Named Entity Recognition
and
Relation Extraction

Bachelor's Thesis
submitted in partial fullfilment of
the requirements for the degree of
Bachelor of Technology
in
Computer Science and Engineering

By

SAI KIRAN NARAGAM
N100638

Under the Guidance of
Mr. A.Udaya Kumar
Asst. Professor
Dept. of Computer Science and Engineering
RGUKT Nuzvid

Rajiv Gandhi University of Knowledge Technologies, Nuzvid.
Krishna District,Andhra Pradesh.

Rajiv Gandhi University of Knowledge Technologies
(A.P. Government Act 18 of 2008)
RGUKT NUZVID, Krishna District - 521202

CERTIFICATE OF COMPLETION

This is to certify that the work entitled, **Entity Recognition and Relation Extraction** is the bona fied work of **NARAGAM SAI KIRAN**, ID No: **N100638**, carried out under my guidance and supervision, for the partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering.

Mr. Ambati Udaya Kumar,
Assistant Professor,
Dept.of CSE.

Rajiv Gandhi University of Knowledge Technologies
(A.P. Government Act 18 of 2008)
RGUKT NUZVID, Krishna District - 521202

CERTIFICATE OF EXAMINATION

This is to certify that the work entitled, **”Named Entity Recognition and Relation Extraction”** is the bonafide work of **NARAGAM SAI KIRAN**, ID No: **N100638** and here by accord our approval of it as a study carried out and presented in a manner required for its acceptance in the partial fulfillment of the requirement for the award of the degree of Bachelor of Technology for which it has been submitted. This approval does not necessarily endorse or accept every statement made, opinion expressed or conclusion drawn, as a recorded in this thesis. It only signifies the acceptance of this thesis for the purpose for which it has been submitted.

Mr. Ambati Udaya Kumar,
Assistant Professor,
Dept.of CSE.

Mr. Amit Patel,
Lecturer,
Dept.of CSE.

Rajiv Gandhi University of Knowledge Technologies
(A.P. Government Act 18 of 2008)
RGUKT NUZVID, Krishna District - 521202

DECLARATION

I **NARAGAM SAI KIRAN**, with ID No:**N100638** hereby declare that t he project report entitle **Named Entity Recognition and Relation Extraction** done by me under the guidance of **Mr. Ambati Udaya Kumar,M.Tech** is submitted for the partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering during the academic session August 2015 April 2016 at RGUKT Nuzvid.

I also declare that this project is a result of my own effort and has not been copied or imitated from any source. Citations from any websites are mentioned in the references.

The results embodied in this project report have not been submitted to any other university or institute for the award of any degree or diploma.

N. SAI KIRAN,
N100638.

Date : ______________
Palce: ______________

Acknowledgements

I would like to thank my parents,guide and friends

Abstract

Abstract goes here.

Contents

Chapter 1

Introduction

When we have large amount of (previous)data we might want to extract some useful information out of it, and use it as summary. or we can predict the future events by learning from the data at hand.
Most of the time data that is available for use in un-structured form like Natural Language Text rather than structured form like tables. It is easy to extract required information or answer a question if the data we are working on has structured form.

But it is difficult to handle unstructured data. Because Natural Language Processing(NLP) that works on unstructured data is still developing.

The amount of natural language text that is available in electronic form is truly staggering, and is increasing every day. However, the complexity of natural language can make it very difficult to access the information in that text[1].

If we instead focus our efforts on a limited set of questions or ”entity relations,” such as ”where are different facilities located, ” or ”who is employed by what company,” we can make significant progress.[1]

Here we are trying to understand the given text and find the limited relevant parts of it. This is what the researchers called as **Information Extraction**.

1.1 Aim

Identifying named entities and working out the relationship between them using hand-written rules with regular expressions. For most of the questions often the answers be named entities.

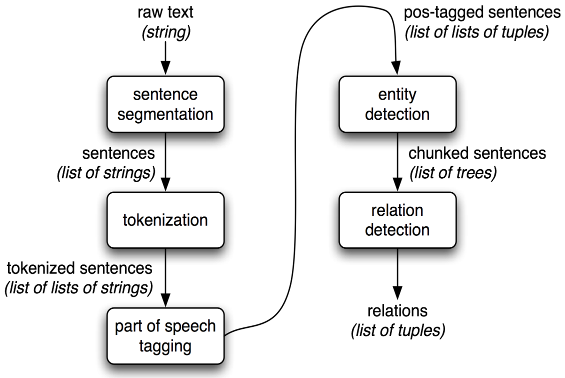


Figure 1.1: Simple Pipeline Architecture for an Information Extraction System[1]

Chapter 2

Named Entity Recognition

2.1 Introduction

Named Entity Recognition is an important sub task of *Information Extraction* ,in this we are going to find and classify (into different classes like PERSON,ORGANIZATION and LOCATION etc.). concrete names of people,organizations,locations and quantities etc.

We are interested in *Named* Entity Recognition. Because not all entities are attached with a name (specific).

For the literature survey on named entity recognition, please refer[9].

2.2 Named Entity Recognition as Tagging

Bikel et. al[1] mapped the Named the Entity Recognition problem very directly into tagging problem. Where where they considered all the named entity classes and an extra NOT-A-NAME tag.

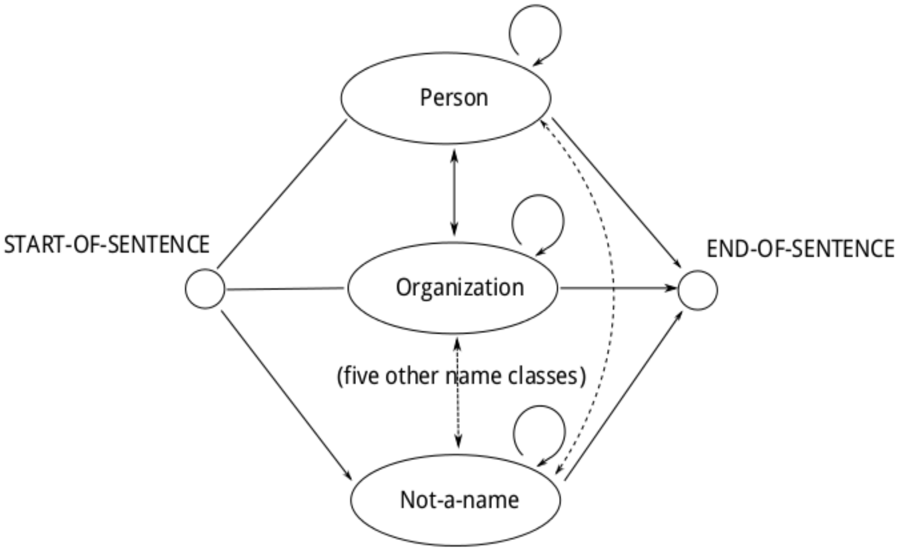


Figure 2.1: Stage Diagram of NER as Tagging by Bikel et. al

[1] http://ilk.uvt.nl/~toine/research/bikel-1999.pdf

They used hand-tagged corpus to train their model(Hidden Markov Model[6]) and some word-features to deal with low-frequency words.

2.3 Named Entity Recognition with PoS tagging & Chunking

Now we have lot of parts of speech tagged corpora (especially for english) as we can use it for machine translataion and many other applications. Here we are going to use PoS tagging for NER.

After having natural language senetences with their underlying *tag sequences* we group the tags into named entities.

2.3.1 PoS Tagging[2]

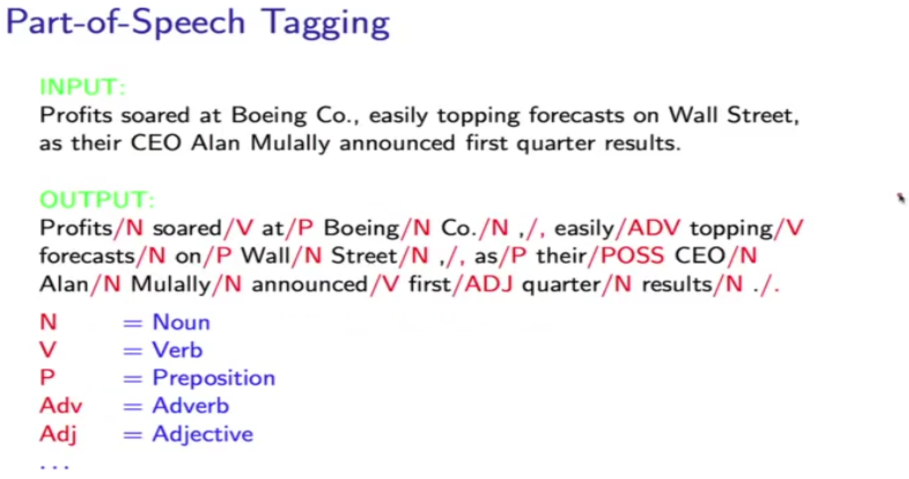
Parts of speech tagging problem is to determine the parts of speech of a particular instance of word. The intuition of PoS tagging is presented in below image 2.2[3].

Tags may vary depending on corpus we are dealing with. For example, tag set of The Brown Corpus[4] and P.O.S tag set of The Penn Treebank[5] To check in NLTK, execute and *nltk.help.brown_tagset()*,*nltk.help.upenn_tagset()* respectively for the brown corpus tagset and Penn Treebank tagset.

If we want to write a tagger then we need *large amount of labled corpus*[6]. Which in this case are Penn Tree Bank tagged corpus or Brown corpus.

For more insights on writing a parts of speech tagger in NLTK, please refer[2]

For theoritical understanding of a tagger. We learned how a Hidden Markov Model tagger[6] works. Here for this problem we used NLTK's default implementation of tagger(*nltk.tag()*) as it is recommeneded for better results.



Part-of-Speech Tagging

INPUT:
Profits soared at Boeing Co., easily topping forecasts on Wall Street, as their CEO Alan Mulally announced first quarter results.

OUTPUT:
Profits/N soared/V at/P Boeing/N Co./N ,/, easily/ADV topping/V forecasts/N on/P Wall/N Street/N ,/, as/P their/POSS CEO/N Alan/N Mulally/N announced/V first/ADJ quarter/N results/N ./.

N	= Noun
V	= Verb
P	= Preposition
Adv	= Adverb
Adj	= Adjective
...	

Figure 2.2: Parts-of-Speech Tagging

[2] For more details:
Tagging Problems, and Hidden Markov Models of [7] and POS Tagging of [8]

[3] Slide from The Tagging Problem of [7]

[4] https://www.comp.leeds.ac.uk/ccalas/tagsets/brown.html

[5] https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

[6] For more insights of Machine Learning techniques I feel, [4] is a good source

2.3.2 Named Entity Chunking[7]

After tagging comes the named entity chunking. We'll group the pos tags into named entities (if possible intuit its class). Chunker is also a tagger that is trained on some corpus. One of the most useful sources of information for NP-chunking(Noun Phrase-chunking) is part-of-speech tags. This is one of the motivations for performing part-of-speech tagging in our information extraction system

Here for this problem we used NLTK's default implementation of named entity chunker (*nltk.ne_chunk()*) as it is recommended for better results. It can be used for multi- class(PERSON, LOCATION, ORGANIZATION and GPE) or binary class(NP).

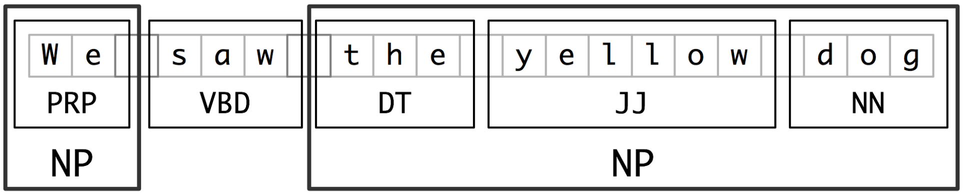


Figure 2.3: Chunking

For more insights on develping a chunker, please refer [1]. Here it described how to create a basic chunker and a chunker that can learn from data for good performance.

Morphology of words to identify Noun Phrases. And to identify the class of Noun Phrase(Named Entity) the chunker will use context of the Noun Phrase. There are many formats to represent Named Entities, those are IB(Every token is **I**n the chunk or **B**egining of the chunk), IOB(Every token is **I**n the chunk or **O**ut of the chunk or **B**egining of the Chunk),tree representation etc.

[7] For more details: refer [1]

Chapter 3

Relation Extraction

Relation Extraction is an important component of Information Extraction.

Using the Named Entities and clever patterns we extract relation. These rules can get high precision as they are specific.

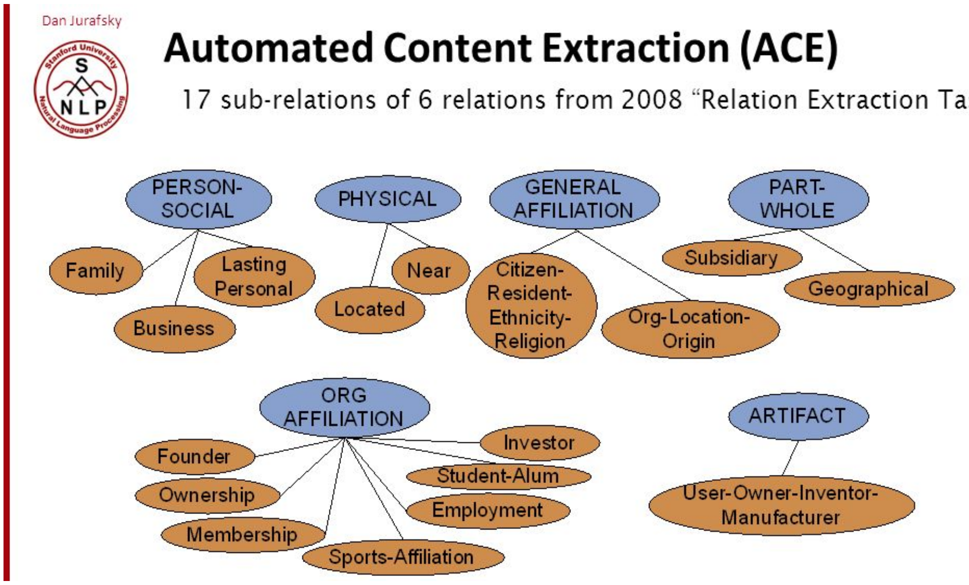
We will focus on the simpler task of extracting **relation triples**. Relation triples are of the form *(Named Entity,Relation,Named Entity)*.

We will use *relextract* module of *Python* for this.

Rule based relation extraction We can create new structured knowledge bases by relation extraction Questions that are asked in natural language can be converted in to a query to a structured knowledge base.

So, Here is a question for us. *Which relations should we extract ?* It depends on how many classes of entities we are able to extract in *Named Entity Recognition*.

But, a set of relations comes from the *Automated Content Extraction (ACE)* task. 3.1[1]



Dan Jurafsky

Automated Content Extraction (ACE)

17 sub-relations of 6 relations from 2008 “Relation Extraction Task”

Figure 3.1: ACE Relation set

[1] Slide from: https://class.coursera.org/nlp/lecture/138

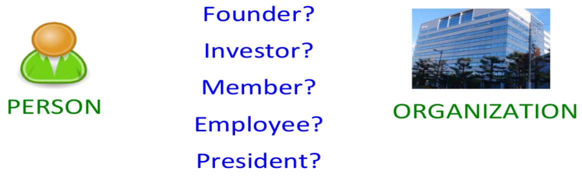


Figure 3.2: Relation between PERSON and ORGANIZATION named entity classes

Figure 3.2[2] gives basic intuition of relation between entities.

3.1 How to extract relations ?

We can use:

- Hand-written patterns
- Supervised,semi-supervised and unsupervised machine learning

We are using *only* Hand-written patterns to extract relations as it is simplest way. Here we are trying to extract relations between *specific entities*.

We used **regular expressions** [5] in Python to write patterns to extract relations.

3.1.1 Procedure followed

- First identify the named entities (POS tagging then Chunking).
- Then we group a Noun phrase with its left context.
  Now we'll have document as list of tuples.
  Ex: `[(String1,Named Entity1), (String2,Named Entity2) , (String3,Named Entity3) , ... ]`
  Here `Named Entity1,Named Entity2,Named Entity3` are tree representations of Noun Phrase.
  Now we take two consecutive tuples and add them to create *semi*relation dictionaries.
  That dictionary contains **key**,**value** as described in table 3.1
- Apply hand-written rules on filler,right context and left context to extract relations

For more information on relation extraction, please refer [1] for more information. The rules we wrote for this project can be found here[3]

3.1.2 Positives of Hand-written rules

- String patterns tends to be high-precision.
- Works well for specific domains/entities.

[2] https://class.coursera.org/nlp/lecture/139

[3] https://github.com/saikiran638/MyProjects/blob/master/FinalYearProject/RelationRules.py

KEY	**VALUE**
`filler`(text between two Named Entities)	`String2` which is leftcontext of Named Entity2
`lcon`(left context of the relation)	`String1` which is leftcontext of Named Entity1.
`objclass`(Class of object of the relation)	`Root of the Named Entity2 tree structure`
`objsym`(Normalized text of object with no white space	`Normalized object with underscore in palce of space.`
`objtext`	`objtext`
`rcon`(right context of the relation)	`String3`, which is right context of the Named Entity 2
`subjclass`(Class of subject of the relation)	`Root of the Named Entity1 tree structure.`
`subjsym`(Normalized text of suject with no white space)	`Normalized subject with underscore in palce of space.`
`subjtext`	`subject text`
`untagged_filler`	`filler with no POS tags`

Table 3.1: Explanation of Key,Value pairs of `semi`relation dictionary

3.1.3 Negatives of Handiwritten rules

- It's difficult to think of all possible patterns.
- We don't want to fix the entities for relation extraction.

Chapter 4

Results and Observations

4.1 Observations

Named Entity Chunker provided in Python's NLTK (*nltk.ne_chunk*) considers morphology of words while chunking. Make sure that words are not normalized(HMMTagger requires words to be normalized) while chunking.

Trained HMMTagger(for PoS tagging) available in Python's NLTK with *treebank* tagged corpus and tried chunking but performance is not as good as NLTK's Recommended PoS tagger (*nltk.tag*). So, we used NLTK's Recommended PoS tagger for tagging sentences. It was found that NLTK's current recommended tagger is 'Averaged Perceptron Tagger'(It might change over time).

4.2 Results

This project is hosted on `github`[1] , you can access source code and documents of it.

For testing we used *BBC*'s corpus.

[1] https://github.com/saikiran638/MyProjects/tree/master/FinalYearProject

Chapter 5

Conclusion and Future Work

5.1 Conclusion

From the text we are going to extract information should a formal writing. For informal writings it won't work well as every steps assumes the formal nature of the text.

5.2 Future Work

Indian languages have very less *tagged* corpus compared to English. We require large amount of *tagged corpus* to train classifiers.

- To apply Named Entity Recognition & Information Extraction for Indian languages.

Bibliography

[1] **Extracting Information from Text**
http://www.nltk.org/book/ch07.html

[2] **Categorizing and Tagging Words**
http://www.nltk.org/book/ch05.html

[3] **Information Extraction in NLTK**
http://www.nltk.org/howto/relextract.html

[4] Prof.Yaser Abu-Mostafa,California Institute of Technology
Learning from Data(Online Course)

[5] **Regular Expression in Python**
https://docs.python.org/2/howto/regex.html#regex-howto

[6] Prof. Michael Collins,Columbia University
Tagging Problems, and Hidden Markov Models
http://www.cs.columbia.edu/~mcollins/hmms-spring2013.pdf

[7] Prof. Michael Collins, Columbia University
Natural Language Processing(coursera)
https://class.coursera.org/nlangp-001/lecture

[8] **Dan Jurafsky,Christopher Manning**
Week 4 - Relation Extraction of **Natural Language Processing**(coursera lectures),
https://class.coursera.org/nlp/lecture

[9] Rahul Sharnagat
Named Entity Recognition: A Literature Survery
http://www.cfilt.iitb.ac.in/resources/surveys/rahul-ner-survey.pdf