Natural Language Processing

**Project Report on**

“NER Chatbot”

**Academic Year:** First Semester 2024-2025

**Course No:** SSZG599

Group No: 8

**Team Members:**

**Sriram Rokkam [2023MT12251]**

**Anurag Maharshi [2023MT12125]**

**Vigneshwaran K R [2023MT12091]**

**Godlaveeti Anil Govind [2023MT12272]**

**Korukonda Sowmya [2023MT12203]**

Table of Contents

[1. Project Overview: 3](#_Toc180991455)

[2. Model Architecture 3](#_Toc180991456)

[2.1 CRF Architecture 3](#_Toc180991457)

[2.2 BERT Architecture. 3](#_Toc180991458)

[3. Solution Design (with Comparison of CRF and BERT) 4](#_Toc180991459)

[1. Data Preparation: 4](#_Toc180991460)

[2. Feature Engineering & Extraction: 4](#_Toc180991461)

[3. Data Pre-Processing: 4](#_Toc180991462)

[4. Model Training: 4](#_Toc180991463)

[5. Evaluation: 5](#_Toc180991464)

[4. How to Execute: 5](#_Toc180991465)

[Setup Instructions 5](#_Toc180991466)

[Prerequisites 5](#_Toc180991467)

[Dataset 5](#_Toc180991468)

[Model Training 5](#_Toc180991469)

[Model Testing 5](#_Toc180991470)

[5. Test Results and Observations: 6](#_Toc180991471)

[Observations: 6](#_Toc180991472)

[6. Summary: 6](#_Toc180991473)

[7. Recommendation: 6](#_Toc180991474)

[8. Team Contributions 6](#_Toc180991475)

[9. References and Credits 7](#_Toc180991476)

[CRF (Conditional Random Fields) 7](#_Toc180991477)

[1. Original Paper: 7](#_Toc180991478)

[2. Tutorial: 7](#_Toc180991479)

[3. Scikit-learn Implementation: 7](#_Toc180991480)

[BERT (Bidirectional Encoder Representations from Transformers) 7](#_Toc180991481)

[1. Original Paper: 7](#_Toc180991482)

[2. GitHub Repository: 7](#_Toc180991483)

[3. Hugging Face Transformers: 8](#_Toc180991484)

[\*\*\* END \*\*\* 8](#_Toc180991485)

# 1. Project Overview:

In the rapidly advancing field of Natural Language Processing (NLP), Named Entity Recognition (NER) plays a pivotal role in understanding and extracting structured information from unstructured text. Our project focuses on building a comprehensive NER model that detects and classifies named entities within textual data, such as dates, names, locations, and organizations, organized into predefined categories relevant to various applications. To make this functionality accessible, we have developed a user-friendly chatbot interface, allowing users to input sentences or paragraphs and receive insights into recognized entities in real-time.

Token classification, the core task of our project, entails assigning specific labels to each token (word or phrase) in a given text sequence. We explored two primary NLP architectures:

* CRF ( Conditional Random Fields ) - Statistical model
* BERT (Bidirectional Encoder Representation Transformers) – Transformer model.

Each approach has been trained on a well-structured dataset with columns for sentences, words, part-of-speech tags (POS), and NER labels, enabling the models to understand both the syntactic and semantic aspects of the language.

# 2. Model Architecture

## 2.1 CRF Architecture

Conditional Random Fields (CRF) is a statistical model widely used for structured prediction tasks, like Named Entity Recognition (NER). CRFs are particularly effective when the goal is to make predictions for sequences of data, as they capture the relationships between tags in the sequence. Unlike simpler classifiers that predict each tag independently, a CRF takes into account the tags of neighbouring words, making it ideal for tasks where the context matters.

## 2.2 BERT Architecture.

BERT, or Bidirectional Encoder Representations from Transformers, is a model developed by Google in 2018 to understand text by reading both left-to-right and right-to-left.

This bidirectional approach helps BERT understand context more accurately than traditional NLP models. Pretrained on large datasets like Wikipedia, BERT learns general language features, which can then be fine-tuned for tasks like Named Entity Recognition (NER) using smaller **labelled** datasets.

# 3. Solution Design (with Comparison of CRF and BERT)

## 1. Data Preparation:

1. The program begins with data preparation by defining a class to group words, part-of-speech (POS) tags, and entity (NER) tags by sentence.
2. Organizing data into sentences ensures that each sequence can be processed individually, a critical step for sequence-based models both CRF and BERT.
3. Data preparation is a common step for both CRF and BERT Model.

## 2. Feature Engineering & Extraction:

1. **CRF model :**
   * For CRF, the `word2features` function extracts specific linguistic features for each word, including lowercase form, suffixes, casing (uppercase, title-case), and whether it’s numeric.
2. **BERT model :**
   * For BERT, instead of manual feature engineering, it utilizes pre-trained embeddings, capturing a word’s context directly within its bidirectional transformer architecture, which considers the entire sentence’s context.

## 3. Data Pre-Processing:

To ensure compatibility with each model, sentences are converted into input features, labels, and tokens.

1. **CRF model :**

For CRF, this involves manually crafted features (`sent2features`).

1. **BERT model :**

For BERT leverages tokenized inputs and pre-trained embeddings to represent each word and its position in the sentence.

These processed inputs are then used to create input (`X`) and target (`y`) sets for both models.

## 4. Model Training:

The dataset is split into training and testing sets in 80:20 Ratio. The training for both the models is to learn to identify and tag named entities within sentences by recognizing patterns and dependencies.

Note: We have created separate programs for Training and Evaluating for each both BERT and CRF Models, detailed in Section 3 How to Execute.

1. **CRF model :**

The CRF model is trained with specific hyperparameters, such as L1 (`c1`) and L2 (`c2`) regularization values, to balance model performance.

1. **BERT model :**

The BERT model is fine-tuned on the NER dataset, leveraging its pre-trained contextual knowledge to adapt to NER-specific entities.

## 5. Evaluation:

Predictions are made on the test set, and performance is evaluated using the F1 score for both models. The weighted F1 score provides an overall metric of each model’s effectiveness in recognizing entities across varied sentence structures.

1. **CRF model :** For CRF, predictions are based on the features derived from the test set.
2. **BERT model :** For BERT’s predictions utilize contextual embeddings.

# 4. How to Execute:

## Setup Instructions

To set up the project and install the required packages, follow these steps:

`pip install -r requirements.txt`

## Prerequisites

Make sure you have Python 3.x and `pip` installed on your system. You can check if you have them installed by running:

python --version

pip --version

\*\*Note:\*\* You can install the dependencies using: 'requirements.txt' file provided

## Dataset

The dataset used is a custom NER dataset provided in CSV format with columns:

- `sentence\_id`: Unique identifier for sentences.

- `words`: The words in each sentence.

- `labels`: The named entity labels corresponding to each word.

## Model Training

To Train the selected models execute the below programs:

* BERT Model:

Execute file '01.NER\_BERT\_Training\_F1.ipynb' ([link](https://github.com/sriramrokkam/NER_NLP_Assignment/blob/main/01.NER_BERT_Training_F1.ipynb))

* CRF Model :

Execute file '02.NER\_CRF\_Training\_F1.ipynb ([link](https://github.com/sriramrokkam/NER_NLP_Assignment/blob/main/02.NER_CRF_Training_F1.ipynb))

## Model Testing

To Train the selected models execute the below programs:

* BERT Model:

Execute file '01.NER\_BERT\_Chatbot\_F1.ipynb' ([link](https://github.com/sriramrokkam/NER_NLP_Assignment/blob/main/01.NER_BERT_Chatbot_F1.ipynb))

* CRF Model :

Execute file '02.NER\_CRF\_Chatbot\_F1.ipynb' ([link](https://github.com/sriramrokkam/NER_NLP_Assignment/blob/main/02.NER_CRF_Chatbot_F1.ipynb))

GitHub Repository: <https://github.com/sriramrokkam/NER_NLP_Assignment/blob/main/README.md>

# Test Results and Observations:

|  |  |  |
| --- | --- | --- |
| **Metric/Model** | **CRF** | **BERT** |
| **F1 Score** | 0.62 | 0.7976 |
| **Accuracy** | 0.97 | 0.87 |
| **Precision** | 0.68 | 0.8258 |
| **Recall** | 0.59 | 0.7713 |

## Observations:

1. F1 Score: BERT (0.7976) > CRF (0.62) — BERT balances precision and recall better.
2. Accuracy: CRF (0.97) > BERT (0.87) — but accuracy alone isn’t as reliable for NER tasks.
3. Precision & Recall: BERT scores higher, meaning it’s better at correctly identifying and capturing entities.

# Summary:

1. In the evaluation of models for Named Entity Recognition (NER), BERT clearly outperforms CRF in key metrics, making it the preferred choice.
2. **F1 Score**: BERT achieved an F1 score of 0.7976, significantly higher than CRF's 0.62, indicating better balance between precision and recall.
3. **Accuracy**: While CRF has higher accuracy at 0.97 compared to BERT's 0.87, accuracy alone is not a reliable measure for NER effectiveness.
4. **Precision & Recall**: BERT excels in both precision and recall, demonstrating its superior ability to accurately identify and capture entities.

# Recommendation:

1. We recommend using the BERT model for NER tasks due to its higher F1 score, stronger precision and recall, and enhanced contextual understanding, which provide advantages in processing complex language.

# Team Contributions

1. **Sriram [2023MT12251] & Sowmya [2023MT12203]:** Led the development, training, and testing of the BERT model, demonstrating proficiency in natural language processing and deep learning techniques.
2. **Anurag [2023MT12125] & Govind [2023MT12272]:**  Led the implementation, training, and testing of the Conditional Random Fields (CRF) model, showcasing expertise in sequence labelling and statistical modelling.
3. **Vigneshwaran K R [2023MT12091]:** Played a pivotal role in building the chatbot, as well as documenting processes and testing the models developed by both teams, ensuring comprehensive support and integration across projects.

# References and Credits

<https://www.kaggle.com/datasets/debasisdotcom/name-entity-recognition-ner-dataset>

## CRF (Conditional Random Fields)

### 1. Original Paper:

- "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data" by Lafferty, McCallum, and Pereira (2001).

- [Link to Paper](https://cs.cmu.edu/~fmcc/15-712/papers/lafferty01conditional.pdf)

### 2. Tutorial:

- "A Gentle Introduction to Conditional Random Fields" by H. Andrew Schwartz.

- [Link to Tutorial](http://www.ark.cs.cmu.edu/~hughc/CRF\_tutorial.pdf)

### 3. Scikit-learn Implementation:

- Scikit-learn provides an implementation of CRF through the `sklearn-crfsuite` package.

- [Link to Scikit-learn CRF](https://scikit-learn-crfsuite.readthedocs.io/en/latest/)

## BERT (Bidirectional Encoder Representations from Transformers)

### 1. Original Paper:

- "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding" by Devlin et al. (2018).

- [Link to Paper](https://arxiv.org/abs/1810.04805)

### 2. GitHub Repository:

- The official BERT repository by Google, which includes code and pretrained models.

- [Link to GitHub](https://github.com/google-research/bert)

### 3. Hugging Face Transformers:

- A popular library for NLP that includes BERT and other transformer models.

- [Link to Hugging Face](https://huggingface.co/transformers/model\_doc/bert.html)

# \*\*\* END \*\*\*