Macquarie University

Named Entity Extraction in Historical Australian Newspaper Text

A thesis submitted in fulfilment of the requirements
for the degree of **Masters of Research(MRes)**

in the

DBpedia project
School of Computing
Department of Science and Engineering

Honarable Supervisor: Associate Professor Steve Cassidy

Submission Date: 9th October, 2015

Declaration of Authorship

I, A H M Quamruzzaman, declare that this thesis titled, 'Named Entity Extraction in Historical Australian Newspaper Text' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: 

Date: 9th October, 2015

Abstract

A Named Entity Recognition (NER) objective is to extract and to classify atomic entities in text such as proper names (Names and locations), temporal expressions and other specific notation identification. In this project, we will apply NER methods to historical newspaper text taken from the Trove archive in the National Library of Australia. We will present an evaluation of various available NER systems on a hand-annotated sample of newspaper text. We will then present the result of applying the system to the whole corpus of text. Even when the occurrence of a given name is known across a large data set, there may be many individuals who share that name; this is particularly evident in the Trove corpus since it spans a long time period (1803-1959). In the second part of this project we will develop methods to try to classify different individuals with the same name. In particular, we will classify names as either Politician, Entertainer or other based on the documents that they occur in.

Acknowledgements

I would like to thank Associate Professor Dr. Steve Cassidy of Macquarie University for his help and guidance throughout this research with his remarks.

I also like to thanks the other Research Members of Research offices who are always helping us by proving proper guidance.

Contents

List of Figures

List of Tables

Abbreviations

ACE	**A**utomatic **C**ontent **E**xtraction
CoNLL	**C**onference **o**n **N**atural **L**anguage**L**earning
CRF	**C**onditional **R**andomn **F**ield
HMM	**H**idden **M**arkovn **M**odel
IE	**I**nformation **E**xtraction
IR	**I**nformation **R**etrieval
NA	**K**nowledg **A**cquisition
NE	**N**ame **E**ntity
NER	**N**ame **E**ntity **R**ecognition
NLTK	**N**atural **L**anguage **T**ool **K**it
MUC	**M**essage **U**nderstanding **C**onference
MUC6	The Sixth **M**essage **U**nderstanding **C**onference
MUC7	The Senventh **M**essage **U**nderstanding **C**onference
SVM	**S**upport **V**ector **M**achine

Chapter 1

Introduction

Nowadays, Digital information is one of the prominent parts in human life. It contains an unthinkably huge amount of information and this information is increasing very rapidly in daily activities. There have several reasons for the digital explosion and one of the obvious reasons is technology that contains digital devices and different types of information. With digital devices, its capacities are increasing and prices are plummeted so that more people are using devices to provide more information or using existing information. Moreover, digital information are come from different kind of sources such as Medical data, Scientific data, Sport data, Research data, Journals, Newspapers and many more. These data are making them increasingly inaccessible for vast amount of information those are produce by the digital devices. So, one question is come out that "How we can make sense of all digital data and utilized that vast amount of data for the next generation?"

Natural Language processing is a complex job for doing process of document annotation, indexing, translation, summarization, etc with digital information. The fundamental step of Natural language processing is to extract digital information that is known as information extraction (IE). Information Extraction works automatically to extract or discover textual mentions of specific types of entities and relationships into structured information from unstructured documents. It is making the information more suitable that have been written with human languages.

Name Entity Recognition is an Information Extract process that is related to find entities in set of categories. All the categories are pre-defined categories such as the names of persons, organizations, locations, expressions of times, percentage and monetary values, etc. Finding these categories are very important task to extract information from unstructured text. Name Entity Recognition is identifying categories as significant information and processes that information for different purposes such as Question answering, Machine learning, NLP-based search engines, Speech recognition, Machine translation and Knowledge discovery in unstructured texts.

Furthermore, another task of Natural Language processing is document classification. With document classification, we can assign documents into classes or predefine successful categories to identify documents easily for doing classification and efficiently manage those documents based on Name Entities.

In out project, we works on Historical Newspaper because Historical newspaper need to handle huge volume of Optical Character Reader (OCR) counted documents with the bad OCR quality. Optical Character Reade (OCR) is used to digitize the old Analog version (paper copy) into Digital vision (Soft copy) to contribute documents for historical Newspaper research(Packer et al., 2010). Entity recognitions from historical Newspaper documents are getting the new challenge because it is very difficult to process OCR documents than from natively digital data. The real world challenge is pointing out in presence of word errors and lack of proper complete formatting information in scanned documents (OCR) (Miller et al., 2000). Researchers are working on this problem to improve the extraction performance and it became one of challenges to recognize Name Entity (NE) properly from the historical documents. This is why we are motivated for extracting Name Entity (NE) and classification of those documents from Australian Newspaper text.

One of the goal of our project is to apply NER methods to historical newspaper text from the Trove archive (National Library of Australia). We have applied different popular NER systems and presented an evaluation of those NER systems. Moreover, other goal is to develop a system that will classify documents into different predefined classes such as Politician, Entertainer and Others with performance evaluation and statistical analysis based on the Name Entities that we have searched for.

Chapter 2

Literature Review

2.1 Introduction

Named Entity Recognition is widely used in the field of Natural language processing (NLP). In today's world, there has lots of electronic text around us and increasing that type of text gradually and progressively. However, information extraction from that text are become a very complex job and collecting or accessing information properly from that text is very difficult as well. For instance, building general purpose of extraction process from the text is still a long way to achieve a goal.

In 1990, the Sixth Message Understanding Conference (MUC-6) appealed to the researchers to work on information extraction (IE) (Grishman and Sundheim, 1996). At that time, researcher had extracted information related to the company and defence from the unstructured text such as journals, newspapers and magazines. They had tried to recognize atomic entities like names, times and numbers and recognition of identifying entities became an important task of information extraction. Hence, information extraction from named entities became a separate field named as "Named Entity Recognition"(Nadeau and Sekine, 2007). NER is also known as entity extraction, entity chunking, knowledge extraction and entity

identification. A NER system attempts to locate the Named Entity elements in text and classify elements in text to pre-defined categories (Alfred et al., 2014).

The categories of Named Entities are defined by the CoNLL (Conference on Natural Language Learning) and MUC (Message understanding conference). There are different opinions on what categories should be regarded as Named Entities based on different types of language. Moreover, there are few conventions having come out recently and atomic entities are commonly marked up in some categories by those conventions. According to the convention of MUC (Message Understating conference), ENAMEX, NUMEX and TIMEX categories are used in XML format for presenting Name expression, Numerical expression and time expression (AFNER, 2015). Typically, categories are format as follows:

<ENAMEX TYPE="PERSON" >
<ENAMEX TYPE="ORGANIZATION">
<ENAMEX TYPE="LOCATION">
<TIMEX TYPE="TIME">
<TIMEX TYPE="DATE">
<NUMEX TYPE="MONEY">
<NUMEX TYPE="PERCENTAGE">

These can be illustrated with the following examples.

Example-1:

Sentence= "Alex Martin bought 100 books for Newtown Council in year 2015"

$[Alex\ \ Martin]_{ENAMEX=Person}$ bought $[100]_{NUMEX=Quantity}$ books for $[Newtown\ \ Council]_{ENAMEX=Organization}$ in year $[2015]_{TIMEX=Time}$.

Example-2:

Sentence="Australian government gives 120 to 130 million Australian dollars for Universities in 2013"

$[Australian\ Government]_{ENAMEX=Organization}$ gives [120 to130 $million$ $Australian\ dollars]_{NUMEX=Money}$ for $[Universities]_{ENAMEX=Organization}$ in $[2013]_{TIMEX=Time}$

Furthermore, basic categories generally agreed upon based on CoNLL, IREX and MUC, include the following, (Ferro et al., 2007):

- Names (enamex)- Person, Location, Organization, GPE.

Name Entity Type	**Example**
PERSON	David Brown, Mark Dras, Alex Clark
LOCATION	Earlwood, Newtown, Rockdale
ORGANIZATION	CISCO,WHO,UNICEF
GPE	South Asia, Australia, Canada

- Times (timex) –Time and Date.

Name Entity Type	**Example**
TIME	3:50am, 9pm, nine am
DATE	March, 27-05-2015, 30th April 2015

- Numbers (numex)- Percent and Money.

Name Entity Type	**Example**
PERCENT	20.18 %, 80 pct
MONEY	$500AUD, two hundred dollars

However, the following may be considered as categories/subcategories: Distance, Speed, Age, City, River, Country, State/Province, Weight, etc. In addition, categories can differ for a particular NER project and based on the specific requirements of the project (AFNER, 2015). For instance, if needed to dealing with numerical data may need to classify numerical classification on that particular field. Similarly, for geographical classification should need to classify the location

entity for a particular location type.

With an example, we can clearly show that how name entity could be generated from a text:

Example-1:

Sentence= Alex Brooke purchased 50 books from Amazon in 2012.

And after doing name extraction, it producing an annotated block of text that highlights the names of entities:

$[Alex\ \ Brooke]_{Person}$ purchased 50 books from $[Amazon]_{Organization}$ in $[2012]_{Time}$.

Example-2:

Sentence= We arrived in Sydney at 2:50pm, it was a great holiday with cheap ticket price like $1500AUD. I stayed in Shangri-la hotel , New York, USA.

And after doing name extraction, it producing an annotated block of text that highlights the names of entities:

We arrived in $[Sydney]_{Location}$ at $[2:50pm]_{Time}$, it was a great holiday with cheap ticket price like $[\$1500AUD]_{Money}$. I stayed in $[Shangri-la]_{Organization}$ hotel, $[NewYork]_{Location}$ $[USA]_{GPE}$.

2.2 Challenges for NER

Many researchers have contributed to developing NER systems, however, there are lots of challenges and those are:

1. To find out a specific name entity, we need to execute the name entity recognizer over the all documents or files. With the coordination technique, we can execute the recognizer once and used the mapping technique to locate the exact location of name entity for future search. For the existing system of Name Entity Recognizer, it is very hard to implement and became a challenge for future Name Entity Recognition (NER) system.

2. According to the category, it is clearly visible that how can we define categories, however, there has some gray point to be considered named metonymy. As an example:

   Person vs. Artefact: "The king has played football today." vs "Purchase a king size drink for me."

   Location vs. Organisation: " (1) The $[Cricket\ World\ Cup]_{Name}$ organized in $[Australia]_{Location}$ for 2015" vs "$[Australia]_{ORG}$ won the Cricket World Cup in 2015".
   "(2) She met with her family at $[Sydney\ \ Airpor]_{Location}$" vs "$[Sydney\ \ Airpor]_{ORG}$ authorities have their own rule to apply"

   Artefact vs. Company: "We are watching $[BBC]_{Name}$." Vs "$[BBC]_{ORG}$ works with many countries."

3. Label consistency for name entity is one of the prominent challenges. Label-consistency refer the way of identifying tokens in such a way that have the same label assigned or cross-reference. We can mention some example like:

   Example-1:

The words "United States", "USA", "US", "States", "the United States of America" should all recognized in the same output and category as LOCATION.

Example-2:

Rockdale, rockdale, and ROCKDALE should all be identified as the same entity and these would break word-level rule based systems (e.g. find all words that are capitalized). However, most of the recognizer (like NLTK, Stanford) has recognized the above example as a different name entity output. As a result, Non-local dependency is still a challenge in Name Entity Recognition(NER)

4. Name Entity Recognition for the historical newspaper need to handle huge volume of Optical Character Reader (OCR) counted documents with the bad OCR quality. Optical Character Reade (OCR) is used to digitize the old Analog version (paper copy) into Digital vision (Soft copy) to contribute documents for historical Newspaper research (Packer et al., 2010). Entity recognitions from historical Newspaper documents are getting the new challenge because it is very difficult to process OCR documents than from natively digital data. The real world challenge is pointing out in presence of word errors and lack of proper complete formatting information in scanned documents (OCR)(Miller et al., 2000). Researchers are working on this problem to improve the extraction performance and it became one of challenges to recognize Name Entity (NE) properly from the historical documents.

5. Historical spelling on Newspapers or historical journals has made the confusion for identifying NE properly. This problem arises with the historical spelling variation. The spelling variation affects on person and particular places names that are spelled differently in current world than the previous 200 years (Europeana, 2015) (Marrero et al., 2013). So, the spelling variation is noticing us a new challenge for recognizing the Name Entity (NE) properly.

6. In the Multilingual content or text, Name Entity can defer from languages to languages because every language is designated differently with their own culture and structure. In real world experiment, Name Entity methods, technique and evolution are totally different for every language. Some methods and techniques make for Name Entity Recognition in one language can make lower rate of recognition for other languages (Cloud, 2015). So, different recognizing technique, training and evaluation models for different languages structures are producing many challenges.

2.3 Review of Background Bibliography

A Named-Entity Recognition (NER) is part of the process in information extraction and could be used as a process in Text mining. NER is used as a tool to aid user to identifying and detecting entities such as person, location, organization, date, time or numbers.

However, dissimilar languages may have different morphologies or techniques that involved different NER processes. For instance, an English NER process cannot be utilized to process Hindi journals or newspaper because of the dissimilar morphology used in different languages such as English, Spanish and Dutch (Tjong Kim Sang and De Meulder, 2003).

One of the earliest papers presented on the IEEE conference that proposed a system to extract and classify company names from the financial news based on heuristics and handcrafted rules. (Rau, 1991). Afterwards, many improvements have been occurred after 1996 and most of the events such as MUC-7, CONLL and IREX have much contribution on this field.

2.3.1 Information Extraction Architecture:

2.3.1.1 Information Extraction:

In the current technological world, internet is producing lots of textual information and the volume of textual information are significantly growing or increasing in exponentially. Most of the information are coming from different types of sources such as Social Media communication, online news portals, different kind of documents from Government and Private agencies, court rulings and proceeding info, Medical data on research papers, articles, records and alerts, online sales records and many more. With all types of documents, the following queries can be done (Piskorski and Yangarber, 2013). Those are like:

1. News agency needs a detail views for current political situation.
2. Government agencies can discovery knowledge about the future trend of natural disaster situation, current trend of education achievement and current financial situation.
3. Sales companies can trace the people opinion on a newly released product performance.
4. Government intelligent agency can investigate general trend of terrorist activities.
5. Medical research can investigate a new treatment with the old process of treatment.
6. Legal scholars can search for the decision of judges in criminal proceedings.

If we want to get the related information then we need to handle millions of documents to extract the results of such queries. However, all of above type of information are need to process from unstructured documents and unstructured documents make it very difficult to extract or discover valuable and relevant information in structured way. To solve this kind of problem, Information Extraction

(IE) technology has emerged. Information Extraction works automatically to extract or discover structure information from unstructured documents for making the information more suitable that have been written with human languages.

As an example of Information Extraction:

News Article= "At least 104 people have been killed when a gas cylinder exploded in a restaurant in central India. The death toll rose rapidly from an initial count of 20 after rescuers recovered dozens more bodies from the debris of the destroyed restaurant and neighbouring structures in the town of Petlawad in Jhabua district. "

This article has taken from Sky news on Gas cylinder explosion in India. After Information Extraction process, we can get the information in the following format.

Type	**Crisis**
Subtype	Accident
GPE	Central India
Death Count	104
Location	Petlawad, Jhabua
Accident Type	gas cylinder

TABLE 2.1: Example of automatically extracted information from a Sky news article on a Gas cylinder explosion (Piskorski and Yangarber, 2013)

The state of the art in NLP (Natural Language Processing) is still a long way from being able to build general-purpose representations of meaning that should be extracted from unrestricted text.

2.3.1.2 Extraction process:

Figure 2.1 describes the architecture of a simple information extraction process. At the beginning of the process, a document has been taken as a raw text and split into sentences using sentence segmentation process. With the process of sentence segmentation, we will get all separate sentences from text and process each sentence with word tokenization process in the next step of processing. In

word tokenization, we will separate a list of words from each sentence for part-of-speech tagging to process Named Entity Recognition and relational recognition.

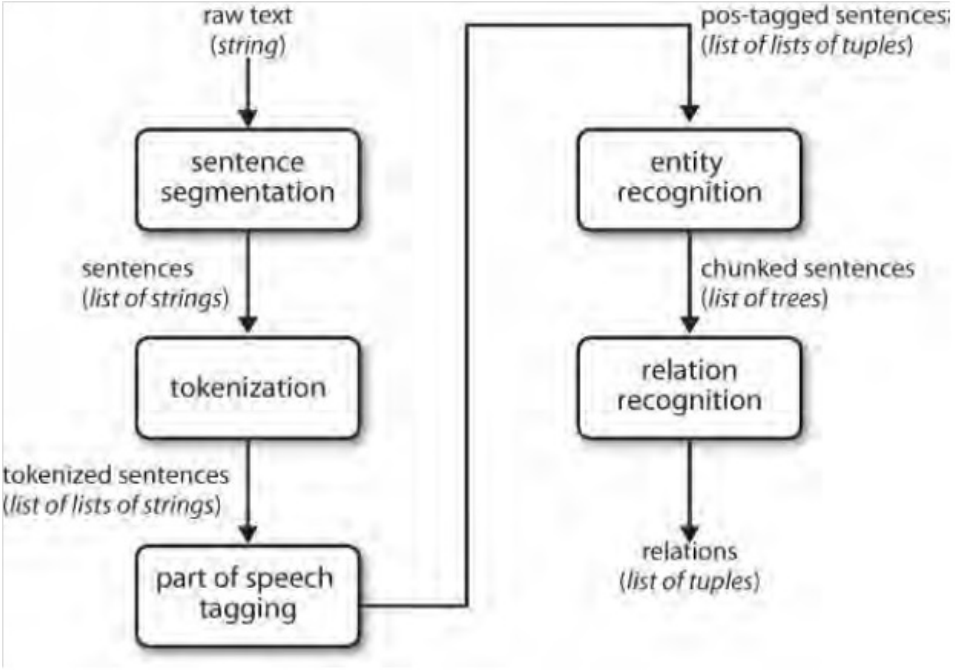


FIGURE 2.1: Information Extraction Architecture (E. and Loper, 2009)

Step-1: Sentence Segmentation:

Sentence segmentation is known as Sentence boundary detection, Sentence boundary disambiguation or Sentence segmentation (textminingonline.com, 2015). In Natural Language processing, it is the first job to find out where sentences begin and end. Afterwards, all segmented sentences have sent to the next step for word tokenization. Moreover, to perform sentence segmentation, the basic rules is simply checking each punctuation mark and find where the punctuation is label as boundary. It has used the model that contains abbreviation words, collocations and words that start sentences (Steven Bird and Loper, 2015b). An example has given below, how we generate list of sentences from a paragraph:

> **Paragraph=** "The leaders of the World will be making snide jokes about Australian Democracy. Frankly, four PMs in 28 months. Even

> in South America there is more political stability. We need a Republic based on American constitutional lines to stop this lunacy." (Herald, 2015)

After processing the above text for sentence segmentation, we can get the following output which is contains a list of separate sentences.

> **List_of Sentences=**['The leaders of the World will be making snide jokes about Australian Democracy.', 'Frankly, four PMs in 28 months.', 'Even in South America there is more political stability.', 'We need a Republic based on American constitutional lines to stop this lunacy.']

Step-2: Tokenization:

Tokenization is the process of breaking sentences into words, symbols or any meaningful elements where a sentence is an instance of sequence of characters. This process is very important or essential part of all text processing for extracting information like Name Entity Recognition (NER). Word tokenize takes a sentence as a string and produce meaningful words as output. Moreover, a very simple method has used to generate tokens by finding out the white space (space or line break, or by punctuation), however, we can use Regular Expression to generate more complicated tokens (Steven Bird and Loper, 2015c). An example has given below, how we generate list of words from a sentence:

> **Sentence=** "The leaders of the World will be making snide jokes about Australian Democracy." (Herald, 2015)

After processing the above Sentence for Tokenization, we can get the following output which is contains a list of separate words.

> **List_of_Words=** ['The', 'leaders', 'of', 'the', 'World', 'will', 'be', 'making', 'snide', 'jokes', 'about', 'Australian', 'Democracy', '.']

Step-3: Parts-of-Speech (POS) tagging:

In text analysis process, one of the most important parts is classifying words from sentence into the Part-of-Speech (POS) and labels them according to Part of Speech. So, this process is said Part of Speech tagging. It also has known as POS tagging, word classes or lexical categories. However, most of the researchers are simply mention as POS tagging. Beginning of the process to classify words, a predefined collection of tags are used for proper labelling of Part-of-Speech (POS) tagging. Theses collection tags are known as tagset".

Furthermore, after finishing all text processing into generating words, Part-of-Speech (POS) tagging method has been used to process those words with proper tagging. Two types of algorithms are used in POS tagging. One is said, "Rules based methods" that works with a large number of hand-crafted rules that is one of the first and mostly used methods. Other one is Probabilistic methods that work on tagged corpus to train model to process part-of-speech tagging (Brill, 1992). An example has given below, how we can generate parts of speech from List of words where the list of words collected from above section:

> **List_of_Words=** ['The', 'leaders', 'of', 'the', 'World', 'will', 'be', 'making', 'snide', 'jokes', 'about', 'Australian', 'Democracy', '.']

After execution the Part-of-Speech (POS) procedure, we can get the following words with proper tagging for that list of above words.

> **POS_tagging=** " [('The', 'DT'), ('leaders', 'NNS'), ('of', 'IN'), ('the', 'DT'), ('World', 'NNP'), ('will', 'MD'), ('be', 'VB'), ('making', 'VBG'), ('snide', 'NN'), ('jokes', 'NNS'), ('about', 'IN'), ('Australian', 'JJ'), ('Democracy', 'NNP'), ('.', '.')"

In the above POS tagging, the following tag set is used.

Tag	**Meaning**	**Example**
DT	Determiner	The
NNS	Noun plural	'leaders'
IN	Preposition	'of", 'about'
NNP	Proper noun, singular	'World', 'Democracy'
MD	Modal	'will'
VB	Verb	'be'
VBG	Verb, Gerund/present participle	'making'
NN	Noun	'snide'
JJ	Adjective	'Australian'

TABLE 2.2: Example of Part-of-Speech tag sets

Step-4: Entity Recognition:

Chunking is the next step in name entity recognition after finishing the part of speech tagging. In part-of-speech tagging, we can just take a word token as input and develop output with Part-of-Speech (POS). This annotation task is simple one to one mapping. Similarly, a chunker is works over part-of-speech tagging for segmentation and label multi-level token sequences. In another way, we can say, chunking is the process of extracting short phrases in a particular pattern from a part-of-speech tagged sentence. Chunking can be like Verb-phrase chunking, Noun-phrase chunking, Prosodic chunking, etc. In Named Entity Recognition, we are dealing with Noun-phrase (NP) chunking to find out Nouns for identifying Named Entities.

With the following figure, we can express chunker in boxes. In the figure, small box is used to express word tokenization and part-of-speech (POS) tagging. At the same time, big box is selected a subset of the tokens as a chunk.

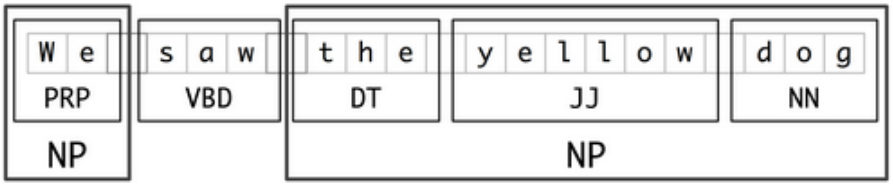


FIGURE 2.2: Chunking: segmentation and label multi-level token sequences (Steven Bird and Loper, 2015a)

NP-Chunking:

NP-Chunking will search for individual noun phrase from the tokenized text. As an example:

> **Sentence=** "John Peterson saw a beautiful animal with the telescope."

After, applying the NP-Chunker, we can get the following output with chunk annotation.

> **NP-Chunking =** " $[John\ \ Peterson]_{NP}$ saw $[a\ \ beautiful\ \ animal]_{NP}$ with $[a\ \ telescope]_{NP}$."

After the Noun Phrase (NP) chunking, we can call gazetteers or dictionaries that are related to the Name, Organization and Location entity for matching or identifying noun phrase to recognize the Named Entity (ClearTK, 2015).

After finishing this process of entity recognition, we can get the following output for the above NP-Chunk output.

> **Output(NE):** Name Entity PERSON : ['John Peterson']
> Name Entity ORGANIZATION : []
> Name Entity GPE : [ ]

Step-5: Relation Recognition:

Once, we have identified the name entities from unstructured text then our next step can be find out the relational extraction task. That is also said relational recognition. This is basically looking for the relation among the recognized name entities. This task is alike to name entity recognition with some additional activities like detecting relation or multiple relationships within the entities (E. and

Loper, 2009). We can express relational recognition by using table in the following way:

Subject	Relation	Object
X	Y	Z

TABLE 2.3: Table Structure of Relation Recognition

where, X and Z represent Name Entity and Y is the string of words that counted as relationship between two entries. As an example of Relation Recognition:

Subject	**Relation**	**Object**
Bill Gates	Person-works in	Microsoft Inc
Opera House	Located-In	Circular Quay, Sydney
p53	Is -a	Protein

TABLE 2.4: Example: Relation Recognition

Relational recognition is working with several of formats and languages. As an example: RDF (Resource Description Framework) is relational representation for the Web data. Furthermore, the dramatic increasing of information in every day, relation recognition needs to work with large textual documents (Hong, 2005).

2.3.2 Features for NER

In the Named Entity System, features are descriptor or characteristic attributes of a word. Many features are used for the purpose of Named Entity recognition and classification.

For instance, a Boolean variable associate with a Binary variable can be define to characterise the words as true for Noun when it find any exact matching with the rules otherwise indicating as a false value as other Objects (Verb, Pronoun, Adjectives, Adverbs, Preposition, Conjunctions and Interjections) where rules are defined earlier to execute this process (Nadeau and Sekine, 2007). In the following example, we can define rules and apply that rules to find out the feature vector for a specific text where each word identity as Noun or other Object.

Rules:

1. Each word can be defined by one or more attribute like Numerical, Boolean, Nominal value, Noun as N and O for other type of Objects.
2. Numeric value corresponding to the length of specific word.
3. Set the Boolean "true" when system find the word as Noun and set "false" if word is other then Noun.
4. Nominal values corresponding to the specific word on the process.

Text="Macquarie University is a leading University in Australia"

This text can produce the following feature vector as result using the above rules.

Output="(9, true, Macquarie, N) (10, true, University, N) (2, false, is, O) (1, false, a,O) (7, false, leading, O) (10, true, University, N) (2, false, in, O) (9, true, Australia, N)"

2.3.2.1 Word Level feature space:

Character markups of words are related to the word level feature. They have described word case, numerical value, special character, punctuation, etc. in figure 2.3.

In the following table, we can categorize words level features with an example.

For example: **Text=** "Alex Martin purchases games of FIFA-XI from eBay"

From the above text, we can get the following words that match with word level features in table 2.5.

Features	Examples
Case	- Starts with a capital letter - Word is all uppercased - The word is mixed case (e.g., ProSys, eBay)
Punctuation	- Ends with period, has internal period (e.g., St., I.B.M.) - Internal apostrophe, hyphen or ampersand (e.g., O'Connor)
Digit	- Digit pattern - Cardinal and Ordinal - Roman number - Word with digits (e.g., W3C, 3M)
Character	- Possessive mark, first person pronoun - Greek letters
Morphology	- Prefix, suffix, singular version, stem - Common ending
Part-of-speech	- proper name, verb, noun, foreign word
Function	- Alpha, non-alpha, n-gram - lowercase, uppercase version - pattern, summarized pattern - token length, phrase length

FIGURE 2.3: Word-level features (Nadeau and Sekine, 2007)

Word	Word level Features
Alex Martin	- Case (Start with capital latter) - Function (n-gram) - Character (First person, Noun)
purchases	- Parts of Speech (Verb)
games	- Parts of Speech (Noun)
FIFA-XI	- Digital pattern(Roman Number) - Punctuation, - Case (Word is all Upper case)
eBay	- Case (The word is mixed case)

TABLE 2.5: Example of Word level feature

2.3.2.2 Digit pattern

Using digit, we can express a range of information such as dates, percentage, identifier, intervals, amount of money, time etc. As an example, two digits and four digits number can be use for particular pattern of year such as day/month/15 or day/month/2015. Moreover, if the two digits or four digits associated with "s" then it can used for decade such as 90s or 1900s (Nadeau and Sekine, 2007).

2.3.2.3 Common word ending

In English text, there are many words ending with some word affixes. This type of affixes indicates a profession or category to easily recognize the words classification. For an example, system can identify human profession by checking the affixes like "ist" that can be match with "journalist", "cyclist" and "motorist". In Name Entity, there also has some common suffix feature (Nadeau and Sekine, 2007). For instance, those are:

1. For Person name, we can use some common affixes that are like "own", "ack" and "ung".
2. For Organization, we can use some common affixes that are like "ex", "tech" and "soft".
3. For Location, we can use some common affixes that are like, "town", "ford" and "ham".

The following table is showing example of name, organization and location those are affixes with common word ending.

Entity Name	affixes	Example
Person	- "own" - "ack" - "ung"	- Gerald W. Brown, John Brown, Oliver Brown - Alan Black, Alex Black, Amy Black - Alan Young, Andrew Young, Bill Young
Organization	- "ex" - "tech" - "soft".	- Jitex, Sofex, Miniex - Softech, Hevntech, Supertech - Microsoft, Sunsoft, GebiusSoft
Location	- "bury" - "ford" - "ham".	- Canterbury, Dewsbury, Pendlebury - Bradford, Ampleforth, Bideford - Lewisham, Gillingham, Chatham,

TABLE 2.6: Example of Common word ending features

2.3.2.4 List lookup features:

"Gazetteer", "Lexicon" and "Dictionary" are used as a list look up features. These looks up features are very useful for Name Entity Recognition system. Moreover,

Gazetteers and Dictionaries do not depend upon any other annotation, matching pattern or any algorithms. These has been only based on the textual contents of documents and usually work as a standalone lookup tools to find out occurrences of string from the predefined lists (Nguyen et al., 2013)(Saha et al., 2008).

Moreover, if a word is specifically matching with the list that is predefined as the name dictionary can express as a Name Entity. For instances, Dhaka is an entity element of the list of cities means it is indicating as a location entity (Nadeau and Sekine, 2007).

2.4 Techniques and Algorithms to solve the NER problem

The aim of NER system is to recognize the unknown atomic entry from text in a systematic way. In the recognising process, different kinds of recognition and classification rules implies with positive and negative feature on NER for a large collection of annotated data. In the CoNLL forum and MUC-7 research community, they have presented several kinds of learning techniques based on rules based automatic learning system and handcrafted rules.

2.4.1 Supervised learning (SL) methods

In current research, most of the dominant features are supervised learning in Named Entity recognition. Supervised learning system is worked on annotated entries from the training corpus for tagging words of a test corpus that means, it can be used for mapping test data with training data. The performance of the supervised learning depends on the vocabulary transfer in both training and testing corpus. A supervised learning algorithm works on the training data and produces a function that will map the testing data. There are many researcher described SL techniques those are Decision Trees , Hidden Markov Models (HMM) (Bikel

et al., 1997),Conditional Random Fields (CRF) (McCallum and Li, 2003), Support Vector Machines (SVM) (Asahara and Matsumoto, 2003) and Maximum Entropy Models (ME). These are all different approaches works on a system that reads a large annotated corpus, memorizes lists of words entities and creates disambiguate rules for recognizing Named Entity (Nadeau and Sekine, 2007). A supervised learning is working on classification and prediction on documents where deducing a classification is useful and easy to determined based on the pre-determined classification (aihorizon, 2015).

2.4.2 Semi-supervised learning methods

Semi-supervised learning (SSL) is a relatively recent feature of learning process with very small degree of supervision in specific time duration. SSL is using unlabeled data along with the label data to learn better predicative model. This learning system works on regression and classification tasks based on the various assumptions like clustering, smoothness and manifold on the unlabeled data (Gunopulos et al., 2011).

In SSL, bootstrapping is used as the main technique that involves small degree of supervision by human for starting the learning procedure. For instance, if we make a system that will collect the different city names in Australia from given newspaper and user might ask to provide a small number of city names as a seed to recognize and classify cities from the text. Afterwards, the system will start recognizing and classifying the city names. Then, we can search for other instance of city names that will come along in similar contexts. By repeating this process, we can recognized and classify a large amount of Australian city names from a large number of texts (Nadeau and Sekine, 2007).

In 1999, E. Riloff and Jones has defined concept of mutual bootstrapping that work with growing set of atomic name entities and a set of texts. To make it so robust, they had added a another level of bootstrapping that said "meta-bootstrapping"

that would retain most reliable entities generated by mutual bootstrapping and repeat the process. This mutual bootstrapping algorithm has generated both semantic lexicon and extraction patterns that produce more accurate dictionaries in each level of iteration. As a result, this algorithm has produced a high quality of dictionary for various semantic categories (Riloff et al., 1999).

2.4.3 Unsupervised learning methods

Unsupervised learning method in NER is basically little bit different and harder implementation. In this learning system, there is no supervision or given seeds or no pre-discovered classification. In this approach, system needs to teach agent by giving some sort of reward system to indicate system success rather than giving explicit categorizations (Nadeau and Sekine, 2007). For example, decision problem framework is generally getting training and makes decision for maximizing rewards rather than producing a classification. In the field of NER, success is completely based on how well the agent can work to finding out maximum recognition of entity. Moreover, another type of unsupervised learning that is called "Clustering". The primary goal of this clustering is simply to find out similarities in the training data set and finding out the maximum functional activities. For instance, clustering individuals based on demographics that might result in a clustering of the male in one group and the female in another (aihorizon, 2015).

2.5 Evaluation of NER

Evaluation is essential to find out the progress of NER system. NER system usually evaluates depending on human linguistic output comparing with the system generated output. There is a lots of techniques are described by lots of researchers but most researchers are following two main evalution methods such as MUC and ACE events.(Ferro et al., 2007)

2.5.1 MUC Evaluation:

This is a technique that works on scoring system based on two axes, firstly it works to find out correct type and secondly, it has the ability to find out exact text.(Riloff et al., 1999) All of the evaluation process are following three things for measuring both TYPE and TEXT. Those are:

1. The number of correct answers (Correctly identified or True positive (TP)).
2. The number of incorrectly identified answer (False positive (FP)) and
3. The number of possible entities in the solution that are incorrectly rejected (False negative).

The micro-averaged f-measure(MAF) is known as MUC scouring system that has used as single measure of performance produce from the harmonic mean of precision (positive predictive value) and recall (sensitivity value) calculated on both axes over the all entity slots (Grishman and Sundheim, 1996).

Equation for Precision(Positive Predictive Value) is PPV=TP/(TP+FP) where TP and FP stands for True Positive and False Negative respectively.

Furthermore, Equation for Recall is TPR=TP/(TP+FN) where TPR, TP and FN stands for True Positive Rate, True Positive, False Negative respectively.

Moreover, the harmonic means of two numbers are never become higher than the geometrical mean. Hence, the F-measure leans to privilege balanced systems (Ferro et al., 2007). Equation for F-measure,

```
F-Measure=2*((Precision*Recall)/(Precision+Recall))
```

2.5.2 ACE evaluation

ACE is more difficult scoring process than MUC Evaluation and Exact-match evaluations. There are various issues included with this evaluation such as wrong type and partial match. In the evaluation, each atomic entity type has a parameterized weight and it contribute maximum proportion of final score. In addition, customizable costs (COST) have been applied for determining false alarms, type errors and missed entities. Moreover, partial matches only counted when a significant number of proportions of character are match with the name entity.(Nadeau and Sekine, 2007)

2.6 Summary

The Name Entity Recognition is a thriving research field for the last twenty years. NER is looking a simple task but faces a number of challenges. The main aim is to find entities while entities are very difficult to discover and once it is found then another difficult task are to classify properly from the text such as newspaper, journals and magazine. Sometimes few name entity recognition also making lots of complexity when they are extracting information and classifying name entities from the text. As an example, locations and person names can be the same and follow similar formatting so that it is very hard to classify them in exact format.

The amount of natural language text increasing day by day and corpus has given us available data set in electronic form. Moreover, researchers are contributing lots of ideas, methods and processes for making this procedure easier and reduce different sorts of complexity. However, the complexity of natural language can make it very difficult to access the information from general text. For these sorts of complexity, state of the art in NLP is still tremendous job and a far way from building general-purpose representations of meaning from unrestricted text.

Chapter 3

Name Entity Recognition on Trove Newspaper Text

3.1 Trove

Trove is an Australian online library that is organized, developed and maintained by National Library of Australia collaborating with the National and State Library of Australasia. This is basically a database aggregator, search engines, a platform, collaboration, content repository and a community. The key feature of Trove is to provide faceted search on Australian contents like Full-Text books, Letters, Archives, Diaries, People, Journals, Maps, Music, Newspapers, Pictures and Websites for finding and retrieving information easily interacted with contents and social engagement for Australians.(Trove, 2015)

At present, Trove has enable their library data to be shared in the services that has currently provide 90 million items of metadata collaborating with more than 1000 libraries, museums, galleries, archive and other organizations inside and outside of Australia. This is the search engine that only work on the materials available on trove communities. Trove materials cannot access via the other traditional web search engines like Google, Yahoo or Bing. (Holley, 2015)

3.2 Why works on Trove Archive?

Trove is a good source to get all Australian newspaper in one place with proper data. Trove is providing access and retrieving information over 17 million pages over 900 Australian newspapers from each state and territory. All of the news paper have collected from the earliest newspaper in Australia in 1803 to till now including some community language newspapers (National Library, 2015). Every day, new digitized published newspaper are added to the archive and enriching the Australian historical achievements.

In my project, the main actives are to make the newspaper information available for the public based on their query. The first step to find the specific newspaper article or related information is to identify Name, Location and Organization on some Newspaper archive or data-set. As of 1 September 2015, there are 18,366,616 pages consisting of 180,934,631 articles available for different use. For the availability of all popular newspaper in Australia, I have chosen the Trove archive for collecting my desire data set.

3.3 Name Entity Recognition

From the literature review, we have got some brief ideas about the Named Entity Recognition and my first task is to work on the process of name entity recognition. For doing Name Entity Recognition (NER), we have written several kind of program on python programming language for finding Name Entities (Person, Organization and Location). We have utilized three different types (different organizational recognizers) of recognizers with my prepared corpus for identifying Name Entity Recognition, those are, NLTK (Natural Language Toolkit), Senna Tagger and Stanford Tagger. Moreover, We have made our own corpus by collecting files from Trove database as data set for doing our experiment on entity recognition.

In the following section, we are going to describe all of my process for Name Entity Recognition (NER) with three different types of recognizers. For each type of Name Entity Recognizer, we were followed the next steps to recognize name entity from my trove archive files/corpus. Those are:

1. Text from the Trove File into the segmentation of sentences
2. From sentences into the tokenization of words
3. Parts of Speech tagging for all words that we produce from sentence tokenization.
4. Using the Recognizer (NTLK, Stanford, Senna Tagger) to recognize the name entity from the Parts of Speech tagging and NP-chunking.

3.3.1 Getting Ready for Test data

Our main task is to identify Name Entity from text. For doing our experiment, we have chosen 100 files from Trove archive related with the two types of sources and had made our two sets of corpus with 50 files each. Two types of files were collected on the basis of recent digital Newspaper data and other set of files are collected from old Newspaper (OCR generated raw data).

At the beginning of test with our written program (Name Entity recognizer), we have used first corpus with 50 files (recent digital Newspaper data) to identify all name entities such as persons, organizations and locations. Moreover, in the second experiment, we have applied our program on other corpus files (OCR generated raw data) for comparing the evaluation of two types.

In our experiment, we used two set of text data (50 files each) that contain unstructured text. By using baseline program, we have generated Name Entities (Person, Organization and Location) from those two sets of text data files. On the other hand, we have produce hand annotated data using those two sets (50 files each) of text data files for evaluating the system outputs.

These hand annotated 100 files (two sets) have been pre-generated for evaluation with the following CoNLL IOB format. The following table is sample of hand annotated data that has generated from a sentence of my test file.

Words	**Label**
Hon	O
Sharon	PERSON-B
Bird	PERSON-I
Is	O
House	ORGANIZATION-B
Of	ORGANIZATION-I
Representative	ORGANIZATION-I
In	O
Australia.	LOCATION

Table 3.1: Example of NER hand annotated file sample data

3.3.2 Evalution of my test data

Evaluation is vitally necessary to find out the progress of Name Entity Recognition (NER) system. NER system usually evaluates depending on human linguistic output comparing with the system generated output.

In our experiment, we have produced list of Name Entity from 100 files (two sets) as test data and we have made hand annotation list of Name Entity from that 100 files to check the system success. Moreover, for doing evaluation, we have calculated the following evaluation on our hand annotated data and system producing data,

1. Equation for Precision (Positive Predictive Value), PPV=TP/(TP+FP)
2. Equation for Recall (True Positive Rate), TPR=TP/(TP+FN)
3. Equation for f-measure, F-Measure=2*((Precision*Recall)/(Precision+Recall))

Here, TP, FP and FN stands for True Positive, False Positive and False Negative respectively.

In the above 3 equations, we have used TP, FP and FN that was calculated from hand annotation data and system generated data. By the following way, we calculate TP, FP and FN.

- TP was calculated by finding the common Entity between the hand annotation data and system generated data.
- FP was calculated by finding the Entities that those are exist in system generated output but not in hand annotated data.
- FN was calculated by finding the Entities that those are exist in hand annotated data but not in system generated data.

3.3.3 NLTK

NLTK is stands for Natural Language Toolkit. NLTK is comprehensive collection of python programs, modules, datasets, collection of corpus (support different languages) graphical demonstrations, sample data and tutorials that give the supports to the developers and researchers for natural language processing and text analysis including customization and optimization (NLTK.org, 2015). NLTK has written by Steven Bird, Ewan Klien and Edvard Loper. Originally, they designed NLTK for teaching purpose for the development of Natural Languages processing (NLP).

NLTK is using a supervised machine learning algorithms that is known as MaxEnt classifier. This classifier maintains a uniform distribution of empirical data from a corpus that has been manually annotated and trained on data from ACE (Automatic Content Extraction) (Johnson, 2013).

However, NLTK is suitable for linguists, students, engineers, researchers, educators and industry users and it is free, easy to use, well documented, modular and extensible for building research systems and platforms.

3.3.3.1 Creating a Baseline for NER System using NLTK

In this section, out task was to create a Baseline for Name Entity Recognition (NER) that will find out Name Entity (NE) from the unstructured text.

In our baseline program, we have processed two sets of test data (50 files each) and run through the baseline program to produce Name Entity on PERSON, ORGANIZATION and LOCATION.

For creating baseline program, we have used NLTK that is the tool for Language processing. This NLTK contains several kinds of methods to perform each step of recognizing process. We have design our program in our own way with collection of those methods to get the proper output.

At the beginning of the program, it is works with two sets of test data and produces the Name Entities. With all of collected Name Entities, our task is to evalute the automatic generated outputs with hand annotation outputs. In the next section, we are showing output of our evaluation results. In the following example, a sample outputs has given for an unstructured single file Name Entity Recognition with our baseline program.

Output (based on one sample file):

- **Name:** ['Her Majesty Queen Elizabeth II', 'Royal Highness', 'Wales Railways', 'Challis House Martin Place Sydney']
- **ORG:** ['Duke', 'Railways', 'Roil', 'Royal Tour', 'Royal']
- **Location:** ['Edinburgh', 'Farm Cove', 'Australian', 'Royal', 'New South Wales', 'Sydney']

3.3.3.2 Evaluation of test data set

In the evaluation phase, whatever the Name Entities are recognized by the system, we compare every output with hand annotated data. In addition, evaluation denote frequencies of instances being true positives (TP), false positives (FP), true negatives (TN), or false negatives (FN) to calculate Precision, Recall and f-measure.

	Precision	**Recall**	**F-Measure**
PERSON	0.23	0.38	0.29
ORGANIZATION	0.06	0.29	0.1
LOCATION	0.27	0.42	0.33
Overall	**0.18**	**0.38**	**0.24**

TABLE 3.2: Output of NLTK result

Furthermore, from our experiment, evaluation graph for NAME, ORGANIZATION and LOCATION individually shows Precision, Recall and f-measure with the NLTK toolkit. In every case, precision is very lower and below 30% (Max 27 %) in Name, Organization and Location that means all of the entities identify by the system, originally very few are correct Name entities.

Moreover, in every case, precision is lower than the recall for Person, Organization and Location entity recognition. Precision is very poor with many false positive and Recall is little bit better but missed 60% of Names.. This graph tells us, NLTK has labelled some of the truly Name Entity correctly, however, it has labelled a whole bunch of innocuous entities as Name Entity.

After process of individual evaluation, we have calculated combine evaluation (evaluation with all Entities together). In combine evaluation, Name entity Recognition (NER) system is providing the following results for NLTK Toolkit based on Person, Organization and Location.

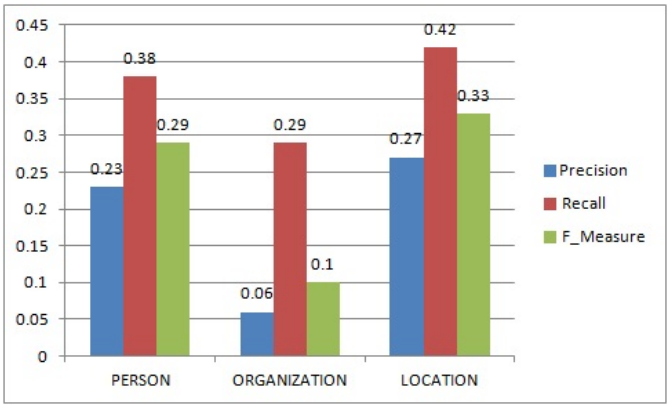


FIGURE 3.1: Evaluation graph for NAME (Person), ORGANIZATION and LOCATION Individually with NLTK

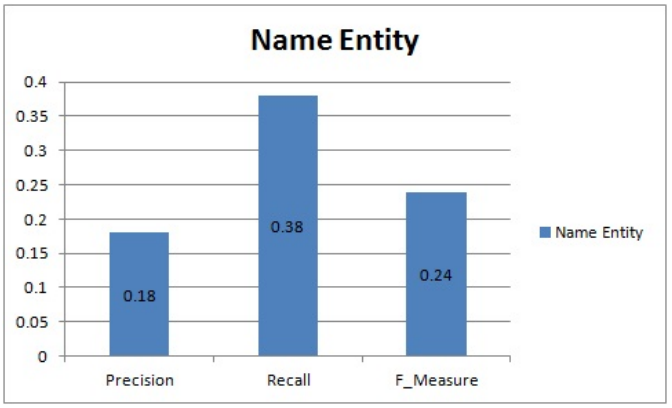


FIGURE 3.2: Evaluation graph for Name Entities by NLTK

In the summary with all entities recognition in our experiment, we saw precession, recall and f-measure are very low (less than 40%) that indicate very low rate of entity recognition by the NLTK toolkit.

3.3.4 Senna Tagger

Senna Tagger is specific software that represents different techniques to manipulate unstructured text in Natural Language Processing (NLP) system. This tagger can describe syntactic information such as POS (Parts-of-Speech) tagging, chunking and parsing. It can also provide semantic information such as semantic role levelling, entity extraction, disambiguation and anaphora resolution (Collobert, 2015) (Al-Rfou and Skiena, 2013).

Moreover, SENNA builds on the idea of deep learning of extracting useful features that has mentioned above from the unlabeled text. For doing this job, Senna tagger has simple interface and high speed processing on small memory footprint.

3.3.4.1 Why do we need Senna Tagger

In our first experiment with NLTK, we have got very poor results. For this reason, we did the experiment with same data sets (two sets) through the Senna Tagger to get better performance. For measuring the output of Senna tagger, we have used the same evaluation technique that has been described at the beginning of this chapter.

3.3.4.2 Getting ready for Senna Tagger

At the beginning of our experiment, our first task was to install the entire relevant package that I need to process Name entity recognition with Senna Tagger. As I have install NLTK before and NLTK is giving us full supports of Senna tagger so that I have used Senna tagger from NLTK package rather than installing the other package. Thus, we can design baseline program for Named Entity Recognition (NER) in our own way to process our test data set with unstructured text.

3.3.4.3 Creating a Baseline for NER System using Senna Tagger

In this section, our task was to create a Baseline for Named Entity Recognition (NER) that will find out Name Entity (NE) from the unstructured text using Senna Tagger. In our baseline program created with Senna Tagger, we have processed two sets of our test data (50 files each) and run through the baseline program to produce or identify Name Entity on PERSON, ORGANIZATION and LOCATION.

For Language processing with Senna Tagger, we have used several kinds of methods to perform each step of the recognition process that are following Information

Extraction architecture for NER. At the end, our task is to evolute the program with the automatic generated outputs and hand annotation outputs.

3.3.4.4 Evaluation of test data set

With the same data sets (two sets) that have collected from Trove News archive, we have done our experiment with Senna Tagger to produce or identify Name Entities. All of the outputs was measured based on true positives (TP), false positives (FP), true negatives (TN), or false negatives (FN) to calculate Precision, Recall and f-measure.

	Precision	**Recall**	**F_Measure**
PERSON	0.28	0.65	0.39
ORGANIZATION	0.23	0.49	0.32
LOCATION	0.62	0.63	0.65
OVERALL	**0.34**	**0.63**	**0.44**

TABLE 3.3: Output of Senna Tagger result

Furthermore, in our evaluation graph (figure 3.3) for NAME, ORGANIZATION and LOCATION individually shows Precision, Recall and f-measure that have generated with the Senna Tagger. In Name and Organization recognition, precision is very lower and below 30% in Name and Organization and more than 60 % in Location recognition that means the Senna Tagger baseline program doing well to identify Location rather than Name and Organization entity.

Moreover, in Name and Organization recognition, precision is lower than the recall for person and Organization entity recognition. This graph (figure 3.3) tells us, Senna Tagger has labelled some of the truly Entity (Person and Organization) correctly, however, it has labelled a whole bunch of innocuous entities as Name Entity. On the other hand, Senna tagger identify Location entities that showing high precision and higher recall with much more improvement of entity identification.

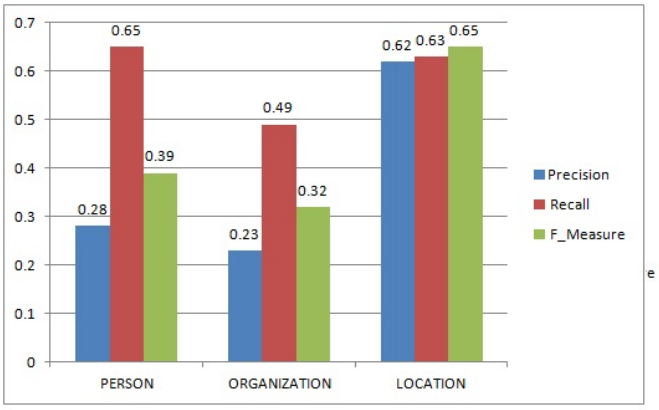


FIGURE 3.3: Evaluation graph for NAME (Person), ORGANIZATION and LOCATION Individually with Senna Tagger

After process of individual evaluation, we have calculated combine evaluation (evaluation with all Entities together). In combine evaluation, Name entity Recognition (NER) system is providing the following results (figure 3.4) for Senna Tagger based on Person, Organization and Location.

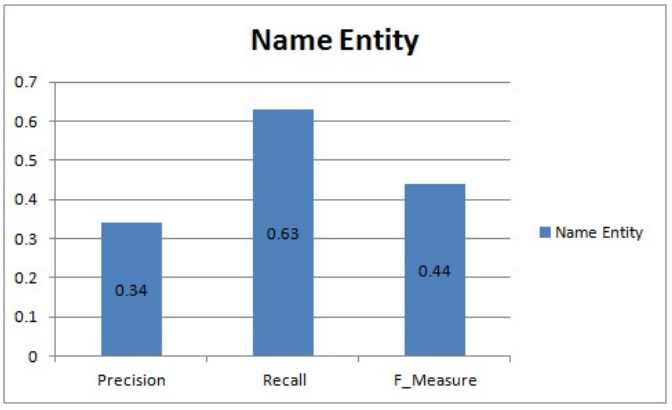


FIGURE 3.4: Evaluation graph for Name Entities by NLTK

In the summary with all entities recognition, in our experiment, we saw precession, recall and f-measure are very low (less than 40%) that indicate very low rate of entity recognition by the Senna Tagger like NLTK toolkit, However, perform much better than NLTK.

3.3.5 Stanford Tagger

Stanford Tagger is specific software that represents different techniques to manipulate unstructured text and applied to various natural language processing tasks including Part-of-Speech tagging (POS), chunking, named entity recognition, and semantic role labelling. Stanford Tagger is Natural Language tools that is develop by The Stanford Natural Language Processing Group in Stanford University (Group, 2015) (Gimpel et al., 2011).

3.3.5.1 Why do we need Stanford Tagger

In our experiment with NLTK and Senna Tagger, we got very poor results altogether except some improvement in Location entity recognition only with Senna Tagger. For this reason, we did the experiment with same data sets (two sets) through the Stanford Tagger to get better performance. For measuring the output of Stanford tagger, we have used the same evaluation technique that has been described at the beginning of this chapter.

3.3.5.2 Getting ready for Stanford Tagger

At the beginning of my experiment, my first task was to install the entire relevant package that I need to process Name entity recognition. At the beginning, we download and install the Stanford tagger from the Stanford Natural Language Processing Group web site and install it for python programming language. This toolkit has all types of methods for python programming language so that we can design baseline program for Named Entity Recognition (NER) in our own way to process our test data set with unstructured text.

3.3.5.3 Creating a Baseline for NER System using Stanford Tagger

In this section, out task was to create a Baseline for Name Entity Recognition (NER) that will find out Name Entity (NE) from the unstructured text using

Stanford Tagger. In our baseline program created with Stanford Tagger, we have processed two sets of our test data (50 files each) and run through the baseline program to produce or identify Name Entity on PERSON, ORGANIZATION and LOCATION.

For Language processing with Stanford Tagger, we have used several kinds of methods to perform each step of the recognition process that have been following Information Extraction architecture for NER. At the end, our task is to evaluate the program with the automatic generated outputs and hand annotation outputs.

3.3.5.4 Evolution of test data set

With the same data sets that have collected from Trove News archive, we have done our experiment with Stanford Tagger to produce or identify Name Entities. All of the outputs was measured based on true positives (TP), false positives (FP), true negatives (TN), or false negatives (FN) to calculate Precision, Recall and f-measure.

Furthermore, in our evaluation output (and graph) for NAME, ORGANIZATION and LOCATION individually shows Precision, Recall and f-measure with the Stanford Tagger. From the output of Stanford Tagger, we can see nice improvement in recognizing Name Entity. Graph is showing Name (Person), Organization and Location recognition have a very good score comparing to the NLTK and Senna Tagger. In the following table, we can see Precision and Recall for all type of Name Entity Recognition (Person, Organization and Location) are more than 75% on average. So, higher precision says, labelled data are recognized as true Entity (75% on average) and higher recall says system has labelled a high amount of innocuous entities as Name Entity (75% on average).

	Precision	**Recall**	**F_Measure**
PERSON	0.81	0.86	0.83
ORGANIZATION	0.7	0.71	0.7
LOCATION	0.86	0.78	0.82
OVERALL	**0.8**	**0.8**	**0.8**

TABLE 3.4: Output of Stanford Tagger result

FIGURE 3.5: Evaluation graph for NAME (Person), ORGANIZATION and LOCATION Individually with Stanford Tagger

After process of individual evaluation, we have calculated combine evaluation (evaluation with all Entities together). In combine evaluation, Name entity Recognition (NER) system is providing the following results for Stanford Tagger based on Person, Organization and Location.

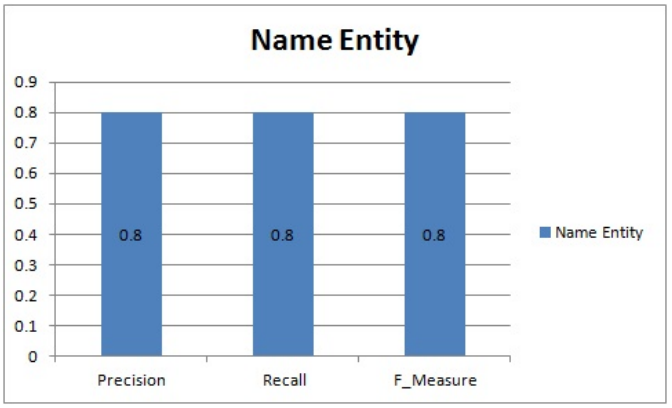


FIGURE 3.6: Evaluation graph for Name Entities by NLTK

In the summary with all entities recognition, in our experiment, we saw precession, recall and f-measure are very high (80%) that indicate very high rate of entity recognition by the Stanford Tagger comparing to the NLTK toolkit and Senna Tagger.

From the table 3.5, we can conclude that two set of data (50 files each) producing the closest result that means, Older Newspaper (OCR) and New Collection of Newspaper (Digitized format) both are doing better with Stanford tagger.

		Data Set-1	**Data Set -2**
NLTK Toolkit	**Precision**	0.18	0.21
	Recall	0.38	0.34
	F-measure	0.24	0.25
Senna Tagger	**Precision**	0.34	0.35
	Recall	0.63	0.69
	F-measure	0.44	0.46
Sanford Tagger	**Precision**	0.80	0.77
	Recall	0.80	0.82
	F-measure	0.80	0.79

TABLE 3.5: Evaluation Comparison with NLTK, Senna and Stanford Tagger

3.4 Discussion

At the beginning of our experiment with the two data sets that we have collected from the trove Newspaper archive, we have chosen three most available and most productive Natural Language Toolkits for identifying Name Entities. Those are NLTK, Senna Tagger and Stanford Tagger.

With all of the Toolkits, we have done the experiment and find out which toolkit is best for our historical newspaper data sets. We found, NLTK and Senna Taggers are not doing well with our data sets and producing very low rate of Precision, Recall and f-measure even though in some cases (Location Entity Recognition) Senna Tagger was showing better result than NLTK. However, this improvement does not show the overall higher level of Entity recognition.

Comparatively, Stanford Tagger has given us a very good level of scoring. Precision, Recall and f-measure are showing us very high rate of Entity recognition on average 80% in our both test data sets. The table (Table 3.5) is giving a quick

comparison with the two data sets and tested Natural Languages Toolkits that we have used.

Furthermore, In our experiment, we found that Stanford Tagger is working well with our historical Newspaper data sets on experiments. From the Stanford tagger documentation, this tagger is specially trained on the Newspaper Corpus and lead to improve evaluation on Newspaper text where the other tagger (NLTK and Senna tagger) are used for many different purposes. Overall performance with Stanford tagger is identical with our experiment data sets and we hope, our bulk data set will work fine with better performance.

3.5 Conclusion:

Overall, all of the experiments with data sets and Natural Language toolkit, we have got a lot of experience on processing techniques for Name Entity Recognition (Person, Organization and Location). Moreover, we have found that Stanford Tagger is working well with our historical Newspaper data sets on experiments. We have found from the Stanford tagger documentation that this tagger is specially trained on the Newspaper Corpus and lead to improve evaluation on Newspaper text.

For this reason, we have decided to use Stanford tagger for our future experiment with bulk amount of text as data set to extract identity recognition and other future improvements.

Chapter 4

Document Classification on Trove Newspaper Texts

4.1 Introduction

Document classification is a part of Machine Learning (ML) process in the field of Natural Language Processing (NLP). This classification task is for allocating a document into the classes or categories. The main purposes to categorize a document are making it easier to manage and sort the documents. As an example, libraries has needed to classify documents into the categories to find out a document easily (kdnuggets, 2015).

Moreover, based on the classification problem, document classifier may be worked on different classes. Document can be classified as text, music, images, news sites, blogs and subject of the document (Type, Author, Publisher, Year etc)(kdnuggets, 2015).

In the Trove Newspaper archive, if we want to search document with a specific name then we will get hits for many different individuals who are sharing the same name. What we would like to do is identify the different people with that specific

name in the documents returned by the search. So if we search for the "Sally Smith", we will get the following sample of list of documents returned.

- DR SALLY CHAPTER I.
- NATIONAL BASEBALL AUSTRALIAN LEAGUE
- TALES OF THE BRAVE.
- NATIONAL ATHLETICS TASMANIA ULTRAMARATHON, etc.

Trove as a large archive of Newspaper with long period, it is common to have same name in different documents. As an example, if we found Sally Smith in one document as a Politician then in other document can be found as hairdresser. To make a solution of this kind of problem, we like to be able to automatically group documents together that will find the same person. We can try to make a group with the documents based on the specific name. For this reason, we are going to do classification experiment on Trove Newspaper for grouping the documents.

In our experiment, we choose two professions (Entertainer and Politician) and non-matching group as Others and try to classify documents into these categories. If we can do this, we can take the results from a search for a name and classify the documents so that we can try to identify different individuals. From the above sample, we might be able to classify the documents on entertainer, Politician and Others and group them together.

4.2 Document Classification

4.2.1 Getting ready for experiment

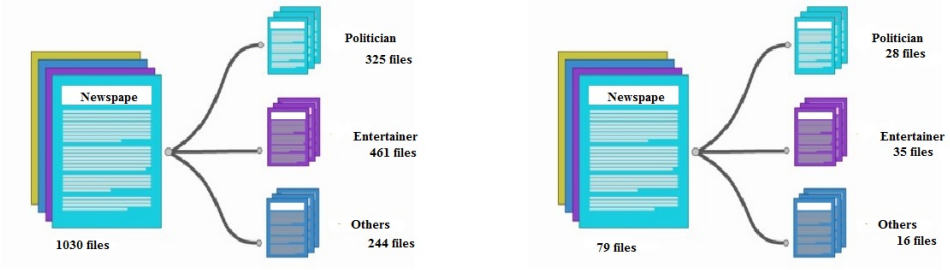
In the document classification, two types of techniques are used such as supervised and unsupervised. In our experiment, we used supervised classification technique with predefined categories and sets of documents. Categories can be change depends on the scenario and text. In our experiment with Newspaper texts, we have

used some popular predefined categories. Categories are labelled into three classes such as Politician, Entertainer and Others.

In our data collection, we searched for some known names that could be expected to be politicians and entertainers and selected 100 documents from each search. Those names are: Walter Cooper (Theatre), George Coppin (Theatre, also politician), Maggie Moore (Actor), Harry Rickards (Theatre owner), Roy Rene (Actor, comedian), Barry Humphries (Actor), Prime Ministers (Barton, Bruce, Curtin, Fisher, Menzies), John Smith (probably other - very common name). Afterwards, we classified all of those documents by reading the document text into the three categories (Politician, Entertainer and Others).

In the final phase of data collection, we have used 1109 classified files in this experiment where 1030 files used for provide training as training data set to a classifier and 79 files used for classifying documents into the specified categories for testing purpose as testing data set. As supervised classification, we have used 1030 files that have collected from three labels or categories for train a classifier. Three categories contain 461 files for Entertainer, 325 files for Politician and 244 files for Others group. For the testing purpose, 79 files are collected from three categories with 35 files for Entertainer, 28 files for politician and 16 for Others group.

Moreover, we have used another unknown 100 files to make classification into the three categories for testing purposes.



(A) Amount for Train files (B) Amount for Test files

FIGURE 4.1: Distribution of Train and test files

4.2.2 Create a program as a trainer

In this section, out first task was to create a program that will work for the document classification.

To build our program, we used python programming language and NLTK (Natural Language Toolkits) Package. With our programming, we have written all of our codes including the plug-ins connections, evaluation and visualization process. For the plug-ins, those are come from NLTK toolkits (NLTK data mining package) to produce results only.

With our program, we have used 1109 files (Historical Newspaper documents) and files has divided into the two sets. First set contains 1030 files to train up the classification trainer and other 79 files have taken as test set for evaluation purpose so that train classifier can identify document into a specified class to generate results.

At the beginning of the process in our program, we have created a classifier using those 1030 files for automatically tag new documents (79 files) with the appropriate labels. First, we have been created a list of documents that have labelled with the appropriate categories. Next, we defined feature extractor for list of documents to provide knowledge for the classifier so that classifier can handle which aspects of data need to pay attention. Afterwards, we have defined feature of each words that has indicating the words existence into the documents for topic identification. After finished the step, we have trained the classifier to level the test data set (79 files) and measuring the accuracy on the test set.

Moreover, we have classified the document by comparing the number of matching terms in the document vectors. In the real world, there has numerous complex algorithms exist for classification such as Support Vector Machines (SVMs), Naive Bayes and Decision Trees. For the comparison, which classifier is doing well with our test data set, we have used the following classifier. Those are:

- **Naive Bayes classifier:** Naive Bayes classifiers are highly scalable and requiring a number of variables (features/predictors) in a learning problem. This classifier is based on Bayes' theorem with independence assumptions between predictors.

- **MNB classifier:** The multinomial Naive Bayes classifier is a linear classifier with multiple event model and multinomial distribution for all the pairs (Predictors).

- **BernoulliNB classifier:** This is the extension of MNB classifier. In the multivariate Bernoulli event model, features are independent Booleans (binary variables) describing inputs. The difference is that BernoulliNB is designed for binary/boolean features where MNB works with occurrence counts only.

- **Logistic Regression classifier:** This classifier is a regression model and work with categorical dependent variable. Algorithm is working on binary dependent variable that can only works with two value such as pass or fail, win or loss or present or absent.

- **SGDClassifier classifier:** This classifier is works with discriminative learning of linear classifiers under convex loss functions where linear classifier are Support Vector Machines and Logistic Regression.

- **SVC classifier :** This is a supervised learning methods used for classification and regression tasks by a hyperplane. This classifier has developed from statistical learning theory.

- **LinearSVC classifier:** This classification is similar to SVC classifier and implemented with libliner rather than libsvm with the more flexibility with large number of samples.

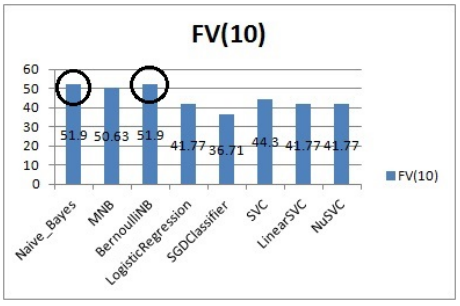
- **NuSVC classifier:** This is also similar to SVC based on libsvm and uses a parameter to control the number of support vectors.

4.2.3 Result

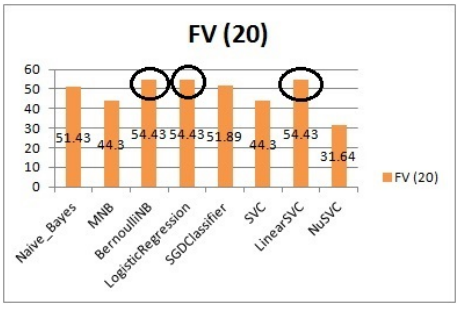
After execute our program with the training data for providing training to different classifiers that has mentioned before, our task was to find out the accuracy for the test data set. In our experiment, we have tested our data for different size of feature vectors in the training season such as 10, 20, 50, 100, 250, 500, 750, 1000 collecting top most words. We wanted to see which optimum features are giving us the highest result. The following table (Figure 4.2) belongs to the accuracy measurements that we have found after running trained classifiers with the different amount of feature vectors.

	FV(10)	FV (20)	FV (50)	FV (100)	FV (250)	FV (500)	FV (750)	FV (1000)
Naive_Bayes	**51.9**	51.43	52.16	**53.16**	**50.63**	**54.43**	**55.69**	**50.63**
MNB	50.63	44.3	43.04	44.3	44.3	44.3	49.36	44.3
BernoulliNB	**51.9**	**54.43**	**54.43**	**53.16**	**50.63**	**54.43**	**55.69**	**50.63**
LogisticRegression	41.77	**54.43**	46.84	44.3	44.3	41.77	46.83	43.04
SGDClassifier	36.71	51.89	46.84	37.97	21.51	50.63	29.11	45.57
SVC	44.3	44.3	44.3	44.3	44.3	44.3	44.3	44.3
LinearSVC	41.77	**54.43**	48.1	44.3	44.3	44.3	46.83	43.04
NuSVC	41.77	31.64	34.18	20.25	39.24	45.57	46.83	29.11

FIGURE 4.2: Accuracy for different type of classifiers with different amount of features



(A) Feature Vector (FV) with top 10 words

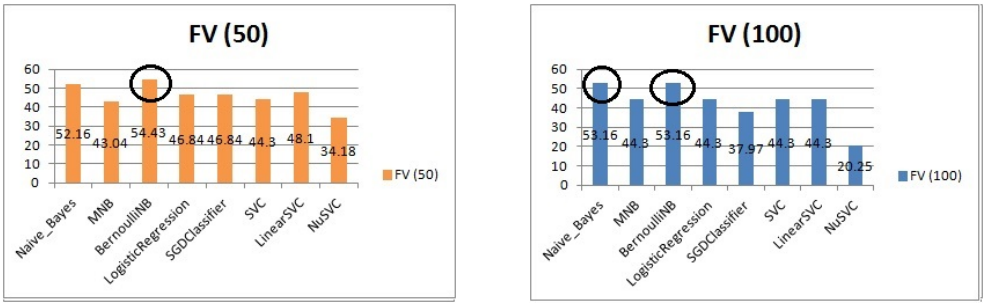


(B) Feature Vector (FV) with top 20 words

FIGURE 4.3: Accuracy for feature vector with top 10 and 20 words

From the figure 4.3(A), we have seen that the feature vector with top most 10 words generating highest value of 51.9 for Naive Bayes classifier and BernoulliNB classifier but other classifiers are generating lower scores than 51%. Another highest value of 54.43 generated (Figure 4.3(B)) by the feature vector with top most

20 words with BernoulliNB classifier, LogisticRegression classifier and LinearSVC classifier where as the other classifiers are generating lower value than 52%.

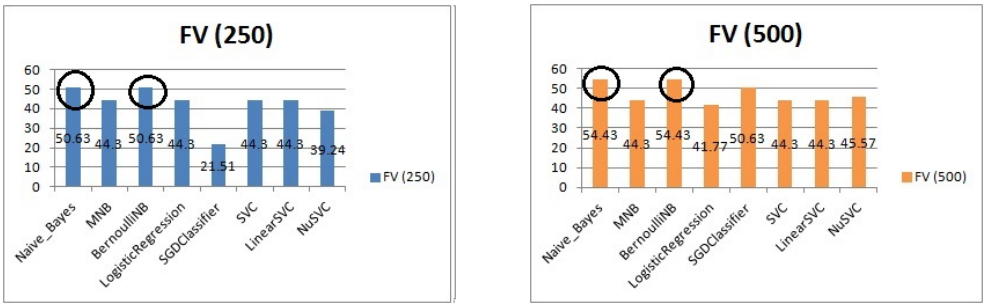


(A) Feature Vector (FV) with top 50 words

(B) Feature Vector (FV) with top 100 words

FIGURE 4.4: Accuracy for feature vector with top 50 and 100 words

From the figure 4.4(A), we have seen that the feature vector with top most 50 words generating highest value of 54.43 for BernoulliNB classifier only but other classifiers are generating lower scores less than 49%. Another highest value of 53.16 generated (Figure 4.4(B)) by the feature vector with top most 100 words with Naive Bayes classifier and BernoulliNB classifier where as the other classifiers are generating lower value than 45%.



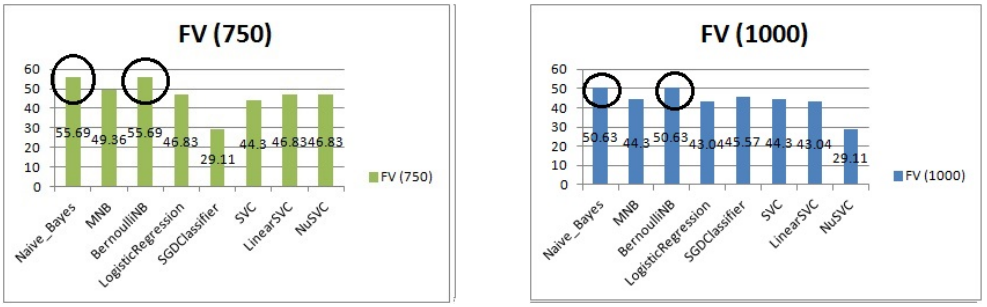
(A) Feature Vector (FV) with top 250 words

(B) Feature Vector (FV) with top 500 words

FIGURE 4.5: Accuracy for feature vector with top 250 and 500 words

From the figure 4.5(A), we have seen that the feature vector with top most 250 words generating highest value of 50.63 for Naive Bayes classifier and BernoulliNB classifier but other classifiers are generating lower scores less than 45%. Another highest value of 54.43 generated (Figure 4.5(B)) by the feature vector with top

most 500 words with Naive Bayes classifier and BernoulliNB classifier where as the other classifiers are generating lower value than 50%.



(A) Feature Vector (FV) with top 750 words

(B) Feature Vector (FV) with top 1000 words

FIGURE 4.6: Accuracy for feature vector with top 750 and 1000 words

From the figure 4.6(A), we have seen that the feature vector with top most 750 words generating highest value of 55.69 for Naive Bayes classifier and BernoulliNB classifier but other classifiers are generating lower scores less than 50%. Another highest value of 54.43 generated (Figure 4.6(B)) by the feature vector with top most 1000 words with Naive Bayes classifier and BernoulliNB classifier where as the other classifiers are generating lower value than 45%.

From the figure 4.7, we have seen that the feature vector with top most 750 words generating highest value of 55.69 for Naive Bayes classifier and BernoulliNB classifier but other classifiers are generating lower scores less than 50%. Next highest value of 54.43 generated by the feature vector with top most 500 words with Naive Bayes classifier and BernoulliNB classifier where as the other classifiers are generating lower value than 45%.

From the graph (figure 4.7), we can observe that there are only two classifiers (such as Naive Bayes classifier and BernoulliNB classifier) giving us highest accuracy of 55.69 for features vectors of 750. For this cause, we can select only one classifier for further analysis and experiment with large documents between two highest score

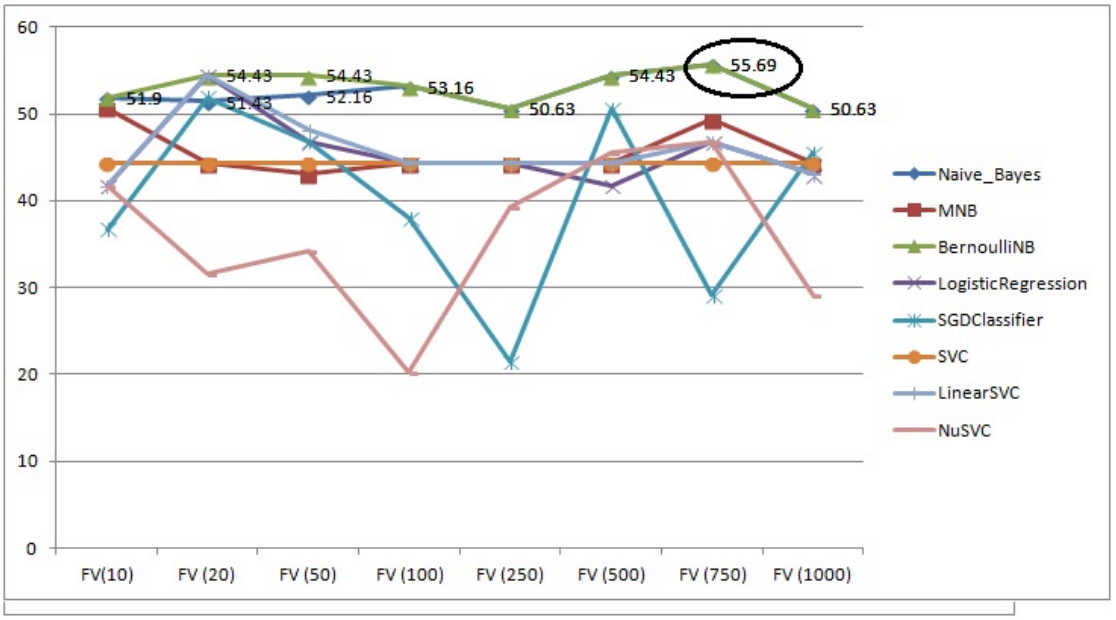


FIGURE 4.7: Highest accuracy (Black Circle) for various set of Feature Vector (FV)

classifier. Therefore, we select Naive Bayes classifier for our future experiment and analysis.

Moreover, in the next phase, we are going to describe the performance of a classification model or a classifier and represent information regarding actual and predictive classifications to evaluate the performance in matrix format. This classification model works on a set of text data where the true data are known. Here, our test data is 79 documents files that we used for finding the accuracy. We have already selected Naive Bayes classifier with 750 feature vector as our optimum result so that we have selected the following confusion matrix related to Naive Bayes classifier with 750 feature vectors where row is reference and column represent test data.

	Entertainer	**Others**	**Politician**	Row Total
Eentertainer	23	0	12	35
Others	10	0	6	16
Politician	7	0	21	28
Column Total	40	0	39	79

TABLE 4.1: Confusion Matrix for Naive Bayes classifier.

The following list of rates that are computed from the above confusion matrix, see table 4.2:

Accuracy	0.7
Misclassification	0.3
True Positive Rate (TPR)	0.56
False positive Rate (FPR)	0.22
Specificity	0.78
Precision	0.56
F1 (Entertainer)	0.61
F1 (Politician)	0.63
F1(Others)	0

TABLE 4.2: List of other statistical results on documents classification

Moreover, from the above contingency table (Table 4.1) (confusion matrix), we have done the statistical analysis that is known as Fisher's exact test. This test give us exact p-value (P= 0.005334) and we found that the value is less than the rejection rate (0.05 or 5%).

4.3 Discussion

At the beginning of the experiment, we have done supervised classification for creating our model based training set. We have predefined our three categories such as Politician, Entertainer and Others and documents within the training data set are tagged with that three category labels. Next, we have done training on the several classifiers and tested our test data set. The result made by several classifiers was varied and we have selected Naive Bayes Classifier as a top value generator (Accuracy=55.69).

In most cases, we found that naive bayes classifier do better than other classifiers. This is because of the following reasons:

- Promising results for textual tasks like Newspaper, HTML, blogs etc.

- Naive Bayes classifier can handle missing data such as if there is a missing value for an attribute then it can be ignored when preparing model and calculated a class value.
- Naive Bayes can re-calculate the probabilities as the data changes (daily, monthly or yearly)
- Naive Bayes classifier with categorical attributes, it is easy to calculate a frequency for each observation.
- Feature selection in Naive Bayes classifier is the selection of data attributes (training dataset) that best characterize a predicted variable.

Afterwards, we have calculated confusion matrix (contingency table) and has given us some other statistical list of data such as Precision, Recall, True Positive Rate, False Positive Rate etc. At the end, we have done Fisher's exact test where p-value calculated for statistical analysis and we got 0.005334 as p-value that said the rejection rate was less than 5%.

4.4 Conclusion

In conclusion, we have represented out thoughts to classifying documents into the categories. We found that with the right features selection and a large enough training data set, we can trained a classifier to classify Newspaper documents with a good accuracy. In this way, we can group documents that we have got from the search and can easily group them together for better search result with related (categories) documents.

Chapter 5

Discussion

An information extraction system has searched entities and relation from large bodies of unstructured text to populate well organized databases. As a result, we can use these databases to find out answer for specific question, NLP-based search, Speech recognition, Machine translation and Knowledge discovery.

In the first part of our thesis contain the literature reviews about the Name Entity recognition. We have pointed out to show the definition of Name Entities and what kind of Name Entities that we have search for our project. Moreover, we have included the challenges of NER and features of Name Entities. We have defined step by step architectural process with some examples. Next, we have included the technique of learning process for identifying Name Entities and techniques of evaluation the whole process.

Second part contains the information about the NER implementation process. Here, we have implemented our Name Entity recognizer with three different popular tools based on NLTK, Senna Tagger and Stanford to process our Newspaper texts. At the beginning of the process, we have created a Baseline program for Name Entity Recognition (NER) that will find out Name Entity (NE) from the unstructured text. In our baseline program, we have processed two sets of test

data (50 files of each) and run through the baseline program to produce Name Entity on PERSON, ORGANIZATION and LOCATION.

We have evaluated our three systems like NLTK, Stanford tagger, Senna Tagger, to find out the best suited Name Entity Recognizer. All of these three systems are working under Supervised Learning (SL) methods. Supervised learning system is working on annotated entries that can be used for mapping test data with training data. At the beginning of the evaluation process, we have collected the Name Entities from two sets of files and evaluated the automatic generated outputs (NE as Test data) with hand annotation outputs (Training data). We have found that the Name Entity recognizer program with Stanford tagger is the best system with Historical Newspaper texts for its optimum result.

In the third part of the thesis, we have developed a method to classify Newspapers. We have done supervised classification for creating our model based training set. We have worked on defining three categories labels such as Politician, Entertainer and Others. Next we have done training on the several classifiers and tested our test data set applying the training on several classifiers and tested out data with them. After the test, we have selected Naive Bayes Classifier as a top value generator (Accuracy=55.69). In addition, we have evaluated the system with confusion matrix (contingency table) to find out the different statistical analysis results and validating results with Fisher Exact Probability test.

Chapter 6

Conclusion

As a part of Information Extraction process, NER is helping us to build knowledge from unstructured text to structure format of identifying objects. Within this project of Name Entity Recognition system, we have worked for producing a system that will be well suited to identify Name Entities properly from the Newspaper texts.

All of the experiments with data sets and Natural Language toolkits, we have got a lot of observations on processing techniques for Name Entity Recognition (Person, Organization and Location). Moreover, we have found that Stanford Tagger is working well with our historical Newspaper data sets on experiments where this tagger is specially trained on the Newspaper Corpus and lead to improve evaluation on Newspaper text. These improvements demonstrate that the challenges for recognizing of the Name Entity (NE) have got a better shape but still need some more improvements.

Moreover, the focus of our task was to build a method that will classify Newspaper documents into the predefined categories. Those categories are Politician, Entertainer and Others. We have found that with the right features selection and a large enough training data set, we can trained a classifier to classify Newspaper documents with a good accuracy. In this way, we can group documents that we have got from the Search or Corpa and can easily group them together for better

search result with related (categories) documents.

In future, our work will be representing meaning from unlimited set of Newspaper texts to store of Knowledge. We can do it with feature based grammars by analysis of sentences. Moreover, we can works on managing Linguistic data that will identify existing wrong format to a suitable format for OCR generated Newspapers.

Bibliography

AFNER (2015). Named Entity Recognition. http://afner.sourceforge.net/what.html.

aihorizon (2015). Machine Learning, Part I: Supervised and Unsupervised Learning. http://www.aihorizon.com/essays/generalai.

Al-Rfou, R. and Skiena, S. (2013). Speedread: A fast named entity recognition pipeline. *arXiv preprint arXiv:1301.2857*.

Alfred, R., Leong, L. C., On, C. K., and Anthony, P. (2014). Malay named entity recognition based on rule-based approach. *International Journal of Machine Learning & Computing*, 4(3).

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

Bikel, D. M., Miller, S., Schwartz, R., and Weischedel, R. (1997). Nymble: a high-performance learning name-finder. In *Proceedings of the fifth conference on Applied natural language processing*, pages 194–201. Association for Computational Linguistics.

Brill, E. (1992). A simple rule-based part of speech tagger. In *Proceedings of the workshop on Speech and Natural Language*, pages 112–116. Association for Computational Linguistics.

ClearTK (2015). ClearTK: Machine Learning for UIMA. https://code.google.com/p/cleartk/wiki/TutorialNamedEntityChunkingClassifier.

Cloud, M. (2015). Recognizing entities in a text: not as easy as you might think! http://www.meaningcloud.com/blog/named-entities-recognition-ner/.

Collobert, R. (2015). Senna Tagger. http://ml.nec-labs.com/senna/.

E., B. S. K. and Loper, E. (2009). *Natural Language Processing with Python*. OReilly Media, Inc, 1 edition.

Europeana (2015). Named Entity Recognition for digitised newspapers. http://www.europeana-newspapers.eu/named-entity-recognition-for-digitised-newspapers/.

Ferro, L., Gerber, L., Mani, I., Sundheim, B., and Wilson, G. (2007). Tides-2005 standard for the annotation of temporal expressions. 2005. *MITRE Corporation*.

Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J., and Smith, N. A. (2011). Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 42–47. Association for Computational Linguistics.

Grishman, R. and Sundheim, B. (1996). Message understanding conference-6: A brief history. In *COLING*, volume 96, pages 466–471.

Group, T. S. N. L. P. (2015). Stanford Log-linear Part-Of-Speech Tagger. http://nlp.stanford.edu/software/tagger.shtml.

Gunopulos, D., Hofmann, T., Malerba, D., and Vazirgiannis, M. (2011). *Machine Learning and Knowledge Discovery in Databases, Part III: European Conference, ECML PKDD 2010, Athens, Greece, September 5-9, 2011, Proceedings*, volume 6913. Springer Science & Business Media.

Herald, T. S. M. (2015). Malcolm Turnbull wins Liberal leadership challenge. http://www.smh.com.au/federal-politics/the-pulse-live/politics-live-september-14-2015-20150913-gjlqy5.html.

Holley, R. (2015). Trove: Innovation in Access to Information in Australia. http://www.ariadne.ac.uk/issue64/holley.

Hong, G. (2005). Relation extraction using support vector machine. In *Natural Language Processing–IJCNLP 2005*, pages 366–377. Springer.

Johnson, M. (2013). http://www.mattshomepage.com/blogs/feb2013/liftingthehood.html. http://www.mattshomepage.com/blogs/feb2013/liftingthehood.html.

kdnuggets (2015). Text Analysis 101: Document Classification. http://www.kdnuggets.com/2015/01/text-analysis-101-document-classification.html.

Marrero, M., Urbano, J., Sánchez-Cuadrado, S., Morato, J., and Gómez-Berbís, J. M. (2013). Named Entity Recognition: Fallacies, Challenges and Opportunities. *Computer Standards & Interfaces*, 35(5):482–489.

McCallum, A. and Li, W. (2003). Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 188–191. Association for Computational Linguistics.

Miller, D., Boisen, S., Schwartz, R., Stone, R., and Weischedel, R. (2000). Named entity extraction from noisy input: speech and ocr. In *Proceedings of the sixth conference on Applied natural language processing*, pages 316–324. Association for Computational Linguistics.

Nadeau, D. and Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26.

National Library, A. (2015). Australian newspaper digitisation program. http://www.nla.gov.au/content/newspaper-digitisation-program.

Nguyen, G., Dlugolinskỳ, Š., Laclavík, M., and Šeleng, M. (2013). Token gazetteer and character gazetteer for named entity recognition. In *8th Workshop on Intelligent and Knowledge Oriented Technologies: WIKT*, pages 1–6.

NLTK.org (2015). Natural Language Toolkit. http://www.nltk.org/.

Packer, T. L., Lutes, J. F., Stewart, A. P., Embley, D. W., Ringger, E. K., Seppi, K. D., and Jensen, L. S. (2010). Extracting person names from diverse and noisy ocr text. In *Proceedings of the fourth workshop on Analytics for noisy unstructured text data*, pages 19–26. ACM.

Piskorski, J. and Yangarber, R. (2013). Information extraction: Past, present and future. In *Multi-source, multilingual information extraction and summarization*, pages 23–49. Springer.

Rau, L. F. (1991). Extracting company names from text. In *Artificial Intelligence Applications, 1991. Proceedings., Seventh IEEE Conference on*, volume 1, pages 29–32. IEEE.

Riloff, E., Jones, R., et al. (1999). Learning dictionaries for information extraction by multi-level bootstrapping. In *AAAI/IAAI*, pages 474–479.

Saha, S. K., Sarkar, S., and Mitra, P. (2008). Gazetteer preparation for named entity recognition in indian languages. In *IJCNLP*, pages 9–16.

Steven Bird, E. K. and Loper, E. (2015a). Extracting Information from Text. http://www.nltk.org/book/ch07.html.

Steven Bird, E. K. and Loper, E. (2015b). Extracting Information from Text. http://www.nltk.org/book/ch06.html.

Steven Bird, E. K. and Loper, E. (2015c). Extracting Information from Text. http://www.nltk.org/book/ch03.html.

textminingonline.com (2015). Dive into nltk, part ii: Sentence tokenize and word tokenize. http://textminingonline.com/dive-into-nltk-part-ii-sentence-tokenize-and-word-tokenize.

Tjong Kim Sang, E. F. and De Meulder, F. (2003). Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 142–147. Association for Computational Linguistics.

Trove (2015). Trove is....., Trove Help Centre. http://help.nla.gov.au/trove/using-trove/getting-to-know-us/trove-is.