**Project Instructions for Named Entity Recognition (NER) Task**

**Objective**

The purpose of this project is to evaluate your ability to design, implement, and test an AI-based system capable of extracting Named Entities (e.g., names, organizations, locations, dates, etc.) from a dataset containing over 1000 conversation sentences.

**Dataset**

* **File Name**: NER\_Training\_Data.xlsx
* **Format**: Excel file with multiple sheets, each containing a collection of sentences.
* **Structure**: Each sentence represents a single conversation input.

**Requirements**

1. **Model Selection and Training**:
   * Select or fine-tune a pre-trained NER model (e.g., SpaCy, Hugging Face Transformers).
   * If fine-tuning, use publicly available datasets and ensure reproducibility.
2. **Data Processing**:
   * Preprocess the input sentences for NER extraction.
   * Ensure compatibility with the NER system (tokenization, text cleaning, etc.).
3. **Named Entity Extraction**:
   * Identify and classify entities into categories such as:
     + **username**: Usernames mentioned in the text.
     + **authorname**: Names of authors.
     + **Address**: Any addresses mentioned.
     + **bookname**: Names of books.
     + **productname**: Names of products.
     + **publishername**: Names of publishers.
     + **categoryname**: Names of categories.
     + **brandname**: Names of brands.
     + **date**: Dates mentioned.
     + **phoneno**: Phone numbers.
     + **orderno**: Order numbers.
     + **trnx ID**: Transaction IDs.
4. **Output Format**:
   * Provide the extracted entities in the following structured format:
     + **Columns**:
       - Message
       - username
       - authorname
       - Address
       - bookname
       - productname
       - publishername
       - categoryname
       - brandname
       - date
       - phoneno
       - orderno
       - trnx ID
   * Example:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Message** | **username** | **authorname** | **Address** | **bookname** | **productname** | **publishername** | **categoryname** | **brandname** | **date** | **phoneno** | **orderno** | **trnx ID** |
| "Helo My Name is Rajib, Cell: 1744898066. I need two books below: 1. "English Conversation Made Natural" by Language Guru, 2. Speak English Like an American" by Amy Gillett" | Rajib | Language Guru, Amy Gillett |  | English Conversation Made Natural, Speak English Like an American |  |  |  |  |  | 1744898066 |  |  |

* Deliver the output as a CSV file.

1. **Evaluation**:
   * Include precision, recall, and F1-score metrics to evaluate the performance of the system.
   * Optionally provide a confusion matrix.
2. **System Documentation**:
   * Clearly document the pipeline: preprocessing, model, and postprocessing steps.
   * Include instructions for replicating the environment (e.g., Python version, dependencies).

**Deliverables**

1. **NER System**:
   * Source code with proper documentation.
   * Clear instructions for setup and execution.
2. **Extracted Entities**:
   * CSV file containing the extracted entities in the specified format.
3. **Performance Report**:
   * Metrics evaluating the system’s performance.
   * Additional insights, if any.
4. **Project Timeline**:
   * Provide an estimated timeline for completion.

**Constraints**

* The system should work for sentences in the provided dataset.
* Ensure computational efficiency and scalability.
* Follow best practices in coding and AI development.

**Submission**

Submit the project deliverables in a compressed folder with the following structure:

Project\_Folder/

├── code/ (Python scripts and modules)

├── data/ (Processed input and output files)

├── report/ (Performance metrics and system description)

├── requirements.txt (Dependencies and setup instructions)

└── README.md (Instructions for running the system)

**Evaluation Criteria**

* **Accuracy**: How well the system extracts entities.
* **Scalability**: Ability to handle a large number of sentences.
* **Documentation**: Clarity and completeness of instructions.
* **Code Quality**: Use of best practices and readability.