# PII Entity Recognition for Noisy STT Transcripts

Plivo Assignment

Kush Patil

# Problem Overview

- Noisy STT transcripts with spoken digits, “dot”, “at”, fillers.
- Detect 7 entity types:
  - `CREDIT_CARD`, `PHONE`, `EMAIL`, `PERSON_NAME`, `DATE`, `CITY`, `LOCATION`
- Mark PII=true for: `CREDIT_CARD`, `PHONE`, `EMAIL`, `PERSON_NAME`, `DATE`
- Output exact **character spans**.
- Priority: **High precision** over recall.
- Latency requirement: p95 $\leq$ **20ms** (CPU).

# Data Generation

- Synthetic STT-style dataset:
  - 900 training examples
  - 150 dev examples
  - Stress-test set
- Simulated STT noise:
  - Spoken digits (“nine eight double three”)
  - Spaced email tokens
  - Spoken credit-card chunks
  - No punctuation and uneven spacing

# Model & Tokenizer

- **Model:** DistilBERT (`distilbert-base-uncased`)
- **Tokenizer:** WordPiece
- Why DistilBERT?
  - Lightweight and fast
  - Strong NER performance
  - Great CPU latency
  - Perfect for BIO tagging + offset mapping

# Code Base Structure

- `dataset.py` — JSONL $\rightarrow$ BIO dataset
- `model.py` — DistilBERT TokenClassifier
- `train.py` — Training loop (focal loss)
- `predict.py` — Inference + validators
- `labels.py` — LABEL2ID, ID2LABEL, PII map
- `eval_span_f1.py` — Span-level evaluation
- `measure_latency.py` — Latency measurement
- `generate_stt_data.py` — Synthetic generator

# Key Techniques

- BIO $\rightarrow$ span reconstruction with offset mapping.
- Entity-specific confidence thresholds.
- Helper validators:
  - Email normalization + regex
  - Spoken-number parsing for PHONE
  - Spoken digits + Luhn check for CREDIT_CARD
- System optimized for **precision-first**.

# Final Metrics (Dev Set)

## Dev Set Metrics

CREDIT_CARD: P=1.000 R=0.125 F1=0.222
DATE: P=1.000 R=0.348 F1=0.516
EMAIL: P=1.000 R=1.000 F1=1.000
PERSON_NAME: P=1.000 R=1.000 F1=1.000
PHONE: P=1.000 R=1.000 F1=1.000

**PII Precision:** 1.000
**PII Recall:** 0.737
**PII F1:** 0.848
**Macro-F1:** 0.748

# Final Metrics (Stress Set)

## Stress Test Metrics

CITY: P=0.000 R=0.000 F1=0.000
CREDIT_CARD: P=1.000 R=1.000 F1=1.000
DATE: P=1.000 R=0.613 F1=0.760
EMAIL: P=1.000 R=1.000 F1=1.000
LOCATION: P=0.000 R=0.000 F1=0.000
PERSON_NAME: P=1.000 R=1.000 F1=1.000
PHONE: P=0.939 R=0.939 F1=0.939

**PII-only F1:** 0.939
**Macro-F1:** 0.671

# Latency Results

- CPU latency (batch size $= 1$):
  - p50: 6.61 ms
  - p95: 8.36 ms
- Far under the 20ms requirement.
- Trade-off:
  - Higher thresholds $\rightarrow$ extremely high precision
  - Slightly lower recall for DATE & CREDIT_CARD

# Conclusion

- Fine-tuned DistilBERT for noisy STT PII NER.
- Achieved near-perfect PII precision.
- Robust validators for spoken/obfuscated formats.
- Excellent CPU latency (¡10ms).
- Clean, modular, production-ready codebase.

**Thank You!**