# **Email Classification System With PII Masking**

## Introduction

Effective email management is critical for any support team seeking to provide timely and accurate responses to customer inquiries. The primary purpose of the email classification system is to automatically categorize incoming support emails into predefined categories, enabling support agents to prioritize and address issues more efficiently. By automating this process, the system not only enhances productivity but also improves customer satisfaction by ensuring that requests are directed to the appropriate team members without delay.

### Importance of Email Categorization

Categorizing support emails allows for:

* **Streamlined Workflow**: By organizing emails into categories such as 'Technical Support', 'Billing Issues', or 'General Inquiries', support teams can quickly assess and respond to requests.
* **Data-Driven Insights**: Classification aids in tracking trends in customer inquiries, enabling the team to identify areas for improvement in products and services.
* **Resource Allocation**: Understanding the volume and type of requests helps allocate resources effectively and manage workload among team members.

### Masking Personal Identifiable Information (PII)

In conjunction with email classification, masking personal identifiable information (PII) is vital for safeguarding customer privacy. PII, such as names, contact details, and payment information, needs to be anonymized to comply with data protection regulations like GDPR and CCPA. The significance of PII masking includes:

* **Compliance with Regulations**: Ensuring that sensitive customer information is protected according to legal standards.
* **Risk Mitigation**: Reducing the chances of data breaches, leading to potential reputational and financial damage.
* **Trust Building**: By prioritizing data privacy, companies foster trust with their customers, encouraging greater engagement and loyalty.

The combination of efficient email classification and robust PII masking forms the foundation of a supportive customer communication framework, ultimately leading to enhanced service delivery and improved customer relationships.

## Problem Statement

The challenge of effectively categorizing support emails lies in the high volume of diverse inquiries that organizations receive daily. With emails ranging from simple queries to complex issues, the need for an efficient classification system is paramount. This system must accurately categorize incoming emails into predefined categories such as **Technical Support**, **Billing Issues**, or **General Inquiries** to route them appropriately. However, the diversity and ambiguity of natural language in these emails complicate the classification process.

### Challenges in Email Classification

1. **Variability in Language**: Customer emails often contain various phrasings, slang, and even typos, making it difficult for automated systems to classify them correctly.
2. **Evolving Context**: The nature of customer inquiries may change over time, necessitating updates to classification criteria and algorithms to ensure accuracy.
3. **Subjectivity of Categories**: Different organizations may have unique interpretations of categories, complicating the standardization of email classification processes.

In addition to ensuring accuracy, the solution must also integrate a robust system for masking Personal Identifiable Information (PII). Given the rising concerns around data privacy, it is essential to process emails without reliance on large language models. Traditional PII masking techniques, such as **Named Entity Recognition (NER)** and **regular expressions (Regex)**, can be implemented to ensure consumer data remains secure and compliant with data protection legislations.

Thus, the problem is not solely about categorization efficiency; it is about balancing accuracy with the necessity of safeguarding sensitive information throughout the process. Addressing these challenges is crucial for developing a reliable email classification system that serves support teams effectively.

## Data Collection & Preprocessing

To develop a robust email classification system, the initial steps of **data collection** and **preprocessing** are vital for ensuring high-quality datasets that can inform the training algorithms.

### Data Collection Steps

**Sourcing Emails**: The dataset was compiled from a combination of previous customer interactions stored in the organization's customer support system. This included both resolved and unresolved tickets to provide a comprehensive representation of inquiries received.

**Data Sanitization**: Before further processing, emails were screened to remove any duplicates and irrelevant entries. Non-representative emails, such as automated replies or spam, were excluded to enhance the dataset's quality.

**Storage Format**: The collected emails were stored in a structured format (e.g., CSV or JSON) to facilitate easy processing later in the pipeline.

### Preprocessing Techniques

Preprocessing is crucial for preparing the email data for feature extraction and model training, ensuring both the integrity of the data and the privacy of personal identifiable information (PII).

Identifying and Masking PII

To safeguard PII while maintaining the usefulness of the data, several techniques were employed:

**Named Entity Recognition (NER)**:

* A machine learning model was trained to identify entities such as names, email addresses, and phone numbers within emails. Libraries like SpaCy or NLTK were utilized for this purpose.
* The extracted entities were replaced with placeholders (e.g., <NAME>, <EMAIL>), ensuring the original data couldn’t be traced back to individuals.

**Regular Expressions (Regex)**:

* For instances where NER might fall short, such as recognizing formatted data patterns (e.g., phone numbers, credit card numbers), regex patterns were defined to identify and mask PII effectively.
* Example regex for email: ([a-zA-Z0-9.\_%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}) replacing all email occurrences with <EMAIL>.

**Custom Methods**:

* In cases where conventional techniques did not yield satisfactory results, custom algorithms were developed to detect context-specific PII. For instance, analyzing the structure of sentences to infer potential names or personal identifiers.

### Secure Storage for Original Data

An essential component of the preprocessing phase involved setting up robust methods for storing the original email data securely:

**Encrypted Database**: Original emails containing PII were stored in an encrypted format in a secured database. This ensured that unauthorized access would not lead to data breaches.

**Access Controls**: Restricted access policies were implemented to limit the number of individuals who could view or restore the original data, ensuring compliance with legal requirements.

By following these meticulous steps in data collection and preprocessing, the stage was set for the subsequent phases of model training and deployment, ensuring efficacy in classifying emails while maintaining stringent privacy standards.

## Model Selection & Training

The selection and training of an appropriate machine learning or deep learning model are critical steps in developing an effective email classification system. This section outlines the process involved in selecting models, training them, and validating their performance.

### Model Selection

When classifying support emails, the primary objective is to choose models capable of addressing the variability and complexity inherent in natural language. Consideration was given to the following algorithms:

**Traditional Machine Learning Models**:

* **Support Vector Machines (SVM)**: Effective for high-dimensional spaces, particularly useful when working with features derived from text data.
* **Random Forest**: A robust ensemble learning method that can handle overfitting and provide excellent classification accuracy with a structured approach.

**Deep Learning Approaches**:

* **Recurrent Neural Networks (RNNs)**: Particularly Long Short-Term Memory (LSTM) networks, which can capture sequential dependencies in text data.
* **Transformers**: Models like BERT (Bidirectional Encoder Representations from Transformers) are particularly adept at contextual understanding, making them suitable for complex text classification tasks.

After evaluating the various models, BERT was selected for its state-of-the-art performance on a variety of NLP tasks. Its ability to understand the context of words within the sentence structure enhances its classification capability, making it an ideal choice for handling diverse customer inquiries.

### Training the Model

In training the selected model, several crucial steps were followed:

**Feature Extraction**:

* Using the BERT tokenizer, the textual data was converted into tokens, which are numerical representations of text. This tokenizer accounts for the nuances of language, allowing the model to understand context better.

**Data Preparation**:

* Input data was split into training, validation, and test sets to ensure that the model was generalized and could perform well on unseen data. Typically, a 70-20-10% split was applied.

**Model Training**:

* Utilizing libraries such as Hugging Face's Transformers, the selected BERT model was fine-tuned on the training data. Hyperparameters, such as learning rate and batch size, were optimized through techniques such as grid search or random search.

### Verification and Validation

To assess the model's performance, validation methods such as cross-validation were applied:

**Performance Metrics**: Evaluation was based on metrics including accuracy, precision, recall, and F1 score. These metrics help gauge how well the model categorizes emails across different classes.

**Confusion Matrix**: Visualization of true positive, true negative, false positive, and false negative rates was performed to diagnose the model's performance comprehensively.

**Fine-tuning**: Based on validation results, additional training cycles were conducted, adjusting model parameters to improve performance.

Through careful model selection, training, and validation, the email classification system was equipped with a robust backbone capable of adapting to various customer inquiries while maintaining a high level of accuracy.

## System Integration

The integration of the email classification system involves a seamless workflow that transitions from email reception to delivering classification results while ensuring personal information is correctly masked. This multi-step process consists of various components working together to provide a secure and effective solution for customer support teams.

### Workflow Overview

**Email Reception**:

* Emails are ingested from the support team’s email channel, typically through API endpoints, which enable automated processing.

**Preprocessing**:

* Incoming emails undergo preprocessing to filter out noise (e.g., spam) and standardize formatting. This includes the conversion to a consistent character encoding (like UTF-8) to ensure compatibility across different systems.

**PII Detection and Masking**:

* Using techniques like **Named Entity Recognition (NER)** and **Regex**, the system identifies and masks personal identifiable information (PII). All detected PII elements are replaced with placeholders (e.g., <NAME>, <EMAIL>) in real-time, preserving the integrity of the data while complying with privacy regulations.

**Feature Extraction**:

* The preprocessed and masked email text is transformed into numerical representations. For instance, the BERT tokenizer segments the text into tokens, ensuring contextual understanding is retained despite the masking.

**Email Classification**:

* The masked data is fed into the selected classification model (e.g., BERT). The model predicts the category of the email based on learned patterns from training data. Each email is assigned predefined categories such as 'Technical Support' or 'Billing Issues.'

**Result Delivery**:

* After classification, the results are formatted and returned to the support team via the API. This might include not just the classification label, but also confidence scores, enabling agents to gauge the reliability of the classification.

### Integrating Masking and Classification

The integration of the masking system and email classification ensures that sensitive information remains protected while accurate categorization is achieved. The pipeline is designed so that PII masking occurs before classification, ensuring that no unmasked data is handled during the processing phase. This two-layered approach provides dual assurance of privacy and efficient support workflow:

**Seamlessness**: By embedding PII detection and masking within the classification pipeline, the system minimizes the risk of accidental data exposure.

**Scalability**: As new types of PII emerge or classification categories evolve, the system can be adjusted with minimal disruption.

This integrated workflow fundamentally enhances the efficiency of support operations, allowing teams to focus on resolving customer issues promptly and securely.

## API Development & Deployment

The development of an API to expose the email classification system is a pivotal aspect of this project, enabling support teams to interact with the classification engine efficiently. We opted for **FastAPI** as our framework due to its performance, ease of use, and native support for asynchronous programming, which is essential for handling multiple requests concurrently.

### API Specifications

The API is designed with the following specifications:

* **Base URL**: /api/classify
* **HTTP Methods**:
  + **POST**: To submit emails for classification
* **Request Headers**:
  + Content-Type: application/json
* **Response Format**: JSON

### Input & Output Formats

Input Format

The input to the API consists of a JSON payload containing the email text. Below is an example of a typical request:

{  
 "email": "I'm having trouble with my billing information. Could you please assist me?"  
}

Expected Output Format

Upon processing, the API returns a JSON response containing the classified category and the confidence score, structured as follows:

{  
 "category": "Billing Issues",  
 "confidence": 0.92  
}

### Exception Handling

The API handles various potential errors, returning appropriate HTTP status codes along with error messages. For example:

* **400 Bad Request**: If the input data is malformed.
* **500 Internal Server Error**: For unexpected server issues.

### Security Measures

To ensure the confidentiality of sensitive data and compliance with regulations:

1. **PII Masking**: As emails are processed, any personal identifiable information is masked before classification, ensuring no sensitive data is exposed.
2. **Authentication**: Implementing JWT (JSON Web Tokens) to authenticate users and secure endpoints prevents unauthorized access.
3. **Rate Limiting**: To prevent abuse, we incorporated rate limiting which restricts the number of requests a user can make in a given timeframe.

By adhering to these specifications, the API is prepared for deployment, allowing seamless integration with various front-end applications and ensuring secure handling of customer support emails.

## Challenges Faced and Solutions Implemented

During the development of the email classification system, several significant challenges were encountered, particularly concerning personal identifiable information (PII) masking, classification accuracy, and API deployment. Below is a detailed overview of these challenges and the solutions implemented to overcome them.

### 1. PII Masking Difficulties

**Challenge**: The sensitive nature of PII posed a complex challenge when processing customer emails. Accurately identifying and masking information like names, email addresses, and phone numbers without losing context was critical.

**Solution**: A combination of **Named Entity Recognition (NER)** and **Regular Expressions (Regex)** was utilized to effectively detect and mask PII.

* **NER Models**: A robust machine learning model was developed to identify various entities within text accurately. Models like SpaCy were employed for their effectiveness in entity recognition.
* **Regex Techniques**: For structured data types, regex patterns provided a failsafe method for detecting PII, ensuring a comprehensive approach to data masking.

### 2. Classification Accuracy Concerns

**Challenge**: Variability in language usage among customers, including slang, typos, and diverse phrasing, often led to decreased classification accuracy. Additionally, the evolving nature of support categories posed further challenges in maintaining precision.

**Solution**: Implementation of a deep learning model, specifically **BERT**, enabled better contextual understanding of the text.

* **Continual Model Training**: The model was fine-tuned continuously with updated datasets reflecting current trends in customer inquiries. Cross-validation ensured the model remained adaptable to changing language patterns.

### 3. API Deployment Issues

**Challenge**: Deploying the API while ensuring optimal performance under high load, and maintaining security protocols was a substantial concern.

**Solution**:

* **FastAPI Framework**: The API was built using FastAPI, known for its speed and efficiency in handling asynchronous requests.
* **Security Protocols**: Measures such as JWT for authentication and strict rate-limiting policies were implemented to secure endpoint access and prevent misuse.
* **Error Handling**: A systematic error handling mechanism ensured that users were informed of any issues during classification, enhancing the overall user experience.

These strategies collectively addressed the significant challenges faced during the development of the email classification system, ensuring a robust solution that prioritizes both efficiency and security.

## Conclusion

The email classification system developed for the support team is pivotal for enhancing operational efficiency while safeguarding user privacy. By categorizing incoming emails into structured categories, support agents can address inquiries more swiftly, leading to improved customer satisfaction. The incorporation of personal information masking ensures compliance with data protection regulations, which is essential in today’s data-driven landscape.

### Future Improvements

Moving forward, several enhancements could be pursued:

* **Advanced Contextual Models**: Exploring state-of-the-art models beyond BERT, such as T5 or GPT, may enhance classification accuracy.
* **Real-Time Learning**: Implementing mechanisms for the model to learn from new interactions in real-time could improve responsiveness to evolving customer language and needs.
* **API Expansion**: Adding features such as sentiment analysis or multilingual support could broaden the system's applicability and effectiveness in diverse business environments.

These improvements could further solidify the system as a crucial asset for customer support operations.