Trial Helper: A Hybrid SMT-LLM System for Conversational Clinical Trial Matching

Vivien Cheng [1] Steve Dou [1]

[1]Department of Computer Science, Stanford University

1. Motivation & Problem

Clinical trial recruitment is broken:

- Only **3–5%** of eligible patients enroll in clinical trials
- **97%** of trials fail to meet enrollment goals on time
- Eligibility criteria span **20+ inclusion/exclusion conditions**

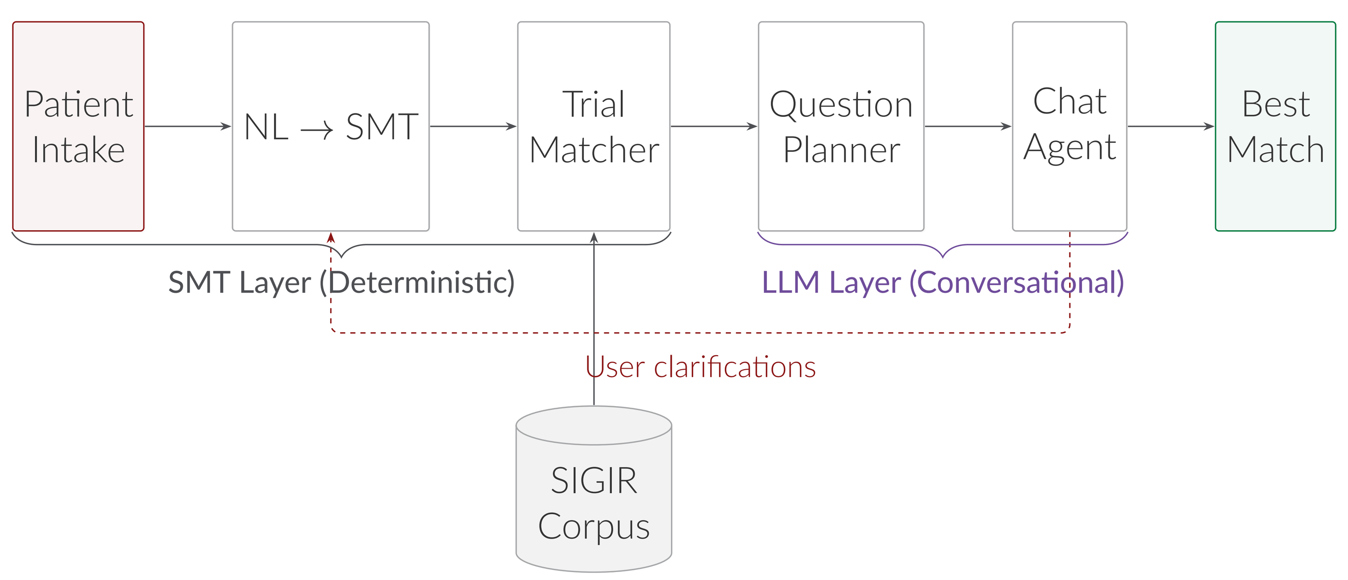
Key Observation: Eligibility has two distinct components:

Hard Constraints	Soft Preferences
Boolean logic (must satisfy)	Patient preferences
e.g., "age ≥ 18", "no infections"	e.g., "prefer oral meds"

Our Approach: SMT for hard constraints + LLM for preferences implies *Guaranteed correctness* with *natural conversation*.

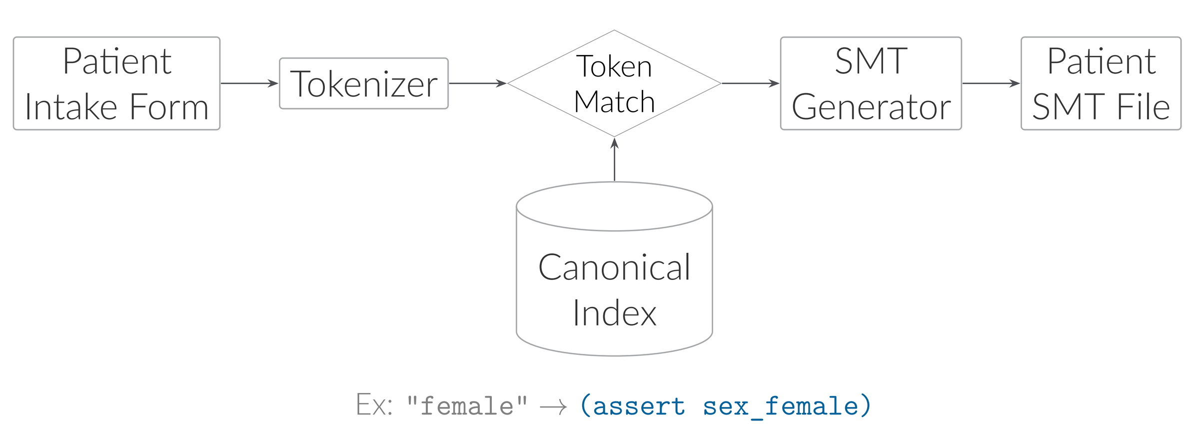
2. Technical Architecture

Complete Trial Matching Pipeline



Step 1: Natural Language to SMT Encoding

This step is inspired by the SMT methodology from CS 224V Lecture 11.



Step 2: Trial Matching

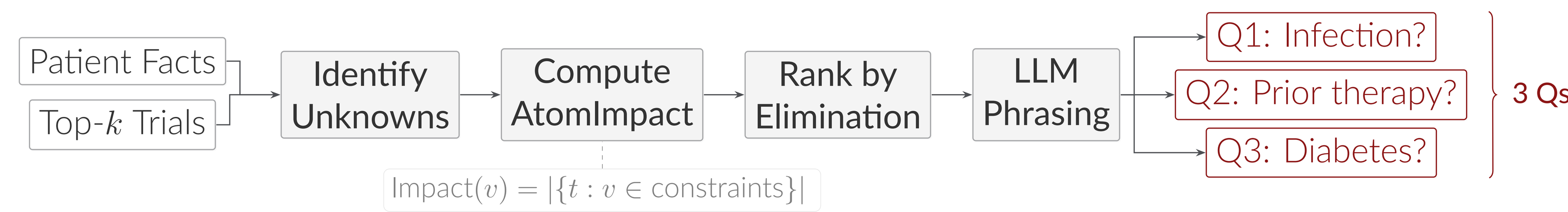
```
# Exclusion: any match ⇒ ineligible
# Inclusion: score = satisfied / total constraints
# Ranking: Exclude → Score → Disease relevance → Top-k
```

2. Technical Architecture (Cont.)

Step 3: Smart Follow-Up Questions (AtomImpact)

Select questions that eliminate the most trials by computing impact for each variable v:

$$\text{Impact}(v) = |\{t : v \in \text{Inc}(t) \cup \text{Exc}(t)\}|$$



Design Characteristics:

- **Deterministic eligibility**: SMT solving guarantees 100% constraint satisfaction—patients only see trials they formally qualify for
- **Conversational refinement**: LLM handles soft preferences (location, procedure comfort, scheduling) through natural dialogue
- **Information-theoretic questions**: AtomImpact metric identifies which yes/no questions most reduce trial set uncertainty
- **Session persistence**: Flask maintains patient state across multi-turn interactions

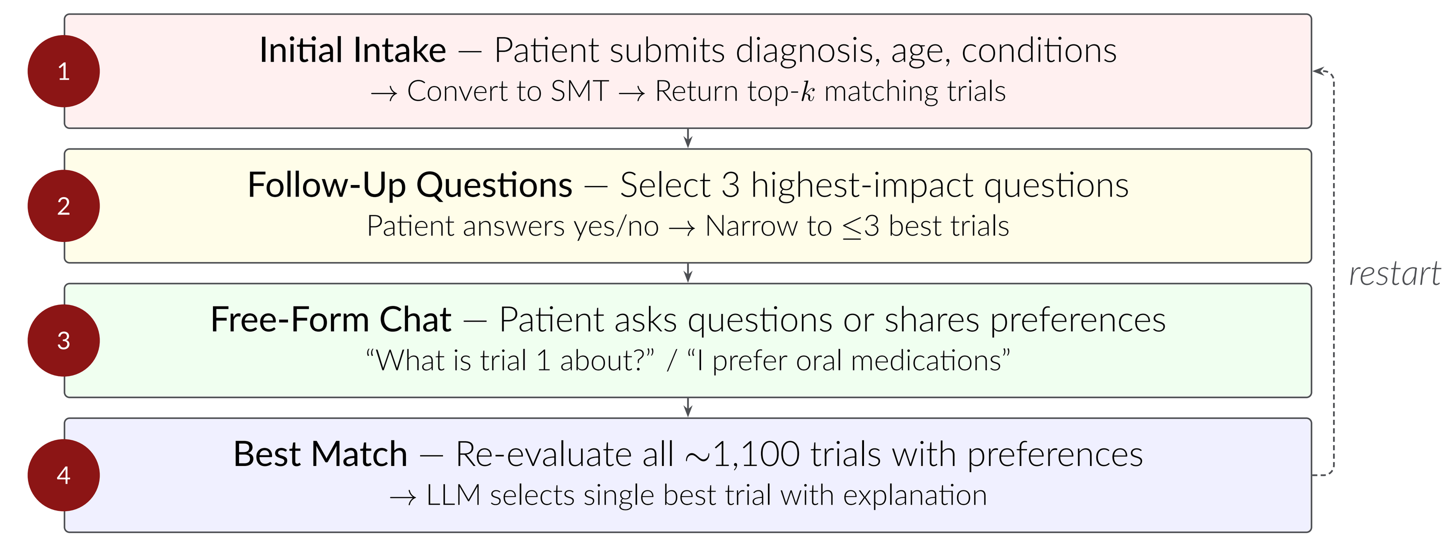
Dataset & Scope:

- SIGIR Clinical Trial corpus: ∼1,100 trials across 8 disease categories
- Pre-computed Boolean constraint projections per trial (.smt2 files)
- Canonical variable index mapping natural language to formal atoms

Chat Agent Intent Classification (`chat_agent.py`):

Intent	Example	Action
question	"What is trial 1 about?"	Answer using trial metadata
preferences	"I prefer oral medications"	Re-rank all trials; return best one
restart	"Start over"	Clear session; show new form
end	"Thank you!"	Farewell; preserve history

3. Conversational Pipeline



4. Demo: Example Interaction

INPUT	**Patient:** Age 58, lung cancer, hypertension. Find 5 trials.
QUESTIONS	**Trial Helper:** Found 5 lung cancer trials. *Loading...* 1. Do you have any active infectious disease? 2. Have you received prior immunotherapy? 3. Do you have uncontrolled diabetes?
RESPONSE	**Patient:** No, no, yes
REFINED	**Trial Helper:** Based on your answers, up to 3 trials remain. Do you have questions or preferences to share?
PREFERENCE	**Patient:** I prefer oral medications over injections.
RESULT	**Trial Helper:** Best match: **NCT02345678** *Why:* Daily pill regimen, no IV infusions, matches your diagnosis.

5. Contributions & Future Work

Technical Contributions:

1. **Hybrid SMT-LLM architecture** separating hard constraints from soft preferences
2. **AtomImpact algorithm** for optimal follow-up question selection
3. **Intent-driven chat agent** with structured preference extraction
4. **End-to-end web interface** demonstrating the complete pipeline

Current Limitations:

- Relies on pre-extracted trial constraints (manual curation required)
- Limited to Boolean eligibility criteria
- No formal user study conducted yet

Future Directions:

- Automated constraint extraction from trial protocols via LLM
- Quantitative user study measuring task completion and satisfaction
- Extension to numeric constraints (lab values, dosages)
- Integration with live ClinicalTrials.gov API

Acknowledgments

We thank Prof. Monica Lam for instruction and Vardhan Agrawal for weekly mentorship. Thanks also to the CS 224V staff for guidance throughout the quarter: Arjun Jain, Arpandeep Khatua, Jon Saad-Falcon, Wesley Tjangnaka, and Yucheng Jiang.

References

1. M. Lam, "Satisfying Natural Language Constraints with SMT," CS 224V Lecture 11, Stanford, 2025.
2. B. Koopman and G. Zuccon, "A Test Collection for Matching Patients to Clinical Trials," *Proc. ACM SIGIR*, pp. 669–672, 2016.
3. L. De Moura and N. Bjørner, "Z3: An Efficient SMT Solver," *Proc. TACAS*, pp. 337–340, 2008.
4. J. Achiam et al., "GPT-4 Technical Report," *arXiv:2303.08774*, 2023.