Technical Handover Package: Text-Based Misinformation Detector

Executive Summary

Project Overview and Humanitarian Objectives: The project aims to develop an AI-based text detector to identify and flag harmful misinformation in social media posts, specifically targeting humanitarian operations in conflict zones. The goal is to enhance the safety and effectiveness of humanitarian efforts by providing timely alerts about potentially harmful content.

Technical Approach and Rationale: The solution employs a lightweight Natural Language Processing (NLP) model designed to operate efficiently on basic hardware, ensuring functionality in environments with limited connectivity. The offline-first deployment strategy allows for local processing and storage, crucial for conflict zones.

Key Implementation Decisions: The decision to use a lightweight NLP model was driven by the need for efficiency on basic hardware. Additionally, the integration of human-in-the-loop mechanisms ensures continuous improvement based on user feedback.

Production Requirements

Security Requirements

Data Security: Implement encryption for data at rest and in transit to protect sensitive information related to humanitarian operations.

Infrastructure Security: Utilize firewalls and intrusion detection systems to safeguard offline systems. Regularly update software to mitigate vulnerabilities.

Compliance Requirements: Ensure adherence to data protection regulations relevant to humanitarian contexts, including GDPR and local laws regarding data privacy.

Performance and Scalability

Expected Performance Benchmarks: Aim for an accuracy of at least 85% in identifying harmful misinformation, with precision and recall metrics above 80%.

Scalability Requirements: Design the system to handle up to 10,000 posts per day, with the

Scalability Requirements: Design the system to handle up to 10,000 posts per day, with the ability to scale based on demand during crisis situations.

Resource Allocation Guidelines: Allocate at least 4GB of RAM and 2 CPU cores for the NLP model to ensure smooth operation on basic hardware.

Error Handling and Resilience

Specific Error Patterns: Monitor for common NLP failure patterns, such as misclassification of benign content as harmful or vice versa.

Graceful Degradation Strategies: Implement fallback mechanisms that allow the system to continue functioning with reduced capabilities in case of model failure.

User-Friendly Error Messaging: Provide clear, non-technical error messages to users, explaining issues in a way that is understandable in humanitarian contexts.

Development Team Requirements

Skills and Expertise

Required Technical Skills: Proficiency in Python, experience with NLP libraries (e.g., SpaCy, Hugging Face Transformers), and familiarity with machine learning frameworks (e.g., TensorFlow, PyTorch).

Humanitarian Domain Knowledge Needs: Understanding of humanitarian operations and the socio-political context of conflict zones to tailor the model effectively.

Team Composition Recommendations: A team of 4-6 members, including 2 data scientists, 1 software engineer, 1 UX/UI designer, and 1 project manager.

Infrastructure and Tools

Development Environment Requirements: Use of Jupyter Notebooks for prototyping, Git for version control, and Docker for containerization.

Deployment Infrastructure for Offline-First: Set up local servers with sufficient storage and processing capabilities to handle the NLP model and data.

Monitoring and Logging Tools: Implement tools like Prometheus for monitoring and ELK Stack for logging to track system performance and issues.

Implementation Timeline

Phase 1: Foundation (4 weeks)

Core NLP implementation

Basic infrastructure setup

Security framework implementation

Phase 2: Integration (6 weeks)

Humanitarian workflow integration

User interface development

Data pipeline implementation

Phase 3: Production (4 weeks)

Production deployment for offline-first

Performance optimization

Monitoring implementation

Risk Assessment and Mitigation

Technical Risks

NLP Specific Challenges: Address potential biases in training data by diversifying sources and continuously evaluating model outputs.

Offline-First Deployment Risks: Ensure robust local storage solutions to prevent data loss during connectivity issues.

Mitigation Strategies: Regularly update the model with new data and feedback to adapt to evolving misinformation trends.

Humanitarian Context Risks

Data Privacy and Protection Risks: Implement strict data governance practices to ensure compliance with ethical standards.

Beneficiary Impact Risks: Engage with affected communities to validate the model's outputs and minimize negative impacts.

Operational Continuity Risks: Develop contingency plans for maintaining operations during

Operational Continuity Risks: Develop contingency plans for maintaining operations during crises or system failures.

Monitoring and Maintenance

Performance Monitoring

Key Metrics for NLP in Humanitarian Context: Track accuracy, precision, recall, and F1 score regularly.

Automated Monitoring Setup: Use automated scripts to generate performance reports and alerts for anomalies.

Alert Thresholds and Responses: Set thresholds for performance metrics that trigger alerts for immediate investigation.

Model Maintenance

Retraining Schedule and Procedures: Establish a quarterly retraining schedule using new data collected from user feedback and real-world performance.

Data Quality Monitoring: Regularly assess the quality of input data to ensure it remains relevant and representative.

Performance Degradation Detection: Implement monitoring for sudden drops in performance metrics to trigger immediate review.

User Support and Training

User Training Requirements for Humanitarian Staff: Conduct training sessions to familiarize users with the system and its functionalities.

Support Documentation Needs: Create comprehensive user manuals and troubleshooting guides tailored for humanitarian contexts.

Feedback Collection Mechanisms: Set up channels for users to report issues and provide feedback on the AI's performance.

Success Metrics and KPIs

Technical Metrics

Performance Benchmarks for NLP: Maintain accuracy above 85%, with precision and recall

Performance Benchmarks for NLP: Maintain accuracy above 85%, with precision and recall above 80%.

System Availability and Reliability: Aim for 99% uptime for the system in offline environments.

Response Time Requirements: Ensure that the system processes posts within 5 seconds on average.

Humanitarian Impact Metrics

Specific Impact Measures for Humanitarian Operations: Track the number of harmful posts flagged and the subsequent actions taken by humanitarian teams.

Operational Efficiency Improvements: Measure reductions in misinformation-related incidents affecting operations.

User Adoption and Satisfaction: Conduct surveys to assess user satisfaction and gather insights for improvements.

Compliance and Documentation

Required Technical Documentation: Maintain detailed documentation of the system architecture, model training processes, and user guides.

Audit and Compliance Requirements: Regularly review compliance with data protection regulations and ethical standards.

Change Management Procedures: Establish protocols for documenting and managing changes to the system and its components.

Emergency Procedures

System Failure Response Protocols: Develop a clear escalation path for addressing system failures, including contact points for technical support.

Data Breach Response Procedures: Implement a response plan for potential data breaches, including notification protocols and mitigation strategies.

Rollback and Recovery Procedures: Create procedures for rolling back to previous versions of the model or system in case of critical failures.