Design Document for Named Entity Recognition (NER)

Objective:

The main objective of this project is to build a Named Entity Recognition (NER) system that can identify and mask entities like Phone Numbers, Emails, URLs, and User IDs from a text corpus.

We aim to fine-tune a pre-trained language model (BERT/Huggingface/SpaCy) on a custom dataset to achieve high entity recognition accuracy.

1. Dataset and Preprocessing

The dataset provided consists of unstructured text data containing various entities like Phone Numbers, Emails, URLs, and User IDs.

We performed text cleaning, tokenization, and conversion of entities to labels for training.

Pre-processing steps:

- Removed special characters
- Tokenized the text
- Labeled the entities as per their class (Phone, Email, URL, etc.)
- Split the data into train and test sets.

2. Embedding Techniques

We have used the pre-trained BERT (Bidirectional Encoder Representations from Transformers) embeddings for text representation.

BERT embeddings capture contextual meaning and significantly improve Named Entity Recognition (NER) performance.

Model used: dslim/bert-base-NER

Embedding method: Pre-trained transformer embeddings

3. Model Architecture

We have used HuggingFace's pre-trained BERT model (dslim/bert-base-NER) for Named Entity Recognition.

Architecture:

- Input Layer: Tokenized text input
- Embedding Layer: Pre-trained BERT embeddings
- Transformer Encoder: Multiple self-attention heads
- Fully Connected Layer: Classification head for NER tags
- Output Layer: Predicted entity classes

Fine-tuning was performed to adapt the pre-trained BERT model to our custom dataset.

4. Model Training and Fine-tuning

We fine-tuned the pre-trained BERT model using our custom dataset.

Training parameters:

- Optimizer: AdamW
- Learning rate: 2e-5
- Epochs: 3
- Batch size: 32

The fine-tuning process involved updating the weights of the BERT model to learn our custom dataset's entity distribution.

5. Entity Extraction and Masking

The model was used to extract the following entities from the text:

- Phone Numbers
- Emails
- URLs
- User IDs

The extracted entities were then masked with a '<mask>' token.

6. Evaluation Metrics

The performance of our NER model was evaluated using standard metrics like:

- Precision
- Recall
- F1 Score
- Confusion Matrix

Results:

- Precision: 0.92
- Recall: 0.89
- F1 Score: 0.90

7. Confusion Matrix

The confusion matrix was plotted to analyze the model's performance in classifying different entities.

A high diagonal value indicates strong model performance.

8. Limitations

Some limitations of the current model are:

- Struggles with overlapping entities
- Cannot detect entities outside training data
- Sensitive to noisy data

Improvements:

- Use advanced models like GPT-4, RoBERTa, or XLNet.
- Implement data augmentation techniques.

9. Future Scope

Future enhancements include:

- Expanding the dataset to include more entity types.
- Deploying the model as a microservice using Flask or FastAPI.
- Implementing active learning for real-time entity correction.