MediCouncil: Multi-Agent LLM Council for Symptom Triage

Done by:

Bhavatharini C (71762233004)

Gayathri SK (71762233014)

Abstract

MediCouncil is an AI-based decision support system that performs symptom triage by combining outputs from a council of specialized Large Language Models (LLMs). The system accepts free-text symptom descriptions along with basic patient context (age, sex, chronic conditions, red-flag indicators) and produces a structured triage output: risk level (Low/Medium/High/Critical), urgency recommendation (self-care, routine visit, urgent care, emergency), confidence score, and explainable reasoning. The key innovation is using a heterogeneous "LLM Council" approach with safety-first consensus rules to reduce single-model hallucinations and under-triage risk while keeping the system reproducible and modular for academic implementation and evaluation.

Problem Statement

Patients and healthcare support systems often struggle to determine the correct urgency of symptoms based on natural-language descriptions. Current digital symptom-checkers and single-model AI triage approaches can be inconsistent and may over-triage non-urgent cases or under-triage severe cases, creating safety and resource-allocation risks. Therefore, a reliable decision-support tool is needed that can analyze symptom descriptions with context, prioritize safety in critical cases, quantify uncertainty, and provide transparent reasoning for triage recommendations.

Objectives

- Build a multi-agent LLM council for symptom triage with three specialized roles (emergency risk, primary-care severity, guideline-based triage).
- Design a safety-first consensus mechanism with emergency override and weighted aggregation for non-emergency cases.
- Provide explainable outputs (agent-wise reasoning summary + key symptom drivers + final rationale).
- Implement explicit confidence scoring using inter-agent agreement and consistency checks.
- Benchmark the council against classical ML baselines (Naive Bayes, Logistic Regression, Random Forest) and a single-LLM baseline.
- Create an end-to-end web application with logging, reproducible configuration, and evaluation reports.

Literature Survey with Identified Research GAPs:

1) **TRIAGEAGENT: Heterogeneous Multi-Agent Framework for Clinical Triage (EMNLP Findings, 2024).**

Link: ***https://aclanthology.org/2024.findings-emnlp.329.pdf***

This paper introduces TRIAGEAGENT, a heterogeneous multi-agent LLM framework designed specifically for clinical triage. The system employs role-playing agents, each focusing on different aspects of triage, and integrates confidence scoring and early-stopping mechanisms to reduce unnecessary computation. Experimental results show TRIAGEAGENT outperforms single-LLM approaches and, in some cases, even experienced healthcare professionals on standardized triage datasets.

Identified Research Gap:

- Primarily evaluated on benchmark triage datasets and not designed for end-user, patient-facing symptom input.
- Limited emphasis on explainability of individual agent reasoning for non-expert users.
- Does not explicitly incorporate safety-first escalation logic suitable for public-facing tools.

Gap Addressed in Proposed Work:
The proposed LLM Council for Symptom Triage extends this idea by supporting free-text patient symptom descriptions, generating human-readable explanations, and applying a safety-first consensus mechanism suitable for real-world decision support.

2) **LLM-Driven Multi-Agent CDSS for Emergency Department Triage & Treatment Planning (arXiv, 2024).**

Link: ***https://arxiv.org/abs/2408.07531***

This study presents a multi-agent Clinical Decision Support System (CDSS) using Llama-3-70B, with agents representing clinical roles such as a triage nurse, emergency physician, pharmacist, and ED coordinator. The system is orchestrated using frameworks like LangChain and CrewAI, demonstrating improved coordination and decision quality in 5emergency department workflows.

Identified Research Gap:

- Designed primarily for hospital and clinician use, not for pre-hospital or patient-initiated triage.
- Requires large proprietary models and infrastructure, limiting reproducibility.
- Lacks lightweight deployment suitable for academic or low-resource environments.

Gap Addressed in Proposed Work:
The proposed system adapts the multi-agent concept into a lightweight, reproducible, patient-facing decision support tool, focusing on symptom-based risk assessment before hospital arrival.

3) Large Language Model Symptom Identification from Clinical Text (JMIR, 2025).

Link: ***https://pmc.ncbi.nlm.nih.gov/articles/PMC12313083/***

This study evaluates LLMs for symptom extraction from unstructured clinical text and compares them against traditional ICD-10 coding systems. Results show LLMs (especially GPT-4) achieve F1-scores above 94%, significantly outperforming rule-based coding approaches.

Identified Research Gap:

- Focuses only on symptom extraction, not on triage decision-making or urgency classification.
- Does not address risk stratification or care recommendation.

Gap Addressed in Proposed Work:
The proposed system builds on accurate symptom extraction and extends it to risk categorization, urgency assessment, and actionable triage guidance.

4) LLMs in Clinical Triage: Opportunities and Challenges (arXiv, 2024).

Link: ***https://arxiv.org/abs/2504.16273***

This paper evaluates LLMs on emergency department triage datasets such as MIMIC. While LLMs outperform traditional ML models, the study identifies a critical issue: under-triage of severe cases, which poses safety risks.

Identified Research Gap:

- Does not propose a concrete architectural solution to mitigate under-triage.
- Lacks a formal safety-first decision logic.

Gap Addressed in Proposed Work:
The proposed LLM Council explicitly introduces emergency override logic and risk-weighted consensus, ensuring severe cases are escalated even if only one agent flags danger.

5) Survey: LLM-Based Multi-Agent Systems in Medicine (TechRxiv, 2025).

Link:*https://www.techrxiv.org/doi/full/10.36227/techrxiv.176089343.36199495/v1*

This survey reviews multi-agent LLM systems across various medical domains. It concludes that council-style architectures improve reliability, reduce hallucinations, and increase decision confidence through debate and aggregation.

Identified Research Gap:

- Primarily a survey paper; does not provide an implementable triage framework.
- Lacks experimental validation in symptom triage scenarios.

Gap Addressed in Proposed Work:
The proposed project operationalizes these survey insights into a working, evaluated symptom triage system.

6) LLMs for Healthcare Decision Support (PMC, 2025).

Link: *https://pmc.ncbi.nlm.nih.gov/articles/PMC12189880/*

This review highlights the effectiveness of LLMs in healthcare decision support when used in assistive roles with human oversight, emphasizing explainability and ethical deployment.

Identified Research Gap:

- Does not propose a specific architectural design for triage.
- Lacks concrete implementation details.

Gap Addressed in Proposed Work:
The proposed system implements these principles via explainable outputs, disclaimers, and decision-support framing.

7) Comparing LLMs vs Clinicians in Diagnosis (PMC, 2025).

Link: *https://pmc.ncbi.nlm.nih.gov/articles/PMC12047852/*

This study compares LLM diagnostic accuracy with clinicians across multiple specialties, reporting 80%+ accuracy, with multi-model ensembles outperforming single models.

Identified Research Gap:

- Focuses on diagnosis, not triage or urgency classification.
- Limited emphasis on safety-critical decision support.

Gap Addressed in Proposed Work:
The proposed work applies ensemble benefits specifically to triage and risk assessment, not diagnosis.

8) LLM Workflows for Clinical Decision Support (PMC, 2025).

Link : *https://pmc.ncbi.nlm.nih.gov/articles/PMC12064692/*

This paper shows that multi-LLM workflows improve specialty referral accuracy and symptom severity assessment.

Identified Research Gap:

- Does not integrate a single unified triage risk output.
- Limited focus on patient-facing systems.

Gap Addressed in Proposed Work:
The proposed system outputs a single, clear risk level and recommended action suitable for patient-facing use.

9) Optimizing Disease Classification through Language Model Analysis of Symptoms (Nature Scientific Reports, 2024).

Link: *https://www.nature.com/articles/s41598-024-51615-5?utm_source=chatgpt.com*

This study demonstrates that NLP-based symptom analysis significantly improves disease classification accuracy.

Identified Research Gap:

- Focused on disease classification, not care urgency or triage.

Gap Addressed in Proposed Work:
The proposed system extends from classification to risk-based triage decisions with urgency outputs.

10) Diagnostic and Triage Accuracy of GPT-3 (PMC, 2023).

Link: ***https://pmc.ncbi.nlm.nih.gov/articles/PMC9915829/***

This study evaluates GPT-3's diagnostic and triage performance, showing promising results but warning against autonomous use.

Identified Research Gap:

- Relies on single-model inference.
- Limited safety mechanisms.

Gap Addressed in Proposed Work:
The proposed system replaces single-model dependence with a multi-agent, safety-first LLM council including consensus and escalation logic.

Tools and techniques to be used:

LLMs (Council Members)

- DeepSeek-R1 (Emergency/Red-Flag Agent): Strong long-context reasoning, used to detect high-severity patterns and red flags.
- GPT-OSS-120B (Primary Care Agent): Used for general severity assessment and common-condition reasoning.
- GLM-4.5V (Guideline/Triage Agent): Used to map case patterns to triage bands and urgency timeframes (can be text-only for this project).

Algorithms and decision logic

- Safety Override Rule: If any agent strongly indicates emergency/high-risk → escalate to High/Critical.
- Weighted Consensus Scoring: Combine normalized agent scores, e.g., Emergency (0.5), Guideline (0.3), Primary care (0.2).

- Confidence Scoring: Agreement-rate based confidence (how aligned the agents are) + conflict detection (flag low-confidence cases for human review).

Classical ML baselines (for benchmarking)

- Naive Bayes: Simple, fast baseline for text/symptom features.
- Logistic Regression: Interpretable linear baseline; good for calibrated probabilities.
- Random Forest: Non-linear baseline for structured symptom flags + demographics.

Statistical tools & evaluation methods

- Confusion Matrix: Error breakdown across triage classes.
- Precision / Recall / F1-score (per class): Especially focus on recall for High/Critical to avoid false negatives.
- ROC-AUC / PR-AUC (if binary emergency vs non-emergency evaluation is used).
- Inter-agent agreement metrics: agreement rate; optionally Cohen's Kappa / Krippendorff's Alpha for consistency reporting.
- Significance test (optional): McNemar's test to compare council vs baseline on paired predictions.

Modules:

1. User Interface Module: symptom input form + display of risk/urgency/confidence/explanation + disclaimer.
2. Input Validation & Preprocessing Module: normalize text, validate fields, construct standardized case JSON.
3. LLM Council Orchestrator Module: call three LLM agents (parallel/asynchronous), enforce JSON output schema.
4. Consensus & Safety Engine Module: emergency override + weighted scoring + final triage mapping.
5. Explainability Module: summarise key reasons from each agent + final combined explanation.

6. Baseline ML Module: classical ML models training/inference for benchmarking.
7. Evaluation & Reporting Module: metrics computation, plots/tables, error analysis.
8. Logging & Audit Module: store inputs, agent outputs, final decision, timestamps, and evaluation logs.

Methodology:

1. Dataset preparation: collect public symptom datasets; create a labeled triage dataset (or map symptom patterns to triage labels) and generate additional synthetic cases for coverage of emergency and non-emergency scenarios.
2. Feature engineering (for baselines): represent symptoms via TF-IDF or bag-of-words; combine with structured flags (age group, comorbidities, red flags).
3. Council prompting: design three role prompts with strict JSON schema outputs; test prompts on a small validation set and refine for consistent formatting.
4. Inference pipeline: for each case, run three agents → parse outputs → apply safety override → compute weighted consensus → compute confidence → generate explanation.
5. Benchmarking: run council, single-LLM baseline, and ML baselines on same test set; compute metrics and compare.
6. Deployment: implement web UI + FastAPI backend; add logging, rate limiting, and clear disclaimers.

Testing Plan / Performance Metrics:

Testing plan

- Unit testing: JSON output parsing, scoring functions, override logic, confidence computation.
- Integration testing: end-to-end request → council → consensus → response.
- Robustness testing: incomplete inputs, contradictory symptoms, long text, spelling variations.

- Safety testing: curated emergency cases (chest pain + radiation, stroke signs, severe breathing difficulty, meningitis signs) to ensure escalation triggers correctly.

Performance metrics (report these)

- Emergency Recall (Critical metric): fraction of true emergency/high-risk cases correctly escalated.
- Macro F1 / Weighted F1: overall classification quality across classes.
- False Negative Rate for High/Critical: must be minimized.
- Agent Agreement Rate: used as a confidence indicator.
- Latency per request: average response time (important due to multi-model calls).

Final Outcomes and Output:

Final outcomes

- A working decision-support web application implementing a multi-LLM council for triage.
- A reproducible evaluation report comparing council vs single-LLM vs classical ML baselines.
- A safe-by-design consensus mechanism with explicit emergency escalation and uncertainty handling.

Output format (example)

- Risk Level: Low / Medium / High / Critical
- Urgency Recommendation: self-care / routine visit / urgent care (2–4h) / emergency now
- Confidence: High/Medium/Low (based on agreement)
- Explanation: short combined rationale + agent-wise key reasons
- Disclaimer: “Decision-support only; consult medical professional.”

Identified Dataset:

Use a combination of public datasets and curated synthetic cases for robust coverage:

- Disease Symptoms and Patient Profile dataset (Kaggle) – symptoms + demographics for case creation.
- Disease and Symptoms dataset (Kaggle) – broad symptom-condition coverage to generate varied symptom combinations.
- Hospital Triage and Patient History dataset (Kaggle/public) – reference for triage-style labels and distributions.
- MIMIC-IV-ED (credentialed access) – gold-standard ED triage dataset for advanced evaluation if available.