006" has similar extraction pattern. The first extraction pattern only explores the part-of-speech and also the word structure without considering the position of the word in the sentence, while in the second extraction pattern, LG gives different connectors to both of the numbers based on their position in the sentence. The first number "200" is given left connector `Jp` and right connector `Mp`, while "2006" is given `IN` connector on the left. From those connectors it can be seen that those words have different position in the sentence and might be classified as different entity too.

Table 4.6 Extraction pattern using syntactical feature

Example statement	South Africa recorded 221 mine deaths last year, up from **200** in **2006**.	Remarks
Extraction pattern with word features	00-[CD]	For both 200 and 2006
Extraction pattern with syntactical feature from LG connectors	[IN][ ][CD][ ] [Jp][ ][CD][Mp]	For 2006 For 200

Figure 4.5 shows the graph of precision, recall and F-measure score in each iteration for the date entity and Table 4.7 provides the detailed score in each iteration including the numbers of patterns obtained. Starting from the 2nd iteration, the precision score fluctuated around 79% to 84%. Since then, it decreased slightly to 83% in the final iteration. The date entity has relatively high precision score, as LG provide a specific link for connecting date entities i.e. name of month, name of day, year number, etc with other words like preposition. In addition, LG also has reserved connectors for connecting between those date entities. The list of the LG connectors for date entity is provided in Table 3.2 in Chapter 3. Unfortunately, the available connectors are used to recognize common structure of dates thus this limitation is the evidence to the lower recall score for date entity i.e. 66%. Although we expect the combination of LG connectors and part-of-speech would increase the performance, surprisingly, date entities presented in text format such as the type of holidays-Christmas Eve, Muslim Eid, type of seasons-winter, summer, autumn are unrecognizable.

Table 4.7 Date score using syntactical feature

Iteration	Pattern	Precision	Recall	F-Measure
1	4	65.21%	6.49%	11.81%
2	5	82.97%	16.88%	28.05%
3	9	79.41%	23.37%	36.12%
4	10	79.34%	31.60%	45.20%
5	10	81.19%	41.12%	54.59%
6	13	82.14%	49.78%	61.99%
7	14	84.14%	59.74%	69.87%
8	18	82.60%	65.80%	73.25%
9	18	82.60%	65.80%	73.25%

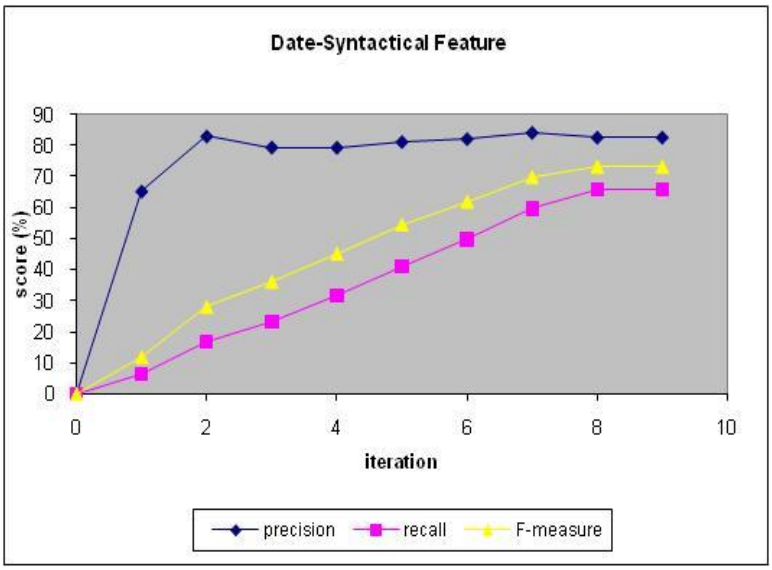

Figure 4.5 Date score using syntactical feature

Next experimental test was done for the location entity. The graph in Figure 4.6 shows that the precision score fluctuated until the 5th iteration and from the 6th iteration it started to decrease steadily. We identified a mistake on the entity boundary identification process that reduced the precision score. In some cases, LG parser gives incorrect connectors to the words. For example, according to MUC-7 Named Entity Identification guidelines [25], directional modifiers like north, south, east, west, etc are considered as a part of named entity if they are intrinsic part of location's official

names i.e. North Dakota. In this entity, LG is expected to assign G connector between words. This connector is used to link between two proper nouns, however for the entity "northern Germany" which appeared in our test data, LG has mistakenly assigned G connector between the word "northern" and "Germany" and tagged both words as one named entity. Furthermore, there are some words that LG failed to identify and left the word without any connector. In addition, we often found LG failed to identify locative designator in location name, such as the word "airport" that commonly followed an airport name.

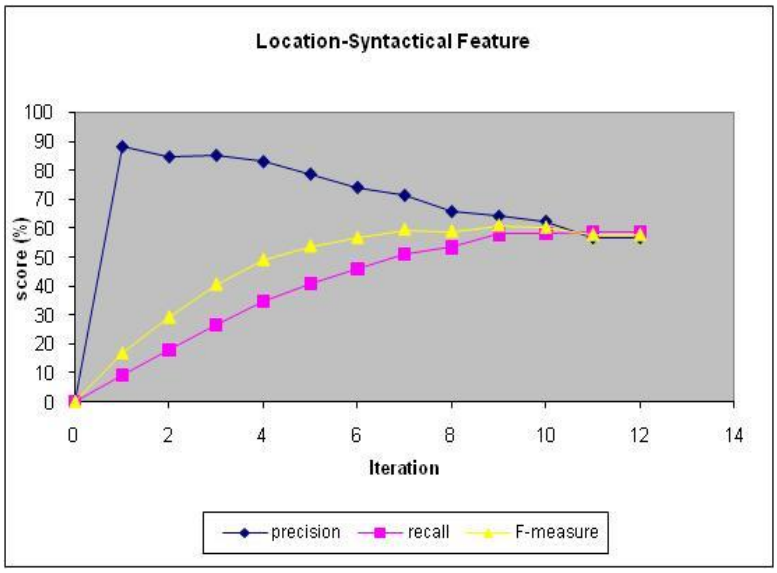

Figure 4.6 Location score using syntactical feature

The system performance for person entity can be seen in Figure 4.7. The graph shows that the system was able to identify the person entity with final precision around 77%. The precision score is considered as stable, starting from the first iteration until the final iteration; the precision decreased only 10% with a slight fluctuation.

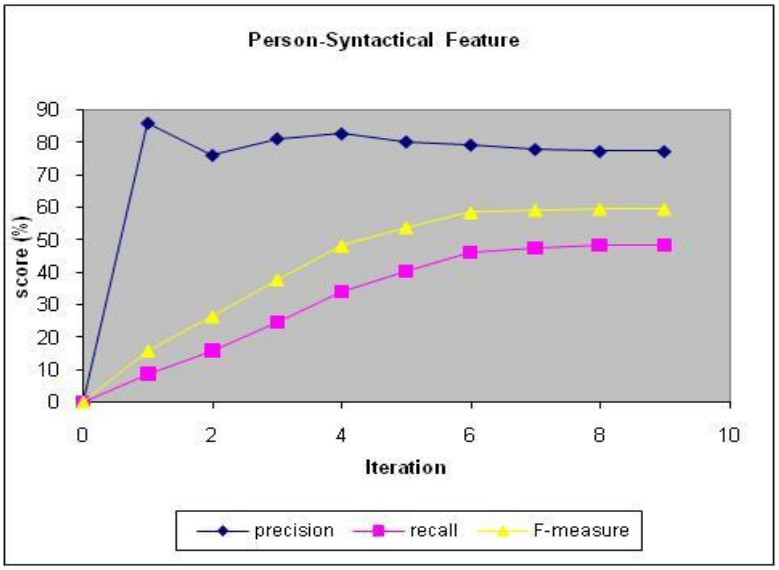

Figure 4.7 Person score using syntactical feature

There are 11 extraction patterns successfully generated after the last iteration. In the person name recognition process, several common syntactical features have been identified. Indirect speech is one of syntactical features that is commonly found in the accident news. Indirect speeches are used to report what witnesses have actually said. The names of witnesses are commonly available either in full name or only the last name, with or without the occupational title. Figure 4.8 shows the example of indirect sentence found in accident news. In this sentence, LG assigned `Ss` connector to the right of the word, `G` connector between the first and the last name ("Tom" and "Boughner") and `GN` connector to link between the last name ("Boughner") and the occupational title ("Sergeant").

"They were treated for minor burns injuries & were released from the hospital in good condition," Sergeant Tom Boughner told Reuters by telephone from the crash site.

Figure 4.8 Example of sentence with indirect speech

Moreover, other common syntactical feature that is used as indication of the person name is the appositive modifier. This feature is used to modify the nouns. In the accident news usually person name is modified using the appositive modifier. An example is illustrated in the Figure 4.9.

Last name is assigned Wd and GN connector on the left while Ss on the right. As mentioned in [22], Wd is used to link the main clause back to the LEFT-wall and nouns that have Wd- connector (in the left side) usually will have Ss connector in other side. In Figure 4.9, the appositive modifier is preceding the modified noun. In some sentences, appositive modifier might be found to follow the modified noun as in this example: "*Bruce Landsberg, president of the AOPA Air Safety Foundation, said in a statement that...".* In this sentence, the appositive modifier for "Bruce Landsberg" is the "president of the AOPA Air Safety Foundation".

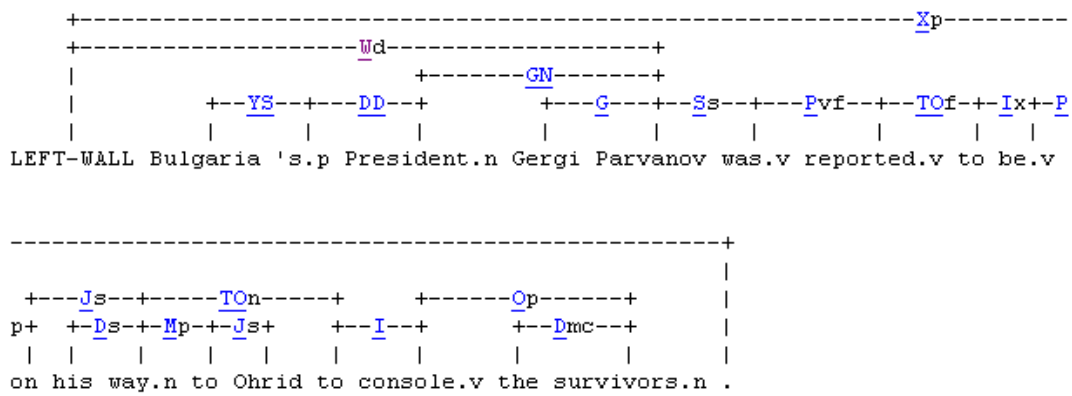

Figure 4.9 LG linkage on sentence with appositive modifier

Based from the above explanation, it can be concluded that by adding syntactical feature to the extraction pattern, it has significantly increased the system performance especially the precision. LG provides a different connector for each syntactical feature, and this has made the extraction pattern unique. In the beginning we have observed that the small number of seed used in the initial training iteration of the Self-Training algorithm contributed to the lower performance. However, due to the unavailability of extraction pattern, several named entities still can't be recognized.

4.3.3 NER with Syntactical and Word Features

The final experiment is to combine both word features and syntactical features. Our previous experimental result obviously showed that extraction pattern with word features has good coverage as evidence by the recall score. However, a lower precision score was recorded.

On the other hand, extraction pattern with syntactical features from LG parser would boost up the performance by increasing the precision score but not the recall score. Thus the idea to combine the word features and syntactical features to construct extraction pattern is expected to create a significant improvement. Table 4.8 shows the performance of the NER system which utilized syntactical and word features and Table 4.9 shows the results for all types of feature set used in our experimental setup.

Table 4.8 Proposed NER result using extraction pattern constructed from word and syntactical feature

Entity	**Precision**	**Recall**	**F-Measure**	**Number of Iteration**	**Number of Pattern**
DATE	84%	69%	76%	10	20
LOCATION	64%	56%	60%	13	59
PERSON	71%	51%	59%	6	25

Table 4.9 Performance result of proposed NER approach using three different feature set

	Word Features			**Syntactical Features**			**Word+Syntactical Features**		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
DATE	50%	75%	60%	83%	66%	74%	84%	69%	76%
LOC	39%	65%	49%	57%	59%	58%	64%	56%	60%
PER	35%	62%	45%	77%	48%	59%	71%	51%	59%

Note: P: Precision, R: Recall, F: F-measure

As expected, there is a slight increment for the date entity on both recall and precision score. Word feature is playing a role on the rise of the precision. Consider this two following sentences:

Sentence 1: *"Twenty-five people were killed & 130 injured last year when a Los Angeles commuter train collided with a freight train."*

Sentence 2: *"Suspects charged in connection with the 2007 attack are being tried in the north western city of Novgorod."*

Although "Twenty-five" in Sentence 1 and "2007" in Sentence 2 has similar syntactical feature and part-of-speech tagset, however, they have a different word pattern. Thus, when syntactical feature is combined with word feature, it would recognize "2007" as a date and leave "Twenty-five" untagged. The use of a small sized dictionary as a quick list look-up containing the name of months and days contributed to the slight increment of the recall score. This dictionary is useful when a pattern of named entity is not included in the pattern pool. An example of this case is illustrated in the following figure:

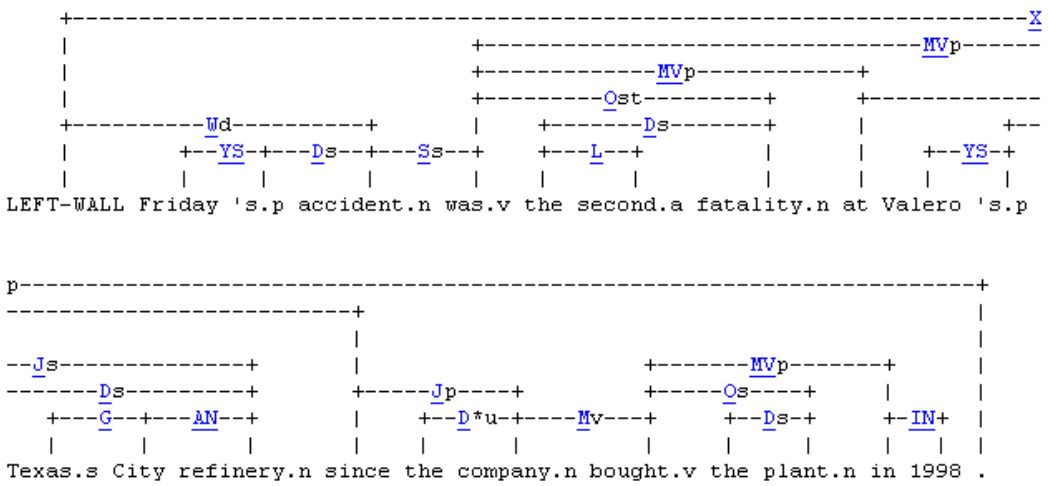

Figure 4.10 LG linkage

In Figure 4.10, manually it is easy to recognize "Friday" as a date entity. However, since the extraction pattern of that entity is rarely found, therefore it can't be included in the pattern pool and the named entity itself can't enter the named entity candidate pool. Thus, the dictionary takes its role here to solve such problem. Figure 4.11 shows the graphical representation of precision, recall and F-measure for the date entity. There is a significant increment to the precision score on the 2nd iteration. However, in the next iteration, the score decreased up to 79%. In the 3rd iteration onwards, the precision score is flattened off and reached 84% in the final iteration. Here, our system was battling with similar problem as in the previous experiments. Early mistake made by the classifier has caused the performance to worsen.

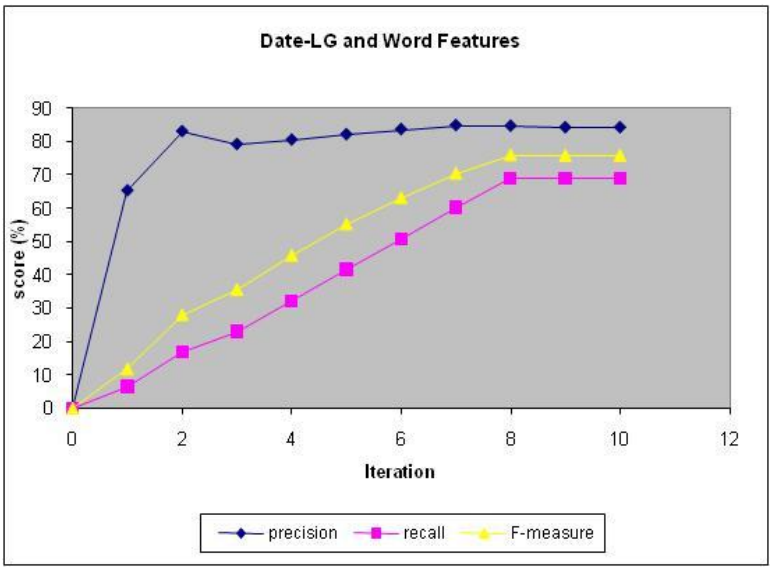

Figure 4.11 Date score using combination of the word and syntactical features

Combination between syntactical and word feature has influenced either the precision score or recall score. The addition of word feature has eliminated the error prediction by comparing the word structure of predicted entities with seed. This condition would increase the precision score but on other hand it decreased the recall score. An example of this case is illustrated in the following sentences.

Sentence 1: *The crane collapse in Houston was the deadliest U.S. crude oil refinery accident since a 2005 explosion at BP Plc's (BP.L) giant refinery in Texas City, Texas, killed 15 workers & injured 180 other people.*

Sentence 2: *"The Turks worked in Germany & were on their way to Turkey when the incident happened", police said.*

"BP" in Sentence 1 has a similar syntactical structure with "Germany" in the Sentence 2. LG parser assigns `Js` connector on the left side of both words. As a result, "BP" will be tagged as the location name when it is not. When word features is added to the feature set, it will differentiate both of those words through their word structure. NER system then tagged "Germany" as the location name and left "BP" untagged. Obviously, this step would increase the precision score but at the same time it also decreased the recall score. As an example, the word "U.S." in the following sentence: *"Many Western aircraft rely on U.S. made engines & part."* will be left

untagged since it has different word structure although it has similar syntactical structure as “Germany”.

Figure 4.12 shows the graphical representation of precision, recall and F-measure score for location entity. The precision starts at 89% but decreased sharply to 64% while recall only reach 56% in the final iteration. Even though the F-measure on this experiment is better than the second experiment, but the main problem remained unsolved. Incorrect prediction of seed in the initial iteration and incorrect entity boundary prediction are the source of the problems.

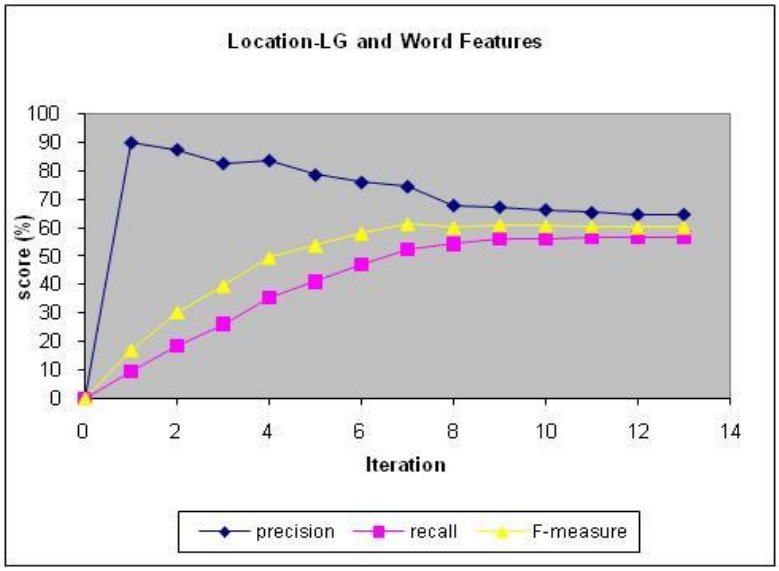

Figure 4.12 Location score using the combination of the word and syntactical features

The performance result for the person entity shows a different graphical trend as in Figure 4.13. As compared to the second experiment, the combination between syntactical and word features produced a slight increment on the recall score but a decrement on the precision score. We believe this is due to the usage of last name as the labelled person entity to recognize unlabelled entity. Using labelled person name does have an advantage. Those names with low *RlogF* or *AvgLog* score could enter the candidate pool. On other hand, as we didn’t rescore each entity then the incorrect prediction will be included in the candidate pool as well which would decrease the precision score. Figure 4.13 shows the performance result of the person entity. In the 1st iteration, the precision score achieved 96% which is considered good but decreased rapidly in the 4th iteration. While in the recall score no significant increment is noticed.

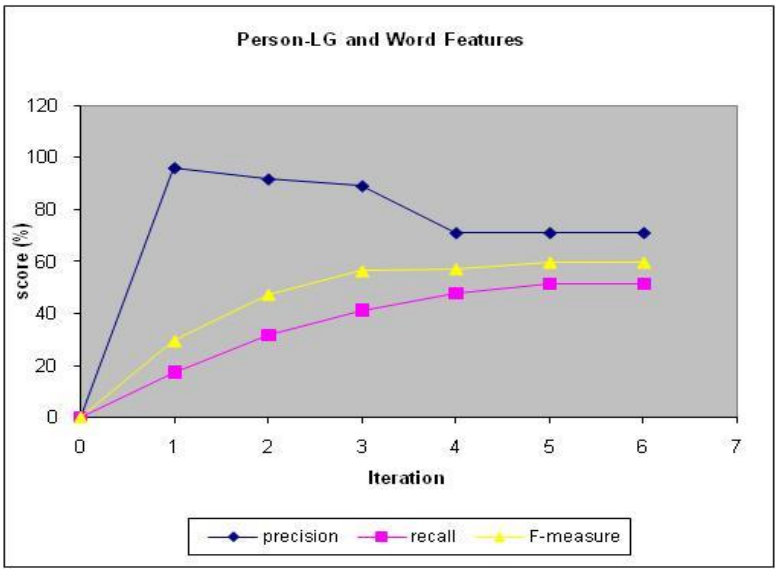

Figure 4.13 Person score using the combination of the word and syntactical features

Based on our experimental result, we proposed a combination of syntactical and word features to yield a better system performance. Hence, rescoring on each predicted entity is needed as a solution for incorrect prediction. We also present our result data in different graph representation. We provide another graph to compare each metric across method, i.e. precision for date entity using all method, recall for person entity using all method. The graphs can be seen in the appendix section.

4.4 Comparison with Other NER Systems

In this section we provide a comparison result between our proposed approach with three free NER systems we have identified: LT-TTT2, Stanford NER and LingPipe NER. Free or non-commercial systems are chosen as they provide full access to their codes and data for research purpose. These systems used different techniques and features as compared to our proposed NER approach. LT-TTT2 is a rule based NER system, while Stanford NER and LingPipe NER are both supervised NER with different classifier algorithm. To our best knowledge, there is no free semi-supervised NER system available to date. These three systems are chosen for benchmarking purposes. Moreover, by comparing the proposed approach with different systems there is an opportunity to explore potential improvement in order to gain similar or

better result. In the next paragraphs, we provide information on each of the NER system.

LT-TTT2

LT-TTT2 [108] is a rule-based NER system developed by Language Technology Group (LTG) University of Edinburgh. The system is available for download and has fee-free use (no modification) license. This system is composed from several modular tools. Each tool has specific function and it can be combined with other tools in UNIX pipeline. This system was implemented using an XML tools called LTXML2, also developed by LTG group. LTXML2 is a tool for XML manipulation. LT-TTT2 takes a plain text file as an input and produced an XML file as the output. There are 6 modular tools used on LT-TTT2 which are *preparetxt, tokenise, postag, lemmatise, nertag and chunk.* Each component does a specific job, as an example, *preparetxt* is used to convert plain text into basic XML format, *postag* component will add part-of-speech tag to each word, etc. The most important part of LT-TTT2 is *nertag* which is used to recognize and mark up several categories of named entity including numex (for money and percentage), timex (including dates and times), enamex (contains of persons, location and organizations) and miscellaneous entities which are not included in those three categories. Figure 4.14 shows the *nertag* pipeline.

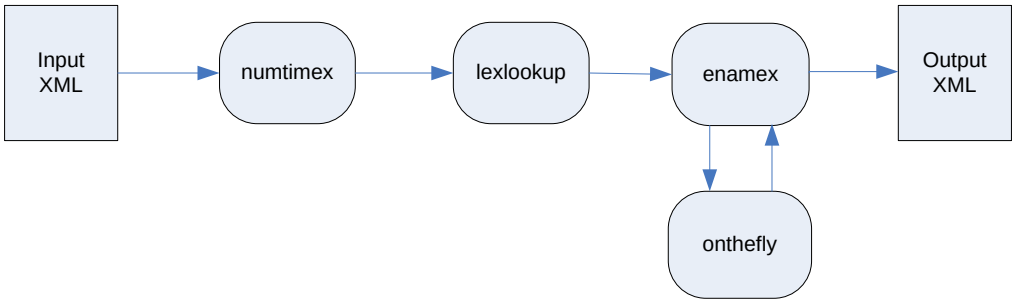

Figure 4.14 nertag pipeline [107]

The *nertag* component consists of three sub processes, *numtimex* to recognize numex and timex element, *lexlookup* to apply dictionary lookup for names and *enamex* to identify enamex elements. To identify those named entities, *nertag* used a list of grammatical rules and various kinds of lexicon. Each named entities element has different grammatical rules. The rule itself is constructed from a set of word features e.g. capitalization, digit, part-of-speech, etc. In total, there are more than 10

different lexicons used in the system including a list of possible times expression, organization names, female first names, male first names, list of country names etc. There are several research works which utilized LT-TTT2. For instance, works reported in [109, 110] used LT-TTT2 for recognizing named entities on the biomedical text. Another work used this NER system to identify named entities on the digitised historical text [111].

Stanford NER

The second system is Stanford NER [112] from NLP group of Stanford University. Stanford NER used one of statistical NLP models called Conditional Random Field (CRF). In addition, Stanford NER also employed Gibss Sampling, a simple Monte Carlo algorithm to solve the limitation of CRF model that only represents local structure. Stanford NER used a list of feature including current word, previous word, next word, current word character n-gram, current part-of-speech tag, surrounding part-of-speech tag sequence, current word shape, surrounding word shape sequence, presence of word in left window (size 4) and presence of word in right window (size 4). The system itself comes with two serialized models/ training data. The first model was trained on data from CoNLL, MUC-6, MUC-7, and ACE that can identify person, organization, and location entities. Moreover, this model was also trained on both US and UK newswire. The second model was trained on the CoNLL 2003 Shared Task training data that labels for person, organization, location and misc. For the purpose of comparison reported in this thesis, we choose the second model.

LingPipe

LingPipe [113] is a text processing tool that uses computational linguistic from *alias-i*. Besides being used for recognizing named entity, LingPipe also can be used for other purposes like classifying Twitter search result and suggesting correct spellings of queries. For recognizing named entities, LingPipe is applying supervised learning and also direct methods like regular expression matching and dictionary matching. LingPipe NER has been trained on MUC-6 corpus and it could extract name of people, locations and organizations in English news texts, and also biological named entities in biomedical text.

For our evaluation, we used the demo version of LingPipe named entity recognition with first-best output. Each named entity will be marked with an ENAMEX element with the TYPE indicating the category of named entity. Basically, LingPipe provides three trainable chunkers that can be used as name entity recognizer which are CharLmHmmChunker, CharLmRescoringChunker and TokenShapeChunker. Each chunker used different method and resulted different performance too. Here, we used Char LmHmmChunker which utilizes Hidden Markov Model (HMM) to handle the tagging part. Among others, this chunker is the simplest and has good recall but least accurate.

The comparison summary between our approach and the three named entity recognizer is available in the Table 4.10. Five factors are identified, studied and presented in this table: the technique, feature, training data, identified entity and also the license.

Table 4.10 NER Systems comparison summary

Parameter	**LTG Edinburgh**	**Stanford NER**	**LingPipe (Demo Version)**	**Proposed Approach**
Technique	Rule based NER	Supervised learning with CRF and Gibbs sampling algorithm	Supervised learning with HMM algorithm Dictionary matching Regular expression matching	Semi supervised technique (with Self -Training and classifier from Basilisk Algorithm)
Features	Totally used more than 10 dictionaries lookup, Used a list of grammatical rules which are constructed from a set of word features	Current , previous and next word, current word character n-gram, current POS tag, Surrounding POS tag sequence, current word shape, surrounding word shape sequence, presence of word in left and right window	Not described	Syntactic and word features
Training Data	No training	CoNLL 2003 Shared Task training data	MUC-6 data corpus	6 percent from total testing data set

Identified Entities	Money, percentage, date, time, person, location, organisation and miscellaneous entities which are not included in those categories.	Person, location and organisation	Person, location and organisation	Date, Person and location
License	Fee-Free Use (no modifications) License	Publicly available	Free demo	-

Our final experiment tested the precision, recall and F-measure for each NER systems for the three entities: date, location and person. The result is presented in Table 4.11. Each NER system has to recognize and classify a total of 231 date entities, 559 location entities and 139 person entities from 100 accident news documents. These are the same documents previously used to test our proposed NER approach. However, Stanford NER and LingPipe don't include date entity as one of the recognized entity. They only recognized three entities: person, location and organization.

Table 4.11 Result of three available NER systems and proposed system

	LT-TTT2			**Stanford NER**			**LingPipe**			**Proposed Approach**		
	P	R	F	P	R	F	P	R	F	P	R	F
Date	87%	93%	90%	-	-	-	-	-	-	84%	69%	76%
Location	76%	76%	76%	75%	75%	75%	59%	55%	57%	64%	56%	60%
Person	74%	76%	75%	74%	88%	81%	44%	69%	54%	71%	51%	59%
Average F-measure			**80%**			**78%**			**56%**			**65%**

Note: P: Precision; R: Recall; F: F-measure

For the date entity, LT-TTT2 outperforms our proposed NER approach with 90% on F-measure while ours only gained 76%. However, the precision score of our approach is almost similar to LT-TTT2. The recall score of 93% for LT-TTT2 shows that it has a very good coverage to recognize several date entity that our approach have missed. For example, LT-TTT2 correctly tagged "late on Thursday" as one date entity while our approach tagged only the word "Thursday" as date entity. We believe that is due to the detailed grammatical rules LT-TTT2 has. In addition, its huge dictionary list also played a significant role, as it could recognize all season names and even an abbreviation of the day names. However, in some cases we found that LT-TTT2 also made a mistake by tagging non date entity. For example, in the following sentence:

"The cause of the accident, which took place at about 1100 a.m. (0900 GMT), is not yet known", a police spokesman said.

It was found that LT-TTT2 often mistakenly tagged a cardinal number of time as a year for example word "1100" on the above sentence. While using our approach, at the extraction pattern construction, LG parser will give an `ND` connector between number ("1100") and its numerical determiner ("a.m."). As a year is rarely found with numerical determiner, then that extraction pattern will not be included in the pattern pool and that word is left untagged.

In the location entity, among others, LT-TTT2 again obtained the highest performance result with 76% on the F-measure. Stanford NER follows with 75% while our proposed approach successfully outperforms LingPipe with 60% on the F-measure. On recognizing location entity, in some cases it was found that LT-TTT2 and Stanford NER did a similar mistake. For instance, they failed to include locative designators e.g. airport, mountains, province as the location entity when those words is started with lower case. On the other hand, LingPipe has almost similar recall score as our proposed approach. Unfortunately, LingPipe has incorrectly tagged many entities that affected its precision score. For example it often tagged adjectival forms of location names like "German" and "Indonesian" as location entity.

Stanford NER achieved the highest score for the person entity with 81% on the F-measure. It successfully identified 88% of all person entity in the text. While LT-TTT2 follows with 75% and LingPipe with 54% on the F-measure. Our proposed NER approach outperforms LingPipe and reaches higher precision score. Mainly, all the NER systems including our approach did a similar mistake i.e. mistakenly tagged location names as the person entity.

In conclusion, basically our proposed NER approach has almost similar precision with either LT-TTT2 or Stanford NER. However, the main problem of our approach remained i.e. less coverage of extraction pattern that affected the recall score. Among others, the rule based NER LT-TTT2 produced the best result performance with 80% on average F-measure. This fact is consistent with several research works [8, 46] which claimed rule based NER has performed better over the other methods. A very

large dictionary and also a grammatical rule created by linguist experts are the supporting evidence why this technique maintains its outstanding performance. On the other hand, Stanford NER which uses a supervised learning approach could yield almost similar performance as the rule based NER. Surprisingly, for person entity it outperforms the rule based technique. A powerful CRF algorithm as the classifier and a set of word features give a significant influence to the performance result. A large set of training data has made the learning successful.

However, LingPipe which is also a supervised NER failed to reach the same performance result as Stanford NER. From our observation, there is no exact reason why both of these supervised NERs producing contrasting result. However, several NLP works [114-116] showed that CRF produces better performance than HMM. Surprisingly, our proposed NER approach which used semi-supervised learning has outperformed LingPipe with an overall F-measure of approximately 9%. Our proposed approach has a better precision score as compared to LingPipe despite of the lower recall score. The overall result of our proposed approach has proven that the combination of semi-supervised algorithm with word and syntactical features from LG parser has a great potential to gain similar or even a better result than rule based or supervised learning. This combination seems to constitute a good mechanism, which overcomes some of the weaknesses of available NER systems. Semi-supervised learning comes with its simplicity, less training data and offers a considerable result. On the other hand, LG parser provides a comprehensive syntactical feature through its connectors. This connector can overcome the shortcomings of previous research works that used a limited syntactical feature. These experimental results help us to achieve our objectives as stated in Chapter 1.

4.5 Chapter Summary

In this chapter, we presented our proposed NER approach experimental results with detailed analysis on it. We started by describing our data set in section 4.1 and the evaluation mechanism used in section 4.2. In section 4.3, three experiment results are provided.

In this section we aimed to show the effectiveness of adding syntactical features from LG parser into NER feature set. From those experiments, we can draw a conclusion that syntactical feature is very effective to be used as NER feature. In addition, we also realized that connector from LG parser has the potential to be utilized either as feature set to construct extraction pattern or determine entity boundary. In the section 4.4 we compared our performance result with three available NER systems which utilize different techniques as compared to our approach. From the comparison, we consider that our semi-supervised approach is promising even though it only uses two features: syntactical and word features.

CHAPTER 5

CONCLUSIONS AND FUTERE WORKS

5.1 Conclusion

The thesis has shown that our main objectives are fulfilled; to evaluate NER performance by applying syntactical structure from Link Grammar as the NER feature. In addition, three supporting objectives have successfully address the existing problem in previous NER works and also offer an alternative feature set to create an extraction pattern. We also make a claim of three major contributions that have the potential to be used in other NER systems. First, we demonstrate the usage of connector from LG parser to identify named entity candidates and determine the entity boundary (Chapter 3, section 3.2.1). Second, we present an extraction pattern construction from LG connector and a set of word features (Chapter 3, section 3.2.2). Finally, our third contribution is a semi-supervised Named Entity Recognition (NER) for accident domain (Chapter 3, section 3.3). Benchmarked against three available NER systems, our proposed NER approach successfully outperforms one of the three systems and produced almost similar precision score compared with the other two (Chapter 4). Due to the fact that there exist no currently available NER work on the accident domain, this thesis comes as the first work that explores that domain. We have shown that accident domain (in this case news documents) is different with other domain like business. Thus it needs a different approach to process that document. Although our proposed NER approach is tested on accident documents, but there is a possibility to make it as an adaptive system in the future.

We have explored a set of connector provided by Link Grammar (LG) parser. With its capability to capture approximately seven hundred definitions ranging from noun-verb agreement, question, imperatives, complex to irregular verbs and many more, made LG as a comprehensive syntactical feature for NER system. LG assigns

different connector to each word in the sentence based on the syntactical structure of the word. Syntactical structure from LG parser has been found capable of providing sufficient evidence to identify a sort of information inside a sentence. Other than that, linkage pattern from LG parser also can also be used to resolve co-reference problem.

Furthermore, this thesis has supported previous NER research works [56-59] which claimed that semi-supervised technique can be considered as the promising method for NER system. The experiment result shows that the performance of semi-supervised NER is comparable to the other techniques. Semi-supervised learning as one of machine learning technique is a solution for the problems found in rule based technique i.e. the difficulties to construct complete dictionary and create a detailed grammatical rules. In addition, when training data is difficult to be obtained, semi-supervised learning is the right method to be applied. Self-Training algorithm as one of semi-supervised technique is a simple yet a powerful algorithm. Self-Training algorithm offers a simple procedure without changing or reducing the performance of learner function used.

5.2 Limitations

From the experimental result, several problems remain as the limitation of our proposed NER approach. We identify there are two main problems that made the performance result low. The first problem is the unavailability of extraction pattern that made our system can't recognize several named entities. This problem is happened due to limited NER feature used in our extraction pattern. As explained in the previous chapter, our extraction pattern only used a small sized of word features and also syntactical features from LG parser. The usage of syntactical features has been proven could boost the system performance. However, in some cases it is found that LG failed to recognise some words or mistakenly assign wrong connector.

The second problem is found in the Semi-Supervised learning part. In the beginning we have observed that the small number of seed used in the initial training iteration of the Self-Training algorithm contributed to the lower performance. The limitation of Machine Learning technique is also found here. The system has

difficulty to detect complex entities that never found before in the training data. In the next section, we describe two future works that might be potential to resolve the limitation of our system.

5.3 Future Works

We believe that the limitations presented in the previous section are potential topics to be further explored. We identified two interesting research areas that could provide further extension to the proposed work in order to better improve the NER performance result.

5.3.1 Employing More NER Features

Our thesis has proven that syntactical and word features are two effective features to be used as NER features. Word feature which describes the structure of a word has the capability to identify named entities with higher coverage. On the other hand, LG connectors as the syntactical features made the extraction pattern unique and resulted on the higher precision. However, in our thesis we only employed limited word features including capitalization, punctuation, digit, part-of-speech and also the previous identified named entity. We believe that this limited word features influence the system performance. In addition, from the comparison result, it can be seen that Stanford NER that employs a large number of word features could achieve a better performance than ours. However, as reflected in the CoNLL-2003 result, the choice of the NER features is very important. Because of that, we come with the conclusion that one way to get a higher NER performance is by employing a greater number of NER features that suit best with the domain.

5.3.2 Hybrid System

As mentioned in Chapter 2, supervised learning has shown very good performance result. However rule-based system still surpassed the performance of supervised learning especially on the specific domain.

The strong point of Supervised Learning is that it can produce accurate prediction and give true label for unlabeled data, but this condition can be achieved if it is given enough training data. The common problems are labelled data is not available in great quantity and generating labelled data can be very expensive [23]. In addition, manual creation of rules is laboured intensive and time consuming. Moreover, the fact that a complete dictionary is difficult to obtain also becomes the limitation of this technique. Recently, there has been a proposal to construct a new approach by combining machine learning and rule based approach. The approach which is called a Hybrid NER is expected to solve limitations found in those two approaches by combining the strongest point in each method [46].

Hybrid NER is able to recognize the named entities using a list of grammatical rules, list of dictionary and also employing the machine learning technique at the same time. A research work on [117] is an example of Hybrid NER system. First, it used a list of sure-fire rules that consist of combination between internal and external evidence of named entity. After that, the system performs a probabilistic partial match of the identified entities. In this step, a pre-trained maximum entropy model is used. The process does not end up here; some other sure-fire rules are applied again. This NER approach has resulted a very good performance both on precision and recall. When machine learning technique is applied, there might be named entity that can't be recognized due to the small training data. As such, grammatical rules and dictionary list will help the NER system to identify the unlabelled named entities. More over, the NER system doesn't have to utilize a big sized of dictionary list and grammatical rules, since it is only to complement the machine learning technique. We believe that by applying Hybrid NER approach especially in accident domain could improve the system performance.

REFERENCES

[1] National Transportation Safety Board. (2002, September 18). *NTSB Aviation Accident Database.* [Online]. Available: http://www.ntsb.gov.

[2] R. Grishman, "Information Extraction: Techniques and Challenges," in *Proc. SCIE '97 International Summer School on Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology*, Frascati, Italy, 1997, pp. 10-27.

[3] E. Marsh, "TIPSTER information extraction evaluation: the MUC-7 workshop," in Proc. of The MUC-7 Workshop, 1998©Association for Computational Linguistics. doi:10.3115/1119089.1119092.

[4] R. Grishman and B. Sundheim, "Message Understanding Conf.- 6: A Brief History," in *Proc. of International Conf. on Computational Linguistic*, Copenhagen, Denmark, 1996, pp. 466-471.

[5] S. Sekine, K. Sudo, and C. Nobata, "Extended named entity hierarchy," in *Proc. of The Third International Conf. on Language Resource and Evaluation*, Canary Island, Spain, 2002.

[6] G. Wei, "Named Entity Recognition and An Application to Document Clustering," M.S. thesis, Dalhousie Univ., Halifax, Nova Scotia, 2004.

[7] R. Grishman, "Event Extraction: Learning from Corpora,", NYU, July, 2003.

[8] D. Appelt and D. Israel, "Introduction to Information Extraction Technology," presented at International Joint Conf. on Artificial Intelligent, Stockholm, Sweden, 1999.

[9] D. Nadeau, "Semi-Supervised Named Entity Recognition: Learning to Recognize 100 Entity Types with Little Supervision," Ph.D. dissertation, Dept. Computer Science, Univ. of Ottawa, Ottawa, Canada, 2007.

[10] E. F. T. K. Sang and F. D. Meulder, "Introduction to the CoNLL-2003 shared task: language-independent named entity recognition," in Proc. of The Seventh Conf. on Natural language learning at HLT-NAACL 2003, 2003©Association for Computational Linguistics. doi: 10.3115/1119176.1119195.

[11] National Transportation Safety Board. (2009). *NATIONAL TRANSPORTATION SAFETY BOARD NTSB Form 6120.1 PILOT/OPERATOR AIRCRAFT ACCIDENT/INCIDENT REPORT*. [Online]. Available: http://www3.ntsb.gov/aviation/6120_1web.pdf.

[12] Occupational Safety and Health Administration. (2001, January 19). *Reporting fatalities and multiple hospitalization incidents to OSHA*. [Online]. Available:

http://www.osha.gov/pls/oshaweb/owadisp.show_document?p_table=standards&p_id=12783

[13] A. Orecchioni, N. Wiratunga, S. Massie, S. Chakraborti, and R. Mukras, "Learning Incident Causes," in *4th Textual CBR Workshop,* 2007.

[14] T. Ekman and A. Nilsson, "Identifying Collision in NTSB Accident Summary Reports," Department of Computer Science, Lund University, Sweden, June 13 2002.

[15] S. Dupuy, A. Egges, V. Legendre, and P. Nugues, "Generating A 3D Simulation of A Car Accident from A Written Description in Natural Language : The CarSim System " in *Proc. of the Workshop on Temporal and spatial information processing,* Toulouse, France, 2001, pp. 1-8.

[16] R. Yangarber, W. Lin, and R. Grishman, "Unsupervised learning of generalized names," in *Proc. of The 19th International Conf. on Computational linguistics,* Taipei, Taiwan, 2002, pp. 1-7.

[17] A. Cucchiarelli and P. Velardi, "Unsupervised named entity recognition using syntactic and semantic contextual evidence," *Journal Computational Linguistics,* vol. 27, pp. 123-131, 2001.

[18] M. Behrang and H. Rebecca, "Syntax-based semi-supervised named entity tagging," in *Proc. of the ACL 2005 on Interactive poster and demonstration sessions,* Ann Arbor, Michigan, USA, 2005, pp. 57-60.

[19] M. Collins and Y. Singer, "Unsupervised models for named entity classification," in *Proc. of the Joint SIGDAT Conf. on Empirical Methods in Natural Language Processing and Very Large Corpora,* Maryland, USA, 1999, pp. 100-110.

[20] D. D. Sleator and D. Temperley, "Parsing english with a link grammar," Dept. of Computer Science, Carnegie Mellon Univ., Tech. Rep. CMU-CS-91-196, 1991.

[21] I. Muslea, "Extraction patterns for information extraction tasks: A survey," in *Proc. of the AAAI-99 Workshop on Machine Learning for Information Extraction,* Orlando, Florida, USA, 1999, pp. 1-6.

[22] D. D. Sleator. (2004, December). *Link Grammar*. [Online]. Available: http://www.link.cs.cmu.edu/link/

[23] S. P. Abney, "Semisupervised Learning for Computational Linguistic," in *Computer Science and Data Analysis Series,* London, UK: Chapman & Hall/CRC, 2007.

[24] M. Dzirutwe. (2009, August 2). *Zimbabwe bus crash kills 33 people –report*. [Online]. Available:http://uk.reuters.com/article/2009/08/02/uk-zimbabwe-accident idUKTRE5710RB20090802

[25] N. Chinchor, "MUC-7 Named Entity Task Definition," In *Proc. of the 7th Message Understanding Conference,* Fairfax, Virginia, 1998.

[26] C. Lee, Y.-G. Hwang, and M.-G. Jang, "Fine-grained named entity recognition and relation extraction for question answering," in *Proc. of the 30th annual international ACM SIGIR Conf. on Research and development in information retrieval,* Amsterdam, The Netherlands, 2007, pp. 799-800.

[27] A. Toral, E. Noguera, F. Llopis, and R. Muñoz, "Improving Question Answering Using Named Entity Recognition," in *Natural Language Processing and Information Systems*, Salford, UK: SpringerLink, 2005, pp. 181-191.

[28] D. Moll, M. van Zaanen, and D. Smith, "Named entity recognition for question answering," in *Proc. of the 2006 Australasian Language Technology Workshop*, Sydney, Australia, 2006, pp. 51-58.

[29] J. M. G. Hidalgo, F. C. Garcia, and E. P. Sanz, "Named Entity Recognition for Web Content Filtering," in *Natural Language Processing and Information Systems*: Salford, UK: SpringerLink, 2005, pp. 286-297.

[30] F. Paradis and J.-Y. Nie, "Filtering Contents with Bigrams and Named Entities to Improve Text Classification," in *Proc. of Asia Information Retrieval Symposium*, Jeju Island, Korea, 2005, pp. 135-146.

[31] B. Babych and A. Hartley, "Improving machine translation quality with automatic named entity recognition," in *Proc. of the 7th International EAMT workshop on MT and other Language Technology Tools: Improving MT through other Language Technology Tools: Resources and Tools for Building MT*, Budapest, Hungary, 2003, pp. 1-8.

[32] N. Kok Wah, F. S. Tsai, C. Lihui, and G. Kiat Chong, "Novelty detection for text documents using named entity recognition," in *6th International Conf. on Information, Communications & Signal Processing*, Singapore, 2007, pp. 1-5.

[33] D. Shen, J. Zhang, G. Zhou, J. Su, and C.-L. Tan, "Effective adaptation of a Hidden Markov Model-based named entity recognizer for biomedical domain," in *Proc. of the ACL 2003 workshop on Natural language processing in biomedicine*, Sapporo, Japan, 2003, pp. 49-56.

[34] S. Burr, "Biomedical named entity recognition using conditional random fields and rich feature sets," in *Proc. of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications*, Geneva, Switzerland, 2004, pp. 104-107.

[35] T. Yoshimasa and T. Jun'ichi, "Boosting precision and recall of dictionary-based protein name recognition," in *Proc. of the ACL 2003 workshop on Natural language processing in biomedicine*, Sapporo, Japan, 2003, pp. 41-48.

[36] T. Ohta, Y. Tateisi, and J.-D. Kim, "Genia corpus: an annotated research abstract corpus in molecular biology domain," in *Proc. of the Second International Conf. on Human Language Technology Research*, San Diego CA, USA, 2002, pp. 82-86.

[37] S. Pyysalo et al., "BioInfer: a corpus for information extraction in the biomedical domain," *BMC Bioinformatics*, vol. 8, no. 1, pp. 50, 2007.

[38] N. Chinchor, "Overview of MUC-7/MET-2," in *Proc. of the 7th Message Understanding Conf.*, Fairfax, VA, USA, 1998.

[39] S. Sekine and Y. Eriguchi, "Japanese Named Entity Extraction Evaluation - Analysis of Results," in *Proc. of the 18th International National Conf. on Computational Linguistics*, Saarbrucken, Germany, 2000, pp. 1106-1110.

[40] E. F. T. K. Sang, "Introduction to the CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition," in *Proc. of The 6th Conf. on Natural Language Learning*, Taipei, Taiwan, 2002, pp. 1-4.

[41] R. Malouf, "Markov models for language-independent named entity recognition," in *Proc. of the 6th Conf. on Natural language learning*, Taipei, Taiwan, 2002, pp. 1-4.

[42] W. J. Black and A. Vasilakopoulos, "Language independent named entity classification by modified transformation-based learning and by decision tree induction," in *Proc. of the 6th Conf. on Natural language learning*, Taipei, Taiwan, 2002, pp. 1-4.

[43] A. McCallum and W. Li, "Early results for named entity recognition with conditional random fields, feature induction and Web-enhanced lexicons," in *Proc. of the seventh Conf. on Natural language learning at HLT-NAACL 2003*, Edmonton, Canada, 2003, pp. 188-191.

[44] P. McNamee and J. Mayfield, "Entity extraction without language-specific resources," in *Proc. of the 6th Conf. on Natural language*, Taipei, Taiwan, 2002, pp. 1-4.

[45] G. Doddington, A. Mitchell, M. Przybocki, L. Ramshaw, S. Strassel, and R. Weischedel, "The Automatic Content Extraction (ACE) Program Tasks, Data, and Evaluation," in *Proc. of LREC*, Lisbon, Portugal, 2004, pp. 837-840.

[46] A. Mansouri, L. Suriani Affendey, and A. Mamat, "Name Entity Recognition Approach," *International Journal of Computer Science and Network Security*, vol. 8, no.2, pp. 339-344, February 2008.

[47] C. Nedellec, "Machine Learning for Information Extraction," Inference and Learning Group Universite de Paris.

[48] E. Riloff, "Automatically constructing a dictionary for information extraction tasks," in *Proc. of The 11th National Conf. on Artificial Intelligence*, Washington D.C, USA, 1993, pp. 811-816.

[49] X. Zhu and A. B. Goldberg, "Introduction to Semi-Supervised Learning," in *Synthesis Lectures on Artificial Intelligence and Machine Learning*, San Rafael, CA :Morgan & Claypool Publishers, 2009, vol. 3, pp. 1-130.

[50] D. M. Bikel, S. Miller, R. Schwartz, and R. Weischedel, "Nymble: a high-performance learning name-finder," in *Proc. of the Fifth Conf. on Applied Natural Language Processing*, Washington, D.C., USA, 1997, pp. 194-201.

[51] G. Zhou and J. Su, "Named entity recognition using an HMM-based chunk tagger," in *Proc. of the 40th Annual Meeting on Association for Computational Linguistics*, Philadelphia, PA, USA, 2002, pp. 473-480.

[52] D. Shen, J. Zhang, G. Zhou, J. Su, and C.-L. Tan, "Effective adaptation of a Hidden Markov Model-based named entity recognizer for biomedical domain," in *Proc. of the ACL 2003 workshop on Natural language processing in biomedicine*, Sapporo, Japan, 2003, pp. 49-56.

[53] G. Paliouras, V. Karkaletsis, G. Petasis, and C. D. Spyropoulos, "Learning Decision Trees for Named-Entity Recognition and Classification." In *Proc. of*

the 14th European Conference on Artifical Intelligent, Berlin, Germany, 2000.

[54] H. Isozaki, "Japanese named entity recognition based on a simple rule generator and decision tree learning," in *Proc. of the 39th Annual Meeting on Association for Computational Linguistics*, Toulouse, France, 2001, pp. 314-321.

[55] Y. Shinyama and S. Sekine, "Named entity discovery using comparable news articles," in *Proc. of the 20th international Conf. on Computational Linguistics*, Geneva, Switzerland, 2004, Art. No. 848.

[56] B. Wang, B. Spencer, C. X. Ling, and H. Zhang, "Semi-supervised self-training for sentence subjectivity classification," in *Proc. of the Canadian Society for computational studies of intelligence, 21st Conf. on Advances in artificial intelligence*, Windsor, Canada, 2008, pp. 344-355.

[57] B. Plank, "A comparison of structural correspondence learning and self-training for discriminative parse selection," in *Proc. of the NAACL HLT 2009 Workshop on Semi-Supervised Learning for Natural Language Processing*, Boulder, Colorado, 2009, pp. 37-42.

[58] Z. Kozareva, "Bootstrapping Named Entity Recognition with Automatically Generated Gazetteer Lists," in *Proc. of the Eleventh Conf. of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, Trento, Itali, 2006, pp. 15-21.

[59] J. Suzuki and H. Isozaki, "Semi-supervised Sequential Labeling and Segmentation Using GigaWord Scale Unlabeled Data," in *Proc. of ACL-08: HLT*, Columbus, Ohio, USA, 2008, pp. 665-673.

[60] X. Zhu, "Semi-supervised learning literature survey. Computer Science Dept.," Univ. of Wisconsin, Madison, Tech. Rep. 1530, 2006.

[61] D. Yarowsky, "Unsupervised word sense disambiguation rivaling supervised methods," in *Proc. of the 33rd Annual Meeting of the Association for Computational Linguistics*, Massachusetts, USA, 1995, pp. 189-196.

[62] H. Schutze, "Dimensions of meaning," in *Proc. of The 1992 ACM/IEEE Conf. on Supercomputing*, Washington, D.C., USA, 1992, pp. 787-796.

[63] Z. Kozareva, B. Bonev, and A. Montoyo, "Self-training and Co-training Applied to Spanish Named Entity Recognition," in *MICAI 2005: Advances in Artificial Intelligence*, Monterrey, Mexico, 2005, pp. 770-779.

[64] V. Ng and C. Cardie, "Weakly supervised natural language learning without redundant views," in *Proc. of The 2003 Conf. of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, Edmonton, Canada, 2003, pp. 94-101.

[65] M. Thelen and E. Riloff, "A bootstrapping method for learning semantic lexicons using extraction pattern contexts," in *Proc. of the ACL-02 Conf. on Empirical methods in natural language processing*, Philadelphia, USA, 2002, pp. 214-221.

[66] E. Riloff, "Automatically generating extraction patterns from untagged text," in *Proc. of the Thirteenth National Conf. on Artificial Intelligence*, Portland, Oregon, USA, 1996, pp. 1044-1049.

[67] R. Jones, A. McCallum, K. Nigam, and E. Riloff, "Bootstrapping for text learning tasks," in *IJCAI-99 Workshop on Text Mining: Foundations, Techniques and Applications*, Stockholm, Sweden, 1999, pp. 52-63.

[68] S. Patwardhan and E. Rilof, "Effective Information Extraction with Semantic Affinity Patterns and Relevant Regions," in *Join Conf. on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, Prague, Czech Republic, 2007, pp. 717-727.

[69] Z. Jingbo, C. Wenliang, and Y. Tianshun, "Using Seed Words to Learn to Categorize Chinese Text," in *Proc. of 4th International Conf. ESTAL 2004*, Alicante, Spain, 2004, pp. 464-473.

[70] E. Agichtein and L. Gravano, "Snowball: extracting relations from large plain-text collections," in *Proc. of the Fifth ACM Conf. on Digital libraries*, San Antonio, Texas, USA, 2000, pp. 85-94.

[71] E. Riloff and R. Jones, "Learning dictionaries form information extraction by multi-level bootstrapping," in *AAAI*, Orlando, Florida, USA, 1999, pp. 1044-1049.

[72] E. Riloff and J. Wiebe, "Learning extraction patterns for subjective expressions," in *Proc. of the 2003 Conf. on Empirical Methods in Natural Language Processing (EMNLP-2003)*, Sapporo, Japan, 2003, pp. 105-112.

[73] S. Patwardhan and E. Riloff, "Learning domain-specific information extraction patterns from the Web," in *Proc. of the Workshop on Information Extraction Beyond The Document*, Sydney, Australia, 2006, pp. 66-73.

[74] E. Riloff, "An Empirical Study of Automated Dictionary Construction for Information Extraction in Three Domains," *Journal Artificial Intelligence - Special volume on empirical methods*, vol. 85, pp. 101-134, August 1996.

[75] W. Cohen and S. Sarawagi, "Exploiting dictionaries in named entity extraction: combining semi-Markov extraction processes and data integration methods," in *Proc. of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, Seattle, WA, USA, 1998, pp. 89-98.

[76] G. Krupka and K. Hausman, "IsoQuest, Inc.: Description of the NetOwl Extractor System as Used for MUC-7," in *Proc. of Seventh Machine Understanding Conference*, Fairfax, VA, USA, 1998.

[77] C.-n. Seon, Y. Ko, J.-s. Kim, and J. Seo, "Named Entity Recognition using Machine Learning Methods and Pattern-selection Rules," in *Proc. of the Sixth Natural Language Processing Pacific Rim Symposium*, Tokyo, Japan, 2001, pp. 229-236.

[78] J. Piskorski. "Named-entity recognition for Polish with SProUT". In Leonard Bolc, Zbigniew Michalewicz, and Toyoaki Nishida, editors, *Lecture Notes in Computer Science Vol 3490 / 2005: Intelligent Media Technology for Communicative Intelligence: 2nd International Workshop, Warsaw, Poland, September 13–14, 2004. Revised Selected Papers*, pages 122–133.

[79] W. Liao and S. Veeramachaneni, "A simple semi-supervised algorithm for named entity recognition," in *Proc. of The NAACL HLT 2009 Workshop on Semi-Supervised Learning for Natural Language Processing*, Boulder, Colorado,

USA, 2009, pp. 58-65.

[80] T. Brants, "TnT-A Statistical Part-of-Speech Tagger," in *Proc. of the Sixth Applied Natural Language Processing Conf.*, Seattle, Washington, USA, 2000, pp. 224-231.

[81] E. Brill, "Some Advances in TransformationBased Part of Speech Tagging," in *Proc. of the Twelfth National Conf. on Artificial Intelligence*, Seattle, Washington, USA, 1994, pp. 722-727.

[82] K. Toutanova and C. D. Manning, "Enriching the knowledge sources used in a maximum entropy part-of-speech tagger," in *Proc. of the 2000 Joint SIGDAT Conf. on Empirical methods in natural language processing and very large corpora: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics*, Hong Kong, 2000, pp. 63-70.

[83] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, "Feature-rich part-of-speech tagging with a cyclic dependency network," in *Proc. of the 2003 Conf. of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, Edmonton, Canada, 2003, pp. 173-180.

[84] E. Brill and J. Wu, "Classifier combination for improved lexical disambiguation," in *Proc. of the 17th international Conf. on Computational linguistics*, Montreal, Quebec, Canada, 1998, pp. 191-195.

[85] R. Florian, A. Ittycheriah, H. Jing, and T. Zhang, "Named Entity Recognition through Classifier Combination," in *Proc. of the seventh Conf. on Natural language learning at HLT-NAACL 2003*, Edmonton, Canada, 2003, pp. 168-171.

[86] G. Schneider, "A Linguistic Comparison Constituency, Dependency, and Link Grammar," M.S. Thesis, University of Zurich, Zurich, Switzerland, 1998.

[87] R. Rosenfeld, "Two decades of statistical language modeling: where do we go from here," in *Proc. of the IEEE*, vol.88, no.8, pp. 1270-1278, August 2000.

[88] P. Venable, "Modeling Syntax for Parsing and Translation," Ph.D. dissertation, *School of Computer Science, Computer Science Dept.,* Carnegie Mellon Univ., Pittsburgh, PA, USA, 2003.

[89] T. B. Aji, "Annotated disjunct for machine translation," Ph.D. dissertation, Computer and Information Sciences Dept., Universiti Teknologi PETRONAS, Perak, Malaysia, 2010.

[90] J. Ding, D. Berleant, J. Xu, and A. W. Fulmer, "Extracting biochemical interactions from MEDLINE using a link grammar parser," in *Proc. of 15th IEEE International Conf. on Tools with Artificial Intelligence*, Sacramento, California, USA, 2003, pp. 467.

[91] T. Brehony and K. Ryan, "Francophone Stylistic Grammar Checking (FSGC) Using Link Grammars," In *Computer Assisted Language Learning*, vol. 7, no. 3, pp. 257-269, 1994.

[92] Y.-H. Wang, W.-N. Wang, C.-C. Huang, T.-W. Chang, and Y.-H. Yen, "Semantic Representation and Ontology Construction in the Question Answering System," in *Proc. of the 7th IEEE International Conf. on Computer and Information Technology*, Fukushima, Japan, 2007, pp. 241-246.

[93] Y.-H. Wang, W.-N. Wang, and C.-C. Huang, "Enhanced Semantic Question Answering System for e-Learning Environment," in *Proc. of the 21st International Conf. on Advanced Information Networking and Applications*, Niagara Falls, Canada, 2007, pp. 1023-1028.

[94] D. MollÃ, "Dealing with ambiguities in an answer extraction system," in *Workshop on Representation and Treatment of Syntactic Ambiguity in Natural Language Processing*, Paris, 2000, pp. 21-24.

[95] D. Freitag, "Toward general-purpose learning for information extraction," in *Proc. of the 17th international Conf. on Computational linguistics*, Montreal, Quebec, Canada, 1998, pp. 404-408.

[96] T. Mitchell, *Machine Learning*: McGraw Hill, 1997.

[97] H. V. Madhyastha, N. Balakrishnan, and K. R. Ramakrishnan, "Event Information Extraction Using Link Grammar," in *Proc. of 13th International Workshop on Research Issues in Data Engineering: Multi-lingual Information Management*, Hyderabad, India, 2003, pp. 16-22.

[98] D. Freitag, "Information extraction from html: Application of a general learning approach," in *Fifteenth National Conf. on Artificial intelligence*, Madison, Wisconsin, USA, 1998, pp. 517-523.

[98] S. B. Huffman, "Learning information extraction patterns from examples," in *IJCAI-95 Workshop on new approaches to learning for natural language processing*, Acapulco, Mexico, 1995, pp. 127-142.

[100] M. E. Califf and R. J. Mooney, "Relational learning of pattern-match rules for information extraction," in *Proc. of the sixteenth national Conf. on Artificial intelligence and the eleventh Innovative applications of artificial intelligence Conf. innovative applications of artificial intelligence*, Orlando, Florida, USA, 1999, pp. 328 - 334.

[101] H. Cunningham and K. Bontcheva, "Named Entity Recognition," in *RANLP*, Borovets, Bulgaria, 2003.

[102] M. Collins, "Ranking algorithms for named-entity extraction: boosting and the voted perceptron," in *Proc. of the 40th Annual Meeting on Association for Computational Linguistics*, Philadelphia, Pennsylvania,USA, 2002, pp. 489-496.

[103] Y. Sari, M. F. Hassan, and N. Zamin, "Creating Extraction Pattern by Combining Part of Speech Tagger and Grammatical Parser," in *Proc. of International Conf. on Computer Technology and Development, 2009.* Kota Kinabalu, Malaysia, 2009, pp. 515-519.

[104] M. E. Califf and R. J. Mooney, "Relational learning of pattern-match rules for information extraction," in *Proc. of the Sixteenth National Conf. on Artificial Intelligence*, Orlando, Florida, USA, 1999, pp. 328-334.

[105] The Stanford NLP Group. (2008, May 21). *Stanford Tagger version 1.6*. [Online]. Available: "http://nlp.stanford.edu/software/tagger.shtml."

[106] Sourceforge. (2009, May 21). JLinkGrammarParser. [Online]. Available : "http://jlinkgrammar.sourceforge.net/index.html."

[107] C. J. v. Rijsbergen, *Information Retrieval*, Second ed., London: Butterworths,

1979.

[108] Language Technology Group University of Edinburgh. (2008, July 28). LT-TTT2. [Online]. Available: "http://www.ltg.ed.ac.uk/software/lt-ttt2."

[109] B. Alex, B. Haddow, and C. Grover, "Recognising nested named entities in biomedical text," in *Proc. of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing*, Prague, Czech Republic, 2007, pp. 65-72.

[110] X. Wang and C. Grover, "Learning the Species of Biomedical Named Entities from Annotated Corpora," in *Proc. of 6th International Conf. on Language Resources and Evaluation (LREC 2008)*, Marrakech, Morocco, 2008, pp. 1808-1813.

[111] R. Tobin, C. Grover, S. Givon, and J. Ball, "Named entity recognition for digitised historical texts.," in *Proc. of 6th International Conf. on Language Resources and Evaluation (LREC 2008)*, Marrakech, Morocco, 2008, pp. 1343-1346.

[112] J. R. Finkel, T. Grenager, and C. D. Manning, "Incorporating non-local information into information extraction systems by Gibbs sampling," in *Proc. of the 43rd Annual Meeting on Association for Computational Linguistics*, Ann Arbor, Michigan, USA, 2005, pp. 363-370.

[113] Alias-i. (). *LingPipe*. [Online]. Available: http://alias-i.com/lingpipe/index.html."

[114] A. Ekbal and S. Bandyopadhyay, "A Conditional Random Field Approach for Named Entity Recognition in Bengali and Hindi," *Linguistic Issues in Language Technology (LiLT)*, vol. 2, no. 1, pp.1–44, 2009.

[115] Y. Liu, A. Stolcke, E. Shriberg, and M. Harper, "Using conditional random fields for sentence boundary detection in speech," in *Proc. of The 43rd Annual Meeting on Association for Computational Linguistics*, Ann Arbor, Michigan, 2005, pp. 451-458.

[116] N. Nam and G. Yunsong, "Comparisons of sequence labeling algorithms and extensions," in *Proc. of The 24th international Conf. on Machine learning*, Corvalis, Oregon, 2007, pp. 681-188.

[117] A. Mikheev, M. Moens, and C. Grover, "Named entity recognition without gazetteers," in *Proc. of the ninth Conf. on European chapter of the Association for Computational Linguistics*, Bergen, Norway, 1999, pp. 1-8.

PUBLICATIONS

1. Y. Sari, M. F. Hassan, and N. Zamin, “A Hybrid Approach to Semi Supervised Named Entity Recognition in Health, Safety and Environment Reports”, in Proceeding of International Conference on Future Computer and Communication, Kuala Lumpur, Malaysia, 2009, pp.599-602.
2. Y. Sari, M. F. Hassan, and N. Zamin, "Creating Extraction Pattern by Combining Part of Speech Tagger and Grammatical Parser," in *International Conference on Computer Technology and Development, 2009. ICCTD '09,* Kota Kinabalu, Malaysia, 2009, pp. 515-519.
3. Y. Sari, M. F. Hassan, and N. Zamin,"Rule-based Pattern Extractor and Named Entity Recognition: A Hybrid Approach", in Proceeding of International Symposium on Information Technology, Kuala Lumpur, Malaysia, 2010, pp.563-568.

APPENDIX A

Texas refinery was trying to restart a giant industrial boiler when a catastrophic failure killed one worker & injured two others late on ***Friday***, a company spokesman said on ***Saturday*****. Valero Energy Corp** (VLO.N) spokesman **Bill Day** also said that the 245,000 barrel per day (bpd) refinery in **Texas City**, 50 miles (80 km) southeast of **Houston**, was currently operating at planned production levels. The boiler that failed was one of several providing power & steam at the refinery.

Tommy Manis, 40, of **Alvin**, **Texas**, died instantly when the boiler failed, **Day** told **Reuters. Manis** was part of a crew working on the boiler. Local media reports on ***Friday*** night said the boiler exploded, but **Day** said investigators were attempting to determine exactly what occurred. "There was definitely a loud noise" when the boiler failed, he said. "Our sympathies are with Mr. **Manis**' family," **Day** said. "It's a very sad event. For a company with 22,000 employees it's surprisingly tight-knit. These things reverberate throughout the **Valero** community."

Investigators from the **U.S. Occupational Safety & Health Administration** arrived at the refinery on ***Saturday*** morning to begin probing the accident. Of the two workers injured in the failure, one suffered cuts & another fell. Both men spent the night in a local hospital. One of the men is a **Valero** employee & the other works for an outside contractor doing work at the refinery.

The failed boiler was being restarted after it had shut down earlier on ***Friday***, **Day** said. Boilers like the one that failed ***Friday*** night generate steam for use in the petroleum refining process. **F*riday***'s accident was the second fatality at **Valero**'s **Texas City** refinery since the company bought the plant in ***1998***. The previous death was in ***1998***. There was no widespread release of hazardous chemicals in the accident, **Day** said.

Figure A.1 Example of Accident News Article

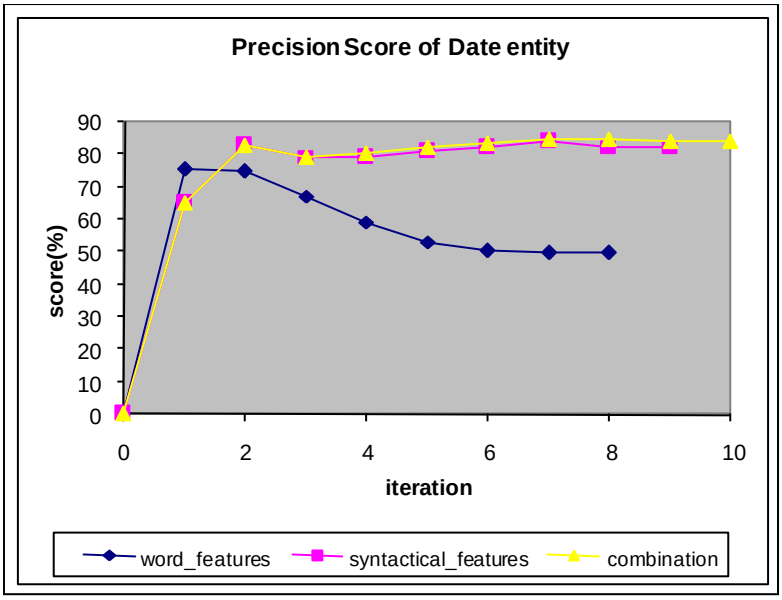

Figure A.2 Precision Score for Date Entity

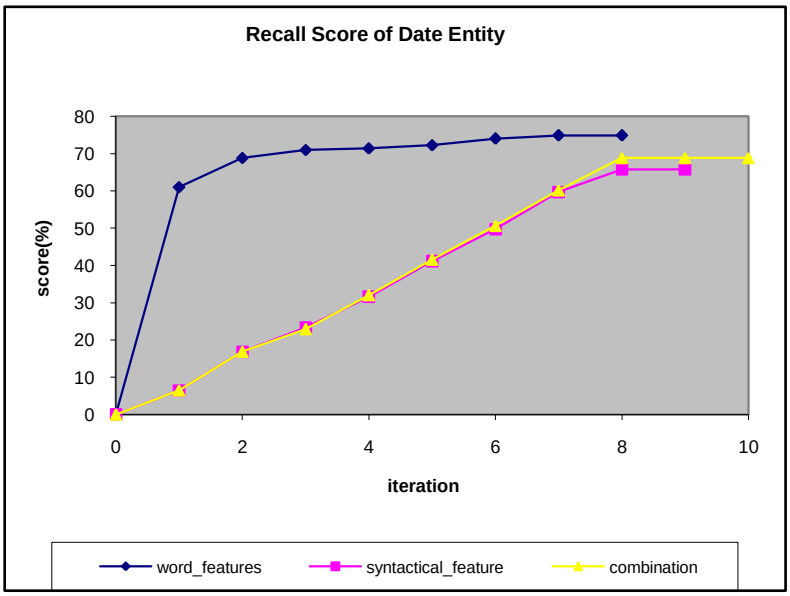

Figure A.3 Recall Score for Date Entity

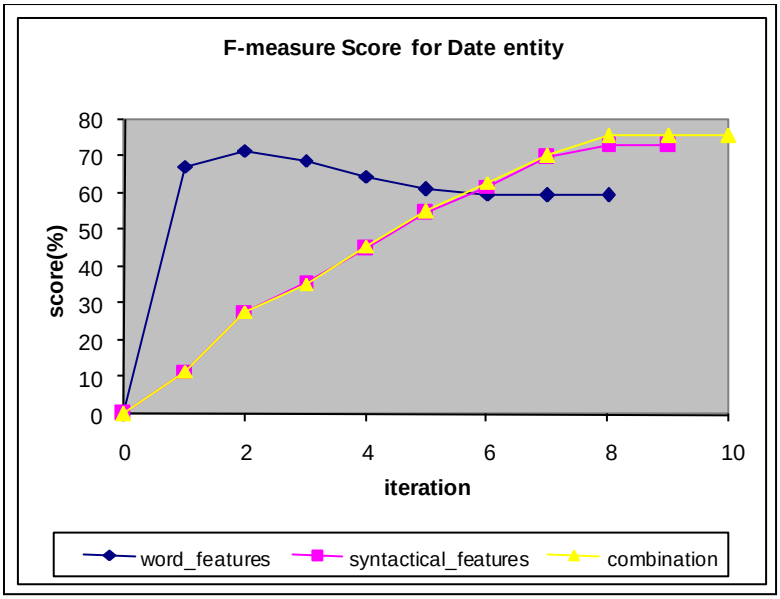

Figure A.4 F-Measure Score for Date Entity

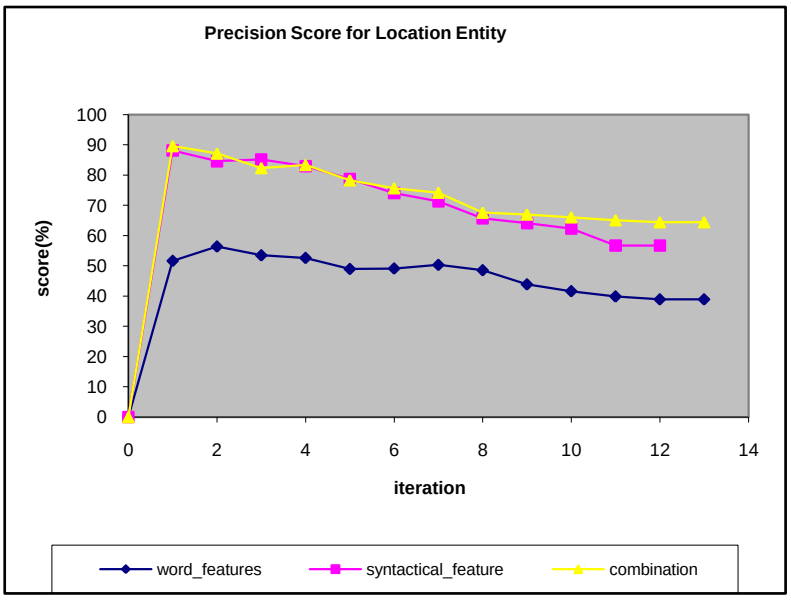

Figure A.5 Precision Score for Location Entity

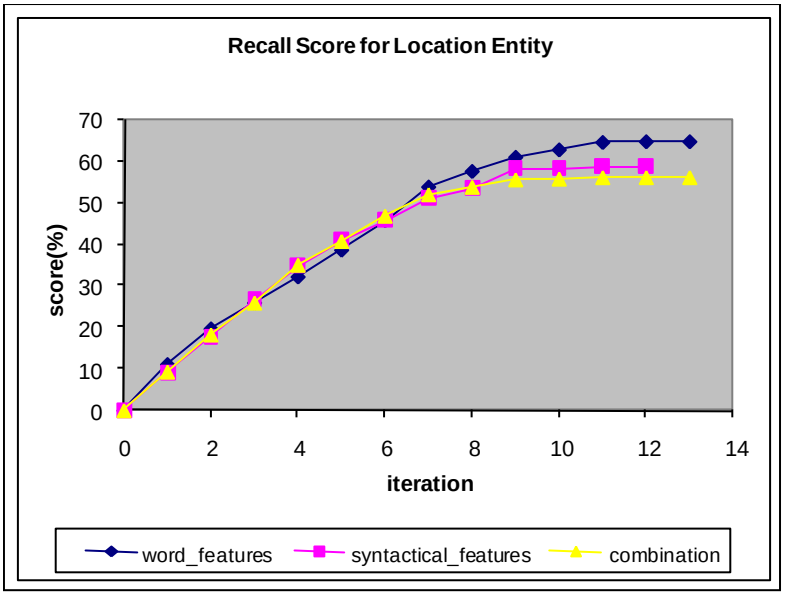

Figure A.6 Recall Score for Location Entity

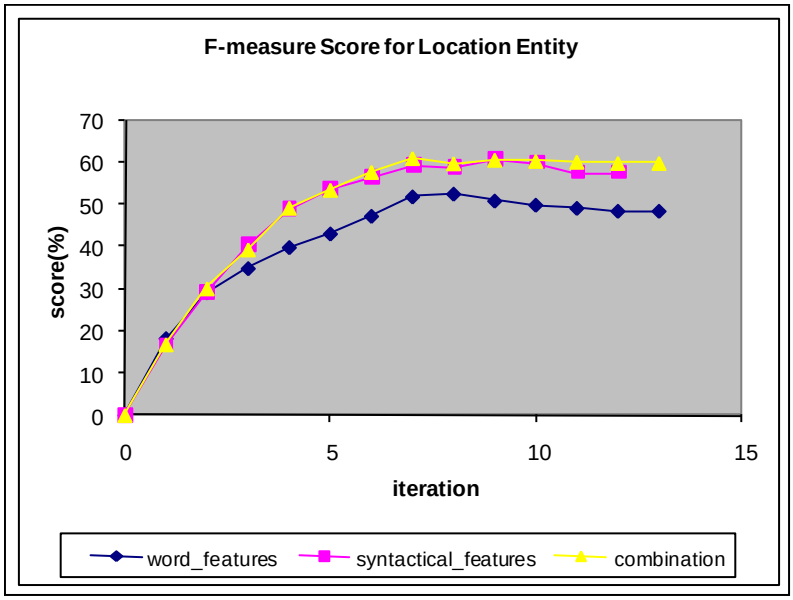

Figure A.7 F-measure Score for Location Entity

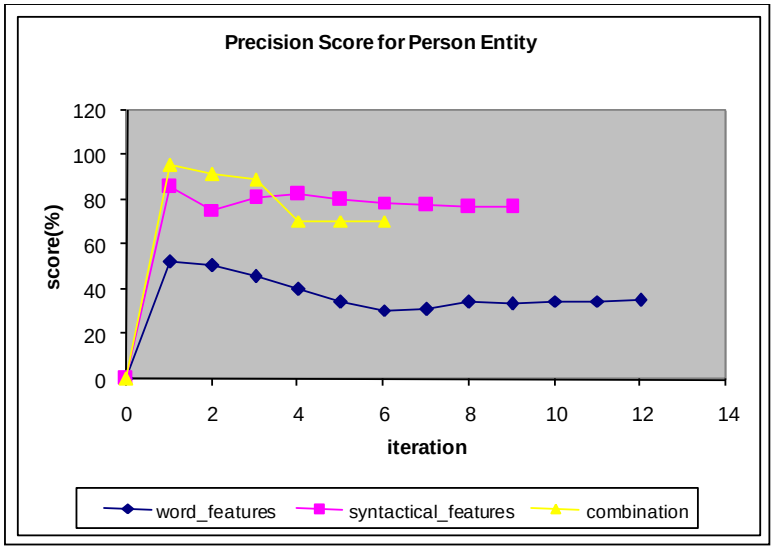

Figure A.8 Precision Score for Person Entity

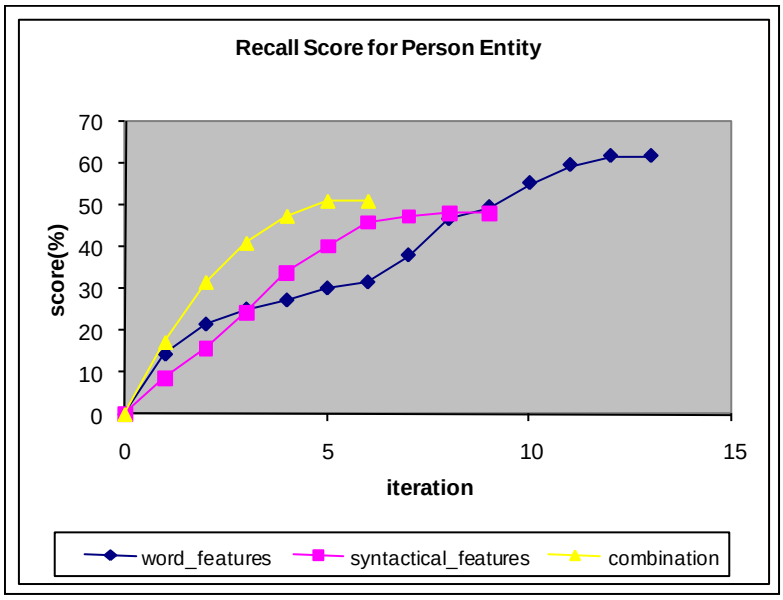

Figure A.9 Recall Score for Person Entity

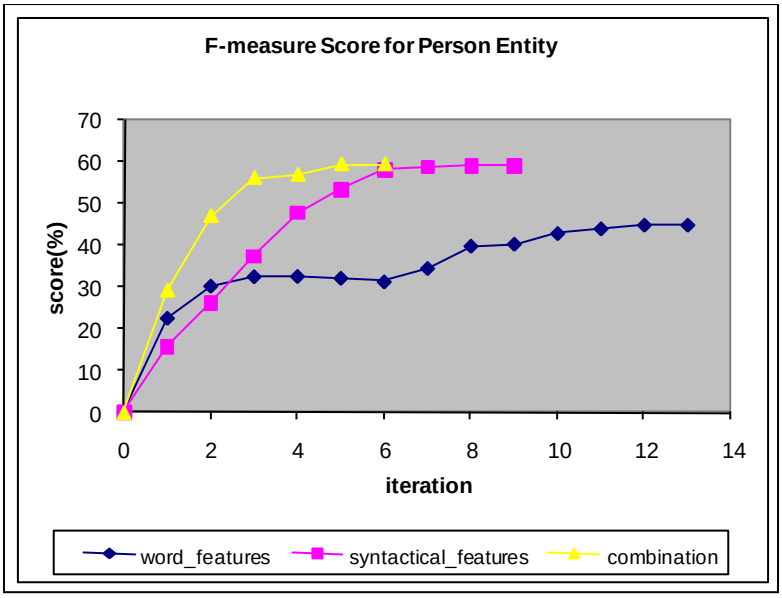

Figure A.10 F-measure Score for Person Entity