# Formal Verification for Natural Language Inference: Exploring the First Order Logic Perspective

Student: Sushma Akoju,
Prof. Mihai Surdeanu
Natural Logic Group

Need for formal verification for Natural Language Inference: Exploring First order logic perspective

# Contents

Formal Verification for Natural Language

# Colab notebook

https://colab.research.google.com/drive/1xAM1BjYvsBSTxfMhMywUiUgnO07vYLbg#scrollTo=QbQ-oXDncJxC

# What is Formal Verification?

Formal Verification is the process of proving or disproving the correctness of an algorithm from a formal specification of the algorithm.

We need to formally verify a claim or an inference for the line of reasoning.

conclusion": "A Czech person wrote a book in 1946."

premises:

"**Miroslav** Venhoda was a **Czech** **choral conductor** who specialized in the performance of Renaissance and Baroque music.",

"Any choral conductor is a musician.",

"Some musicians love music.",

"**Miroslav** Venhoda **published** a book in **1946** called Method of Studying Gregorian Chant."

# Formal Verification of Language Inference

1. Current line of research actively explores providing formal proofs by expressing language as first or higher order logic.
2. Translating Natural language to First Order Logics or Higher Order logics, is a well known challenge.
3. The recent results from Encoder - Decoder model for NL to FOL has proven this challenge to be true and real.
4. The most widely used approaches are to use Lambda calculus to extract FOL representations from an existing parse tree representation.
5. such as Abstract Meaning Representation (AMR) to Higher Order Logic

# Turing Test to verify:  ESSLI (European Summer School 2019)

https://naturallogic.pro/Teaching/esslli19/pdf/1-[sl]NPS4NL.pdf

A - Machine

B - Human

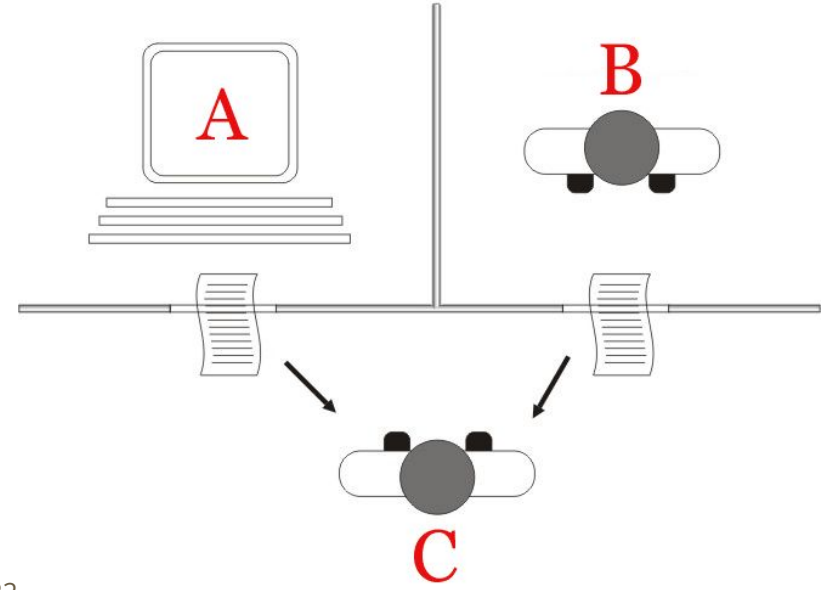
C - Human Interrogator

Exchanging natural language

Conversations between A & B

C - to determine who is Machine &

Who is the human




For normalized environment, the conversations happen over text.

No NLU/Language Model was proven to have passed Turing Test until now (2022).

# Why Formal Logic?

First Order Logic or Higher Order logic

- Are more expressible
- Provide concrete deterministic line of reasoning

Explainable Reasoning:

- Requires white box systems
- Neural Networks are considered black box systems (ESSLI 2019)

# Computational Tree Logic

- Represents path quantifiers
- Uses natural temporal logic

Alternative notations are used for temporal operators.

| | | |
|---|---|---|
| $\Diamond_P$ | $\rightsquigarrow E$ | there Exists a path |
| $\Box_P$ | $\rightsquigarrow A$ | in All paths |
| $\Diamond$ | $\rightsquigarrow F$ | sometime in the Future |
| $\Box$ | $\rightsquigarrow G$ | Globally in the future |
| $\bigcirc$ | $\rightsquigarrow X$ | neXtime |

Ref: http://www.inf.unibz.it/~artale/FM/slide4.pdf

CTL is given by the standard boolean logic enhanced with temporal operators.

"Necessarily Next". $\Box_P \bigcirc \varphi$ is true in $s_t$ iff $\varphi$ is true in every successor state $s_{t+1}$

"Possibly Next". $\Diamond_P \bigcirc \varphi$ is true in $s_t$ iff $\varphi$ is true in one successor state $s_{t+1}$

"Necessarily in the future" (or "Inevitably"). $\Box_P \Diamond \varphi$ is true in $s_t$ iff $\varphi$ is inevitably true in **some** $s_{t'}$ with $t' \geq t$

"Possibly in the future" (or "Possibly"). $\Diamond_P \Diamond \varphi$ is true in $s_t$ iff $\varphi$ may be true in **some** $s_{t'}$ with $t' \geq t$

# Probabilistic Computational Tree Logic

A probability measure on the set of paths with a common prefix of length n

is given by the product of transition probabilities along the prefix of the path

(similar to HMM)

# Markov Logic Neural Networks (MLNs) vs Probabilistic MLN

Logic handles complexity

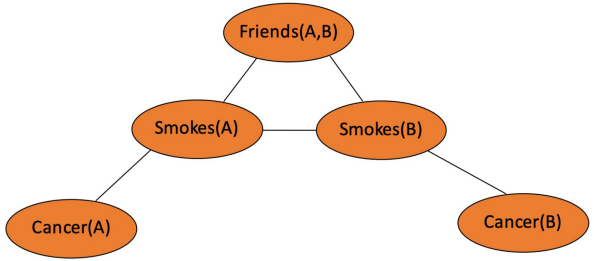
Probability handles uncertainty

Comparision

FOL :

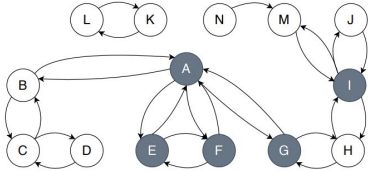
$$\forall x\ Smokes(x) \Rightarrow Cancer(x)$$
$$\forall x, y\ Friends(x, y) \Rightarrow \big(Smokes(x) \Leftrightarrow Smokes(y)\big)$$

MRF:

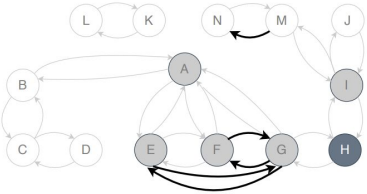



Reference:
https://github.com/sushmaakoju/markov-logic-networks/blob/main/mln-stats-fall-2021-final.pdf

# Neural Markov Logic Neural Networks




(a) The training KB.

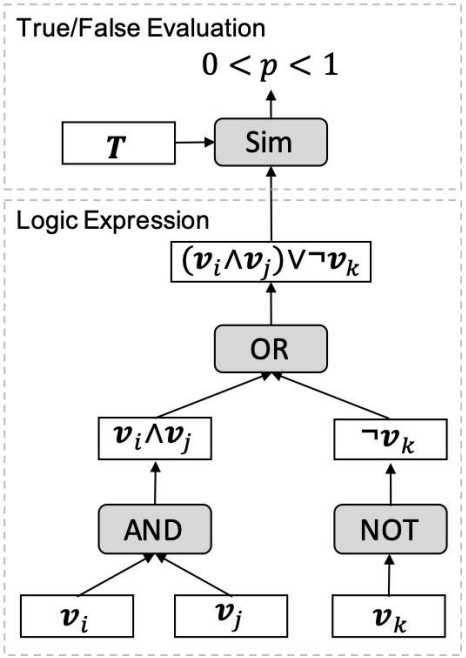



(b) The completed KB.

# Logic-Integrated Neural Network

| | Logical Rule | Equation | Logic Regularizer $r_i$ |
|---|---|---|---|
| NOT | Negation | $\neg T = F$ | $r_1 = \sum_{w \in W \cup \{T\}} Sim(\text{NOT}(\mathbf{w}), \mathbf{w})$ |
| | Double Negation | $\neg(\neg w) = w$ | $r_2 = \sum_{w \in W} 1 - Sim(\text{NOT}(\text{NOT}(\mathbf{w})), \mathbf{w})$ |
| AND | Identity | $w \wedge T = w$ | $r_3 = \sum_{w \in W} 1 - Sim(\text{AND}(\mathbf{w}, \mathbf{T}), \mathbf{w})$ |
| | Annihilator | $w \wedge F = F$ | $r_4 = \sum_{w \in W} 1 - Sim(\text{AND}(\mathbf{w}, \mathbf{F}), \mathbf{F})$ |
| | Idempotence | $w \wedge w = w$ | $r_5 = \sum_{w \in W} 1 - Sim(\text{AND}(\mathbf{w}, \mathbf{w}), \mathbf{w})$ |
| | Complementation | $w \wedge \neg w = F$ | $r_6 = \sum_{w \in W} 1 - Sim(\text{AND}(\mathbf{w}, \text{NOT}(\mathbf{w})), \mathbf{F})$ |
| OR | Identity | $w \vee F = w$ | $r_7 = \sum_{w \in W} 1 - Sim(\text{OR}(\mathbf{w}, \mathbf{F}), \mathbf{w})$ |
| | Annihilator | $w \vee T = T$ | $r_8 = \sum_{w \in W} 1 - Sim(\text{OR}(\mathbf{w}, \mathbf{T}), \mathbf{T})$ |
| | Idempotence | $w \vee w = w$ | $r_9 = \sum_{w \in W} 1 - Sim(\text{OR}(\mathbf{w}, \mathbf{w}), \mathbf{w})$ |
| | Complementation | $w \vee \neg w = T$ | $r_{10} = \sum_{w \in W} 1 - Sim(\text{OR}(\mathbf{w}, \text{NOT}(\mathbf{w})), \mathbf{T})$ |

$$L_1 = L_t + \lambda_l R_l = L_t + \lambda_l \sum_i r_i$$




Reference: https://dl.acm.org/doi/10.1145/3340531.3411949

# Natural Language Theorem Prover

“Natural” Logic Direction:

- Natural Theorem Prover
- Natural Proof
- Natural Lambda Calculus

# Multiple Premises for NLI

Few NLI systems are able to reason over multiple premises.  (ESSLI 2019)

# LangPro

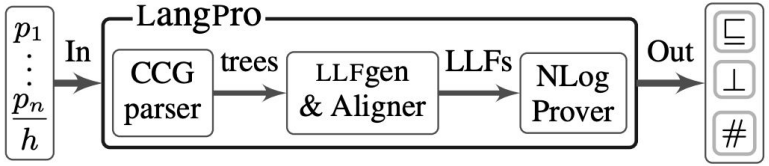
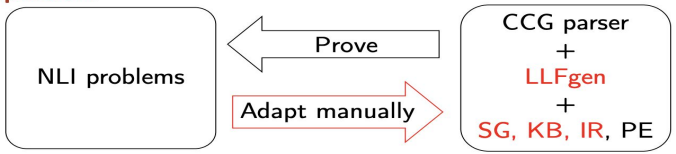



Figure 1: LangPro checks whether a set of premises $p_1, \ldots, p_n$ entails ($\sqsubseteq$), contradicts ($\perp$) or is neutral (#) to a hypothesis $h$.

The prover LangPro is (semi-automatically) trained on the NLI datasets [Abzianidze, 2016a].

- **Adaptation**:

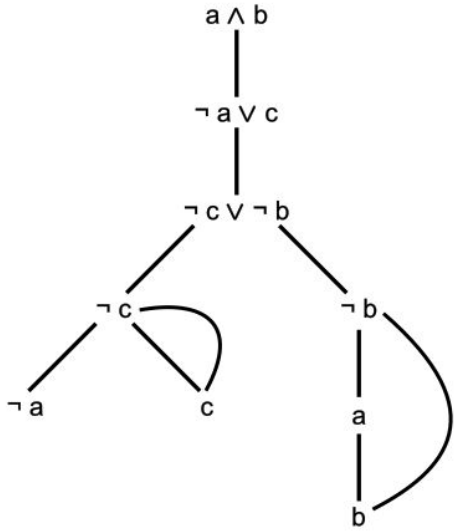



  Used datasets: SICK-trial and FraCaS
- **Development**:
  Finding optimal values for certain parameters of the prover based on its performance on SICK-train.

**NB**: Only C&C parser is used in the learning phase in order to test LangPro for an unseen parser, EasyCCG, later.

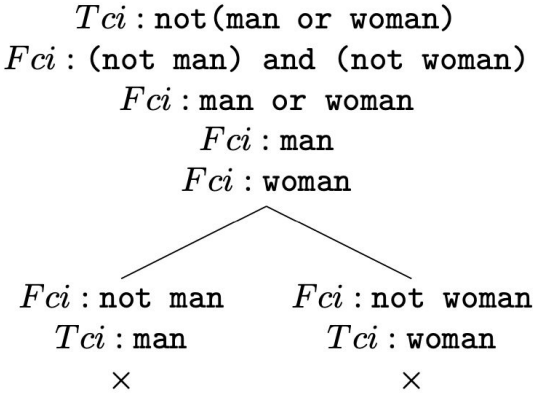
Reference: https://naturallogic.pro/Teaching/esslli19/pdf/5-[sl]NPS4NL.pdf

# LangPro Proofs: Analytic Tableau Method




Reference: https://link.springer.com/content/pdf/10.1007/978-3-642-14287-1_11.pdf

# Lambda Logical Forms (LLFs)(




(1) a. ((a woman)walk)
b. ((if((a woman)walk))((no man)talk))
c. (mary(think((if((a woman)walk))((no man)talk))))
d. ((a woman)(λx(mary(think((if(walk$x$))((no man)talk))))))
e. (few man)λ$x$.(most woman)λ$y$.like$xy$

The terms in (1) were built up in the usual way, but no *logical* constants, such as $=, \forall, \exists, \rightarrow, \wedge, \vee, \neg$ and the like, were used in their composition. The next section will make a connection between some of the non-logical constants used in (1) and logical ones, but this connection will take us from natural representations of linguistic expressions to rather artificial ones. Lambda terms containing no logical constants will therefore continue to have a special status.

Lambda Logical Forms come close to the Logical Forms that are studied in generative grammar. For example, in [13] trees such as the one in (2a) are found, strikingly similar to the $\lambda$-term in (2c).

(2) a. [$_S$[$_{DP}$ every linguist][1[$_S$ John[$_{VP}$ offended $t_1$]]]]
b. ((every linguist)(λ$x_1$(john(offend $x_1$))))

# Summary of tasks for next week

1. Select 10 examples from NLI_XY dataset
2. Chapter 6 (NLI)
    a. Quantifiers
    b. Implications
    c. Negation
3. Test  on the Interfaces:
    - Langpro
    - Nli transformer allennlp
4. Look at numbers, breaking & error patterns
5. Langpro paper discussion

# Cognitive Science Perspective of LMs

https://www.biorxiv.org/content/10.1101/168161v1.full

# Deep Probabilistic Logic

Yet to add...

# Logic Neural Networks (LNNs)

Yet to add...

# Rules vs First Order Logic/Higher Order Logic

Yet to add ..

# Discovering rules from text and new Alignment algorithm

Yet to add ..

# All of the list of literature explored for this direction


1. https://baicsworkshop.github.io/
2. Natural Language Theorem Prover: https://naturallogic.pro/Teaching/esslli19/
3. https://www.biorxiv.org/content/10.1101/168161v1.full
4. https://www.researchgate.net/publication/308317082_Formal_Models_from_Controlled_Natural_Language_via_Cognitive_Grammar_and_Configuration
5. https://ptolemy.berkeley.edu/projects/embedded/research/vis/doc/VisUser/vis_user/node4.html
6. https://engineering.purdue.edu/~givan/papers/aij1.pdf
7. https://people.compute.dtu.dk/ahfrom/Formalized%20First-Order%20Logic.pdf
8. http://demo.clab.cs.cmu.edu/11711fa18/slides/FA18%2011-711%20lecture%2021%20-%20FOPC.pdf
9. https://aclanthology.org/D15-1244.pdf
10. https://naturallogic.pro/Teaching/esslli19/pdf/1-%5Bho%5DNPS4NL.pdf
11. https://arxiv.org/pdf/2203.15099v3.pdf
12. http://phonetics.linguistics.ucla.edu/wpl/issues/wpl17/papers/31_moss.pdf
13. https://www.cse.iitk.ac.in/users/karkare/MTP/2014-15/naman2015logica.pdf
14. https://www.amazon.science/blog/a-gentle-introduction-to-automated-reasoning
14. http://proceedings.mlr.press/v80/xu18h/xu18h.pdf
15. https://dl.acm.org/doi/10.1145/3340531.3411949
16. https://arxiv.org/pdf/2008.09514.pdf
17. https://github.com/rutgerswiselab/NLR
18. https://en.wikipedia.org/wiki/Probabilistic_CTL
19. https://proceedings.mlr.press/v97/fischer19a.html
20. https://iep.utm.edu/duality-in-logic-and-language/
21. https://encyclopediaofmath.org/index.php?title=Duality_principle