NAMED ENTITY RECOGNATION

A Micro Project Report

Submitted by

S. DIVYA SREE
99220041039

B.Tech - CSE,
CYBERSECURITY

Kalasalingam Academy of Research and Education

(Deemed to be University)

Anand Nagar, Krishnankoil - 626 126

[02] [2023]

KALASALINGAM
ACADEMY OF RESEARCH AND EDUCATION
(DEEMED TO BE UNIVERSITY)
Under sec. 3 of UGC Act 1956. Accredited by NAAC with "A" Grade

SCHOOL OF COMPUTING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

Bonafide record of the work done by S.DIVYA SREE - 99220041039 in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Specialization of the Computer Science and Engineering, during the Academic Year Even Semester (2023-24)

Dr. G. Nagarajan
Project Guide
Assistant Professor
Computer Science and Engineering
Kalasalingam Academy of
Research and Education
Krishnan kovil – 626126

Mr. Gnana Kumar
Faculty Incharge
Assistant Professor
Computer Science and Engineering
Kalasalingam Academy of
Research and Education
Krishnan kovil - 626126

Mr. M. Jafer Sathick Ali
Evaluator
Assistant Professor
Computer Science and Engineering
Kalasalingam Academy of
Research and Education
Krishnan Kovil - 626126

Abstract

Named Entity Recognition (NER) is a crucial task in Natural Language Processing (NLP), focusing on detecting and classifying real-world entities in text. This project aims to provide examples and best practices for building NER models using pre-trained BERT models.

Named entity recognition (NER) is a subfield of information extraction, which aims to detect and classify predefined named entities (e.g., people, locations, organizations, etc.) in a body of text. In the literature, many researchers have studied the application of different machine learning models and features to NER. However, few research efforts have been devoted to studying annotation schemes used to label multi-token named entities. In this research, we studied seven annotation schemes (IO, IOB, IOE, IOBES, BI, IE, and BIES) and their impact on the task of NER using five different classifiers. Our experiment was conducted on an in–house dataset that consists of 27 medical Arabic articles with more than 62,000 tokens.

The IO annotation scheme outperformed other schemes with an F-measure score of 84.44%. The closest competitor is the BIES scheme, which scored 72.78%. The rest of the schemes' scores ranged from 60.38% to 69.18%. Although the IO scheme achieved the best results, comparing it to the other schemes is not reasonable because it cannot identify consecutive entities, which the other schemes can do.

Therefore, we also investigated the ability of recognizing consecutive entities and provided an analysis of the running-time complexity.

Contents

Chapter 1

Chapter 1: Introduction

1.1 Background

Named Entity Recognition (NER) is a crucial task in Natural Language Processing (NLP) with applications across various domains. It involves identifying and classifying named entities, such as people, organizations, locations, dates, and other relevant information, within text data. This information can be invaluable for tasks like information extraction, question answering, sentiment analysis, and machine translation.

The increasing volume and complexity of textual information necessitate efficient and robust NER techniques. Traditional approaches relied on handcrafted rules and features, which were often domain-specific and limited in their adaptability. In recent years, deep learning models, particularly transformers like BERT (Bidirectional Encoder Representations from Transformers), have shown remarkable performance in various NLP tasks, including NER. This report explores the application of fine-tuning BERT for Named Entity Recognition.

1.2 Objective

The primary objective of this endeavor is to provide a comprehensive guide on the process of fine-tuning BERT models for NER tasks. While the project centers around the practical aspects, it also aims to lay a robust foundation for users to implement NER in various use cases. The incorporation of state-of-the-art methods, including LSTM-CRF and BERT-based models, underscores the commitment to staying at the forefront of NLP advancements.

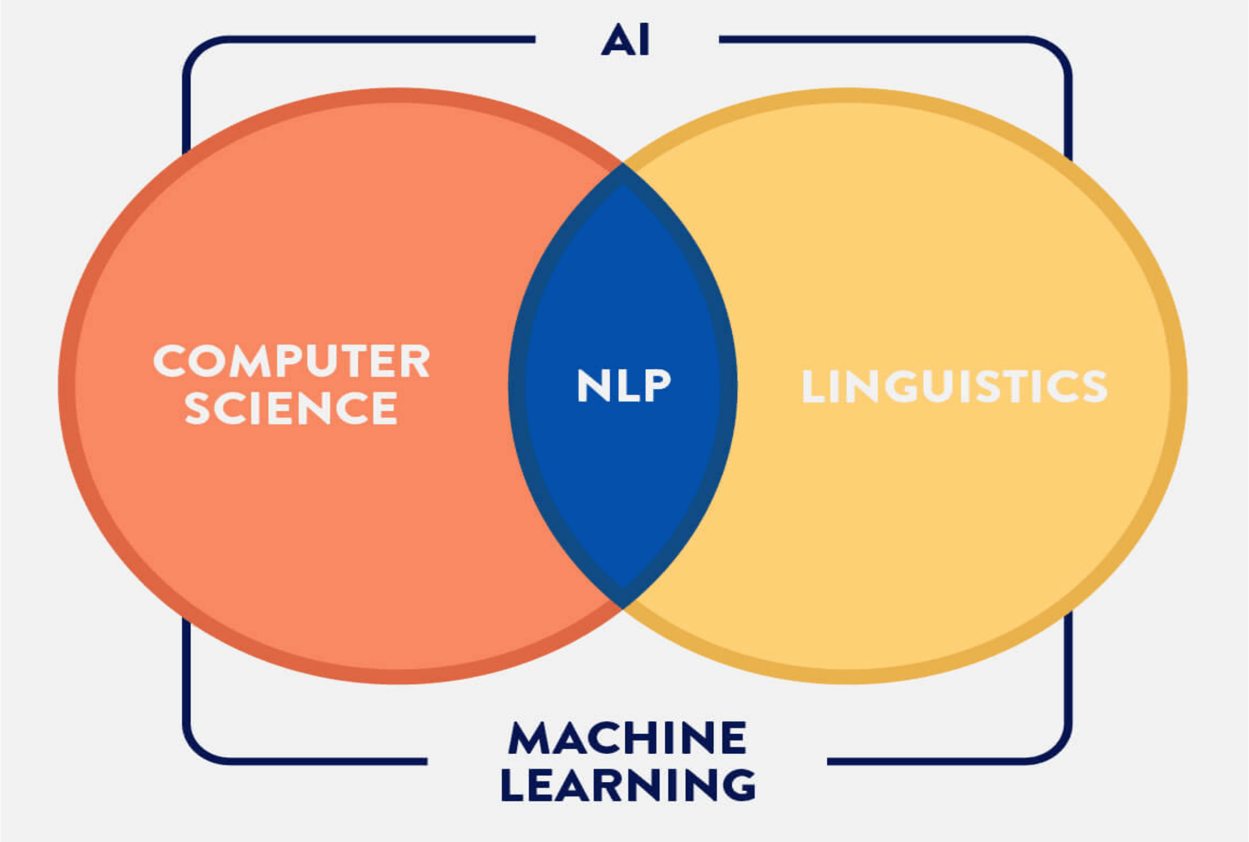


Figure 1.1: Natural language processing

Natural Language Processing (NLP) is a subfield of artificial intelligence that deals with the interaction between computers and humans in natural language. It involves the use of computational techniques to process and analyze natural language data, such as text and speech, with the goal of understanding the meaning behind the language.

Chapter 2

Chapter 2: Named Entity Recognition

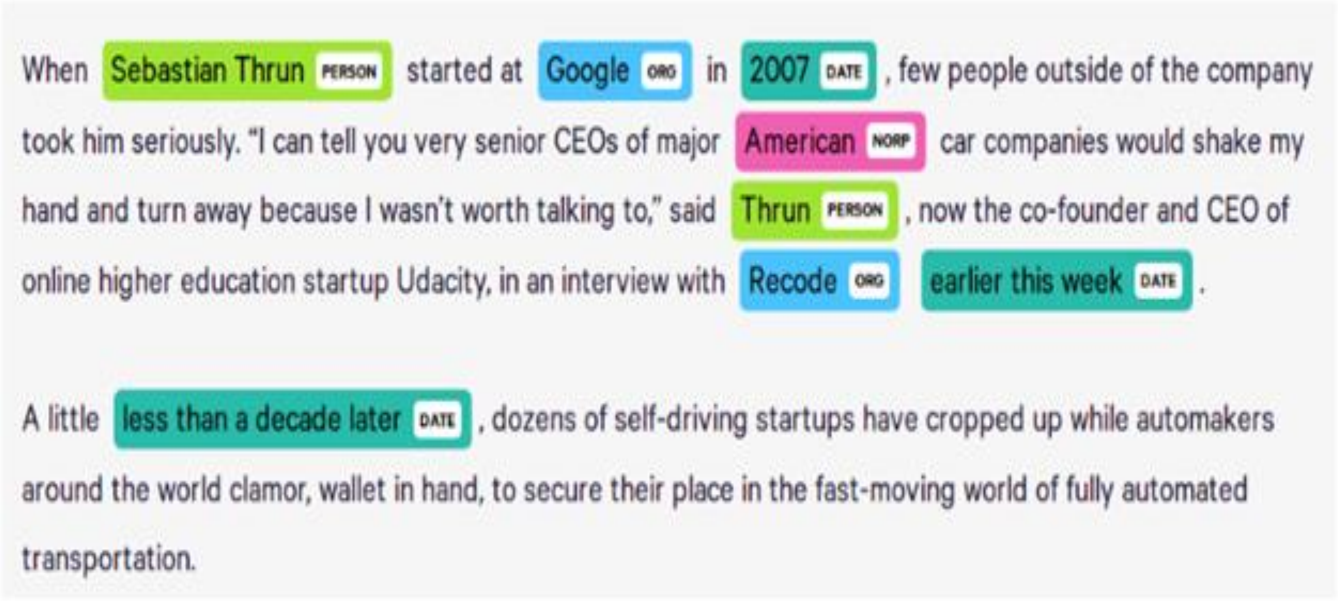
2.1 What is Named Entity Recognition (NER)?

Named Entity Recognition (NER) is a subtask of NLP concerned with identifying and classifying named entities in text. These entities can represent various real-world concepts, including:

- People: Barack Obama, Albert Einstein, Leonardo da Vinci
- Organizations: Apple, Google, United Nations
- Locations: New York City, Mount Everest, River Nile
- Dates: 27th February 2024, 14th July 1789
- Other Entities: Titles (CEO, Professor), monetary values ($100, €50), percentages (25%)

Effectively performing NER allows for extracting valuable information from text data. This extracted information can be utilized for a variety of purposes, such as:

- Information Retrieval: Efficiently searching and filtering documents based on specific entities.
- Question Answering Systems: Accurately responding to questions requiring knowledge of entities.
- Sentiment Analysis: Understanding the sentiment expressed towards specific entities (e.g., companies, products, politicians).
- Machine Translation: Preserving the meaning and identification of entities during translation processes.



When Sebastian Thrun PERSON started at Google ORG in 2007 DATE , few people outside of the company took him seriously. "I can tell you very senior CEOs of major American NORP car companies would shake my hand and turn away because I wasn't worth talking to," said Thrun PERSON , now the co-founder and CEO of online higher education startup Udacity, in an interview with Recode ORG earlier this week DATE .

A little less than a decade later DATE , dozens of self-driving startups have cropped up while automakers around the world clamor, wallet in hand, to secure their place in the fast-moving world of fully automated transportation.

Figure 2.1: named entity

Named Entity Recognition (NER), also known as entity identification, entity chunking, and entity extraction, is a technique in natural language processing (NLP). Its primary goal is to identify and categorize named entities within unstructured text. Let's delve into the details:

Definition:

- NER locates and classifies named entities mentioned in text into predefined categories. These entities can include:
  - Person names
  - Organizations
  - Locations
  - Medical codes
  - Time expressions
  - Quantities
  - Monetary values

2.2 Labeling Schemes

Different NER systems utilize various labeling schemes to categorize identified entities. Here are some common schemes:

- **BIO:** This scheme uses tags like B-PER (beginning of a person entity), I-PER (inside a person entity), O (outside of any entity) to represent the start, continuation, and outside of an entity, respectively.
- **IOB:** Similar to BIO, this scheme uses B-PER, I-PER, and O tags, but also includes an extra tag, "B" (beginning), which marks the beginning of an entity regardless of its position within the sequence.
- **MISC:** This scheme uses a single tag (e.g., MISC) to label every identified entity, irrespective of its type.
- **Custom Schemes:** Some systems may use custom labeling schemes tailored to their specific domain or application. These schemes define specific tags for the types of entities relevant to their domain, such as "LOC" (location) or "ORG" (organization)

The choice of labeling scheme depends on the specific needs of the application and the level of granularity required for entity classification.

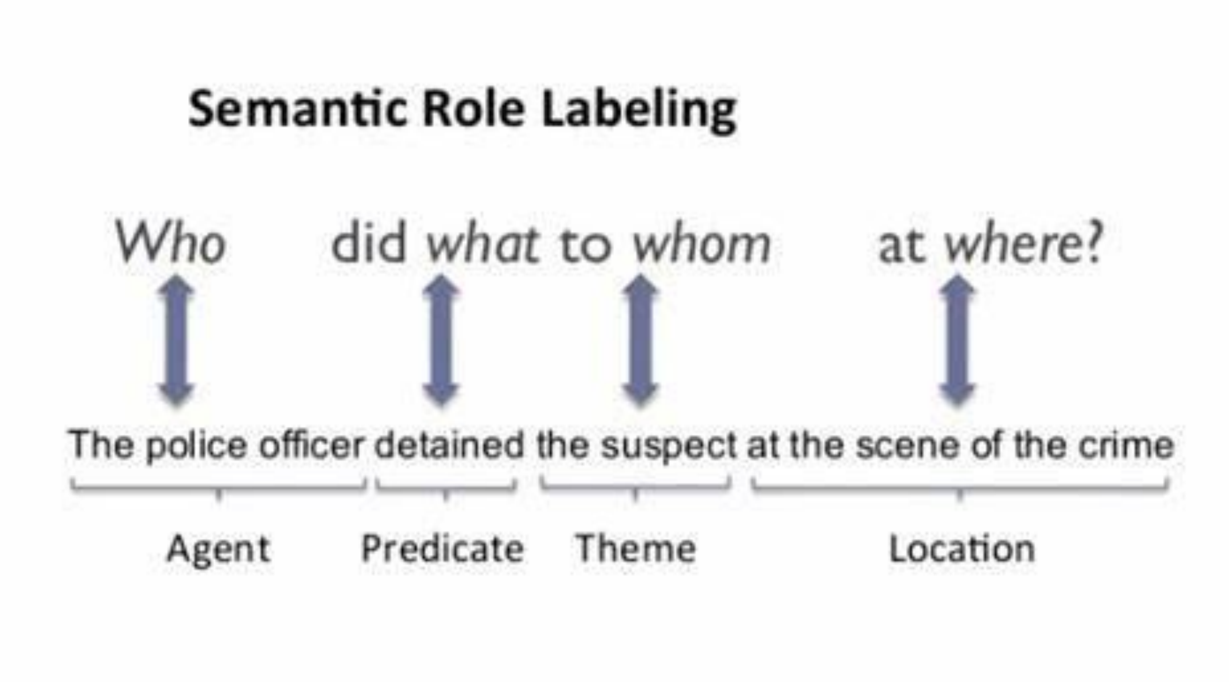


Figure 2.2: sematic Labeling

Chapter 3

Chapter 3 Implementation

Notebook

This section provides a brief overview of the notebooks utilized for implementing the fine-tuning process. The notebooks are assumed to be written in Python and utilize libraries like PyTorch and Transformers for model development.

code

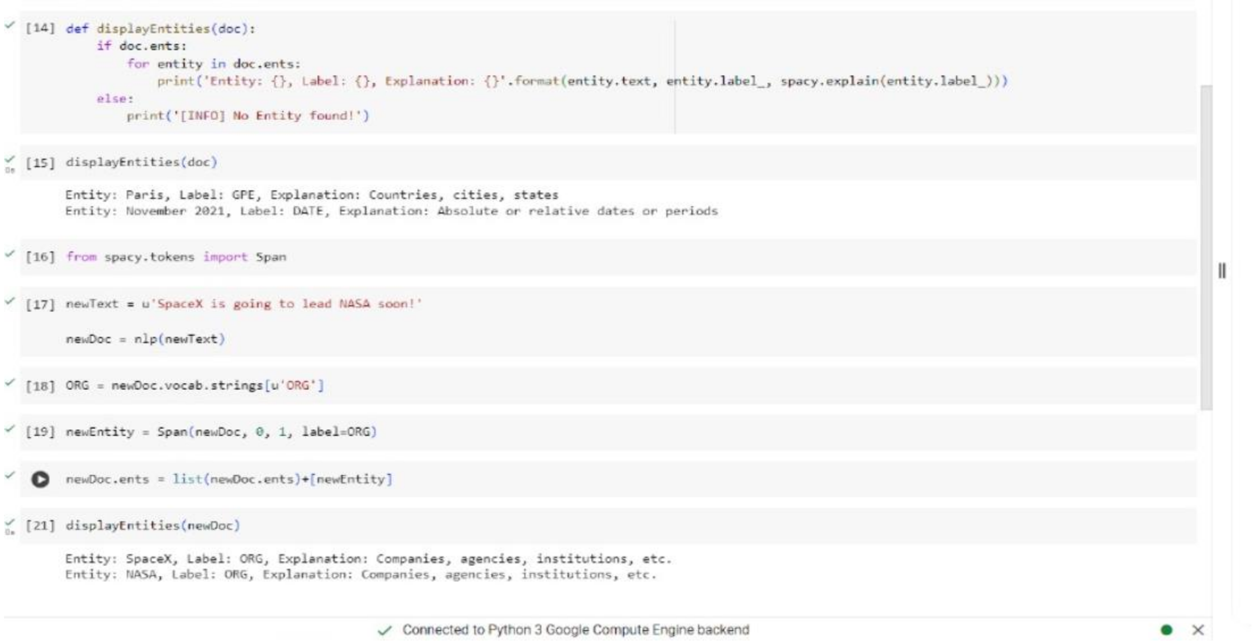
```
import tensorflow as tf
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import torch
if torch.cuda.is_available():
    device = torch.device("cuda")
    print( torch.cuda.device_count())
    print('Available:', torch.cuda.get_device_name(0))
else:
    print('No GPU available, using the CPU instead.')
    device = torch.device("cpu"

!pip install wget
!pip install transformers
```

Data set Training and Testing

```
url_train='https://groups.csail.mit.edu/sls/downloads/movie/engtrain.bio'
url_test='https://groups.csail.mit.edu/sls/downloads/movie/engtest.bio'
import wget
import os
```

```
wget.download(url_train)
wget.download(url_test)

import csv
sentences = []
labels = []

tokens = []
token_labels = []
unique_labels = set()

with open("./engtrain.bio", newline = '') as lines:

    line_reader = csv.reader(lines, delimiter='\t')

    for line in line_reader:

        if line == []:

            sentences.append(tokens)
            labels.append(token_labels)

            tokens = []
            token_labels = []

        else:

            tokens.append(line[1])
            token_labels.append(line[0])

            unique_labels.add(line[0])


[  print(' '.join(sentences[i])) for i in range(10)]


' '.join(sentences[1])


pd.DataFrame({"Word":sentences[1],"Labels":labels[1]})

from transformers import BertTokenizer
import numpy as np
tokenizer = BertTokenizer.from_pretrained('bert-base-uncased')

import matplotlib.pyplot as plt
import seaborn as sns

sns.set(style='darkgrid')
```

```
# Increase the plot size and font size.
sns.set(font_scale=1.5)
plt.figure(figsize=(24,24))
plt.rcParams["figure.figsize"] = (10,5)

# Plot the distribution of comment lengths.
sns.distplot(TokenLength, kde=False, rug=False,color='plum')

plt.title('Sentence Lengths')
plt.xlabel('Sentence Length')
plt.ylabel('# of Sentences');


import spacy

nlp = spacy.load('en_core_web_sm')
text = u'I will visit Paris on November 2021'

doc = nlp(text)
def displayEntities(doc):
   if doc.ents:
       for entity in doc.ents:
           print('Entity: {}, Label: {}, Explanation: {}'.format(entity.text, entity.label_,
spacy.explain(entity.label_)))
   else:
       print('[INFO] No Entity found!')

displayEntities(doc)
from spacy.tokens import Span
newText = u'SpaceX is going to lead NASA soon!'

newDoc = nlp(newText)
ORG = newDoc.vocab.strings[u'ORG']


newEntity = Span(newDoc, 0, 1, label=ORG)

newDoc.ents = list(newDoc.ents)+[newEntity]

displayEntities(newDoc)
```



```
[14] def displayEntities(doc):
         if doc.ents:
             for entity in doc.ents:
                 print('Entity: {}, Label: {}, Explanation: {}'.format(entity.text, entity.label_, spacy.explain(entity.label_)))
         else:
             print('[INFO] No Entity found!')

[15] displayEntities(doc)

     Entity: Paris, Label: GPE, Explanation: Countries, cities, states
     Entity: November 2021, Label: DATE, Explanation: Absolute or relative dates or periods

[16] from spacy.tokens import Span

[17] newText = u'SpaceX is going to lead NASA soon!'

     newDoc = nlp(newText)

[18] ORG = newDoc.vocab.strings[u'ORG']

[19] newEntity = Span(newDoc, 0, 1, label=ORG)

     newDoc.ents = list(newDoc.ents)+[newEntity]

[21] displayEntities(newDoc)

     Entity: SpaceX, Label: ORG, Explanation: Companies, agencies, institutions, etc.
     Entity: NASA, Label: ORG, Explanation: Companies, agencies, institutions, etc.
```

✓ Connected to Python 3 Google Compute Engine backend

3.2 Environment

The local environments used for model development and fine-tuning are discussed in detail. Additionally, the utility scripts located in the utils_nlp folder are explored, shedding light on their role in expediting data preprocessing and facilitating an efficient model-building process

3.3 Dataset

An in-depth analysis of the wikigold dataset used for fine-tuning the BERT model is presented. The dataset's richness in providing a diverse array of entities in English text contributes significantly to the robustness of the trained NER models. This section emphasizes the importance of selecting an appropriate dataset for effective model training.

3.4 Model Evaluation Metrics

Dive into the various metrics employed for evaluating the performance of fine-tuned models. Precision, recall, F1 score, and other relevant metrics are discussed, offering insights into the strengths and potential areas for improvement in the implemented models.

Chapter 4

Conclusion and Future Work

4.1 Conclusion

The culmination of the project is marked by a robust demonstration of the efficacy of fine-tuning pre-trained BERT models for NER tasks. Practical examples and real-world applications showcase the practical utility of the implemented models in the broader context of NLP.

4.2 Future Work

Considerations for future work are explored, extending beyond the immediate implementation. Potential avenues include the expansion of supported pre-trained models, incorporation of multilingual capabilities, exploration of additional datasets, and the integration of emerging NLP methodologies. Discussions also touch upon enhancing model robustness and scalability for broader applicability.

Chapter 5

References

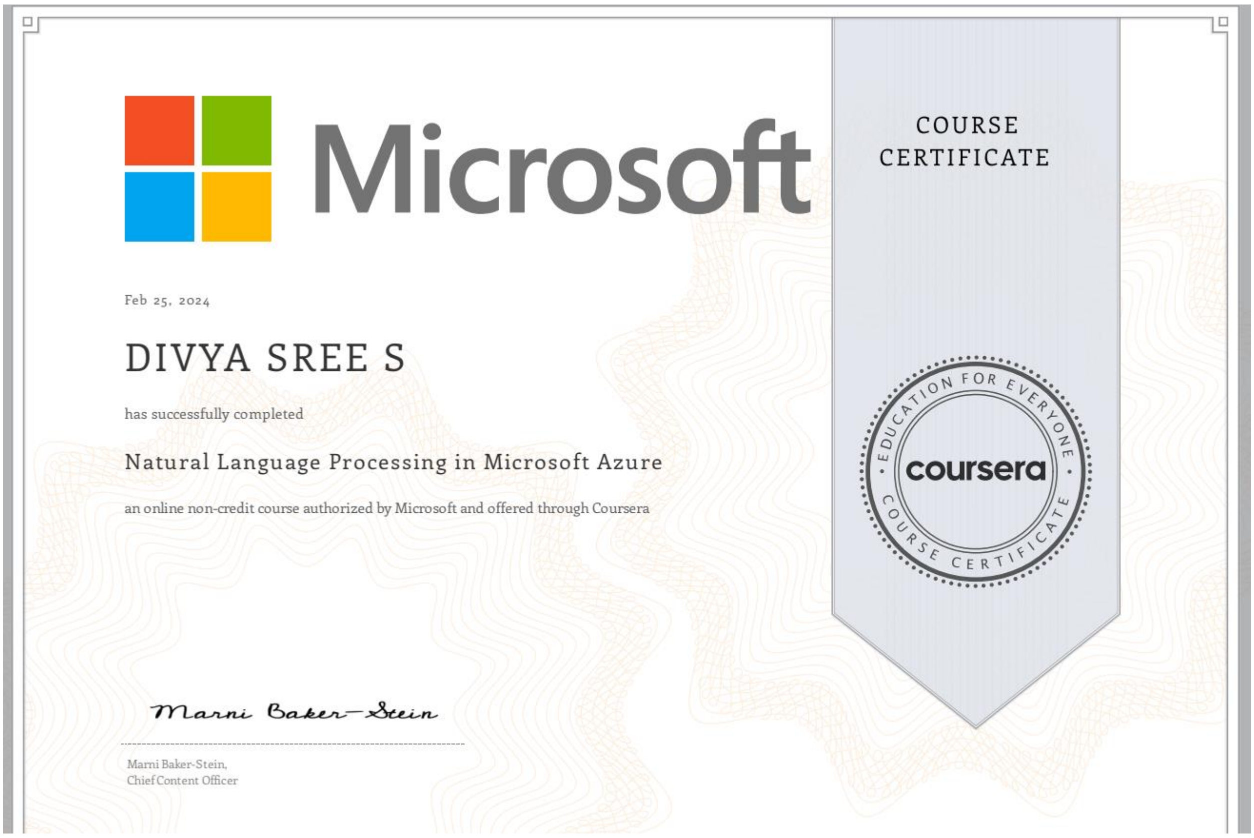
The report refers to various sources, including academic papers, documentation, and external datasets. These references provide a comprehensive understanding of the NER task, the BERT model, and related methodologies.

This project report provides an overview of Named Entity Recognition (NER) using pre-trained BERT models. It covers the background, objectives, implementation details, and conclusions, offering a valuable resource for developers and researchers in the field of Natural Language Processing.

1. A. Harisinghaney, A. Dixit, S. Gupta and A. Arora, "Text and image based spam email classification using KNN Naïve Bayes and Reverse DBSCAN algorithm", In 2014 International Conference on Reliability Optimization and Information Technology (ICROIT), pp. 153-155, 2014, February.
2. S. P. Teli and S. K. Biradar, "Effective Email Classification for Spam and Non-spam", in International Journal of Advanced Research in Computer and software Engineering, vol. 4, 2014.
3. M. Habib, H. Faris, M. A. Hassonah, J. F. Alqatawna, A. F. Sheta and A. Z. Ala'M, "Automatic email spam detection using genetic programming with SMOTE", In 2018 Fifth HCT Information Technology Trends (ITT), pp. 185-190, November 2018.

Chapter 6

Certification



Microsoft

COURSE CERTIFICATE

Feb 25, 2024

DIVYA SREE S

has successfully completed

Natural Language Processing in Microsoft Azure

an online non-credit course authorized by Microsoft and offered through Coursera

EDUCATION FOR EVERYONE coursera COURSE CERTIFICATE

Marni Baker-Stein

Marni Baker-Stein,
Chief Content Officer

Figure 6.1: Certification details