



DATS 6312

Introduction to Natural Language Processing

**Edge-Based PII Detection and Censoring System
Project Proposal**

Pranav Dhawan, Akshit Reddy Palle,
Aakash Singh Sivaram, Dhatrija Sukasi

1. Problem Statement and Rationale

Privacy breaches in LLM interactions represent a critical vulnerability in modern conversational AI systems. Users frequently share sensitive information—credit card numbers, social security numbers, home addresses, medical conditions—without realizing these data points may be logged, transmitted, or inadvertently exposed. Unlike cloud-based solutions that require transmitting potentially sensitive data before filtering, our edge-based approach ensures PII never leaves the user's device, providing a zero-trust architecture for data privacy.

This problem is particularly timely given increasing regulatory scrutiny (GDPR fines, HIPAA violations) and high-profile data breaches. An edge-based solution addresses three key advantages:

- (1) offline functionality without internet dependency,
- (2) low-latency privacy protection,
- (3) compliance with data localization requirements.

Our system will act as a protective middleware layer between user input and cloud-based LLMs, making privacy-preserving AI accessible to all users.

2. Dataset

We will use a mixture of publicly available and synthetic datasets to train and evaluate our PII detection system:

2.1 Primary Dataset: A self-curated synthetic dataset generated using the Faker library and rule-based templates, consisting of over 50,000 conversational examples. The data covers diverse PII types such as names, emails, phone numbers, addresses, SSNs, credit card numbers, dates of birth, and medical conditions.

2.2 Benchmark Datasets:

1. **i2b2 2014 De-identification Challenge Dataset:** Contains de-identified clinical notes with protected health information (PHI).
2. **Enron Email Dataset:** Real-world corporate email communications with naturally occurring PII.
3. **CoNLL-2003 Dataset:** Standard named entity recognition (NER) dataset annotated with person, location, and organization entities.

3. NLP Methods

3.1 Pretrained Transformer Networks (Primary Approach): We will fine-tune DistilBERT or MobileBERT—lightweight transformer models optimized for edge deployment. These models will be adapted for token-level PII classification using Named Entity Recognition (NER) architecture.

3.2 Recurrent Networks (Secondary): A Bi-LSTM with CRF layer will serve as a complementary model for sequential pattern recognition. LSTMs naturally capture dependencies in sequences—useful for multi-word PII like "123 Main Street, Springfield, IL 62701." The CRF layer ensures valid tag sequences (e.g., preventing "B-PERSON I-LOCATION" transitions).

3.3 Rule-Based Models: Regex patterns and Naive Bayes classifiers will handle structured PII with predictable formats:

- a. Email: `[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}`
- b. Phone: `\b\d{3}[-.]\d{3}[-.]\d{4}\b`

c. SSN: \b\d{3}-\d{2}-\d{4}\b

This rule-based layer provides 100% precision for obvious patterns, allowing neural models to focus on ambiguous cases.

We'll implement **SHAP (SHapley Additive exPlanations)** to explain why specific tokens were flagged as PII.

4. Packages that will be used

- **Transformers**: Pretrained model access and fine-tuning
- **spaCy**: Fast NER baseline and linguistic feature extraction
- **NLTK**: Tokenization, POS tagging
- **PyTorch**: Model training and edge optimization
- **Presidio**: PII detection baseline and anonymization strategies
- **Faker**: Synthetic data generation for training
- **SHAP**: Model explanations
- **LIME**: Local interpretability for individual predictions
- **ONNX Runtime**: Cross-platform model deployment

5. NLP Tasks

5.1 Primary Task: Named Entity Recognition (Token Classification)

Multi-class sequence labeling using BIO tagging scheme:

- B-PERSON, I-PERSON (names)
- B-EMAIL, B-PHONE, B-SSN (contact/ID)
- B-ADDRESS, I-ADDRESS (physical locations)
- B-CREDIT_CARD, B-MEDICAL (financial/health)
- O (non-PII)

5.2 Secondary Tasks:

- **Binary classification**: PII-containing vs. clean messages
- **Pattern matching**: Regex-based structured PII extraction
- **Contextual disambiguation**: Distinguishing PII from non-PII in ambiguous cases

- **Anonymization strategy selection:** Choosing appropriate masking techniques (redaction, synthetic replacement, generalization)

6. Performance Metrics

6.1 Detection Performance:

- **Precision & Recall:** Measure accuracy and completeness of PII detection.
- **Entity-level F1-Score:** Evaluate balanced performance based on exact matches for complete PII entities.

6.2 Privacy Assurance:

- **False Negative Rate:** Critical to minimize missed PII instances to ensure privacy protection.

7. Rough Schedule

Week 1: Data Preparation & Preprocessing

- Aggregate and clean datasets .
- Generate synthetic PII examples using *Faker* and rule-based templates.
- Create train/validation/test splits.
- Apply data augmentation.

Week 2: Baseline & Initial Modeling

- Implement rule-based regex patterns for PII detection.
- Train classical models (Naive Bayes, logistic regression).
- Fine-tune *spaCy* NER model.
- Establish baseline performance benchmarks.

Week 3: Advanced & Hybrid Modeling

- Fine-tune *DistilBERT* and *MobileBERT* for token classification.
- Train BiLSTM-CRF model and experiment with ensemble methods (transformer + LSTM + rules).
- Integrate explainability tools (SHAP/LIME).

Week 4: Optimization, Deployment & Demo Development

- Apply model compression techniques.
- Convert models to ONNX for deployment on devices.
- Develop and deploy a demo application showcasing PII detection and masking capabilities.
- Complete documentation and performance analysis.