NL2FOL: Translating Natural Language to First-Order Logic for Logical Fallacy Detection

Presenter

Connor Dilgren

Motivation

- Problem: how to automatically detect logical fallacies in natural language?
- Benefits:
  - Reduce misinformation, manipulative text, propaganda
  - Improve rational discourse
  - Similar role as community notes on X
- Logical fallacies
  - Faulty generalization, ad hominem, circular claim, etc.
  - Lack of evidence/premises
  - Also false premises? Not clear in paper.

First-Order Logic Refresher

- Extension of Zeroth-Order Logic (aka Propositional Logic)
  - Proposition: a statement that is either true or false
  - Propositional connectives (e.g., negation, conjunction, disjunction) connect propositions to form compound propositions, which also evaluate to true or false
  - ZOL example: “College Park is in Maryland”
- First-Order Logic (aka Predicate Logic)
  - Variable: placeholder for an object
  - Quantifiers: describe how many objects there are (for all, there exists)
  - Predicates:
    - Describe properties of objects
    - Take objects as arguments, and evaluate to true or false
  - FOL example: “For all x, if x is in College Park, then x is in Maryland.”

Methodology

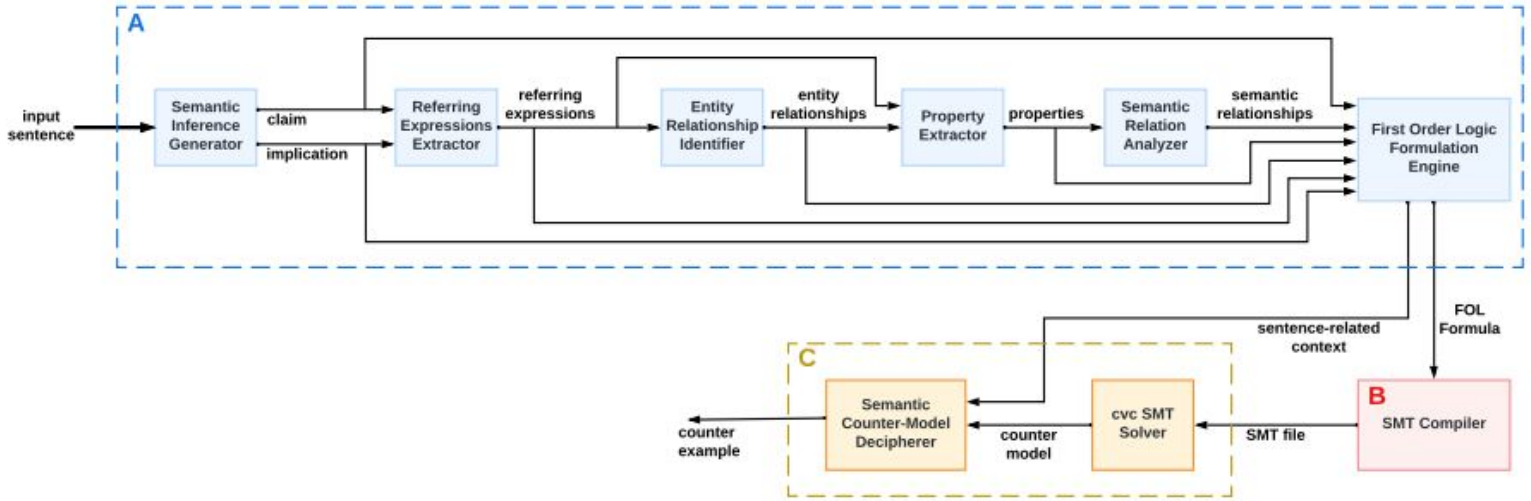


Figure 1: Proposed Logical Fallacy Detection Methodology: *Module A* converts natural language input to a first-order logic formula merged with contextual relationships, *Module B* compiles the negation of a given logical formula to an SMT file with well-defined sorts for variables and predicates, and *Module C* is used to run CVC on the SMT file and if the negation is satisfiable, interpret the counter-model in natural language.

Module A: NL Sentence -› FOL

- Input:
  - A natural language sentence
  - Must contain an implication, and zero or more claims
- Semantic inference module transforms sentence into claim + implication
- Claim + implication split into:
  - Referring expressions: identifies objects
  - Relations between entities: subset, equality, not related, etc.
  - Properties: same as predicates in first-order logic
  - NLI between each pair of properties to get their relationships
- LLM takes all this and generates:
  - First-order logic formulation of the sentence (for SMT compilation)
  - Passes sentence-related context to Module C (for SMT interpretation)

Module A Example

- Original Sentence: My roommate has a pet, and all pets are dogs. Therefore, my roommate has a dog.
- Claims
  - My roommate has a pet
  - All pets are dogs
- Implication: My roommate has a dog
- Referring expressions:
  - My roommate: r
  - Pets: p
  - Dogs: d
- Relationships between entities:
  - Dogs is a subset of pets
  - Pets is a subset of dogs (?)
- Properties
  - HasA(r, p)
  - IsARoommate(r)
  - IsAPet(p)
  - IsADog(d)
- Relationships between properties: none
- First-order logic form:
  - $(\text{HasA}(r, p) \land \text{IsARoommate}(r) \land \text{IsAPet}(p) \land \forall d(\text{IsADog}(d) \rightarrow \text{IsAPet}(d))) \land \forall p(\text{IsAPet}(p) \rightarrow \text{IsADog}(p)) \rightarrow \text{HasA}(r, d) \land \text{IsADog}(d)$

Module B: FOL -› SMT Compiler

- Satisfiability Modulo Theory (SMT) solvers are used to detect logical fallacies in first-order logic
- Authors created first compiler for converting a first-order logic formula to SMT file
- cvc4 solver

Algorithm 1 Logical Formula to SMT Compilation

1. Split the formula across any operator, parentheses, or commas into tokens.
2. Process tokens to instances of operators, variables and predicates. For predicates, identify all arguments and recursively process tokens for the arguments separately.
3. Convert the main logical formula from infix to prefix form. For predicates, recursively convert the arguments to prefix form.
4. Identify sorts of all variables and predicates using unify_sort described in Algorithm 2.
5. Parenthesize the prefix form formula to bring it into SMT format appropriately.
6. Create the SMT file by declaring appropriate sorts, variables and predicates using $(\text{declare} - \text{sort})$ and $(\text{declare} - \text{fun})$. Assert negation of the logical formula. Add $(\text{check} - \text{sat})$ and $(\text{get} - \text{model})$ to the SMT file.

Module C: SMT Solver + Interpreter

- Given SMT file, the SMT solver returns:
  - Satisfiable (original claim is a logical fallacy) or unsatisfiable (original claim is valid), since solver is evaluating the negated FOL formula
  - If satisfiable, a model that makes the formula true, which serves as a counterexample to the original sentence
- SMT solver results are difficult to interpret for non-experts
- LLM outputs an explanation of the counterexample in natural language, given the claim, implication, referring expressions, properties, first-order logical formula, and the counter-model generated by the SMT solver

Experimental Setup

- Main dataset:
  - LOGIC: 2,449 common logical fallacies (no valid sentences)
  - Stanford Natural Language Inference (SNLI) Corpus: 170,000 sets of hypothesis, premise sentences with label of entailment, contradiction, or neutral
    - Equal number of valid statements sampled
- Challenge dataset:
  - LOGICCLIMATE: 1,079 logical fallacies from Climate Feedback website (no valid sentences)
  - SNLI
    - Same as main dataset
    - Equal number of valid statements sampled
- Models
  - BART-MNLI (Zero shot)
  - 5 Pretrained language models (Few shot, one with COT)
  - NL2FOL (Few shot)
    - Llama 2-7B for prompting
    - BART-MNLI for identifying relationships between properties, referring expressions

Results - Main Dataset

- Pretrained models performed better than NL2FOL on LOGIC+SNLI dataset
  - Authors suspect that these models have the LOGIC dataset in their training data
- BART-MNLI performs poorly, labels every sentence as a logical fallacy
- NL2FOL has a high recall and a low precision
  - Actual logical fallacies are rarely labelled valid, and actual valid statements are commonly labelled logical fallacy
  - Proving a statement to be valid is harder than identifying it as a logical fallacy
  - If some semantics or ground truth necessary to prove validity is missing, then it is easy to build a counterexample

Model	Acc	P	R	F1
BART-MNLI (Zero Shot)	0.58	**1**	0.15	0.26
Llama-7b (Few Shot)	0.41	0.45	0.82	0.58
Mistral-7b-Instruct (Few Shot)	0.85	0.85	0.86	0.85
GPT3.5 (Few Shot)	0.88	0.86	0.91	0.89
GPT4 (Few Shot)	0.95	0.97	0.94	0.95
GPT4 (Few Shot with COT)	0.94	0.95	0.94	0.94
Claude 3 Opus (Few Shot)	**0.97**	0.96	**0.98**	**0.97**
NL2FOL (Few Shot)	0.63	0.58	0.92	0.71

Table 2: Model performance on the LOGIC+SNLI dataset, showcasing accuracy (Acc), precision (P), recall (R), and F1 score (F1).

Results - Challenge Dataset

- NL2FOL seems to generalize well
  - The in-context examples come from the LOGIC dataset
- Authors claim this demonstrates that NL2FOL adapts well to real-world text, though this is only one domain-specific context
- Authors claim this is a more fair comparison, though they do not know which datasets are in the pretrained models' training data

Metric	Acc	P	R	F1
BART-MNLI (Zero Shot)	0.57	**1**	0.14	0.25
Llama-7b (Few Shot)	0.31	0.38	0.62	0.47
Mistral-7b-Instruct (Few Shot)	0.62	0.68	0.44	0.53
GPT3.5 (Few Shot)	0.63	0.81	0.39	0.53
GPT4 (Few Shot)	0.64	0.91	0.30	0.45
GPT4 (Few Shot with COT)	0.66	0.90	0.36	0.51
Claude3 Opus (Few Shot)	**0.67**	0.92	0.38	0.54
NL2FOL (Few Shot)	0.66	0.60	**0.94**	**0.73**

Table 3: Comparison of accuracy (Acc), precision (P), recall (R), and F1 score (F1) Metrics for various approaches for the LOGICCLIMATE+SNLI dataset.

Conclusion

- The need for automatic logical fallacy detectors will increase as AI-generated misinformation increases
- NL2FOL is a potential solution, though it labels valid statements as logical fallacies at high rates and should be tested on a more diverse dataset
- Future work
  - Use more advanced LLMs, especially for NL -> FOL conversion
  - Utilize Constrained Decoders to ensure generated output has correct syntax
  - Incorporate NL2FOL into LLMs, to ensure they don’t generate logical fallacies
  - Create dataset containing natural language formulas with annotated first-order logic labels

Social Impact Assessor

Georgios (George) Milis

The authors' self-assessment

Positives:

- Tracking misinformation
- Validating claims
- Accessible NL interpretation of SAT result

Negatives:

- Over-reliance on AI judgment
- Threats to free speech

Neglected positives

- An impartial analyzer of political discourse
- An educational tool for teaching critical thinking

Neglected negatives

- Unavoidable bias of LLMs
- Natural language usually does not translate to first-order logic

Image by ChatGPT. Prompt chain:

1) Generate an image showing that a user is confused because his post "I like this weather" is marked as misinformation by a social media site

(Generated image with sunny weather)

2) Can you change the background so that the weather appears rainy, and the user saying "But I like rainy weather!". Everything else is the same.

Scientific Peer Reviewer

Paul Zaidins

Technical Correctness: 2

- The authors claims an “effective technique to interpret the results of cvc4” one of their major contributions
  - The empirical experiments on classification datasets and therefore do not test this
  - Either give empirical evidence or remove claim
- There is definitely some patterns in the tables worth considering that are simply not addressed
  - False positives are a bigger problem for NL2FOL and false negatives for the tested LLMs

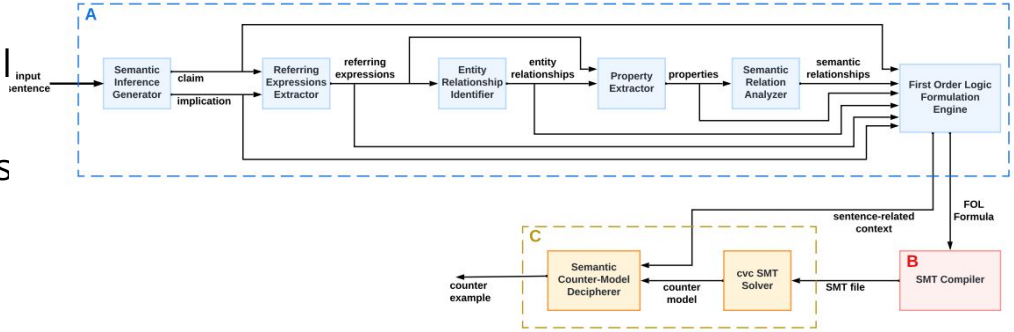
Model	Acc	P	R	F1
BART-MNLI (Zero Shot)	0.58	**1**	0.15	0.26
Llama-7b (Few Shot)	0.41	0.45	0.82	0.58
Mistral-7b-Instruct (Few Shot)	0.85	0.85	0.86	0.85
GPT3.5 (Few Shot)	0.88	0.86	0.91	0.89
GPT4 (Few Shot)	0.95	0.97	0.94	0.95
GPT4 (Few Shot with COT)	0.94	0.95	0.94	0.94
Claude 3 Opus (Few Shot)	**0.97**	0.96	**0.98**	**0.97**
NL2FOL (Few Shot)	0.63	0.58	0.92	0.71

Metric	Acc	P	R	F1
BART-MNLI (Zero Shot)	0.57	**1**	0.14	0.25
Llama-7b (Few Shot)	0.31	0.38	0.62	0.47
Mistral-7b-Instruct (Few Shot)	0.62	0.68	0.44	0.53
GPT3.5 (Few Shot)	0.63	0.81	0.39	0.53
GPT4 (Few Shot)	0.64	0.91	0.30	0.45
GPT4 (Few Shot with COT)	0.66	0.90	0.36	0.51
Claude3 Opus (Few Shot)	**0.67**	0.92	0.38	0.54
NL2FOL (Few Shot)	0.66	0.60	**0.94**	**0.73**

LOGIC (top), LOGICCLIMATE (bottom)

Scientific Contribution: 6, 7

- Provides a Valuable Step Forward in an Established Field (6)
  - A novel approach to automated logical fallacy detection using a chain of LLM queries and allowing some “skip layers
- Establishes a New Research Direction (7)
  - This chain of LLMs approach is worth investigating for general use



Presentation: 2

- A graphical display of the chart information would have been helpful
  - There a few enough LLMs that a scatter plot would have been useful

Recommended Decision: 1

- Only minor technical and presentation issues
- Contributes to the field
- Reviewer Confidence: 2

Archaeologist

Dinithi Wickramaratne

Previous Work

- **QA-NatVer**: Question Answering for Natural Logic-Based Verification (Aly et al. 2023)
  - QA-NatVer is a natural logic inference system that **composes a proof** by casting natural logic into a **question answering framework**.

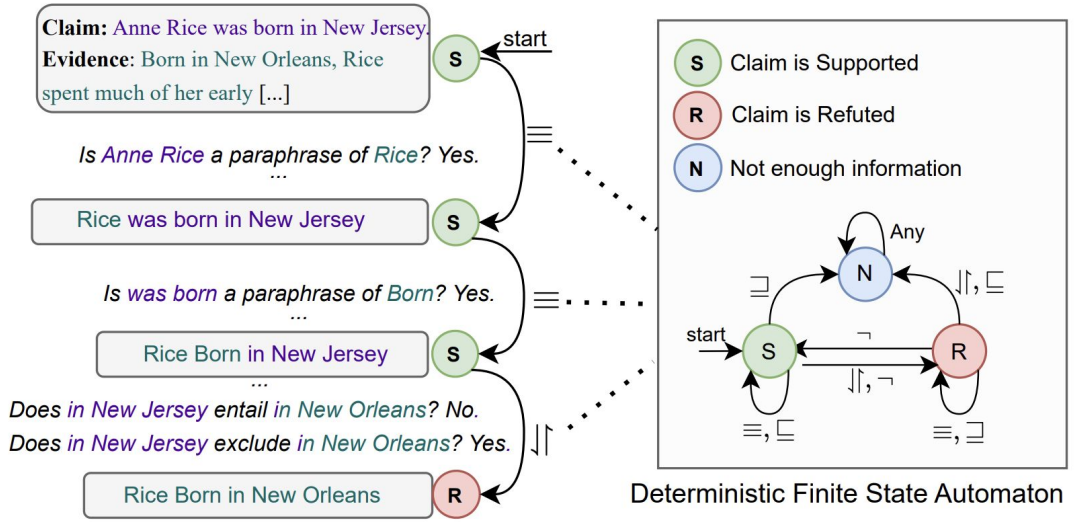
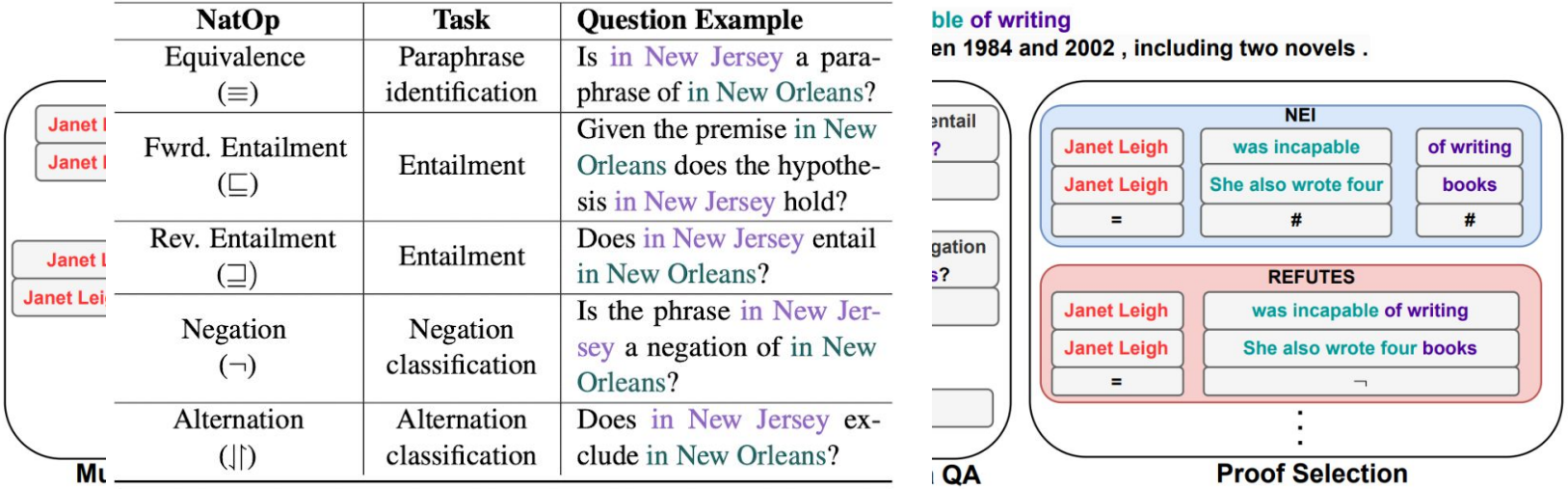


Figure 1: At each inference step, a claim span is mutated into an evidence span via a natural logic operator (NatOp). The current veracity state and mutation operator determine the transition to the next state, via a fine state automaton (DFA). Starting at S, the span *Anne Rice* is mutated via the equivalence operation (≡), resulting in S, according to the DFA. The inference ends in R, indicating the claim's refutation. We use question-answering to predict the NatOps, taking advantage of the generalization capabilities of instruction-tuned language models.

Previous Work (contd.)

NatOp	Task	Question Example
Equivalence (≡)	Paraphrase identification	Is in New Jersey a paraphrase of in New Orleans?
Fwrd. Entailment (⊑)	Entailment	Given the premise in New Orleans does the hypothesis in New Jersey hold?
Rev. Entailment (⊒)	Entailment	Does in New Jersey entail in New Orleans?
Negation (¬)	Negation classification	Is the phrase in New Jersey a negation of in New Orleans?
Alternation (⇃↾)	Alternation classification	Does in New Jersey exclude in New Orleans?



- Generate claim-evidence alignments between overlapping spans of varying length, predict the NatOp for each pair
- Predict NatOps using operator-specific boolean questions (use of instruction-fine tuned language models).
- To select the best proof, combine the answer scores to the questions associated with each proof.

Comparison with NL2FOL

NL2FOL	QA-NatVer
Logical fallacy detection	Identify the relations between claim and evidence
Uses chain of language model	Uses chain of language model
Include ground truth information	Doesn’t explicitly include ground truth information

Subsequent Work

- Understanding Enthymemes in Argument Maps: Bridging Argument Mining and Logic-based Argumentation (Ben-Naim et al. 2024)
  - Cited in the literature review : translating text to logic using LLMs
  - Summary:
    - Argument mining: identify arguments (premises and/or claims), and the support or attack relationships between those arguments.
    - The main challenge addressed : **real arguments often lack explicitly stated premises** necessary for the entailment of claims.
    - Proposes a framework for bridging the gap between argument mining and logic-based argumentation using **enthymemes** in argument maps.
    - The proposed solution combines classical logic and default logic:
      - Classical logic: represent explicit information in the text.
      - Default logic: represent implicit information (enthymemes)

Industry Expert

Pranav Sivaraman

AI-Powered Debate Moderator

- Develop a pipeline for real-time logical fallacy detection
- Builds upon the existing baseline pipeline described in the paper.
- Overall goal is to improve public discourse and critical thinking.

Pros

- **Enhancing public discourse:** Logical fallacies (the concept) are well defined. Identifying certain fallacies can be hard.
- **Promoting critical thinking:** Debaters need to be careful in how they word their arguments.
- **Scalable to other applications:** Methodology describe in the paper could extend to to other applications (Answer Set Programming). Potential for LLMs to generate rules and solvers list all the facts.

Cons

- **Context limitations:** Complex ideas may require more context or elaboration, which is challenging given the context window limitations of current LLMs. The transformation from natural language to first-order logic (FOL) may be inaccurate.
- **Performance concerns:** If the use case is real-time, the pipeline's runtime is a concern. Logical fallacies need to be detected quickly, but the process of prompting the LLM, converting from NL (natural language) to FOL, and then solving can be too slow.
- **Trust and verification:** Verifying that the LLM is generating the correct FOL from natural language statements is difficult, which puts the reliability of the entire pipeline at risk.

NL2FOL: Translating Natural Language to First-Order Logic for Logical Fallacy Detection

Private Investigator

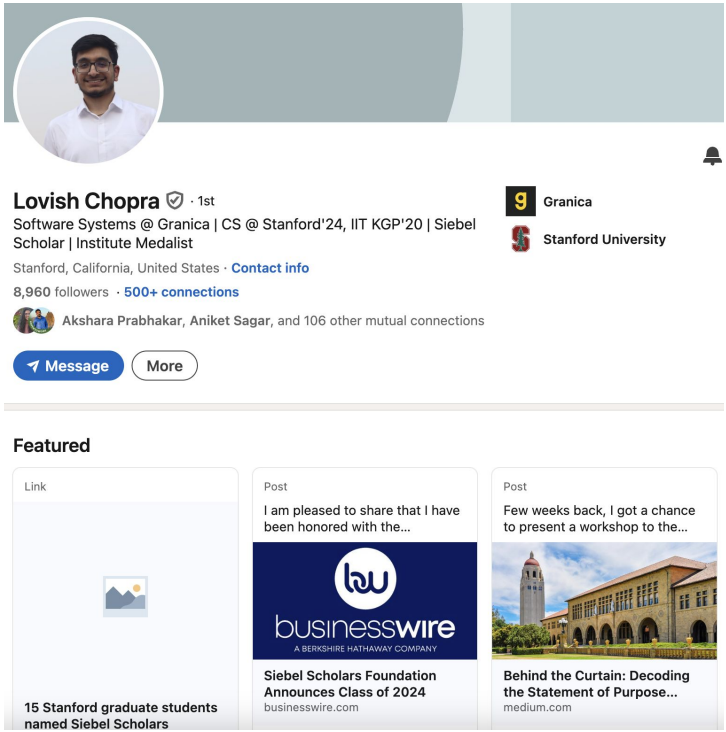
Nishit Anand

Second Author of the Paper - Lovish Chopra

- I will be covering the paper's second author - Lovish Chopra
- I know him since 2+ years
- We connected over LinkedIn two years ago and I've talked to him before too
- I interviewed him regarding the paper over Google Meet and have included his responses

NL2FOL: Translating Natural Language to First-Order Logic for Logical Fallacy Detection

Abhinav Lalwani[1*] **Lovish Chopra**[1*] **Christopher Hahn**[2†] **Caroline Trippel**[1]
Zhijing Jin[3,4] **Mrinmaya Sachan**[4]
[1]Stanford University [2]X, the moonshot factory [3]Max Planck Institute [4]ETH Zürich
{lalwani,lovish}@stanford.edu



LinkedIn

Github

Lovish Chopra
lovishchopra
Follow
MS CS, Siebel Scholar at Stanford | D. E. Shaw | Institute Medallist at IIT Kharagpur
5 followers · 5 following
lovishchopra98@gmail.com
Achievements
x2
Block or Report

Popular repositories

NL2FOL — Public — SMT ☆ 5

TestRoom — Public — Test Room: Test Management System — Java ☆ 1 ⑂ 1

NASA-Space-Voyage-Linked-Data — Public — KMST Project, Autumn 2018 — Python ☆ 1

CS255-Chat-Client — Public — JavaScript ☆ 1

ITRI-Car-Accident — Public — Python

lovishchopra.github.io — Public — Website of lovishchopra — CSS ⑂ 2

256 contributions in the last year

Oct Nov Dec Jan Feb Mar Apr May Jun Jul Aug Sep
Mon
Wed
Fri
Learn how we count contributions
Less More

2024 2023 2022 2021 2020 2019 2018 2017

Contribution activity

October 1, 2024

lovishchopra has no activity yet for this period.

Google Scholar

Lovish Chopra
Stanford University | IIT Kharagpur
Verified email at stanford.edu - Homepage
Systems ML

FOLLOW

GET MY OWN PROFILE

TITLE	CITED BY	YEAR
Parima: Viewport adaptive 360-degree video streaming L Chopra, S Chakraborty, A Mondal, S Chakraborty Proceedings of the Web Conference 2021, 2379-2391	64	2021
NL2FOL: Translating Natural Language to First-Order Logic for Logical Fallacy Detection A Lalwani, L Chopra, C Hahn, C Trippel, Z Jin, M Sachan arXiv preprint arXiv:2405.02318	1	2024
Scalable Load Balanced Web Cache with Dynamic Consistent Hashing L Chopra, MV Agha-Oko, N Kunjal		
ExplainNLI: Translating Natural Language to First Order Logic for Logical Fallacy Detection A Lalwani, L Chopra		

Cited by

	All	Since 2019
Citations	65	65
h-index	1	1
i10-index	1	1

26 13 0
2021 2022 2023 2024

Lovish Chopra

COMPUTER SCIENCE | SOFTWARE DEVELOPER

About Me

In 2020, I graduated with a B.Tech (Hons.) degree in the Department of Computer Science and Engineering from Indian Institute of Technology (IIT), Kharagpur. Since June 2020, I have been working at D.E.Shaw India Pvt. Ltd., where I have been working at mutiple quantitative and software development projects. The domains of my projects revolve around software development, probability and statistics, machine learning and operating systems. I am particularly interested in exploring systems and networks, specially discovering the way how ML can change the way systems work.

Basic Information

AGE:	23
EMAIL:	lovishchopra98@gmail.com
LANGUAGES KNOWN:	English, Hindi

10:51 PM | gwq-cfun-osj

Where has the author studied and worked -

- Did his Bachelors in Computer Science from IIT Kharagpur
- Graduated with Rank 1 in the CS Dept and Rank 2 in the university
- Worked at D.E. Shaw for two years as a Software Developer on backend development and overlooked security of large SQL databases and on financial regression modeling
- Did his MS in CS from Stanford in 2024 with a Distinction in Research, and his Specialization was Systems.
- He was a 2024 Siebel Scholar, a very prestigious scholarship by The Siebel Foundation to recognize talented graduate students
- Currently working as Software Developer at Granica, an AI research and systems startup helping enterprises leverage AI safely and efficiently.

What previous projects might have led to working on this one -

- Paper was done as part of a course project for the course CS 257: Introduction to Automated Reasoning at Stanford.
- Lovish took this course and worked on the paper with his friend Abhinav who is the co-author of the paper.
- Abhinav had previously worked on a paper in 2022 - Logical Fallacy Detection, from which the current paper is inspired.
- Abhinav's background was in NLP, CS Theory and Logic theory as shared by Lovish.
- Lovish was interested in learning about Formal Logic, about how SAT solvers and SMT solvers are built as they have direct impact in the field of Systems, which was Lovish's specialization at Stanford.
- He took the course CS 243: Program Analysis and Optimizations in which they talked about how SAT and SMT solvers are used for system verification.

CS243: Program Analysis and Optimizations Winter 2024

- After that, he decided to explore how these solvers work in further detail, and thus both Lovish and Abhinav took a common course CS 257: Introduction to Automated Reasoning
- Course went into detail on how SAT and SMT solvers work and covered Advanced Formal Logic and Automated Reasoning. In that course, they had to do a course project.
- Abhinav had previously worked on a paper in 2022 - Logical Fallacy Detection, which inspired the current paper. In that paper, the authors try to fit a sentence in a standard logical fallacy template, and then decide whether the sentence is a fallacy or a valid statement.
- Fallacies can be of different types, so they can't always fit into a given template.
- So it's not a very accurate way of identifying logical fallacies. Can we do better? That is the problem statement they started with for the current paper.
- After the course ended, Lovish and Abhinav brought on more people to help them.
- They both refined the paper and more importantly, earlier they used only open-source LLMs like Llama as they could not afford closed LLMs like ChatGPT nor had any credits given by the University
- After onboarding the new folks, they got access to private LLMs like ChatGPT and did more experiments, although the methodology was the same.

CS 257: Introduction to Automated Reasoning

Logical Fallacy Detection

Zhijing Jin, Abhinav Lalwani, Tejas Vaidhya, Xiaoyu Shen, Yiwen Ding, Zhiheng Lyu, Mrinmaya Sachan, Rada Mihalcea, Bernhard Schölkopf

What motivated the author to work on this project -

- This topic has a lot of social implications. Logical fallacy detection can be used to prevent misinformation, which is a pertinent problem in today's society.
- The authors have shown their method's improved performance on LogicClimate, which consists of logical fallacies related to climate change.
- This was Lovish's motivation for working on this project.
- He felt that if we have a robust way of detecting logical fallacies, it would help prevent misinformation, and their model pushes research in this direction.
- The author decided to focus on this topic and helped decrease misinformation.

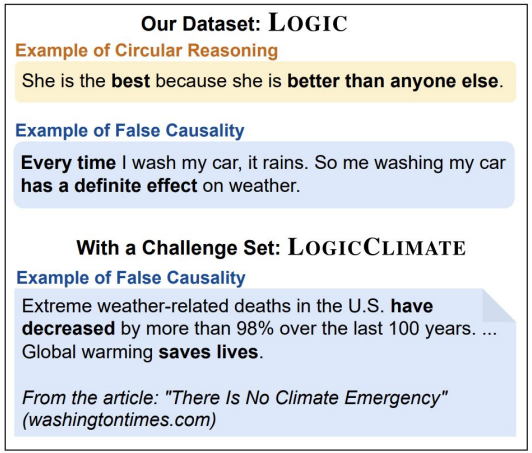
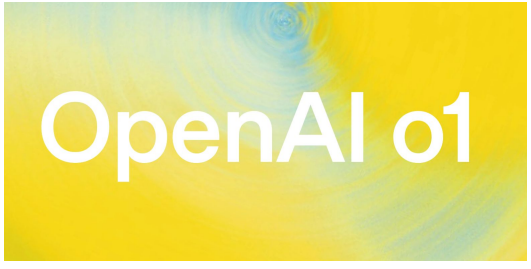


Figure 1: Our dataset consists of general logical fallacies (LOGIC) and an additional test set of logical fallacies in climate claims (LOGICCLIMATE).

What does the author think will be the impact of this paper -

- According to Lovish, if the paper can be made more robust it would be better.
- They worked mostly on open source LLMs. If it can be made more robust, it can help to find logical fallacies in the real world which can help prevent misinformation
- Although, newer models coming up, like OpenAI o1, can perform better inference, but it may not perform very well right now at identifying if a logical formula is a fallacy or not, because even in inference it can mess up sometimes.
- Lovish stated that the good thing about their approach is that it tries to make structure out of unstructured data. He stated that SMT solvers are mostly error-free, so the only question is how robust is their pipeline for converting NL (Natural Language) to FOL (First-Order-Logic), which LLMs help improve a lot.



Fun Facts -

- Lovish is an experienced Ukulele player and likes to play it in his free time
- I also met the Professor with whom Lovish worked and published a paper during his undergrad - Prof. Sandip Chakraborty, during a talk he gave at IIIT Delhi in 2022

Lovish Chopra

FOLLOW

Stanford University | IIT Kharagpur
Verified email at stanford.edu - Homepage
Systems ML

GET MY OWN PROFILE

TITLE	CITED BY	YEAR
Parima: Viewport adaptive 360-degree video streaming L Chopra, S Chakraborty, A Mondal, S Chakraborty Proceedings of the Web Conference 2021, 2379-2391	64	2021
NL2FOL: Translating Natural Language to First-Order Logic for Logical Fallacy Detection A Lalwani, L Chopra, C Hahn, C Trippel, Z Jin, M Sachan arXiv preprint arXiv:2405.02318	1	2024
Scalable Load Balanced Web Cache with Dynamic Consistent Hashing L Chopra, MV Agha-Oko, N Kunjal		
ExplainNLI: Translating Natural Language to First Order Logic for Logical Fallacy Detection A Lalwani, L Chopra		

Cited by

	All	Since 2019
Citations	65	65
h-index	1	1
i10-index	1	1