Automated Tech Support Triage Model

Presented by Abed, Amit & Lior

Automated Tech Support Triage Model

User Case

Classifies customer support tickets, identifies subsystem issues, and determines severity levels automatically.

Why Important

Saves time and resources, speeds up resolution, prioritizes critical issues, routes to right experts.

The Challenge

Limited training data, varied staff experience, manual classification slows identification across diverse systems.

Problem Statement & Approach

Our Goal
Automatically analyze report tickets to predict component origin of the issue and severity level.

Inputs
- User reported issues
- Reporter metadata
- System description

Outputs
- Component/subsystem location
- Severity: Low, Medium, High

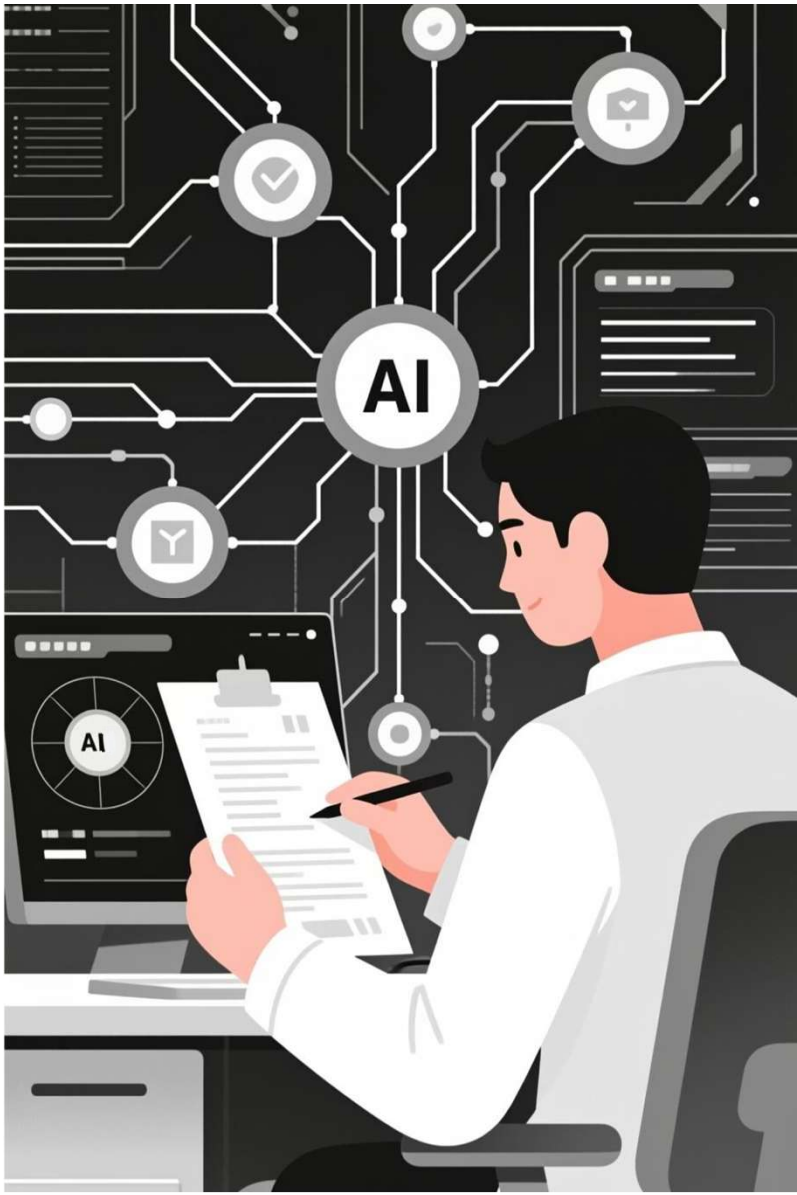
What Makes This Novel

Dual Prediction
Predicts both component and severity, not just single label.

Enhanced Training
Uses multiple methods and larger training datasets than current systems.

Metadata Intelligence
Leverages reporter metadata to gauge reliability and differentiate real issues from user errors. (E.g. experience, role...)

Business Impact Focus
Severity based on domain-specific impact, not generic urgency.



Models & Techniques

Models

- **Distilbert-base-uncased**
  Fine-tuned for ticket analysis
- **Llama-3-8B**
  Fine-tuned for classification

Techniques

- **Supervised Fine-Tuned Classifier**
  For severity determination
- **Zero Shot Classification**
  For component attribution

Fine-Tuning Approach

Attribution Classifier
Build a labeled dataset and evaluate on held-out test data

Severity Classification
Cross-entropy classification loss

Dataset Generation Strategy

01

Create Synthetic Dataset

Simulate typical IT system errors with generated workflow and components

02

Generate Issue List

Develop possible issues for each system component

03

Create User Reports

Generate natural-sounding reports with severity and component attribution

Evaluation & Quality Metrics

Supervised Fine-Tuned Classifier

Data Split

70% train, 15% validation, 15% test

Train & Validate

Use labelled reports, validate to avoid overfit

Test Evaluation

Compute measurements on test set

Zero-Shot Classification

Prepare Dataset

Build labelled dataset for fine-tuning

Consistent Prompting

Run tests individually or batch

Held-out test evaluation

Manual vs predicted labels for component

Key Performance Indicators

Accuracy

Overall correctness

Precision

True positive rate

Recall

Coverage measure

F1 Score

Balanced metric

Confusion Matrix

Detailed breakdown

Macro-F1

Average F1 across classes