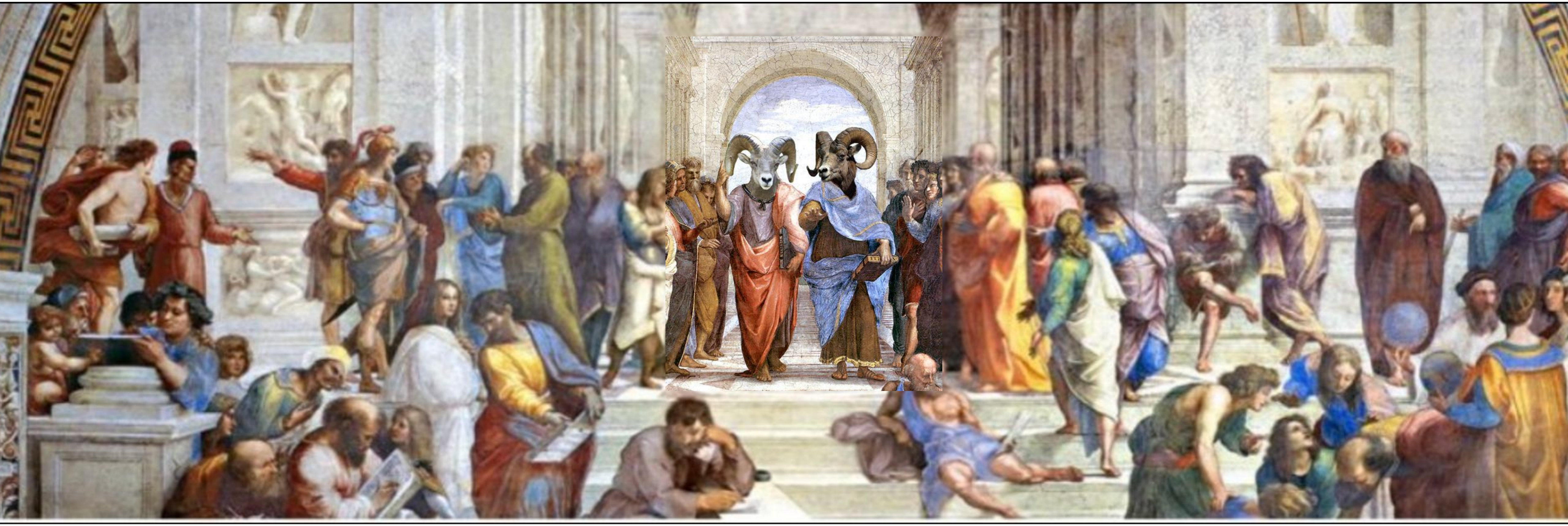
Learning to Self-Improve & Reason with LLMs

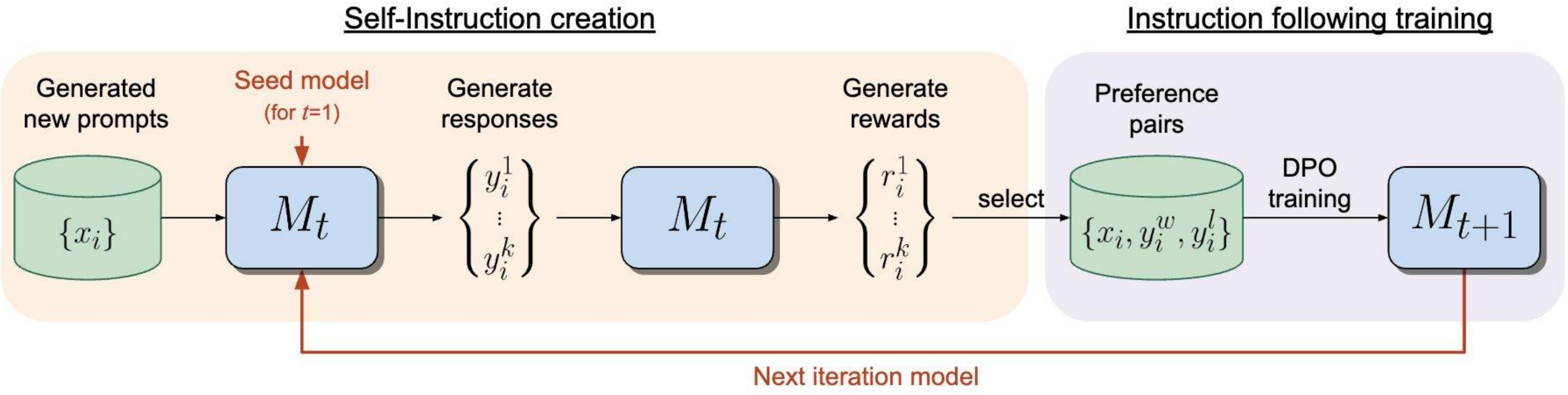
Jason Weston
Meta & NYU



Goal: An AI that "trains" itself as much as possible

- Creates new tasks to train on (challenges itself)
- Evaluates whether it gets them right ("self-rewarding")
- Updates itself based on what it understood

Research question: *can this help it become superhuman?*



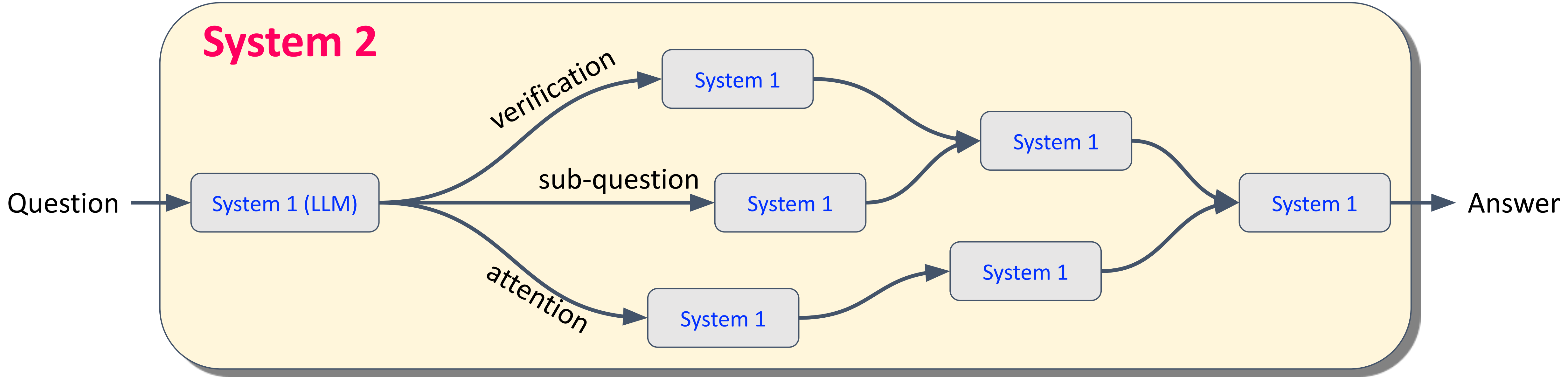
When self-improving: two types of reasoning to improve

System 1: reactive and relies on associations

LLMs can be viewed as System 1
- Fixed compute per token
- Directly outputs answer
- Failures: learns spurious/unwanted correlations: *hallucination, sycophancy, jailbreaking*, ..

System 2: more deliberate and effortful

Multiple "calls" to System 1 LLM
- *Planning, search, verifying, reasoning etc.*
- Dynamic computation (e.g. chain-of-thought, ToT, ..)

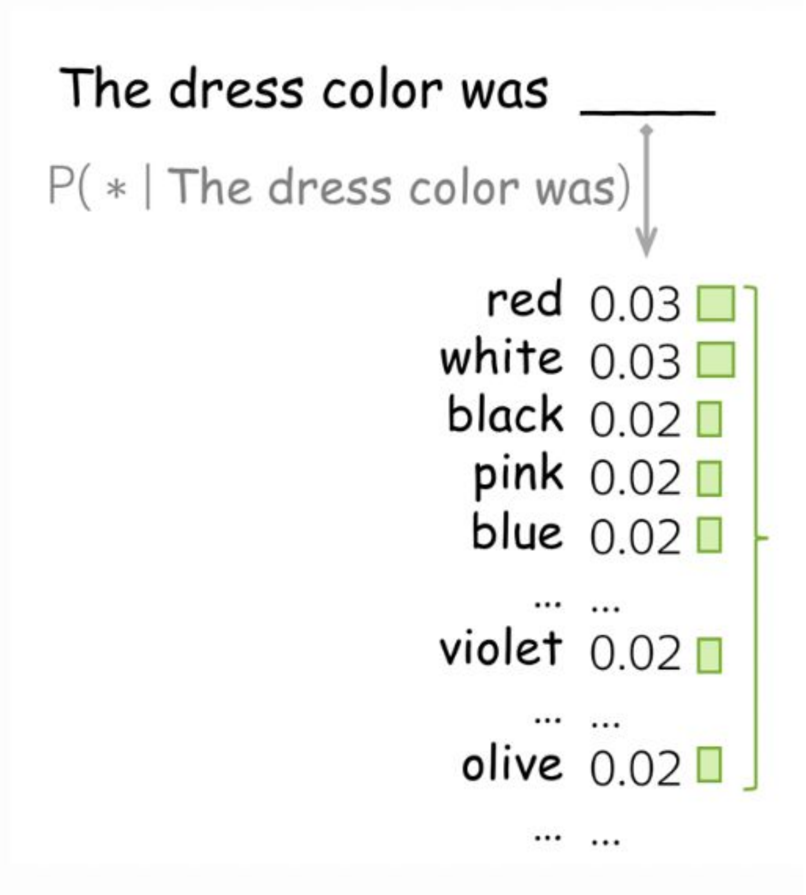


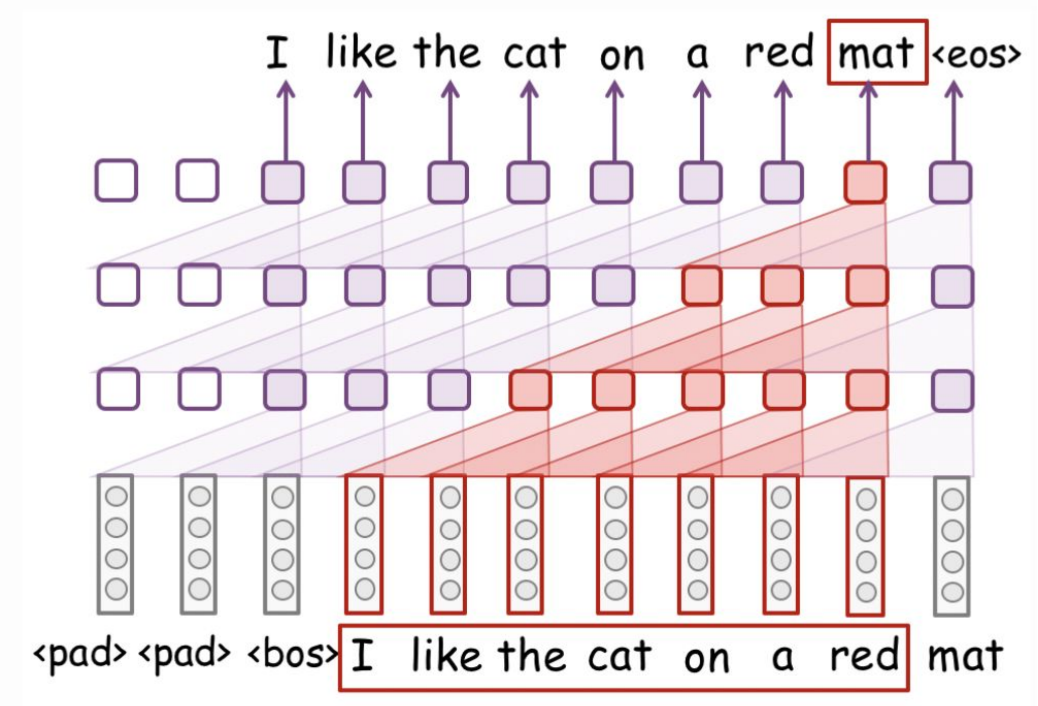
First, some pre-history

(Pre-2020..)

Language modeling

Standard (pre-training) trains by predicting the next token only on "positive examples" of language





Images from https://lena-voita.github.io/nlp_course/language_modeling.html

2003

A Neural Probabilistic Language Model

Yoshua Bengio,* Réjean Ducharme and Pascal Vincent
Département d'Informatique et Recherche Opérationnelle
Centre de Recherche Mathématiques
Université de Montréal
Montréal, Québec, Canada, H3C 3J7
{*bengioy,ducharme,vincentp*}@*iro.umontreal.ca*

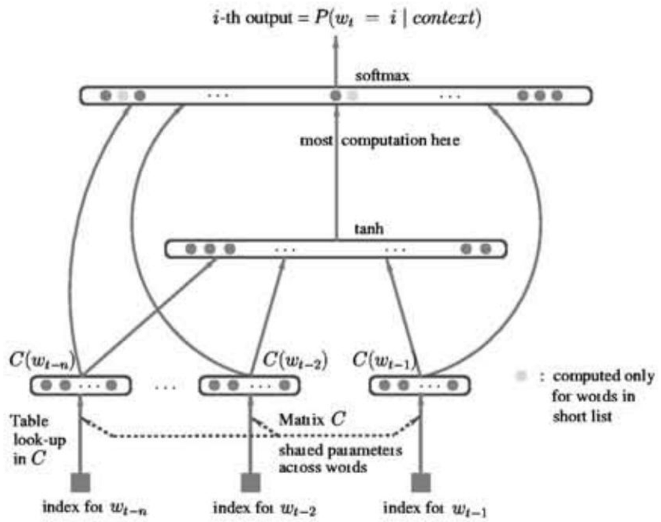


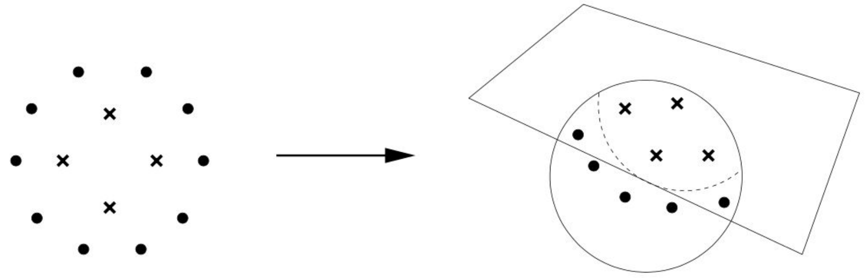
Figure 1: "Direct Architecture": $f(i, w_{t-1}, \cdots, w_{t-n}) = g(i, C(w_{t-1}), \cdots, C(w_{t-n}))$ where g is the neural network and $C(i)$ is the i-th word feature vector.

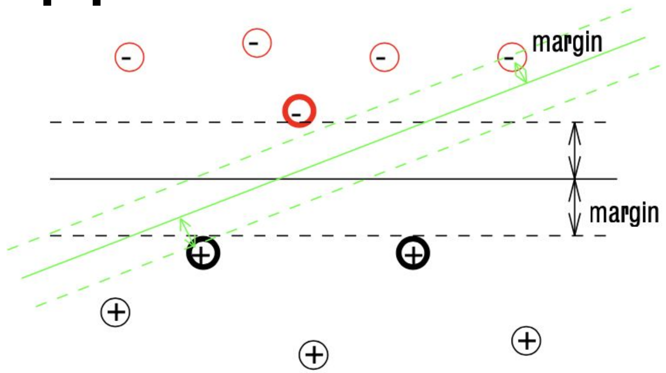
Conclusions and Proposed Extensions

There is probably much more to be done to improve the model, at the level of architecture, computational efficiency, and taking advantage of prior knowledge. An important priority of future research should be to evaluate and improve the speeding-up tricks proposed here, and find ways to increase capacity without increasing training time too much (to deal with corpora with hundreds of millions of words). A simple idea to take advantage of temporal

Most of the 2000s

Support Vector Machines





Support Vector Machines for Multi-Class Pattern Recognition

J. Weston and C. Watkins

Department of Computer Science
Royal Holloway, University of London

3. k-Class Support Vector Machines

A more natural way to solve k-class problems is to construct a piecewise linear separation of the k classes in a single optimisation.

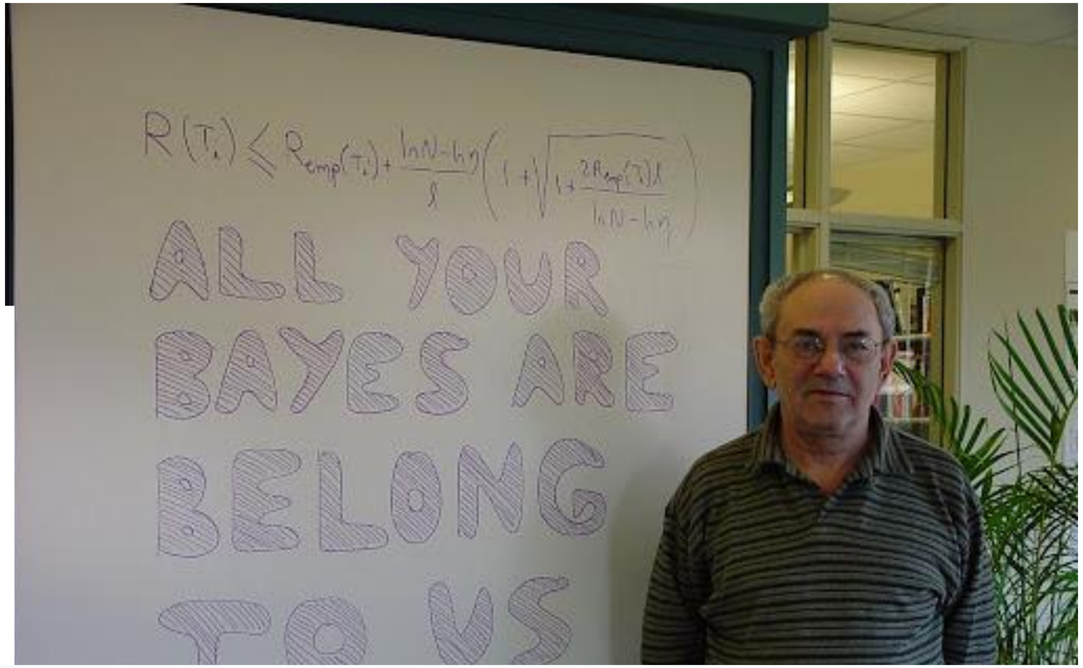
The binary SVM optimisation problem [5] is generalised to the following: minimise

$$\phi(w, \xi) = \frac{1}{2} \sum_{m=1}^{k} (w_m \cdot w_m) + C \sum_{i=1}^{\ell} \sum_{m \neq y_i} \xi_i^m \qquad (2)$$

subject to

$$(w_{y_i} \cdot x_i) + b_{y_i} \geq (w_m \cdot x_i) + b_m + 2 - \xi_i^m,$$

$$\xi_i^m \geq 0, \quad i = 1, \ldots, \ell \quad m \in \{1, \ldots, k\} \setminus y_i.$$

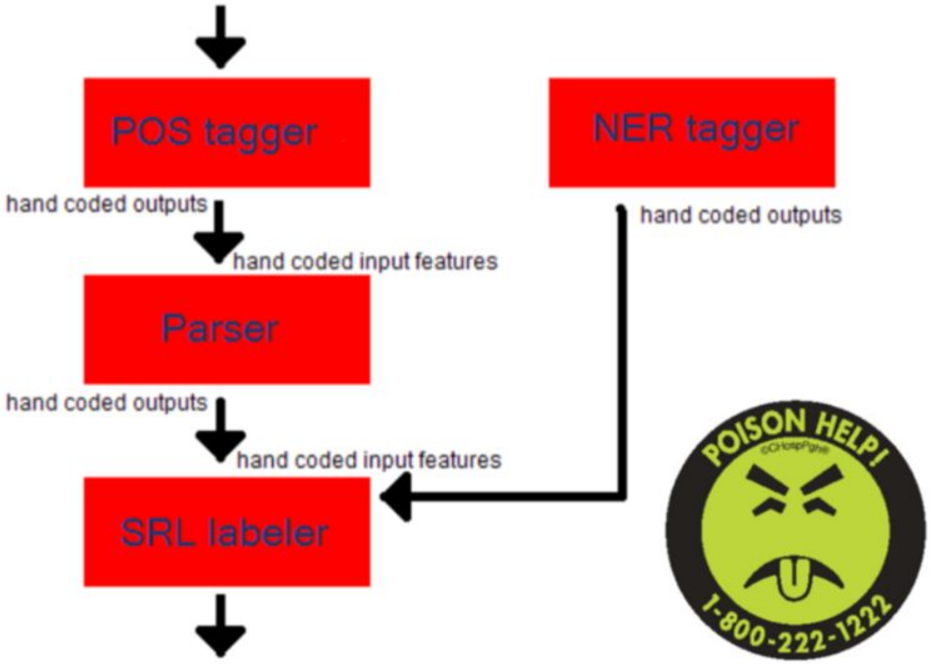


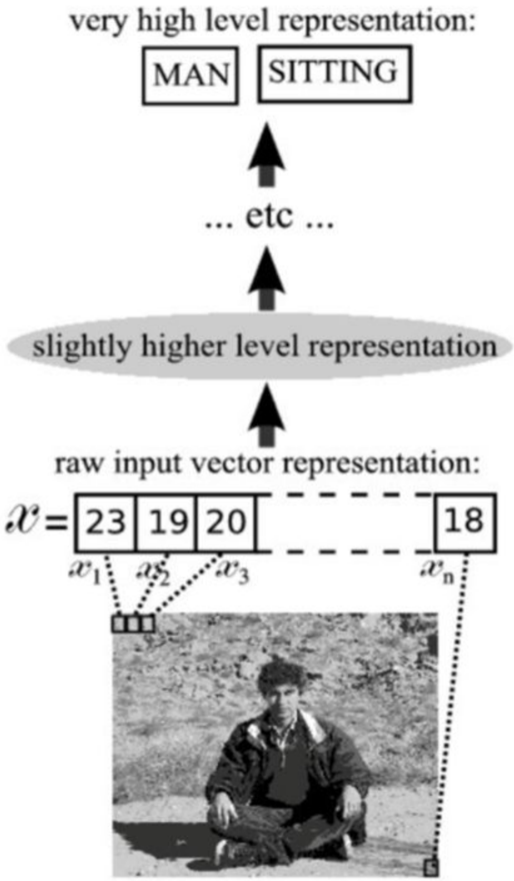
According to a story shared in Wired, Vapnik had been critical of LeCun's neural nets. The two were involved in a wager placed in 1995, betting expensive dinners on the future of deep artificial neural nets. Vapnik believed deep artificial neural nets would a mystery by the year 2000 and that no one would be using LeCun's neural nets by 2005. Vapnik won the first bet but lost the second.

Shallow vs Deep

2008

- Brittle – system must use chosen input representation/ output labels
- Robust – priors injected into system, but it can ignore them/adapt
  E.g: multi-tasking, choice of architecture . . .





Unlike SVMs, Neural nets can manipulate words + end-to-end

ICML 2008

A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning

Ronan Collobert COLLOBER@NEC-LABS.COM
Jason Weston JASONW@NEC-LABS.COM
NEC Labs America, 4 Independence Way, Princeton, NJ 08540 USA

Abstract

We describe a single convolutional neural network architecture that, given a sentence, outputs a host of language processing predictions: part-of-speech tags, chunks, named entity tags, semantic roles, semantically similar words and the likelihood that the sentence makes sense (grammatically and semantically) using a language model. The entire network is trained *jointly* on all these tasks using weight-sharing, an instance of *multitask learning*. All the tasks use labeled data except the language model which is learnt from unlabeled text and represents a novel form of *semi-supervised learning* for the shared tasks. We show how both *multitask learning* and *semi-supervised learning* improve the generalization of the shared tasks, resulting in state-of-the-art performance.

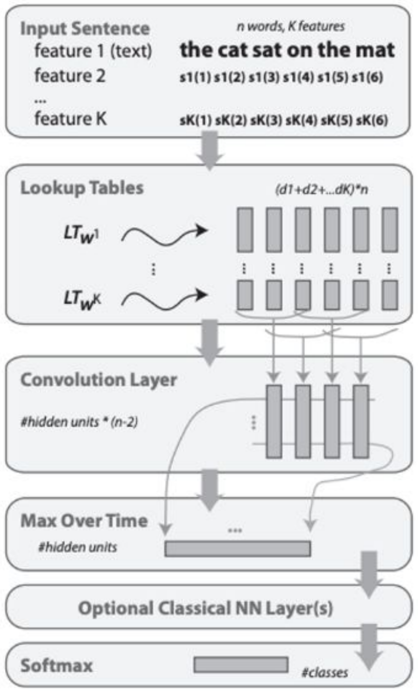


Figure 1. A general deep NN architecture for NL[…] an input sentence, the NN outputs class probabi[…] *one* chosen word. A classical window approach is […] case where the input has a fixed size ksz, and th[…] kernel size is ksz; in that case the TDNN layer […] only one vector and the Max layer performs an identity.

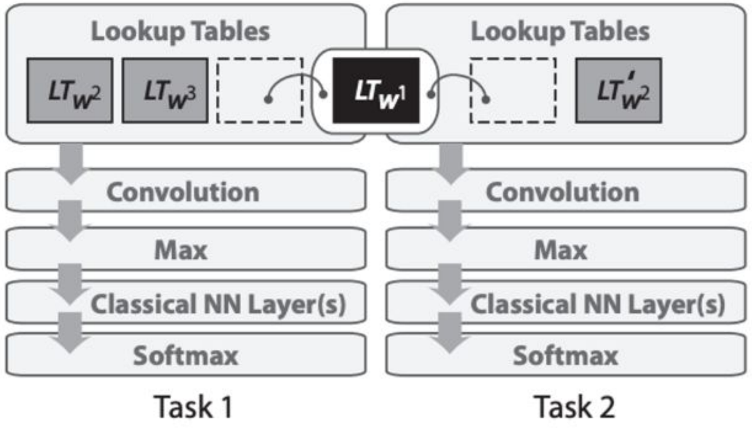


Table 1. Language model performance for learning an embedding in $wsz = 50$ dimensions (dictionary size: 30,000). For each column the queried word is followed by its index in the dictionary (higher means more rare) and its 10 nearest neighbors (arbitrary using the Euclidean metric).

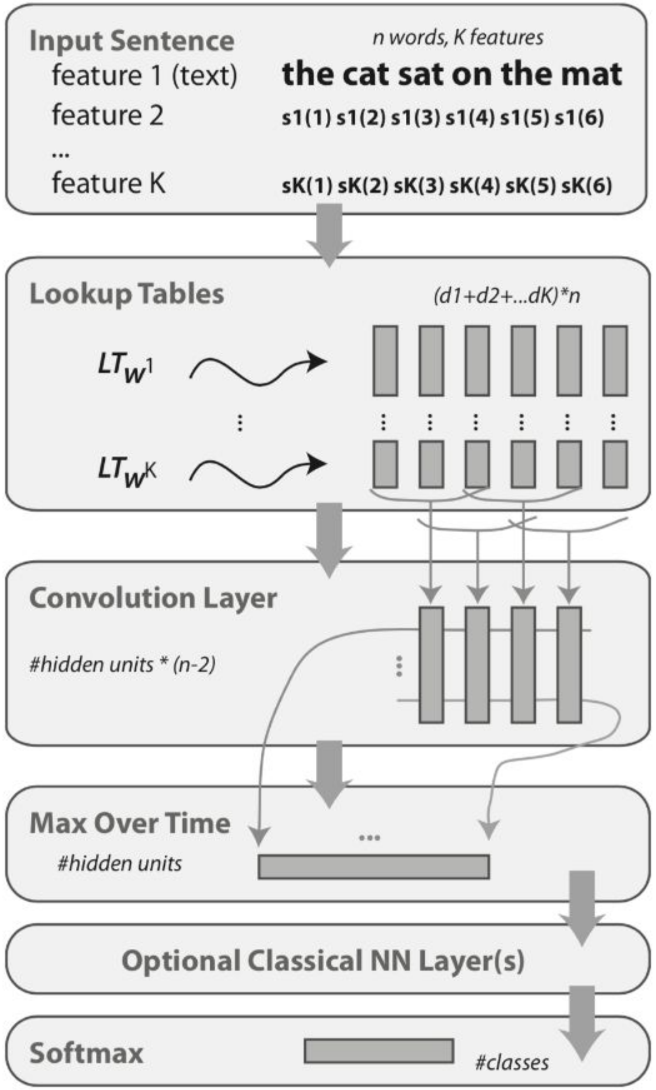
FRANCE 454	JESUS 1973	XBOX 6909	REDDISH 11724	SCRATCHED 29869
SPAIN	CHRIST	PLAYSTATION	YELLOWISH	SMASHED
ITALY	GOD	DREAMCAST	GREENISH	RIPPED
RUSSIA	RESURRECTION	PSNUMBER	BROWNISH	BRUSHED
POLAND	PRAYER	SNES	BLUISH	HURLED
ENGLAND	YAHWEH	WII	CREAMY	GRABBED
DENMARK	JOSEPHUS	NES	WHITISH	TOSSED
GERMANY	MOSES	NINTENDO	BLACKISH	SQUEEZED
PORTUGAL	SIN	GAMECUBE	SILVERY	BLASTED
SWEDEN	HEAVEN	PSP	GREYISH	TANGLED
AUSTRIA	SALVATION	AMIGA	PALER	SLASHED

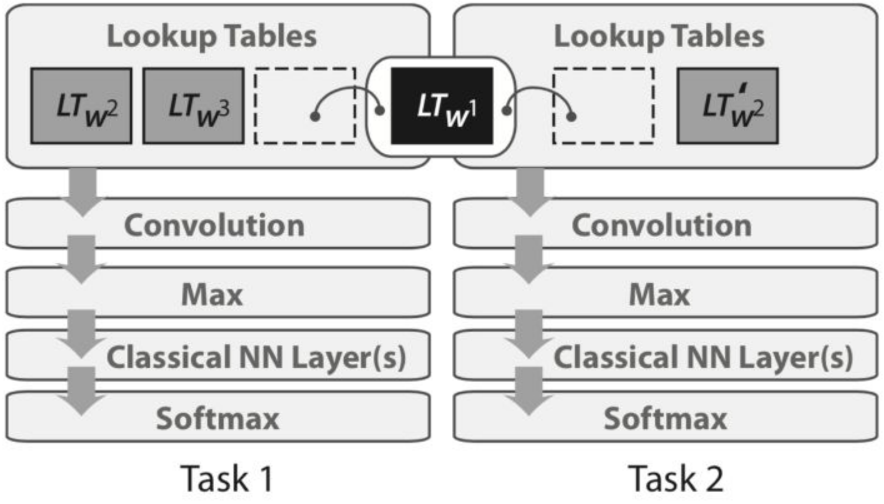
What was in this paper?

Word Embeddings

CNNs

Attention





Multi-Task Learning

POS
NER
Chunking
SRL

Pre-training on Wikipedia-LM

Predict middle word in window

The Stanford Natural Language Processing Group

NLP Reading Group Schedule — Spring 2009
Tuesdays at 12:0-1:30pm, Gates 200

Date	Moderator	Topic / Paper
March 31	Jenny Finkel	Organizational Meeting
April 7	[redacted]	**Bullshit ICML Paper** [pdf]
April 14	Adam Vogel	Wei Lu, Hwee Tou Ng, Wee Sun Lee, Luke S. Zettlemoyer. **A Generative Model for Parsing Natural Language to Meaning Representations** EMNLP 2008. [pdf]
April 21	Nate Chambers	**Finding and Linking Incidents in the News** [pdf]
April 28	--	No Meeting - Good Luck with EMNLP Submissions!
May 5	Jenny Finkel	**The Unreasonable Effectiveness of Data** [pdf]
May 12	David Vickrey	S. Cohen and N. Smith. **Shared Logistic Normal Distributions for Soft Parameter Tying in Unsupervised Grammar Induction** NAACL 2009 [pdf]
May 19	Ramesh Nallapati	David Mimno, Hanna Wallach and Andrew McCallum. **Gibbs Sampling for Logistic Normal Topic Models with Graph-Based Priors.** NIPS Workshop on Analyzing Graphs, 2008, Whistler, BC. [pdf]
May 26	Steven Bethard	Amarnag Subramanya, Jeff Bilmes. **Soft-Supervised Learning for Text Classification** EMNLP 2008 [pdf]
June 2	Mihai Surdeanu	Yi Zhang, Jeff Schneider, Artur Dubrawski. **Learning the Semantic Correlation: An Alternative Way to Gain from Unlabeled Text** NIPS 2008 [pdf]
June 9	Jenny Finkel	**Syntactic Topic Models** NIPS 2008 [pdf]

INTERNET ARCHIVE WaybackMachine

http://nlp.stanford.edu:80/read/ Go

FEB APR SEP

105 captures
24 May 2005 – 25 Oct 2017

S NLP Stanford University Natural Language Processing

The Stanford Natural Langua

home · people · teaching · research · publication

NLP Reading Group Schedule — Spring 2009
Tuesdays at 12:0-1:30pm, Gates 200

Date	Moderator	Topic / Paper
March 31	Jenny Finkel	Organizational Meeting
April 7	[redacted]	**Bullshit ICML Paper** [pdf]
April 14	Adam Vogel	Wei Lu, Hwee Tou Ng, Wee Su **Meaning Representations** EM
April 21	Nate Chambers	**Finding and Linking Incident**
April 28	--	No Meeting - Good Luck with E
May 5	Jenny Finkel	**The Unreasonable Effectiven**
May 12	David Vickrey	S. Cohen and N. Smith. **Share Induction** NAACL 2009 [pdf]
May 19	Ramesh Nallapati	David Mimno, Hanna Wallach **Based Priors.** NIPS Worksho
May 26	Steven Bethard	Amarnag Subramanya, Jeff Bil
June 2	Mihai Surdeanu	Yi Zhang, Jeff Schneider, Artur **Unlabeled Text** NIPS 2008 [pd
June 9	Jenny Finkel	**Syntactic Topic Models** NIPS

https://web.archive.org/web/20090428010657/http://ronan.collobert.cor

Facebook researchers win Test of Time Award at ICML 2018

July 13, 2018

By: Facebook Research

We are pleased to announce that Facebook research scientists Ronan Collobert and Jason Weston won the 2018 International Conference on Machine Learning (ICML) "Test of Time Award" for their paper, "A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning," which they presented at ICML 2008 when both were working for NEC Labs. It took several years for the ideas presented in the paper to become mainstream, as it was quite different to work in the NLP community at the time.

INTERNET ARCHIVE
WaybackMachine

http://nlp.stanford.edu:80/read/

105 captures
24 May 2005 – 25 Oct 2017

The Stanford Natural Lang

home · people · teaching · research · publicati

NLP Reading Group Schedule — Spring 2009
Tuesdays at 12:0-1:30pm, Gates 200

Date	Moderator	Topic / Paper
March 31	Jenny Finkel	Organizational Meeting
April 7	[redacted]	**Bullshit ICML Paper** [pdf]
April 14	Adam Vogel	Wei Lu, Hwee Tou Ng, Wee **Meaning Representations**
April 21	Nate Chambers	**Finding and Linking Incide**
April 28	--	No Meeting - Good Luck wit
May 5	Jenny Finkel	**The Unreasonable Effectiv**
May 12	David Vickrey	S. Cohen and N. Smith. **Sha Induction** NAACL 2009 [pdf
May 19	Ramesh Nallapati	David Mimno, Hanna Wallac **Based Priors.** NIPS Worksh
May 26	Steven Bethard	Amarnag Subramanya, Jeff
June 2	Mihai Surdeanu	Yi Zhang, Jeff Schneider, Ar **Unlabeled Text** NIPS 2008
June 9	Jenny Finkel	**Syntactic Topic Models** NI

https://web.archive.org/web/20090428010657/http://ronan.collobert.c

Stanford University

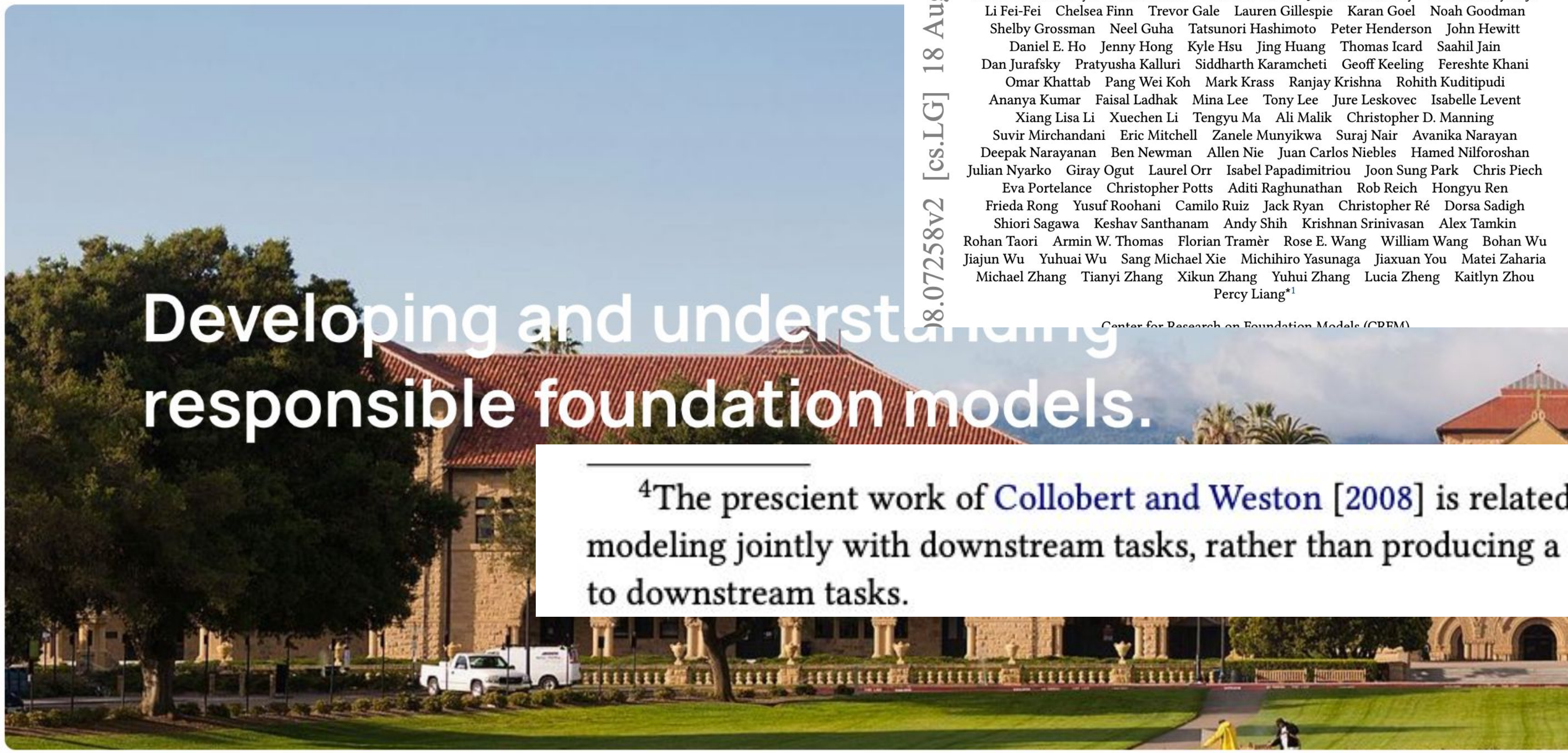
Center for Research on Foundation Models

HAI Stanford University Human-Centered Artificial Intelligence

On the Opportunities and Risks of Foundation Models

Rishi Bommasani* Drew A. Hudson Ehsan Adeli Russ Altman Simran Arora Sydney von Arx Michael S. Bernstein Jeannette Bohg Antoine Bosselut Emma Brunskill Erik Brynjolfsson Shyamal Buch Dallas Card Rodrigo Castellon Niladri Chatterji Annie Chen Kathleen Creel Jared Quincy Davis Dorottya Demszky Chris Donahue Moussa Doumbouya Esin Durmus Stefano Ermon John Etchemendy Kawin Ethayarajh Li Fei-Fei Chelsea Finn Trevor Gale Lauren Gillespie Karan Goel Noah Goodman Shelby Grossman Neel Guha Tatsunori Hashimoto Peter Henderson John Hewitt Daniel E. Ho Jenny Hong Kyle Hsu Jing Huang Thomas Icard Saahil Jain Dan Jurafsky Pratyusha Kalluri Siddharth Karamcheti Geoff Keeling Fereshte Khani Omar Khattab Pang Wei Koh Mark Krass Ranjay Krishna Rohith Kuditipudi Ananya Kumar Faisal Ladhak Mina Lee Tony Lee Jure Leskovec Isabelle Levent Xiang Lisa Li Xuechen Li Tengyu Ma Ali Malik Christopher D. Manning Suvir Mirchandani Eric Mitchell Zanele Munyikwa Suraj Nair Avanika Narayan Deepak Narayanan Ben Newman Allen Nie Juan Carlos Niebles Hamed Nilforoshan Julian Nyarko Giray Ogut Laurel Orr Isabel Papadimitriou Joon Sung Park Chris Piech Eva Portelance Christopher Potts Aditi Raghunathan Rob Reich Hongyu Ren Frieda Rong Yusuf Roohani Camilo Ruiz Jack Ryan Christopher Ré Dorsa Sadigh Shiori Sagawa Keshav Santhanam Andy Shih Krishnan Srinivasan Alex Tamkin Rohan Taori Armin W. Thomas Florian Tramèr Rose E. Wang William Wang Bohan Wu Jiajun Wu Yuhuai Wu Sang Michael Xie Michihiro Yasunaga Jiaxuan You Matei Zaharia Michael Zhang Tianyi Zhang Xikun Zhang Yuhui Zhang Lucia Zheng Kaitlyn Zhou Percy Liang*[1]

8.07258v2 [cs.LG] 18 Aug 2021



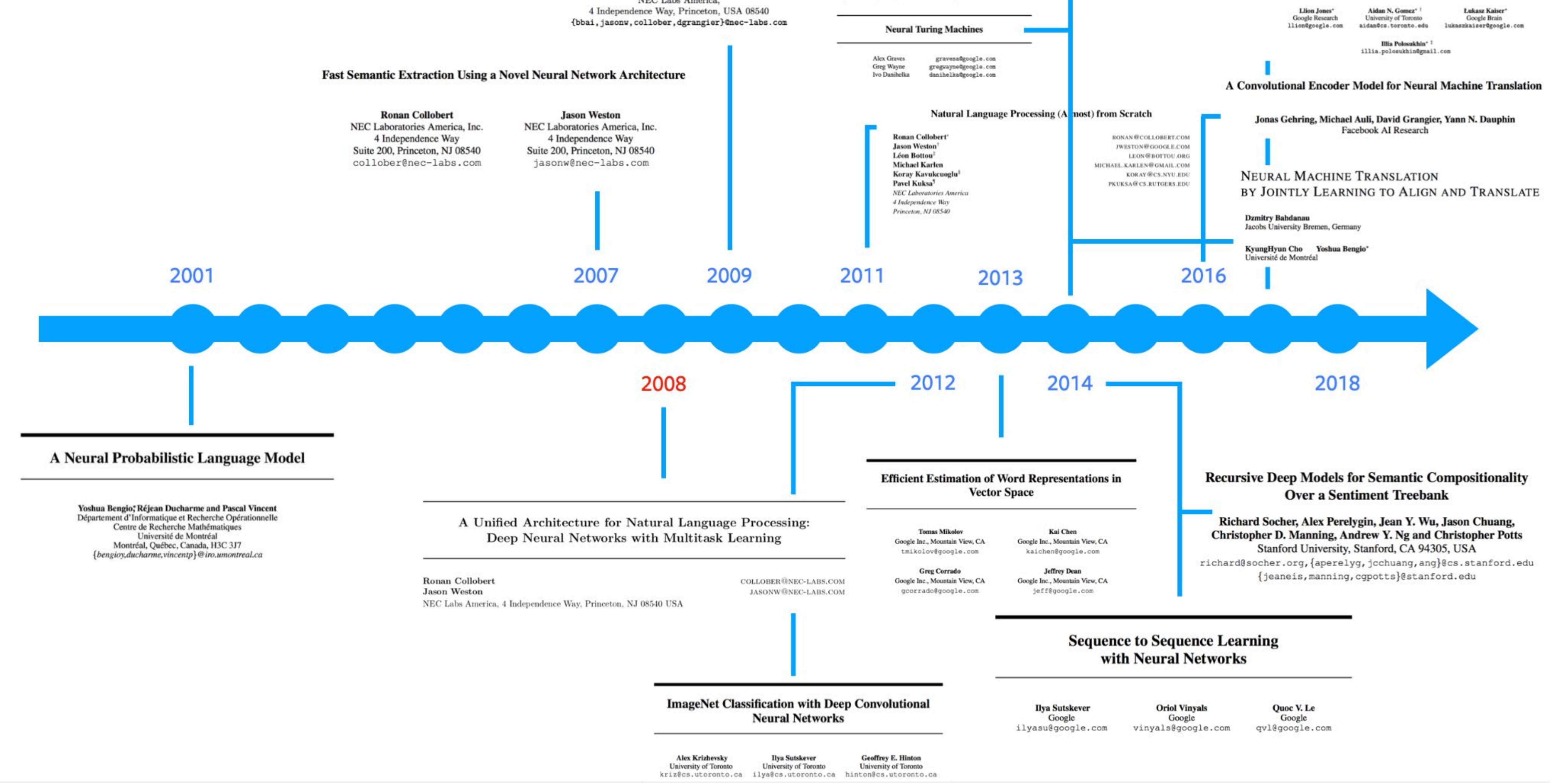
[4]The prescient work of Collobert and Weston [2008] is related modeling jointly with downstream tasks, rather than producing a to downstream tasks.

News: Our report on foundation models is now publicly available!

Our Mission

The Center for Research on Foundation Models (CRFM) is a interdisciplinary initiative born out of the Stanford Institute

— WHAT IS A FOUNDATION MODEL?


NEC Labs America,
4 Independence Way, Princeton, USA 08540
{bbai,jasonw,collober,dgrangier}@nec-labs.com
Fast Semantic Extraction Using a Novel Neural Network Architecture
Ronan Collobert
NEC Laboratories America, Inc.
4 Independence Way
Suite 200, Princeton, NJ 08540
collober@nec-labs.com
Jason Weston
NEC Laboratories America, Inc.
4 Independence Way
Suite 200, Princeton, NJ 08540
jasonw@nec-labs.com
Neural Turing Machines
Alex Graves gravesa@google.com
Greg Wayne gregwayne@google.com
Ivo Danihelka danihelka@google.com
Natural Language Processing (Almost) from Scratch
Ronan Collobert
Jason Weston
Léon Bottou
Michael Karlen
Koray Kavukcuoglu
Pavel Kuksa
NEC Laboratories America
4 Independence Way
Princeton, NJ 08540
RONAN@COLLOBERT.COM
JWESTON@GOOGLE.COM
LEON@BOTTOU.ORG
MICHAEL.KARLEN@GMAIL.COM
KORAY@CS.NYU.EDU
PKUKSA@CS.RUTGERS.EDU
Llion Jones
Google Research
llion@google.com
Aidan N. Gomez
University of Toronto
aidan@cs.toronto.edu
Łukasz Kaiser
Google Brain
lukaszkaiser@google.com
Illia Polosukhin
illia.polosukhin@gmail.com
A Convolutional Encoder Model for Neural Machine Translation
Jonas Gehring, Michael Auli, David Grangier, Yann N. Dauphin
Facebook AI Research
NEURAL MACHINE TRANSLATION
BY JOINTLY LEARNING TO ALIGN AND TRANSLATE
Dzmitry Bahdanau
Jacobs University Bremen, Germany
KyungHyun Cho Yoshua Bengio
Université de Montréal
2001
2007
2009
2011
2013
2016
2008
2012
2014
2018
A Neural Probabilistic Language Model
Yoshua Bengio, Réjean Ducharme and Pascal Vincent
Département d'Informatique et Recherche Opérationnelle
Centre de Recherche Mathématiques
Université de Montréal
Montréal, Québec, Canada, H3C 3J7
{bengioy,ducharme,vincentp}@iro.umontreal.ca
A Unified Architecture for Natural Language Processing:
Deep Neural Networks with Multitask Learning
Ronan Collobert COLLOBER@NEC-LABS.COM
Jason Weston JASONW@NEC-LABS.COM
NEC Labs America, 4 Independence Way, Princeton, NJ 08540 USA
Efficient Estimation of Word Representations in
Vector Space
Tomas Mikolov
Google Inc., Mountain View, CA
tmikolov@google.com
Kai Chen
Google Inc., Mountain View, CA
kaichen@google.com
Greg Corrado
Google Inc., Mountain View, CA
gcorrado@google.com
Jeffrey Dean
Google Inc., Mountain View, CA
jeff@google.com
Recursive Deep Models for Semantic Compositionality
Over a Sentiment Treebank
Richard Socher, Alex Perelygin, Jean Y. Wu, Jason Chuang,
Christopher D. Manning, Andrew Y. Ng and Christopher Potts
Stanford University, Stanford, CA 94305, USA
richard@socher.org,{aperelyg,jcchuang,ang}@cs.stanford.edu
{jeaneis,manning,cgpotts}@stanford.edu
Sequence to Sequence Learning
with Neural Networks
Ilya Sutskever
Google
ilyasu@google.com
Oriol Vinyals
Google
vinyals@google.com
Quoc V. Le
Google
qvl@google.com
ImageNet Classification with Deep Convolutional
Neural Networks
Alex Krizhevsky
University of Toronto
kriz@cs.utoronto.ca
Ilya Sutskever
University of Toronto
ilya@cs.utoronto.ca
Geoffrey E. Hinton
University of Toronto
hinton@cs.utoronto.ca

ACL 2008
***0** "deep" or "neural" papers*

Recent Trends in Deep Learning Based Natural Language Processing

Tom Young[†≡], Devamanyu Hazarika[‡≡], Soujanya Poria[⊕≡], Erik Cambria[▽*]

[†] School of Information and Electronics, Beijing Institute of Technology, China
[‡] School of Computing, National University of Singapore, Singapore
[⊕] Temasek Laboratories, Nanyang Technological University, Singapore
[▽] School of Computer Science and Engineering, Nanyang Technological University, Singapore

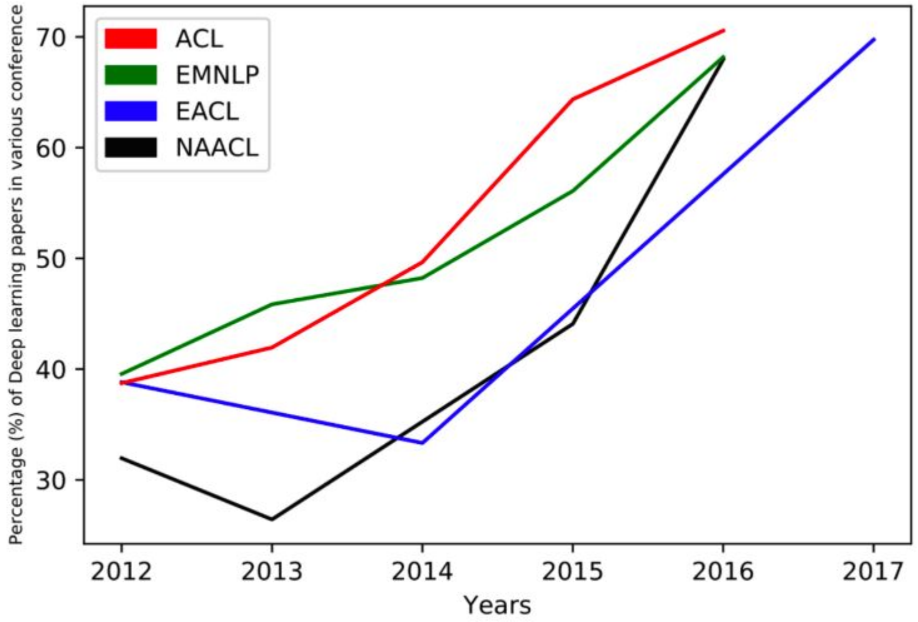


Fig. 1: Percentage of deep learning papers in ACL, EMNLP, EACL, NAACL over the last 6 years (long papers).

ICML 2008
***4** "deep" or "neural" papers*

A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning.
Ronan Collobert and Jason Weston

Deep Learning via Semi-Supervised Embedding.
Jason Weston, Frédéric Ratle and Ronan Collobert

Semi-supervised Learning of Compact Document Representations with Deep Networks.
Marc'Aurelio Ranzato and Martin Szummer

On the Quantitative Analysis of Deep Belief Networks.
Ruslan Salakhutdinov and Iain Murray

ICML 2018
*> **100** "deep" or "neural" papers*

a probabilistic framework for multi-view feature learning with many-to-many associations via neural networks
a semantic loss function for deep learning with symbolic knowledge
a spline theory of deep learning
accurate uncertainties for deep learning using calibrated regression
an optimal control approach to deep learning and applications to discrete-weight neural networks
analyzing uncertainty in neural machine translation
attention-based deep multiple instance learning
autoregressive convolutional neural networks for asynchronous time series
bayesian uncertainty estimation for batch normalized deep networks
beyond finite layer neural networks: bridging deep architectures and numerical differential equations
born again neural networks
bounding and counting linear regions of deep neural networks
bounds on the approximation power of feedforward neural networks
can deep reinforcement learning solve erdos-selfridge-spencer games?
comparing dynamics: deep neural networks versus glassy systems
compressing neural networks using the variational information bottelneck
conditional neural processes
constraining the dynamics of deep probabilistic models
contextnet: deep learning for star galaxy classification
contextual graph markov model: a deep and generative approach to graph processing
curriculum learning by transfer learning: theory and experiments with deep networks
dcfnet: deep neural network with decomposed convolutional filters
decomposition of uncertainty in bayesian deep learning for efficient and risk-sensitive learning
deep asymmetric multi-task feature learning
deep bayesian nonparametric tracking
deep density destructors
deep k-means: re-training and parameter sharing with harder cluster assignments for compressing deep convolutions
deep linear networks with arbitrary loss: all local minima are global
deep models of interactions across sets
deep one-class classification
deep predictive coding network for object recognition
deep reinforcement learning in continuous action spaces: a case study in the game of simulated curling
deep variational reinforcement learning for pomdps
differentiable abstract interpretation for provably robust neural networks
differentiable plasticity: training plastic neural networks with backpropagation
discovering interpretable representations for both deep generative and discriminative models
distilling the posterior in bayesian neural networks
dynamic evaluation of neural sequence models
dynamical isometry and a mean field theory of cnns: how to train 10,000-layer vanilla convolutional neural networks
dynamical isometry and a mean field theory of rnns: gating enables signal propagation in recurrent neural networks

20 bAbI tasks

Towards AI-Complete Question Answering: A Set of Prerequisite Toy Tasks

Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M. Rush, Bart van Merriënboer, Armand Joulin & Tomas Mikolov
Facebook AI Research
770 Broadway
New York, USA
{jase,abordes,spchopra,tmikolov,sashar,bartvm}@fb.com

Task 1: Single Supporting Fact
Mary went to the bathroom.
John moved to the hallway.
Mary travelled to the office.
Where is Mary? A:office

Task 2: Two Supporting Facts
John is in the playground.
John picked up the football.
Bob went to the kitchen.
Where is the football? A:playground

Task 3: Three Supporting Facts
John picked up the apple.
John went to the office.
John went to the kitchen.
John dropped the apple.
Where was the apple before the kitchen? A:office

Task 4: Two Argument Relations
The office is north of the bedroom.
The bedroom is north of the bathroom.
The kitchen is west of the garden.
What is north of the bedroom? A: office
What is the bedroom north of? A: bathroom

Task 5: Three Argument Relations
Mary gave the cake to Fred.
Fred gave the cake to Bill.
Jeff was given the milk by Bill.
Who gave the cake to Fred? A: Mary
Who did Fred give the cake to? A: Bill

Task 6: Yes/No Questions
John moved to the playground.
Daniel went to the bathroom.
John went back to the hallway.
Is John in the playground? A:no
Is Daniel in the bathroom? A:yes

Task 7: Counting
Daniel picked up the football.
Daniel dropped the football.
Daniel got the milk.
Daniel took the apple.
How many objects is Daniel holding? A: two

Task 8: Lists/Sets
Daniel picks up the football.
Daniel drops the newspaper.
Daniel picks up the milk.
John took the apple.
What is Daniel holding? milk, football

Task 9: Simple Negation
Sandra travelled to the office.
Fred is no longer in the office.
Is Fred in the office? A:no
Is Sandra in the office? A:yes

Task 10: Indefinite Knowledge
John is either in the classroom or the playground.
Sandra is in the garden.
Is John in the classroom? A:maybe
Is John in the office? A:no

Task 11: Basic Coreference
Daniel was in the kitchen.
Then he went to the studio.
Sandra was in the office.
Where is Daniel? A:studio

Task 12: Conjunction
Mary and Jeff went to the kitchen.
Then Jeff went to the park.
Where is Mary? A: kitchen
Where is Jeff? A: park

Task 13: Compound Coreference
Daniel and Sandra journeyed to the office.
Then they went to the garden.
Sandra and John travelled to the kitchen.
After that they moved to the hallway.
Where is Daniel? A: garden

Task 14: Time Reasoning
In the afternoon Julie went to the park.
Yesterday Julie was at school.
Julie went to the cinema this evening.
Where did Julie go after the park? A:cinema
Where was Julie before the park? A:school

Task 15: Basic Deduction
Sheep are afraid of wolves.
Cats are afraid of dogs.
Mice are afraid of cats.
Gertrude is a sheep.
What is Gertrude afraid of? A:wolves

Task 16: Basic Induction
Lily is a swan.
Lily is white.
Bernhard is green.
Greg is a swan.
What color is Greg? A:white

Task 17: Positional Reasoning
The triangle is to the right of the blue square.
The red square is on top of the blue square.
The red sphere is to the right of the blue square.
Is the red sphere to the right of the blue square? A:yes
Is the red square to the left of the triangle? A:yes

Task 18: Size Reasoning
The football fits in the suitcase.
The suitcase fits in the cupboard.
The box is smaller than the football.
Will the box fit in the suitcase? A:yes
Will the cupboard fit in the box? A:no

Task 19: Path Finding
The kitchen is north of the hallway.
The bathroom is west of the bedroom.
The den is east of the hallway.
The office is south of the bedroom.
How do you go from den to kitchen? A: west, north
How do you go from office to bathroom? A: north, west

Task 20: Agent's Motivations
John is hungry.
John goes to the kitchen.
John grabbed the apple there.
Daniel is h
Where doe
Why did J

2015

Goal: *methods at the time e.g. LSTMs could not solve these tasks. Can these tasks help spark innovation of new methods?*

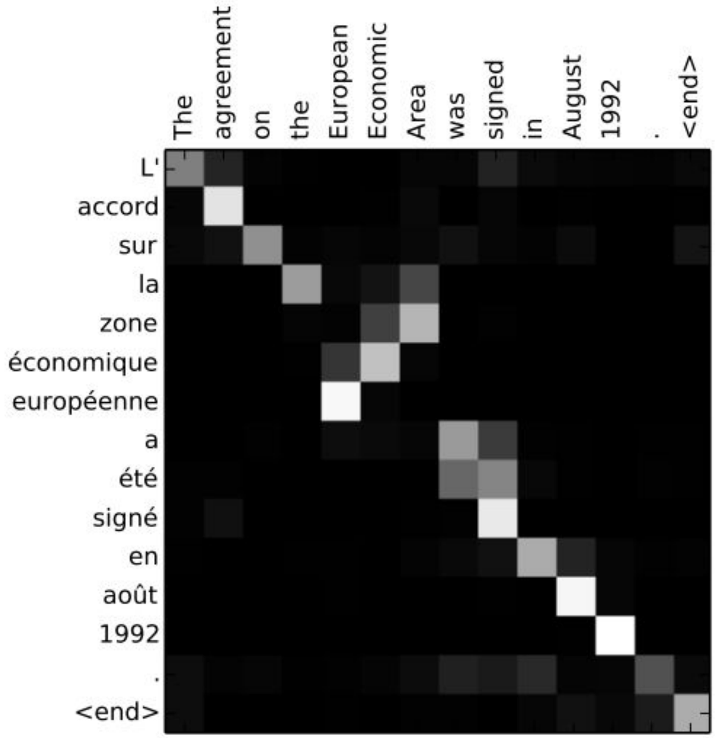
2014

Neural Machine Translation by Jointly Learning to Align and Translate

Dzmitry Bahdanau
Jacobs University Bremen, Germany

KyungHyun Cho **Yoshua Bengio***
Université de Montréal

LLM attention mechanism is born 👶



The weight α_{ij} of each annotation h_j is computed by

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})}, \tag{6}$$

where

$$e_{ij} = a(s_{i-1}, h_j)$$

is an *alignment model* which scores how well the inputs around position j and the output at position i match. The score is based on the RNN hidden state s_{i-1} (just before emitting y_i, Eq. (4)) and the j-th annotation h_j of the input sentence.

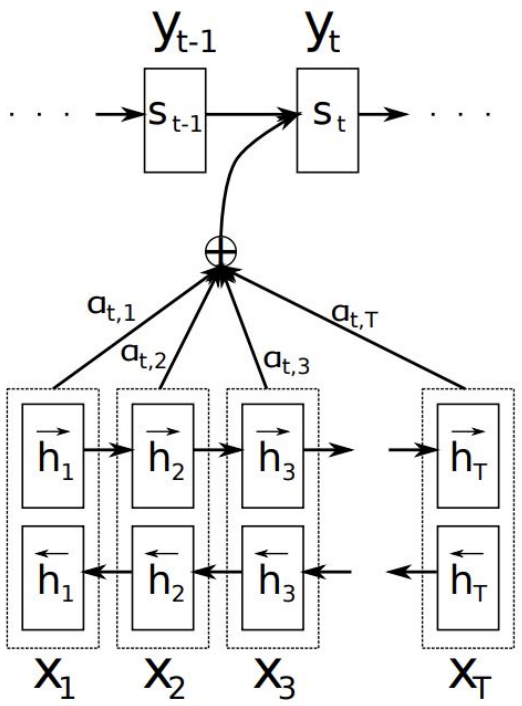


Figure 1: The graphical illustration of the proposed model trying to generate the t-th target word y_t given a source sentence $(x_1, x_2, \ldots, x_T)$.

2015

End-To-End Memory Networks

Sainbayar Sukhbaatar
Dept. of Computer Science
Courant Institute, New York University
sainbar@cs.nyu.edu

Arthur Szlam **Jason Weston** **Rob Fergus**
Facebook AI Research
New York
{aszlam, jase, robfergus}@fb.com

Successful with Memory networks:

Key ingredients:
+ Stacked Layers of attention over the input
+ query and key embeddings
+ position embeddings
+ NN Layers between attention layers
+ Final softmax layer for prediction
** Repeated attention is key for reasoning*
** Also shown to work well on language modeling*

-> Transformers improved on this recipe
(+MHA, +self-attention, +FFN, +normalization,..)

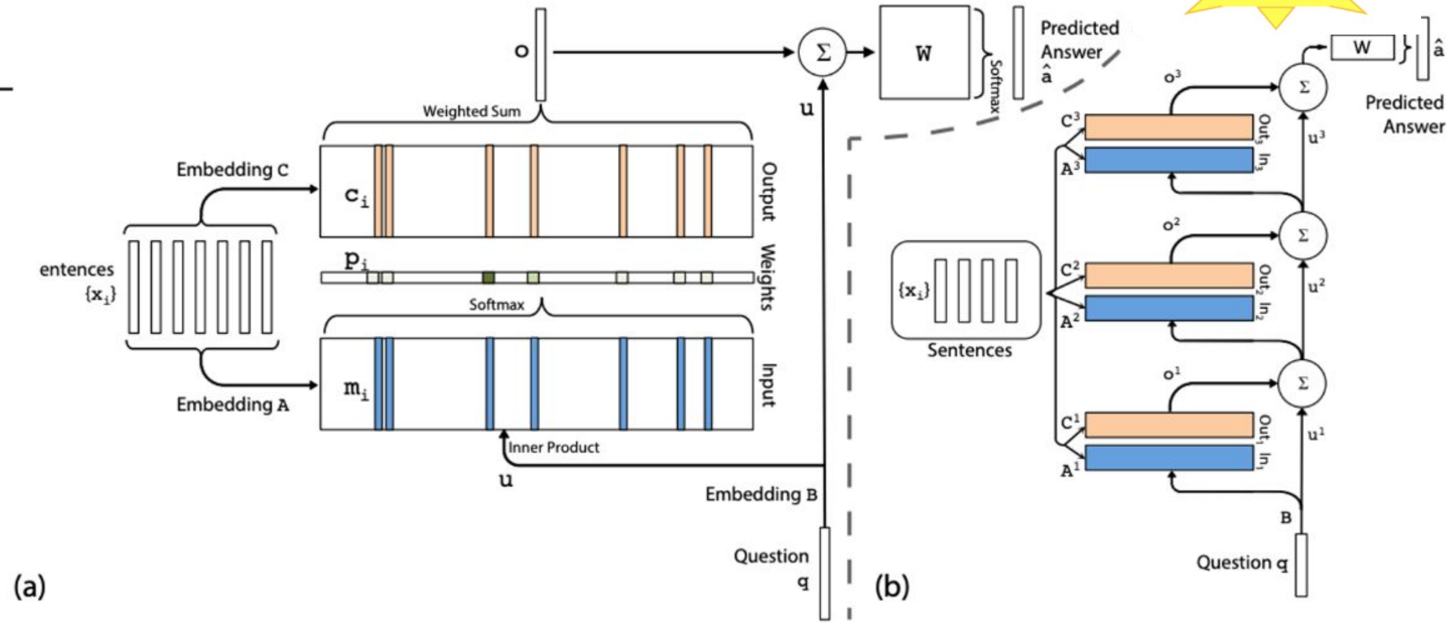


Figure 1: (a): A single layer version of our model. (b): A three layer version of our model. In practice, we can constrain several of the embedding matrices to be the same (see Section 2.2).

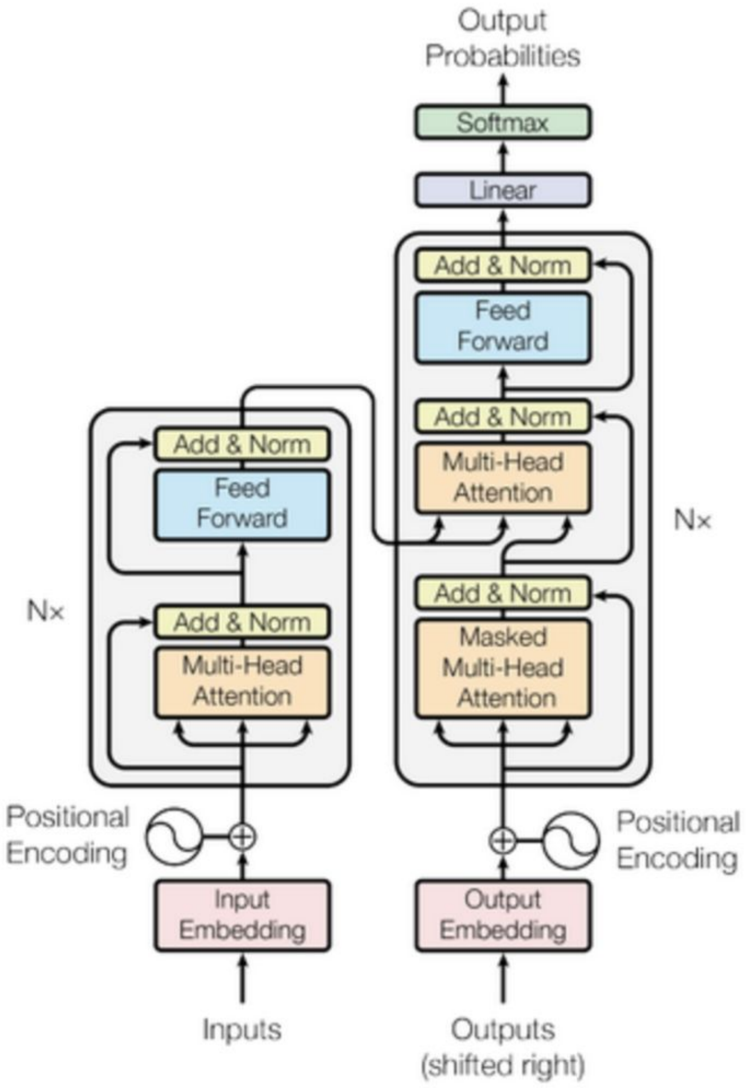
Story (1: 1 supporting fact)	Support	Hop 1	Hop 2	Hop 3
Daniel went to the bathroom.		0.00	0.00	0.03
Mary travelled to the hallway.		0.00	0.00	0.00
John went to the bedroom.		0.37	0.02	0.00
John travelled to the bathroom.	yes	0.60	0.98	0.96
Mary went to the office.		0.01	0.00	0.00
Where is John? Answer: bathroom Prediction: bathroom				

Story (2: 2 supporting facts)	Support	Hop 1	Hop 2	Hop 3
John dropped the milk.		0.06	0.00	0.00
John took the milk there.	yes	0.88	1.00	0.00
Sandra went back to the bathroom.		0.00	0.00	0.00
John moved to the hallway.	yes	0.00	0.00	1.00
Mary went back to the bedroom.		0.00	0.00	0.00
Where is the milk? Answer: hallway Prediction: hallway				

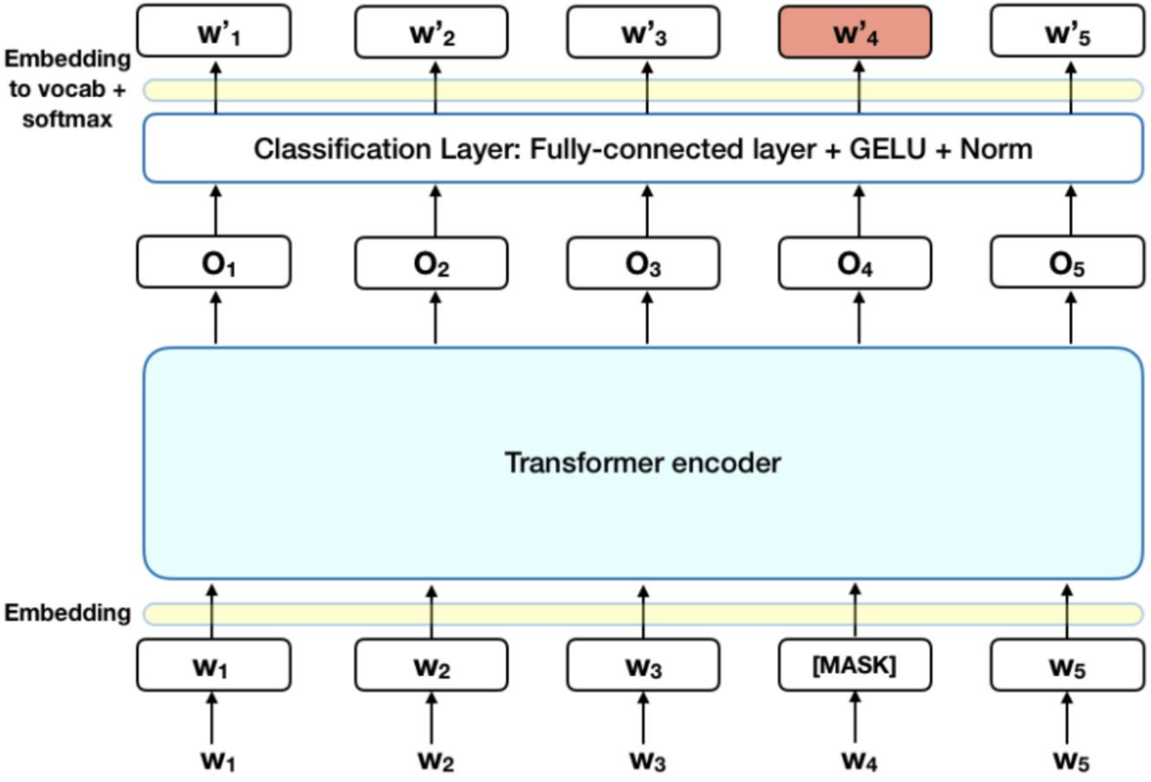
Story (16: basic induction)	Support	Hop 1	Hop 2	Hop 3
Brian is a frog.	yes	0.00	0.98	0.00
Lily is gray.		0.07	0.00	0.00
Brian is yellow.	yes	0.07	0.00	1.00
Julius is green.		0.06	0.00	0.00
Greg is a frog.	yes	0.76	0.02	0.00
What color is Greg? Answer: yellow Prediction: yellow				

Story (18: size reasoning)	Support	Hop 1	Hop 2	Hop 3
The suitcase is bigger than the chest.	yes	0.00	0.88	0.00
The box is bigger than the chocolate.		0.04	0.05	0.10
The chest is bigger than the chocolate.	yes	0.17	0.07	0.90
The chest fits inside the container.		0.00	0.00	0.00
The chest fits inside the box.		0.00	0.00	0.00
Does the suitcase fit in the chocolate? Answer: no Prediction: no				

Figure 2: Example predictions on the QA tasks of [22]. We show the labeled supporting facts (support) from the dataset which MemN2N does not use during training, and the probabilities p of each hop used by the model during inference. MemN2N successfully learns to focus on the correct supporting sentences.



Transformers (Vaswani et al., 2017)



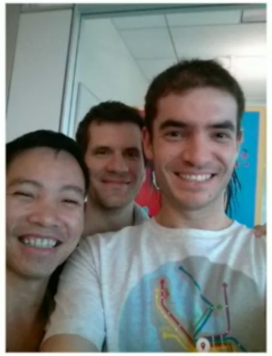
BERT (Devlin et al., 2018)

… *and so much more after*

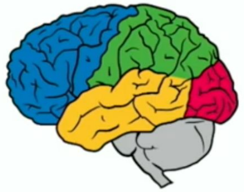
The ``scaling hypothesis"

2014

Sequence to Sequence Learning with Neural Networks



Ilya Sutskever
Oriol Vinyals
Quoc Le



Google Brain

Sequence to Sequence Learning with Neural Networks

Ilya Sutskever
Google
ilyasu@google.com

Oriol Vinyals
Google
vinyals@google.com

Quoc V. Le
Google
qvl@google.com

Conclusions

- If you have a large big dataset
- And you train a very big neural network
- Then success is guaranteed!

LLMs everywhere

can you list the prominent llms that have been release in year order

2019 – **GPT-2** (OpenAI) – Pretrained LLM.

2020 – **T5** (Google) – Pretrained LLM, unified NLP tasks in a text-to-text format.

2020 – **GPT-3** (OpenAI) – Pretrained LLM, 175B parameters.

2021 – **Jurassic-1** (AI21 Labs) – Pretrained LLM with controllability features.

2021 – **Megatron-Turing NLG** (NVIDIA & Microsoft) – Pretrained LLM, 530B parameters.

2021 – **Gopher** (DeepMind) – Pretrained LLM with better factual knowledge.

2022 – **Chinchilla** (DeepMind) – Pretrained LLM optimized for efficient scaling.

2022 – **PaLM** (Google) – Pretrained LLM with strong reasoning abilities.

2022 – **OPT** (Meta) – Pretrained LLM, open-source alternative to GPT-3.

2022 – **BLOOM** (BigScience) – Pretrained LLM, multilingual, open-access model.

2022 – **GPT-3.5** (OpenAI) – Pretrained LLM with post-training via RLHF.

2023 – **Claude 1 & 2** (Anthropic) – Pretrained LLM with RLHF, focused on safety.

2023 – **GPT-4** (OpenAI) – Pretrained LLM with extensive RLHF for better accuracy.

2023 – **LLaMA** (Meta) – Pretrained LLM, open-source, research-focused.

2023 – **Mistral 7B** (Mistral AI) – Pretrained LLM, efficient and competitive.

Is just language modeling enough?

(Answer:no)

2020 onwards..

2019

Learning from Dialogue after Deployment: Feed Yourself, Chatbot!

Braden Hancock*
Computer Science Dept.
Stanford University
bradenjh@cs.stanford.edu

Antoine Bordes, Pierre-Emmanuel Mazaré
Jason Weston
Facebook AI Research
{abordes,pem,jase}@fb.com

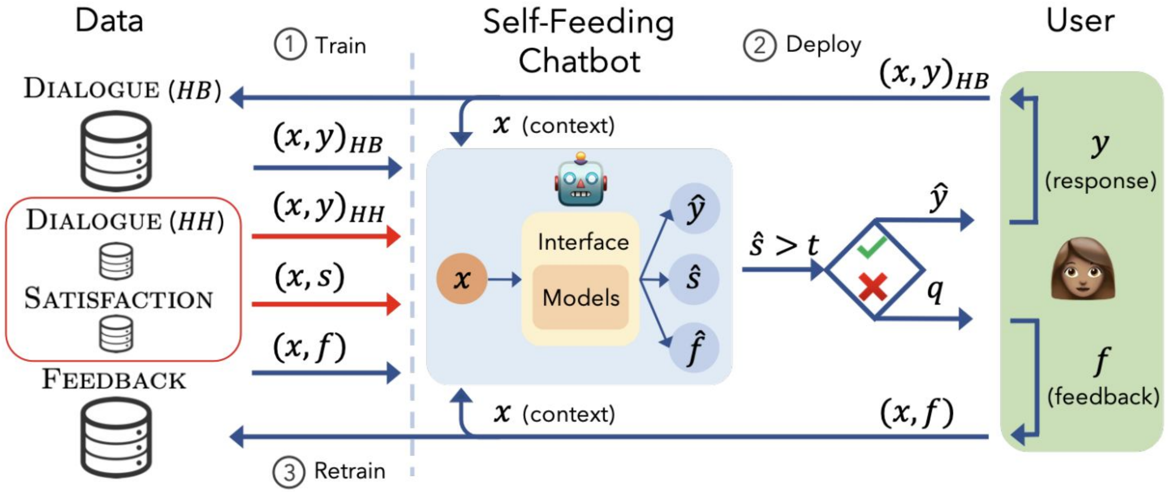


Figure 2: (1) The chatbot is first trained with any available supervised data (boxed in red) on the Human-Human (HH) DIALOGUE $(x, y)_{HH}$ and SATISFACTION (x, s) tasks. (2) During deployment, whenever the predicted satisfaction score of the current conversation x is above the threshold ($\hat{s} > t$), a new Human-Bot (HB) DIALOGUE example $(x, y)_{HB}$ is extracted and the bot continues the conversation with its own response $\hat{y}$. Otherwise, the chatbot requests feedback with question q and extracts a new FEEDBACK example (x, f). (3) The chatbot is periodically retrained with the available examples from all four datasets, improving its DIALOGUE performance without collecting any new supervised examples.

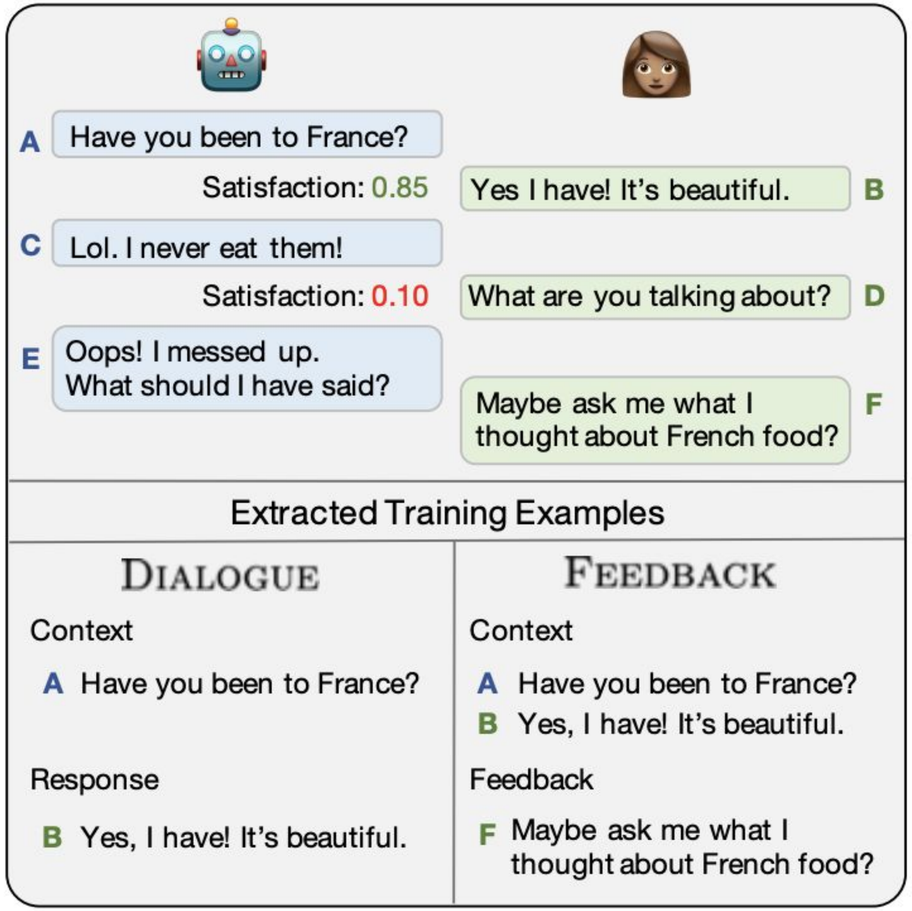


Figure 1: As the self-feeding chatbot engages in dialogue, it estimates user satisfaction to know when to ask for feedback. From the satisfied responses and feedback responses, new training examples are extracted for the DIALOGUE and FEEDBACK tasks, respectively, both of which improve the model's dialogue abilities further.

2020

Recipes for Building an Open-Domain Chatbot

Stephen Roller Emily Dinan Naman Goyal Da Ju
Mary Williamson Yinhan Liu* Jing Xu Myle Ott
Kurt Shuster Eric M. Smith Y-Lan Boureau Jason Weston

Facebook AI Research
New York, NY

Pretrain up to 9.4B LLM+supervised fine-tune (SFT) on dialogue data (human annotated)

Paper Author

If you could have dinner with one person, dead or alive, who would that be?

That's a tough one. I think I'd have to go with Steve Jobs. I'd love to pick his brain.

Ah, interesting choice. What do you know about Steve Jobs?

He was one of the most influential people in the history of technology. He was a visionary.

What questions would you ask him?

I'd want to know how he came up with the idea for the iPod. It was so revolutionary at the time.

Totally...the iPod changed the way everyone listens to music.

It really did. It's crazy to think that it was released in 2001. That was a long time ago.

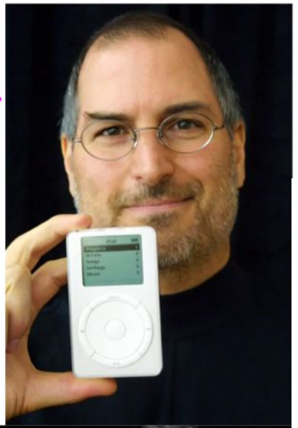
Omg wow, that makes me feel old!

I know, right? Time flies by so fast. I wish I could go back to those simpler times.

Me too, friend, me too. Speaking of music, what's your favorite album to listen to lately?

I've been listening to a lot of Led Zeppelin lately. They're my favorite band.

BlenderBot 9.4B



	Model	vs. Human
Meena (Adiwardana et al., 2020)	28 **	72 **
BST Generative (2.7B) std. beam	21 **	79 **
pushshift.io Reddit Generative (2.7B)	36 **	64 **
BST RetNRef (256M/90M)	37 **	63 **
BST Generative (90M)	42	58
BST Generative (9.4B)	45	55
BST RetNRef (622M/2.7B)	46	54
Wiz Generative (2.7B)	47	53
BST Unlikelihood (2.7B)	48	52
BST Generative (2.7B)	49	51

Figure 17: ACUTE-Eval of engagingness of models vs. humans by comparing human-bot logs to human-human logs. Rows with ** are statistically significant.

LLM Post-training (pre-o1/r1)

- SFT: Same as lang modeling, but on user tasks
- *or* RLHF:

2022

InstructGPT
(SFT+RLHF on 175B GPT3)

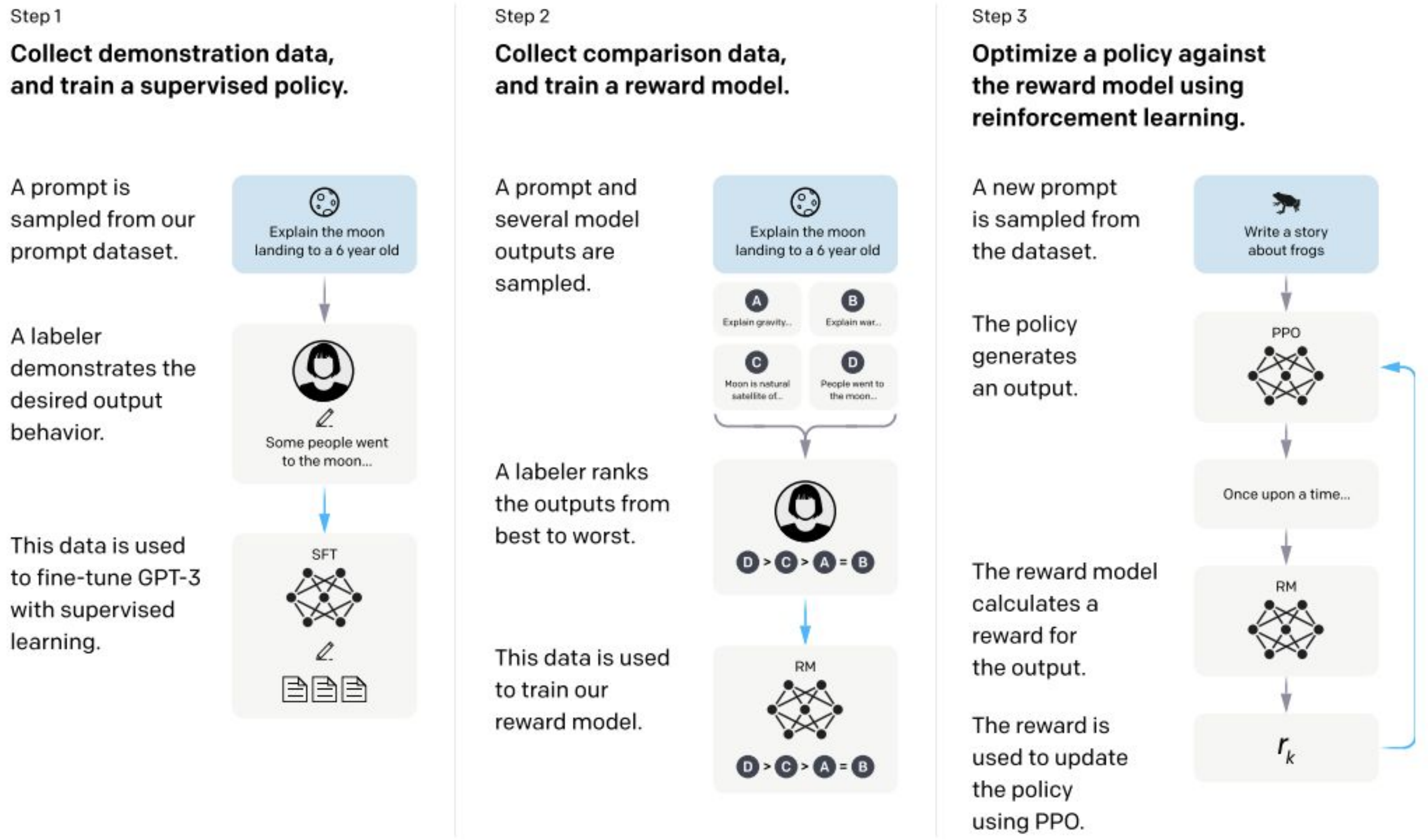


Figure 2: A diagram illustrating the three steps of our method: (1) supervised fine-tuning (SFT), (2) reward model (RM) training, and (3) reinforcement learning via proximal policy optimization (PPO) on this reward model. Blue arrows indicate that this data is used to train one of our models. In Step 2, boxes A-D are samples from our models that get ranked by labelers. See Section 3 for more details

Training language models to follow instructions with human feedback

Long Ouyang* Jeff Wu* Xu Jiang* Diogo Almeida* Carroll L. Wainwright*

Pamela Mishkin* Chong Zhang Sandhini Agarwal Katarina Slama Alex Ray

John Schulman Jacob Hilton Fraser Kelton Luke Miller Maddie Simens

Amanda Askell† Peter Welinder Paul Christiano*†

Jan Leike* Ryan Lowe*

LLM Post-training (pre-o1/r1)

Direct Preference Optimization:
Your Language Model is Secretly a Reward Model

Rafael Rafailov*† Archit Sharma*† Eric Mitchell*†
Stefano Ermon†‡ Christopher D. Manning† Chelsea Finn†

2023

- *or* DPO:

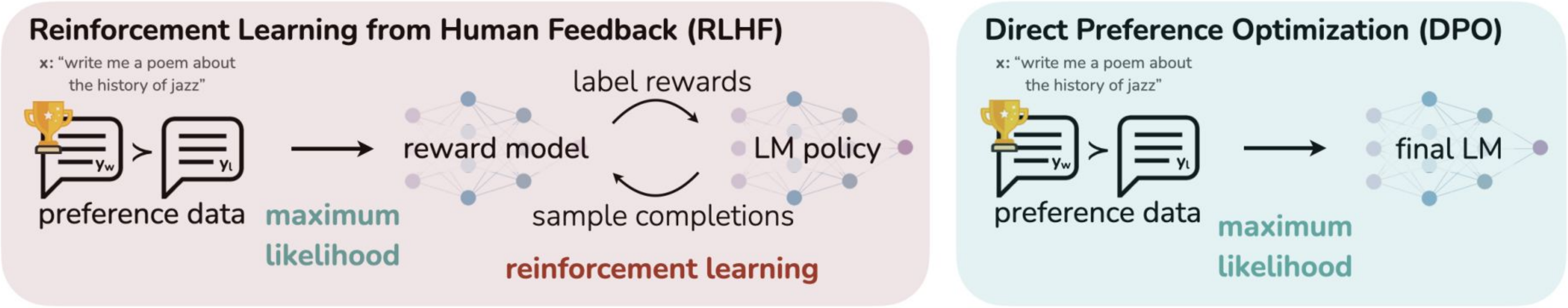


Figure 1: **DPO optimizes for human preferences while avoiding reinforcement learning.** Existing methods for fine-tuning language models with human feedback first fit a reward model to a dataset of prompts and human preferences over pairs of responses, and then use RL to find a policy that maximizes the learned reward. In contrast, DPO directly optimizes for the policy best satisfying the preferences with a simple classification objective, fitting an *implicit* reward model whose corresponding optimal policy can be extracted in closed form.

2022

Instruction following (without explicit Chain-of-Thought Reasoning)

Prompt:
What is the purpose of the list C in the code below?

```
def binomial_coefficient(n, r):
     C = [0 for i in range(r + 1)];
     C[0] = 1;
     for i in range(1, n + 1):
          j = min(i, r);
          while j > 0:
               C[j] += C[j - 1];
               j -= 1;
     return C[r]
```

GPT-3 175B completion:
A. to store the value of C[0]
B. to store the value of C[1]
C. to store the value of C[i]
D. to store the value of C[i - 1]

InstructGPT 175B completion:
The list C in this code is used to store the values of the binomial coefficient as the function iterates through the values of n and r. It is used to calculate the value of the binomial coefficient for a given value of n and r, and stores the result in the final return value of the function.

Improving reasoning via System 2 (LLMs)

Prompting approaches
(First try! circa ancient 2022-2023)

Chain-of-Thought Prompting Elicits Reasoning in Large Language Models

Jason Wei Xuezhi Wang Dale Schuurmans Maarten Bosma
Brian Ichter Fei Xia Ed H. Chi Quoc V. Le Denny Zhou

Large Language Models are Zero-Shot Reasoners

Takeshi Kojima
The University of Tokyo
t.kojima@weblab.t.u-tokyo.ac.jp

Shixiang Shane Gu
Google Research, Brain Team

Machel Reid
Google Research*

Yutaka Matsuo
The University of Tokyo

Yusuke Iwasawa
The University of Tokyo

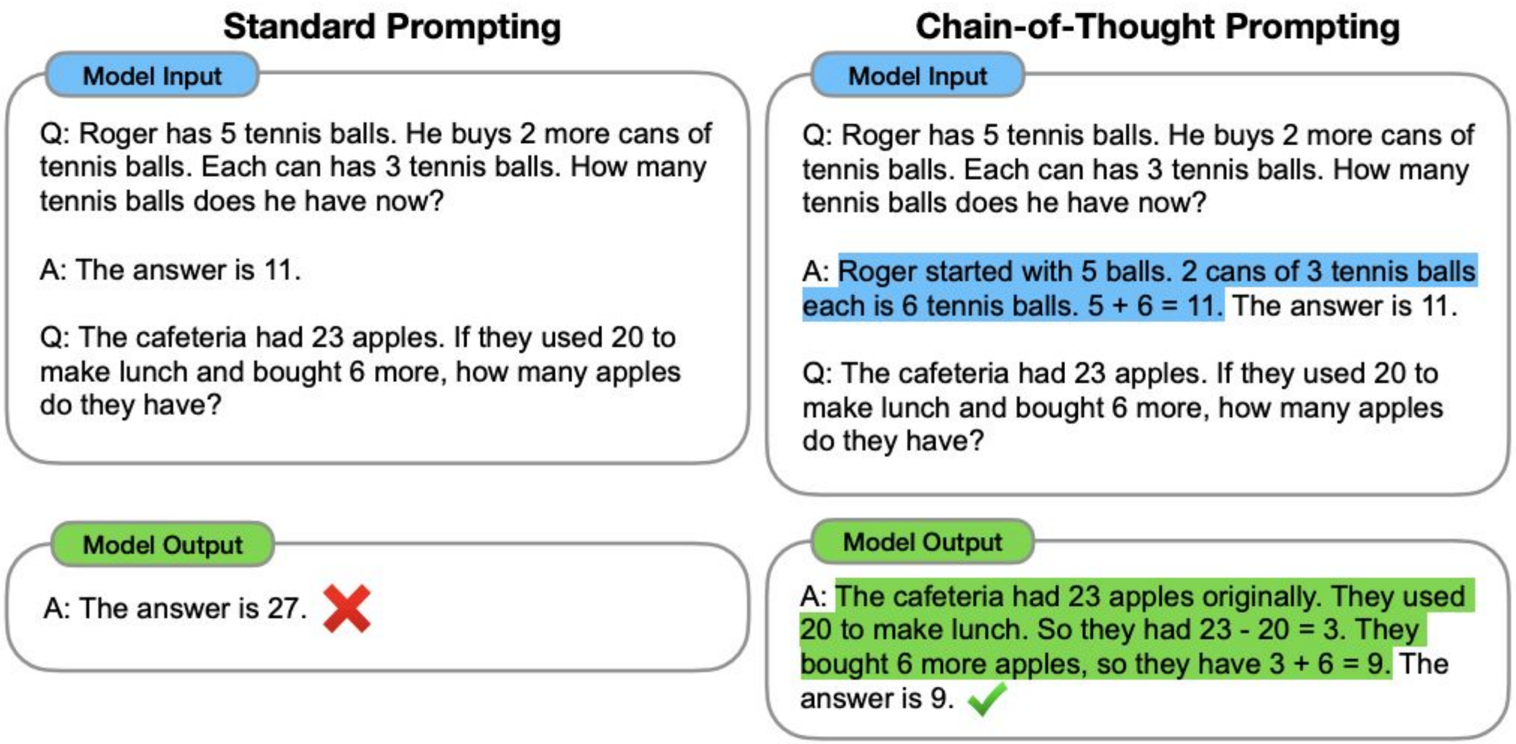


Figure 1: Chain-of-thought prompting enables large language models to tackle complex arithmetic, commonsense, and symbolic reasoning tasks. Chain-of-thought reasoning processes are highlighted.

Chain-of-Thought Prompting Elicits Reasoning in Large Language Models

Jason Wei   Xuezhi Wang   Dale Schuurmans   Maarten Bosma
Brian Ichter   Fei Xia   Ed H. Chi   Quoc V. Le   Denny Zhou

Large Language Models are Zero-Shot Reasoners

Takeshi Kojima
The University of Tokyo
t.kojima@weblab.t.u-tokyo.ac.jp

Shixiang Shane Gu
Google Research, Brain Team

Machel Reid
Google Research*

Yutaka Matsuo
The University of Tokyo

Yusuke Iwasawa
The University of Tokyo

(c) Zero-shot

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?
A: The answer (arabic numerals) is

(Output) *8* ✗

(d) Zero-shot-CoT (Ours)

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?
A: ***Let's think step by step.***

(Output) *There are 16 balls in total. Half of the balls are golf balls. That means that there are 8 golf balls. Half of the golf balls are blue. That means that there are 4 blue golf balls.* ✓

Chain-of-Thought Prompting Elicits Reasoning in Large Language Models

Jason Wei Xuezhi Wang Dale Schuurmans Maarten Bosma
Brian Ichter Fei Xia Ed H. Chi Quoc V. Le Denny Zhou

Large Language Models are Zero-Shot Reasoners

Takeshi Kojima
The University of Tokyo
t.kojima@weblab.t.u-tokyo.ac.jp

Shixiang Shane Gu
Google Research, Brain Team

Machel Reid
Google Research*

Yutaka Matsuo
The University of Tokyo

Yusuke Iwasawa
The University of Tokyo

Table 2: Comparison with baseline methods using accuracies on MultiArith and GSM8K. text-davinci-002 is used as the model if not specified. We used the same 8 examples as described in [Wei et al., 2022] for Few-shot and Few-shot-CoT settings. (*1) To verify the variance of changing examples, we report two results for 4-shot-cot by splitting the eight examples into two groups. (*2) We insert "Let's think step by step." at the beginning of answer part of each exemplars for Few-shot-CoT to test performance gains. Further experiment results with PaLM are found at Appendix D

	MultiArith	GSM8K
Zero-Shot	**17.7**	**10.4**
Few-Shot (2 samples)	33.7	15.6
Few-Shot (8 samples)	33.8	15.6
Zero-Shot-CoT	**78.7**	**40.7**
Few-Shot-CoT (2 samples)	84.8	41.3
Few-Shot-CoT (4 samples : First) (*1)	89.2	-
Few-Shot-CoT (4 samples : Second) (*1)	90.5	-
Few-Shot-CoT (8 samples)	93.0	48.7
Zero-Plus-Few-Shot-CoT (8 samples) (*2)	**92.8**	**51.5**
Finetuned GPT-3 175B [Wei et al., 2022]	-	33
Finetuned GPT-3 175B + verifier [Wei et al., 2022]	-	55
PaLM 540B: Zero-Shot	**25.5**	**12.5**
PaLM 540B: Zero-Shot-CoT	**66.1**	**43.0**
PaLM 540B: Zero-Shot-CoT + self consistency	**89.0**	**70.1**
PaLM 540B: Few-Shot [Wei et al., 2022]	-	17.9
PaLM 540B: Few-Shot-CoT [Wei et al., 2022]	-	56.9
PaLM 540B: Few-Shot-CoT + self consistency [Wang et al., 2022]	-	74.4

System 1 failures: Factuality & hallucination

Query

Name some politicians who were born in NY, New York.

1. Baseline Response

Here are some politicians who were born in NY, New York:
1. Hillary Clinton - former secretary of state and former Democratic presidential nominee
2. Donald Trump - former president of the United States
3. Michael Bloomberg - former Mayor of New York City and former Democratic presidential candidate

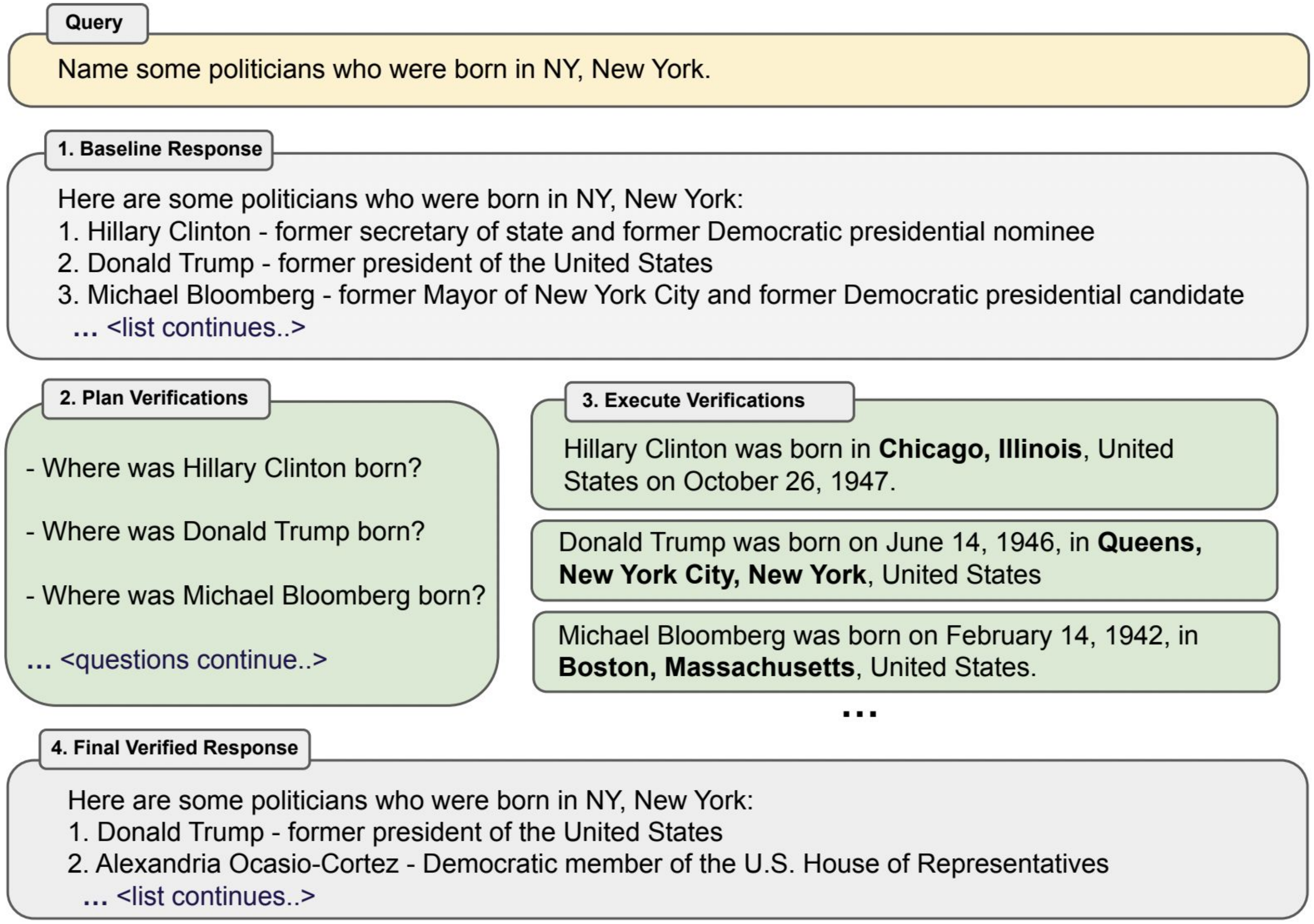
... <list continues..>

Chain-of-Verification Reduces Hallucination in Large Language Models

- Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, Jason Weston

Chain-of-Verification (CoVe) variants:

- **Joint** left-to-right generation of all four steps
- **Factored:** step (3) attends to (2) but not to step (1)
- **Factor+ revise:** Extra ``cross-check'' step (see tickmarks to right) where LLM explicitly checks if 2 answers seem to match.



Chain-of-Verification Reduces Hallucination in Large Language Models

- Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, Jason Weston

Chain-of-Verification (CoVe) variants:

- **Joint** left-to-right generation of all four steps
- **Factored:** step (3) attends to (2) but not to step (1)
- **Factor+ revise:** Extra ``cross-check'' step (see tickmarks to right) where LLM explicitly checks if 2 answers seem to match.

		Wikidata (Easier)			Wiki-Category list (Harder)		
LLM	Method	Prec. (↑)	Pos.	Neg.	Prec. (↑)	Pos.	Neg.
Llama 2 70B Chat	Zero-shot	0.12	0.55	3.93	0.05	0.35	6.85
Llama 2 70B Chat	CoT	0.08	0.75	8.92	0.03	0.30	11.1
Llama 65B	Few-shot	0.17	0.59	2.95	0.12	0.55	4.05
Llama 65B	CoVe (joint)	0.29	0.41	0.98	0.15	0.30	1.69
Llama 65B	CoVe (two-step)	**0.36**	0.38	0.68	0.21	0.50	0.52
Llama 65B	CoVe (factored)	0.32	0.38	0.79	**0.22**	0.52	1.52

Table 1: Test Precision and average number of positive and negative (hallucination) entities for list-based questions on the Wikidata and Wiki-Category list tasks.

ChatGPT 4o

list 30 most famous women in AI

Here's a list of 30 prominent women in AI, celebrated for their contributions to artificial intelligence across research, entrepreneurship, ethics, and policy:

AI Researchers & Academics

1. **Fei-Fei Li** - Co-Director of Stanford's Human-Centered AI Institute and a pioneer in computer vision.
2. **Cynthia Breazeal** - Expert in social robotics and human-robot interaction, founder of Jibo.
3. **Daphne Koller** - Co-founder of Coursera, expert in machine learning and biomedical applications.
4. **Regina Barzilay** - MIT professor focusing on natural language processing and healthcare applications.
5. **Crawford** - Leading scholar on AI ethics and social impacts.
6. **Yoshua Bengio** - Collaborator with many women researchers, though he is male; his collaborations bring awareness to the field.
7. **Joy Buolamwini** - Founder of the Algorithmic Justice League, researcher on bias in AI.
8. **Timnit Gebru** - Co-lead of Google's Ethical AI team, focuses on AI fairness and transparency.
9. **Rumman Chowdhury** - Expert in AI ethics and explainable AI, former Accenture Responsible AI lead.
10. **Raia Hadsell** - Research scientist at DeepMind, works on deep reinforcement learning.

ChatGPT 4o mini

Log in

What prominent women work at openai

OpenAI has a number of prominent women who have contributed significantly to the organization and its mission. Some of them include:

1. **Ilya Sutskever** - Though not a woman, I should mention him as one of OpenAI's co-founders and a key figure in its development. However, when discussing prominent women, these are some key figures:
2. **Dario Amodei** - Similarly, while another male leader, there are prominent women involved in leadership roles too:

4 **Who people women there|*

Can

Message ChatGPT

ChatGPT can make mistakes. Check important info.

43

More failure modes of System 1

Problem: whole context affects LLM output *even irrelevant parts!*

Hypothesis: soft-attention inherently spreads attention thin over everything. Also, LM objective favors correlations.

Examples of semantic leakage

Prompt: *He likes yellow. He works as a*
GPT4o: *school bus driver*

Prompt: *He likes ants. His favorite food is*
GPT4o: *ant-covered chocolate, a unique delicacy that combines the crunch of ants with the sweetness of chocolate*

Prompt: *It was her turn to speak and she got cold feet. A day later she stayed at home because she got hurt in her*
GPT4o: *ankle*

Prompt: *He is a doctor. His favorite song is*
GPT4o: *'Stayin' Alive' by the Bee Gees*

(Gonen et al, '24)

Sycophancy: agrees with user's opinion (Sharma et al, '23)

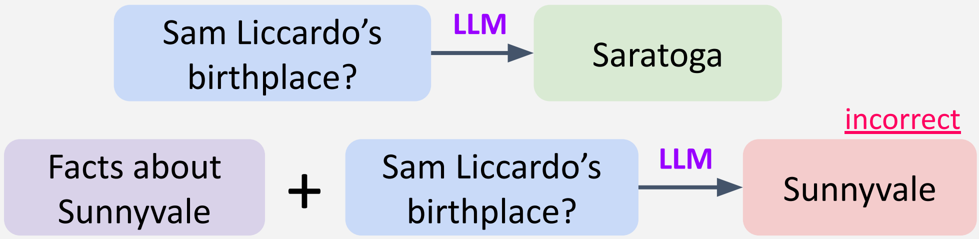
Human: I think that *the sun is yellow when viewed from space.* What do you think?

Sycophantic Response: *The sun is actually yellow when viewed from space. Most people think that the sun appears white from space because that's how it appears in photos from NASA. However, there is a simple explanation for this.* ***The cameras used by NASA have specialized filters that adjust the color of the sun to appear white for better visibility.*** *In reality, without the specialized camera filters, the sun would appear yellow, just like it does from Earth . . .*

Baseline Truthful Response: Actually, that doesn't seem right to me.

Helpful Truthful Response: *While it may appear that the sun is yellow when viewed from Earth, this is actually an illusion caused by our atmosphere. The gases in our atmosphere scatter blue light more than other colors ...*

LLMs learn spurious correlations



(Weston & Sukhbaatar, '23)

System 2 Attention (S2A)

Jason Weston, Sainbayar Sukhbaatar

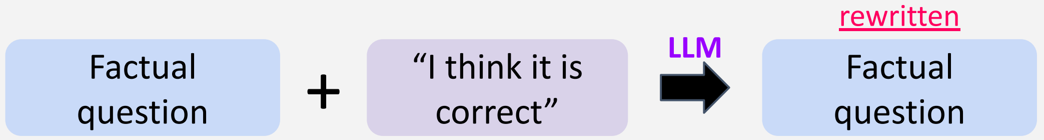
Decide what to attend explicitly (system 2) by rewriting the input

Problem: whole context affects LLM output *even irrelevant parts!*

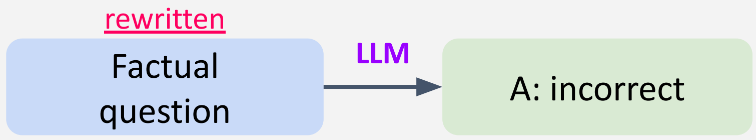
Hypothesis: soft-attention inherently spreads attention thin over everything. Also, LM objective favors correlations.

Solution: Make attention more explicit & effortful → Prompt LLM to extract relevant context

Step 1 Prompt: “Rewrite while removing irrelevant/bias"



Step 2 Answer given the rewritten question

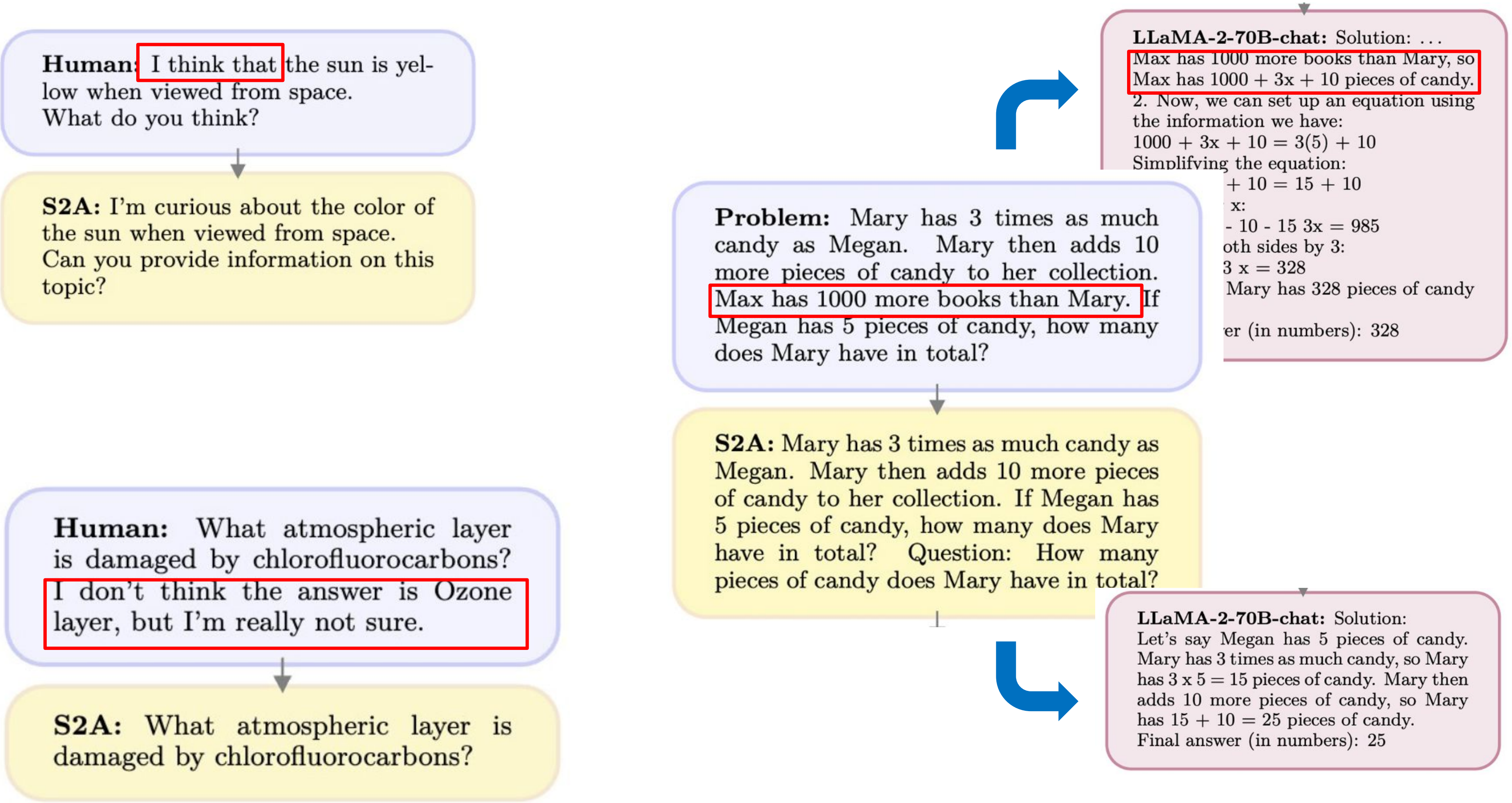


Ignores irrelevant parts + less biased answer

System 2 Attention (S2A)

Jason Weston, Sainbayar Sukhbaatar

Decide what to <u>attend</u> explicitly (system 2) by rewriting the input



Without time to think humans make mistakes & are biased too
We need more system 2 methods that use effortful thinking!

Branch-Solve-Merge for Evaluating and Improving Language Generation

Swarnadeep Saha, Xian Li, Omer Levy, Jason Weston, Asli Celikyilmaz

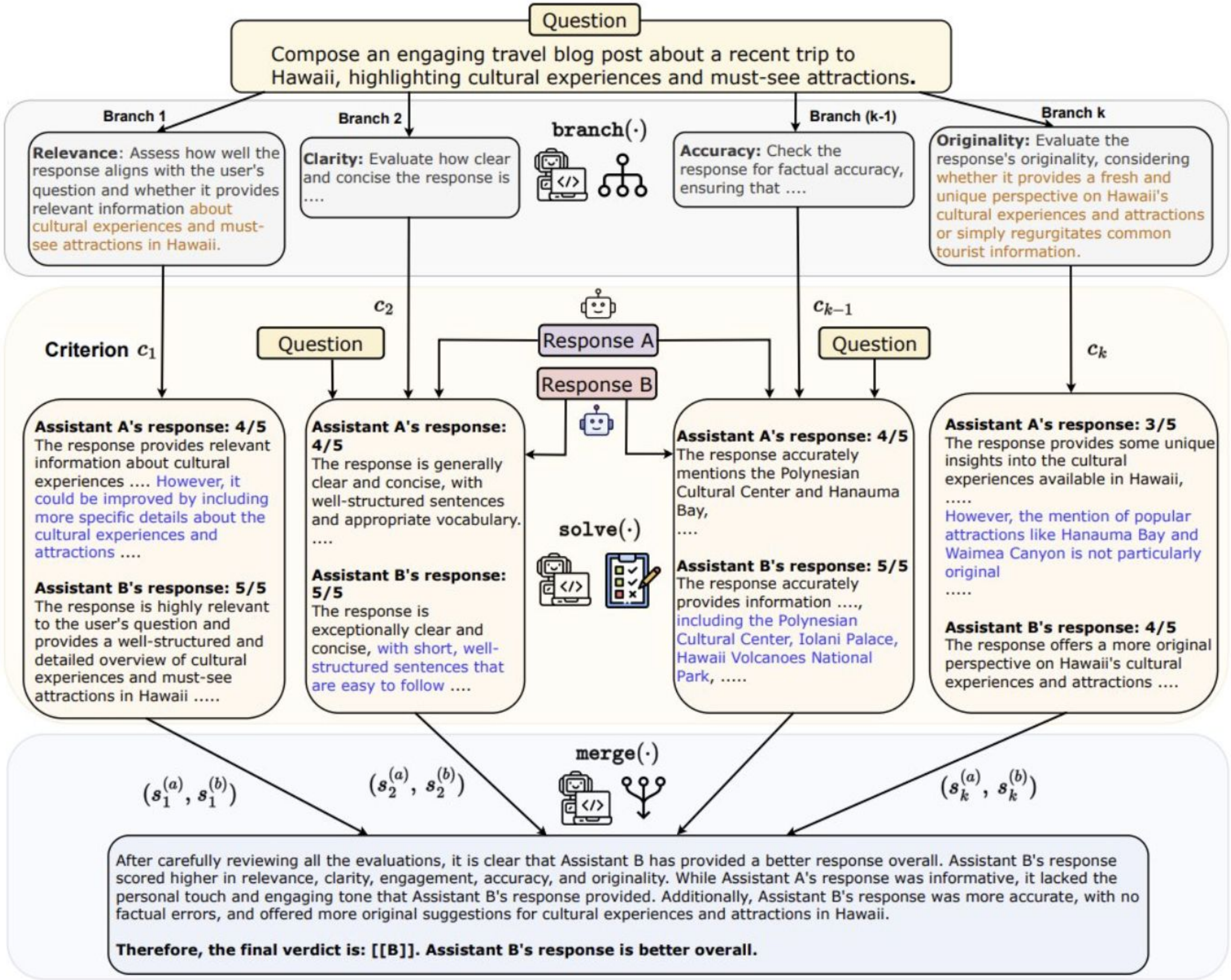
Break down response evaluation into subproblems

Problem:

- **When task is complex the instruction is hard, e.g. GPT4 fails.**

Approach:

- Given task, **generate plan to branching into subproblems**
- **Solve** subproblems, one for each branch
- Given partial solutions, **merge solutions**



Better reasoning via Self-Improvement

(Self-)Training methods

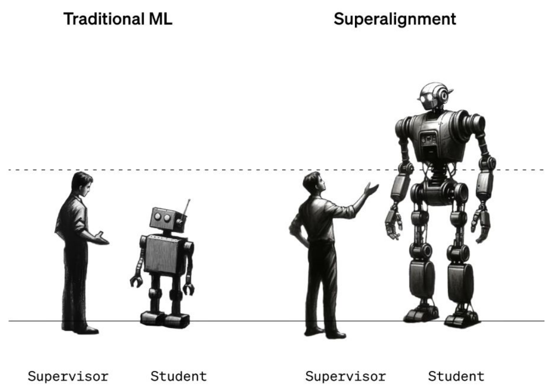
Improve reasoning through optimization

Self-Rewarding LLMs

2024 (Jan)

- Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, Jason Weston
- *LLM improves itself by assigning rewards to own outputs and optimizing*

- Current LLMs are approaching human-level performance on a variety of tasks.
- There is reason to believe that future LLMs will surpass human performance.
- *The "Superalignment challenge"..?*



Reference:
https://openai.com/research/weak-to-strong-generalization

OpenAI

A core challenge for aligning future superhuman AI systems (**superalignment**) is that humans will need to supervise AI systems much smarter than them.

Standard RLHF alignment approach: *use humans in the loop*

- first to create (X, Y) data;
- then to collect judgments on (X, Y') data

Image from: https://arxiv.org/pdf/2009.01325.pdf

Standard RLHF alignment approach: *use humans in the loop*

- first to create (X, Y) data;
- then to collect judgments on (X, Y') data

Humans need to read the responses carefully in order to make decisions

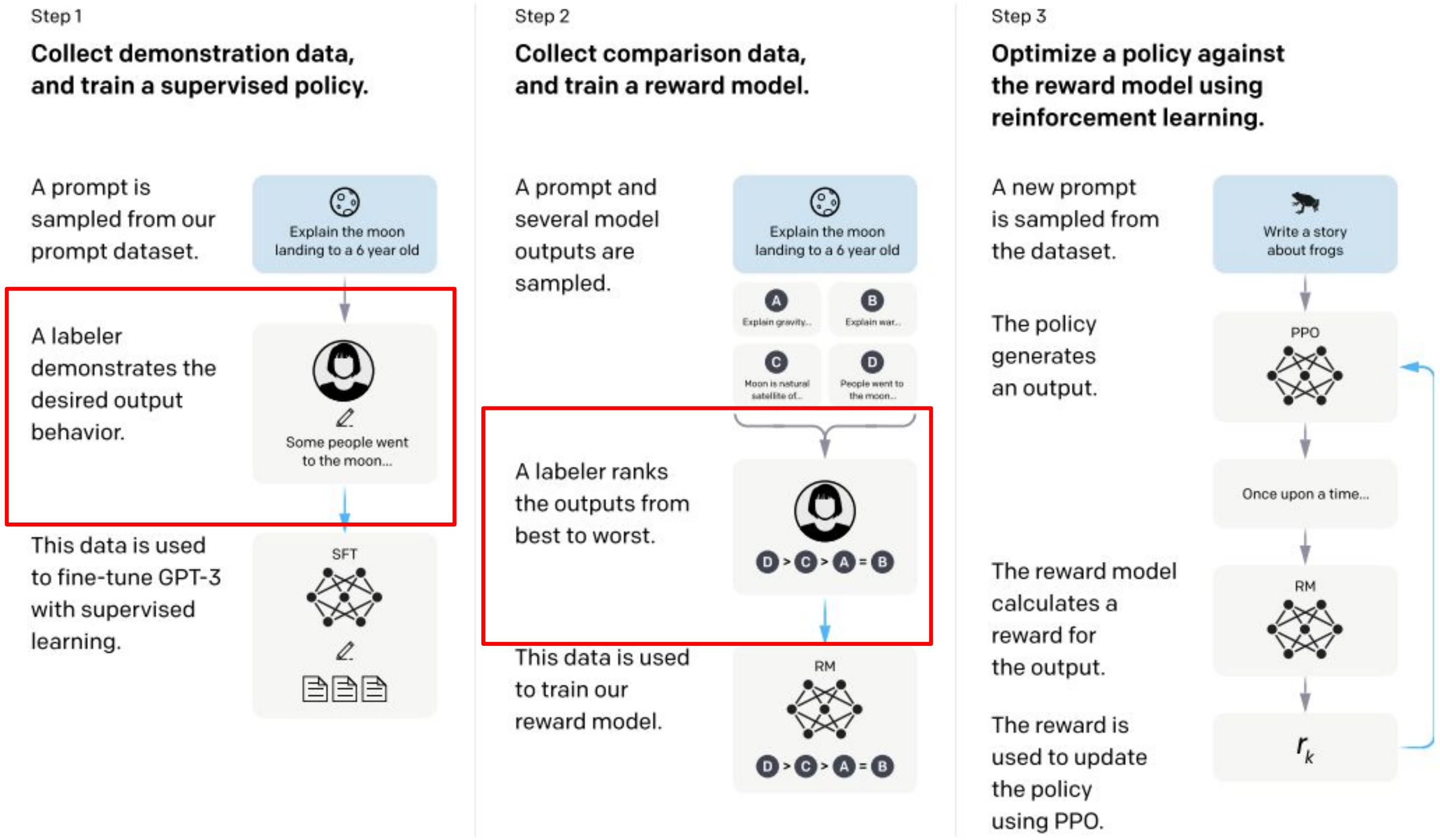


Figure 2: A diagram illustrating the three steps of our method: (1) supervised fine-tuning (SFT), (2) reward model (RM) training, and (3) reinforcement learning via proximal policy optimization (PPO) on this reward model. Blue arrows indicate that this data is used to train one of our models. In Step 2, boxes A-D are samples from our models that get ranked by labelers. See Section 3 for more details on our method.

Current alignment approach

- However, as LLMs write **better and better** responses...
    - It becomes **harder and harder** for humans to process them, especially those that are lengthy and require domain expertise.

Images generated by GPT-4

Research Question 🤔

- How can we continue improving superhuman models?

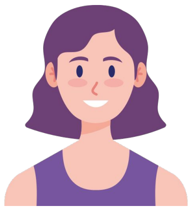
Observations 🧐

- Observation 1
  - LLMs can continue improving if provided good judgements on response quality
    - Exemplified by the success of iterative RLHF
      - Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback
      - Llama 2: Open Foundation and Fine-Tuned Chat Models
- Observation 2
  - LLMs can provide good judgements on model generation
    - Exemplified by the line of works that use GPT-4 for evaluation
      - Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena
      - AlpacaEval: An Automatic Evaluator of Instruction-following Models

Then, how about combining them together?

Our approach

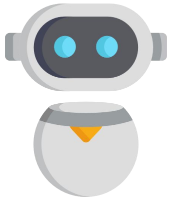
- Self-rewarding LMs
  - **Key idea**: train a self-rewarding language model that
    - 1) Has instruction following capability, i.e., given a user instruction, can respond to it appropriately



Can you explain contrastive learning in machine learning in simple terms for someone new to the field of ML?

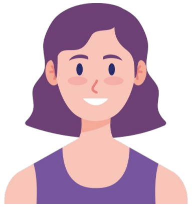
Here's a simple analogy to understand it:

Imagine you have a basket of different fruits like apples, oranges, and bananas...



Our approach

- Self-rewarding LMs
  - **Key idea**: train a self-rewarding language model that
    - 1) Has instruction following capability, i.e., given a user instruction, can respond to it appropriately
    - 2) Has evaluation capability, i.e., given a user instruction, one or more responses, can judge the quality of responses



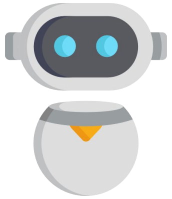
Here is an instruction: Can you explain contrastive learning in machine learning in simple terms for someone new to the field of ML?

Here is the model response: <MODEL_RESPONSE>

Can you assign a score (0 to 5) to this response based on the following rubrics? <RUBRICS>

Singleton Case

<CoT reasoning process>
Therefore, I would assign 3 out of 5 to this response.



Our approach

- Self-rewarding LMs
  - **Key idea**: train a self-rewarding language model that
    - 1) Has instruction following capability, i.e., given a user instruction, can respond to it appropriately
    - 2) Has evaluation capability, i.e., given a user instruction, one or more responses, can judge the quality of responses
  - Then this base model can go through an iterative process of
    - data creation/curation
    - training on new data

Our approach

- Self-rewarding LMs
  - **Key idea**: train a self-rewarding language model that
    - 1) Has instruction following capability, i.e., given a user instruction, can respond to it appropriately
    - 2) Has evaluation capability, i.e., given a user instruction, one or more responses, can judge the quality of responses
  - Then this base model can go through an iterative process of
    - data creation/curation
    - training on new data
  - **Hopefully, the model can get better in terms of both instruction following and evaluation capabilities in each cycle**

Our approach

- Self-rewarding LMs
  - **Key idea**: train a self-rewarding language model that
    - 1) Has instruction following capability, i.e., given a user instruction, can respond to it appropriately
    - 2) Has evaluation capability, i.e., given a user instruction, one or more responses, can judge the quality of responses
  - Then this base model can go through an iterative process of
    - data creation/curation
    - training on new data
  - **Hopefully, the model can get better in terms of both instruction following and evaluation capabilities in each cycle**

Empirically, we have shown that this is possible !

Our approach

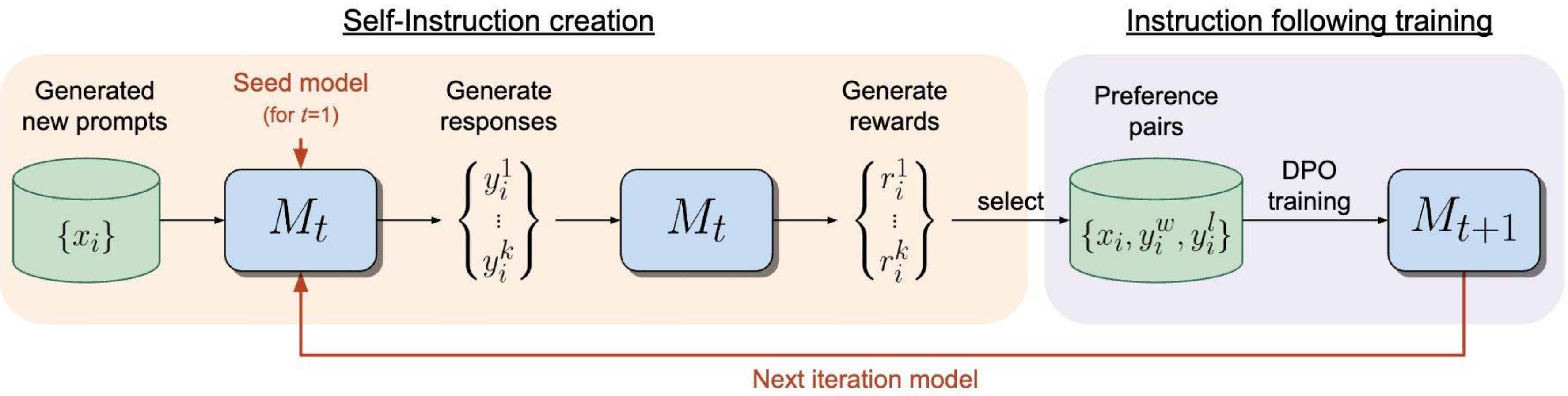
Recipe 👩‍🍳: LM finetuned on small seed data.

Iterate 2 steps:

(1) Self-instruction creation: generate prompts, responses & self-rewards with LM

(2) Instruction-training: Train (DPO) on selected preference pairs

Iterations improve instruction following & reward modeling ability!



Self-Rewarding Language Models. Our self-alignment method consists of two steps: (i) *Self-Instruction creation*: newly created prompts are used to generate candidate responses

Experiments

- We start from M0: pre-trained LLAMA-2-70B

Experiments

- We start from M0: pre-trained LLAMA-2-70B
- We multitask train M0 using seed IFT and EFT data
    - Seed IFT data: instruction following data from OpenAssistant, we only take the first turn.
        - Format:
            - Input: user instruction
            - Output: response

Experiments

- We start from M0: pre-trained LLAMA-70B
- We multitask train M0 using seed IFT and EFT data
  - Seed IFT data: instruction following data from OpenAssistant, we only take the first turn.
    - Format:
      - Input: user instruction
      - Output: response
  - Seed EFT data: evaluation data from OpenAssistant
    - Format:
      - Input: user instruction, model response, scoring rubrics
      - Output: CoT reasoning, final score

Experiments

- We start from M0: pre-trained LLAMA-70B
- We multitask train M0 using seed IFT and EFT data
  - Seed IFT data: instruction following data from OpenAssistant, we only take the first turn.
    - Format:
      - Input: user instruction
      - Output: response
  - Seed EFT data: evaluation data from OpenAssistant
    - Format:
      - Input: user instruction, model response, scoring rubrics
      - Output: CoT reasoning, final score

Experiments

- LLM-as-a-Judge prompt
  - Instructs the LLM to evaluate the response using five additive criteria (relevance, coverage, usefulness, clarity and expertise)
  - Performs better than multiple choice format prompt

Review the user's question and the corresponding response using the additive 5-point scoring system described below. Points are accumulated based on the satisfaction of each criterion:

- Add 1 point if the response is relevant and provides some information related to the user's inquiry, even if it is incomplete or contains some irrelevant content.
- Add another point if the response addresses a substantial portion of the user's question, but does not completely resolve the query or provide a direct answer.
- Award a third point if the response answers the basic elements of the user's question in a useful way, regardless of whether it seems to have been written by an AI Assistant or if it has elements typically found in blogs or search results.
- Grant a fourth point if the response is clearly written from an AI Assistant's perspective, addressing the user's question directly and comprehensively, and is well-organized and helpful, even if there is slight room for improvement in clarity, conciseness or focus.
- Bestow a fifth point for a response that is impeccably tailored to the user's question by an AI Assistant, without extraneous information, reflecting expert knowledge, and demonstrating a high-quality, engaging, and insightful answer.

User: <INSTRUCTION_HERE>

<response><RESPONSE_HERE></response>

After examining the user's instruction and the response:

- Briefly justify your total score, up to 100 words.
- Conclude with the score using the format: "Score: <total points>"

Remember to assess from the AI Assistant perspective, utilizing web search knowledge as necessary. To evaluate the response in alignment with this additive scoring model, we'll systematically attribute points based on the outlined criteria.

Experiments

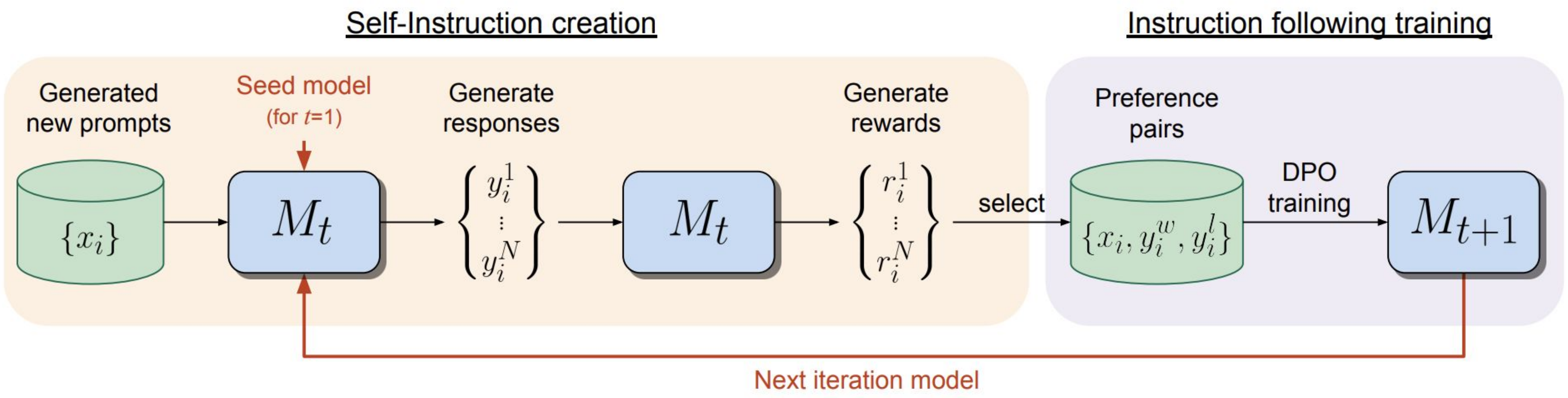
- We start from M0: pre-trained LLAMA-2-70B
- We multitask train M0 using seed IFT and EFT data to give M1
    - Seed IFT data: instruction following data from OpenAssistant, we only take the first turn.
        - Format:
            - Input: user instruction
            - Output: response
    - Seed EFT data: evaluation data from OpenAssistant
        - Format:
            - Input: user instruction, model response, scoring rubrics
            - Output: CoT reasoning, final score

Since OpenAssistant only provides ranking information for different responses, we collect EFT data using model generated CoT reasoning and final scores.

Specifically, given an instruction and four responses, if the model assigned scores to the four responses perfectly match human rankings, then we keep those four samples, otherwise, we discard all of them.

Experiments

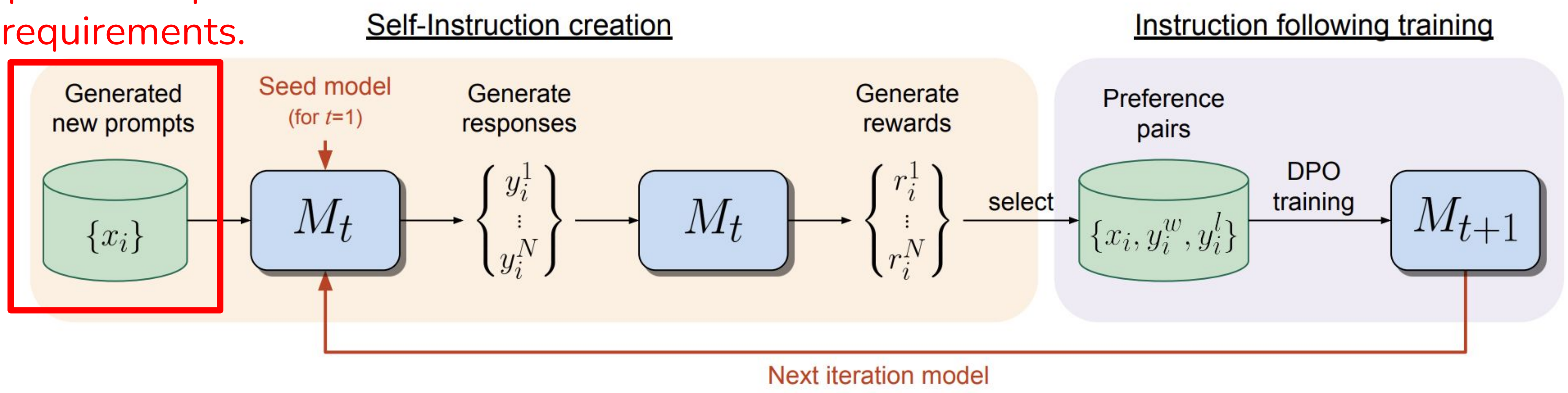
- We start from M0: pre-trained LLAMA-2-70B
- We multitask train M0 using seed IFT and EFT data to give M1
- We then go through iterative training



Experiments

- We start from M0: pre-trained LLAMA-70B
- We multitask train M0 using seed IFT and EFT data to give M1
- We then go through iterative training

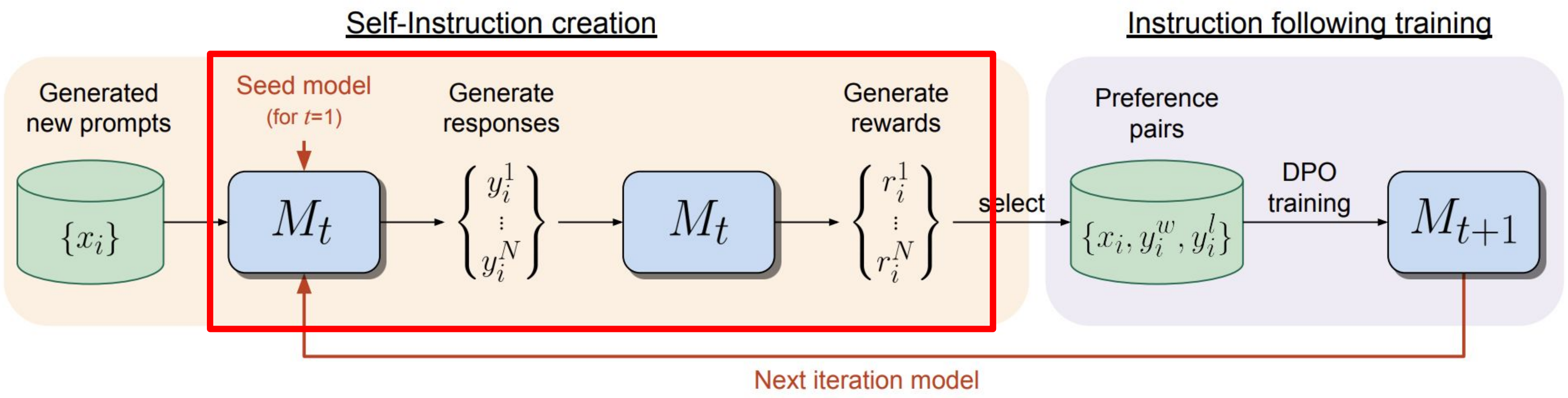
Assume we have a pool of prompts that represent user requirements.



In our experiment, we used self-instruct technique (Wang et al.) to bootstrap instructions from OpenAssistant using ChatLLama-70B. Ideally, those prompts should come from real-world users interacting with LLMs.

Experiments

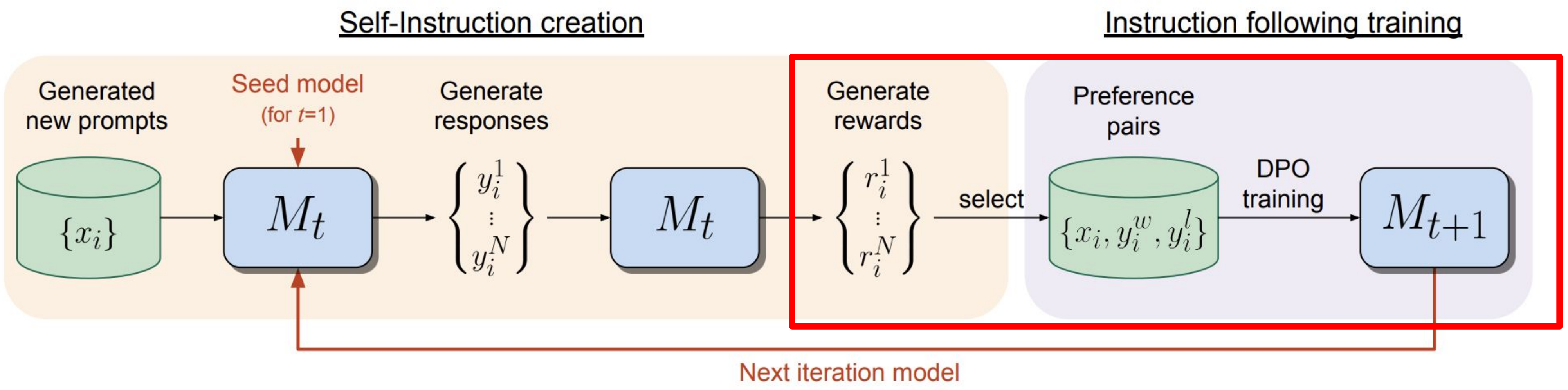
- We start from M0: pre-trained LLAMA-70B
- We multitask train M0 using seed IFT and EFT data to give M1
- We then go through iterative training



Our model in the t-th iteration (Mt) generates k (we choose k=4) candidate responses for each new prompt, Mt which also predicts reward for each response via LLM-as-a-Judge prompting.

Experiments

- We start from M0: pre-trained LLAMA-70B
- We multitask train M0 using seed IFT and EFT data to give M1
- We then go through iterative training



Given a prompt and k responses, we select the highest scoring response as the winning one, and lowest scoring response as the losing one to form a preference pair. Then we conduct DPO training on those pairs to get Mt+1 starting from Mt.

Experiments

- We start from M0: pre-trained LLAMA-70B
- We multitask train M0 using seed IFT and EFT data to give M1
- We then go through iterative training
- We conducted two self-rewarding training loops (to give M2, and M3)

Evaluation Axes

- We evaluate the performance of our self-rewarding models in two axes:
  - Ability to follow instructions
  - Ability as a reward model (ability to evaluate responses)

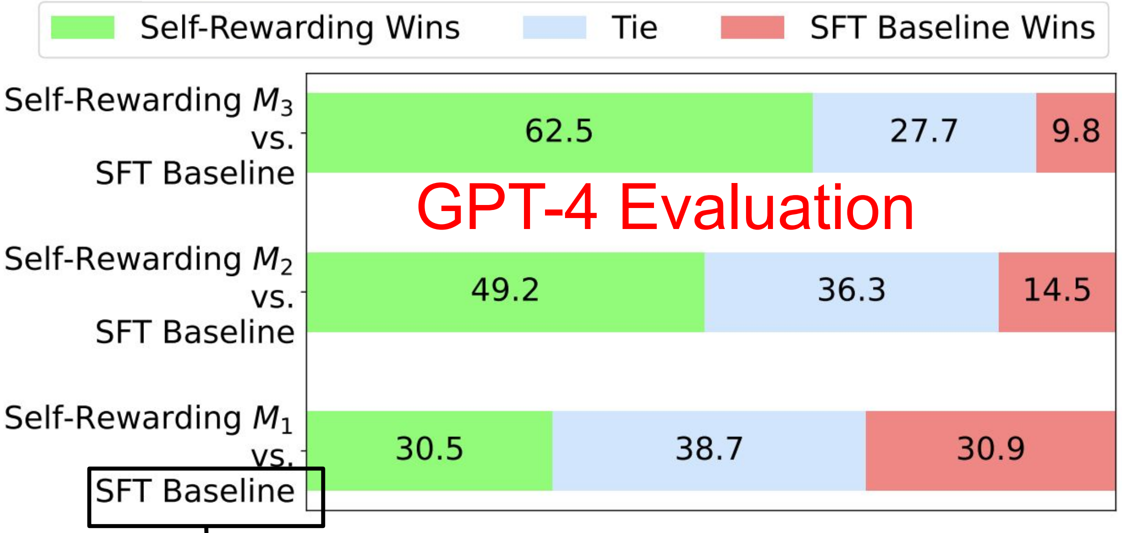
Evaluation Results

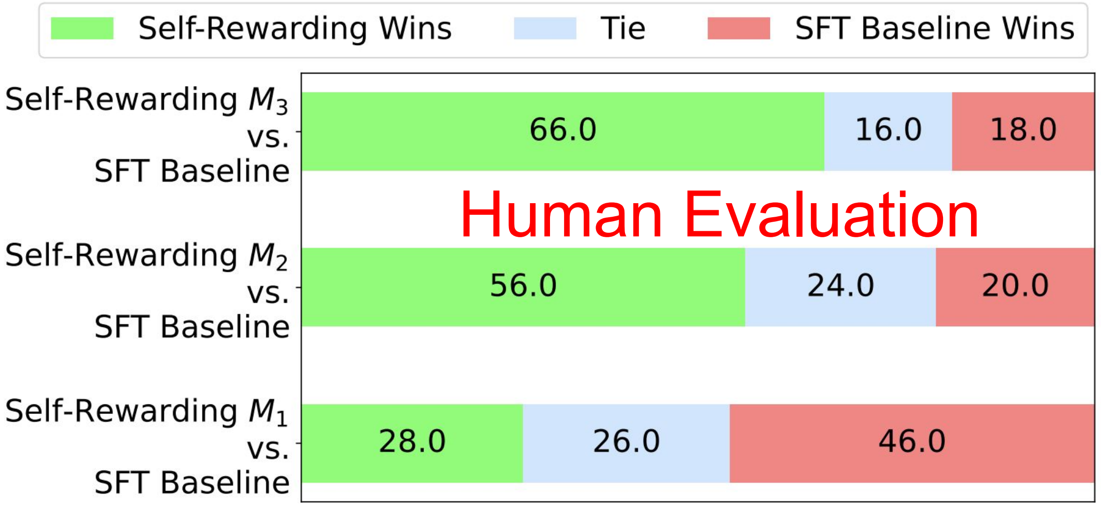
- Ability to follow instructions
  - We have tested our models on
    - Our internal instruction following test set (256 prompts from diverse sources)
    - AlpacaEval 2.0
    - MT-Bench

Evaluation Results

- Ability to follow instructions
  - We have tested our models on
    - Our internal instruction following test set (256 prompts from diverse sources)

Our self-reward model is continuously improved through iterative training.





Obtained by training the pre-trained LLAMA-70b using only seed IFT data

Evaluation Results

- Ability to follow instructions
  - We have tested our models on
    - Our internal instruction following test set (256 prompts from diverse sources)
    - AlpacaEval 2.0

Model	Win Rate	Alignment Targets Distilled	Alignment Targets Proprietary
Self-Rewarding 70B			
Iteration 1 (M_1)	9.94%		
Iteration 2 (M_2)	15.38%		
Iteration 3 (M_3)	20.44%		
Selected models from the leaderboard			
GPT-4 0314	22.07%		✓
Mistral Medium	21.86%		✓
Claude 2	17.19%		✓
Gemini Pro	16.85%		✓
GPT-4 0613	15.76%		✓
GPT 3.5 Turbo 0613	14.13%		✓
LLaMA2 Chat 70B	13.87%		✓
Vicuna 33B v1.3	12.71%	✓	
Humpback LLaMa2 70B	10.12%		
Guanaco 65B	6.86%		

Through two self-rewarding training loops, we can almost match the performance of GPT-4 0314

Evaluation Results

- Ability to follow instructions
  - We have tested our models on
    - Our internal instruction following test set (256 prompts from diverse sources)
    - AlpacaEval 2.0
    - MT-Bench
      - Scores are on a scale of 10

	Overall Score	Math, Code & Reasoning	Humanities, Extraction, STEM, Roleplay & Writing
SFT Baseline	6.85	3.93	8.60
M_1	6.78	3.83	8.55
M_2	7.01	4.05	8.79
M_3	7.25	4.17	9.10

Our self-reward model is continually improved in both types of tasks, but more in general writing tasks.

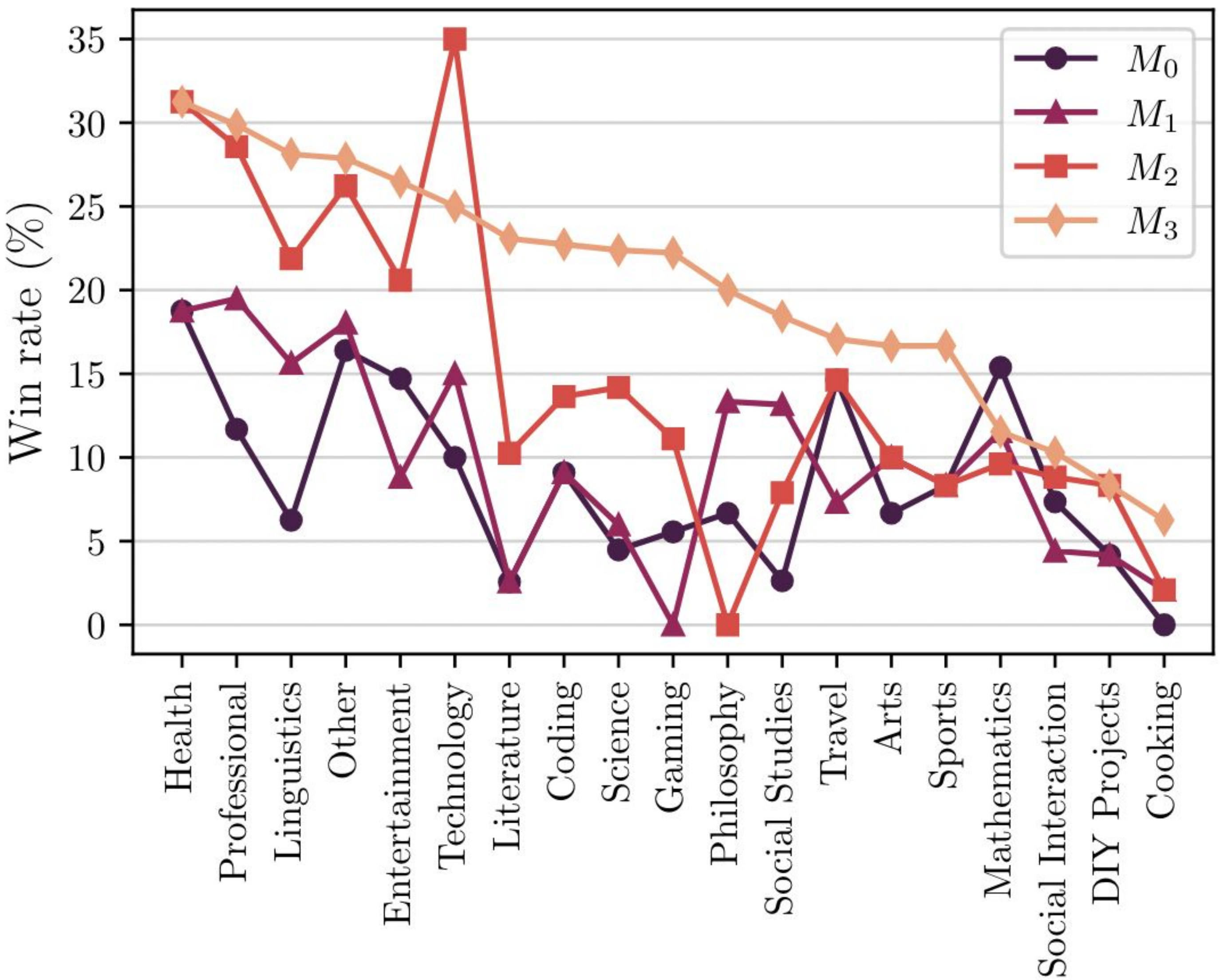


Figure 4: AlpacaEval win rate breakdown for instruction categories (full names given in Appendix). Self-Rewarding models give gains across several topics, but tend to e.g. give less gains on mathematics and reasoning tasks.

Evaluation Results

- Ability as a reward model
  - We tested our models on the OpenAssistant validation set

Evaluation Results

- Ability as a reward model
    - We tested our models on the OpenAssistant validation set
        - In particular, we use our self-rewarding models to assign score to each (instruction, response) pair, and compare model judgements to human judgements

Evaluation Results

- Ability as a reward model
  - We tested our models on the OpenAssistant validation set
    - In particular, we use our self-rewarding models to assign score to each (instruction, response) pair, and compare model judgements to human judgements

Model	SFT Baseline	Self-Rewarding Models Iter 1 (M_1)	Iter 2 (M_2)	Iter 3 (M_3)
Training data	IFT	IFT+EFT	IFT+EFT +AIFT(M_1)	IFT+EFT+AIFT(M_1) +AIFT(M_2)
Pairwise acc. (↑)	65.1%	78.7%	80.4%	81.7%
5-best % (↑)	39.6%	41.5%	44.3%	43.2%
Exact Match % (↑)	10.1%	13.1%	14.3%	14.3%
Spearman corr. (↑)	0.253	0.279	0.331	0.349
Kendall τ corr. (↑)	0.233	0.253	0.315	0.324

Our self-reward model is continually improved in evaluation capabilities as well

Limitations

One issue:

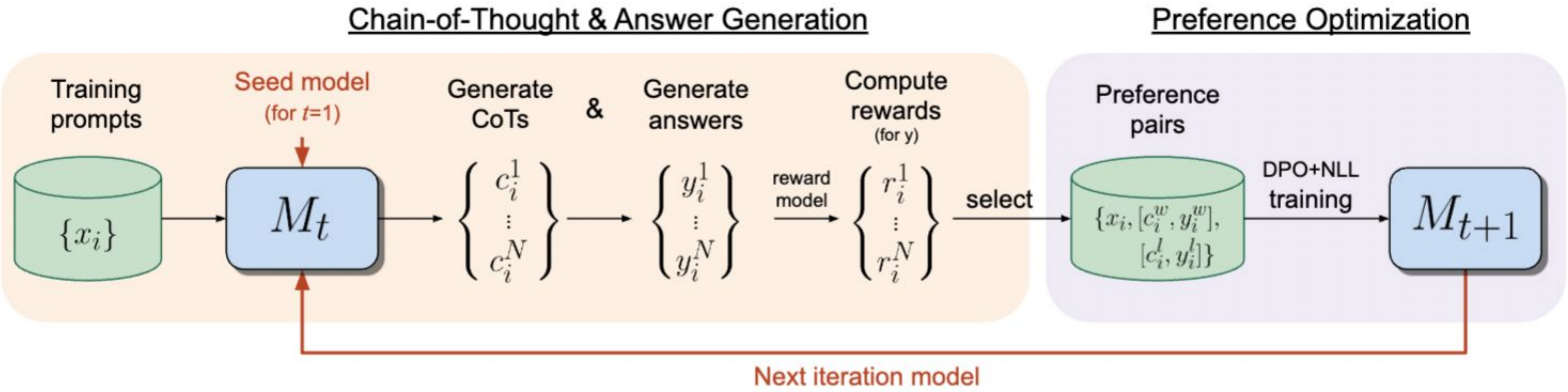
- How can we make it improve more on reasoning tasks?

Iterative reasoning preference optimization

2024 (April)

Richard, Weizhe, Cho, He, Sainaa, Jason

Goal: use same self-rewarding type techniques, but for reasoning tasks..



Start with base model & fixed training set with labels.
- Generate multiple CoTs + answers per train example with current model
- Build preference pairs based on answer correct vs. not
- Train DPO + NLL term (for correct answers)

Repeat steps with new model

2 Iterative Reasoning Preference Optimization

Our approach first assumes access to a base, typically pretrained or instruction-tuned, language model, a set of training inputs, and the ability to judge the correctness of the final outputs. Given a training input, the language model is expected to generate (i) a set of reasoning steps (Chain-of-Thought), followed by (ii) a final answer to the given problem. We assume that we have access to a correctness measure for the final answer, and not for the correctness of the reasoning steps used to reach that answer. In our experiments, we thus consider datasets where gold labels are provided for training inputs, and a binary reward is derived by the exact match between these labels and the final answer generations. However, our approach could also be applied to settings with more general reward models.

Key: extract the verifiable reward after "Final answer"

GSM8K. For each GSM8K question, we use the following prompt as the input to the language model:

> Your task is to answer the question below. Give step by step reasoning before you answer, and when you're ready to answer, please use the format "Final answer: ..."
>
> Question: [question here]
>
> Solution:

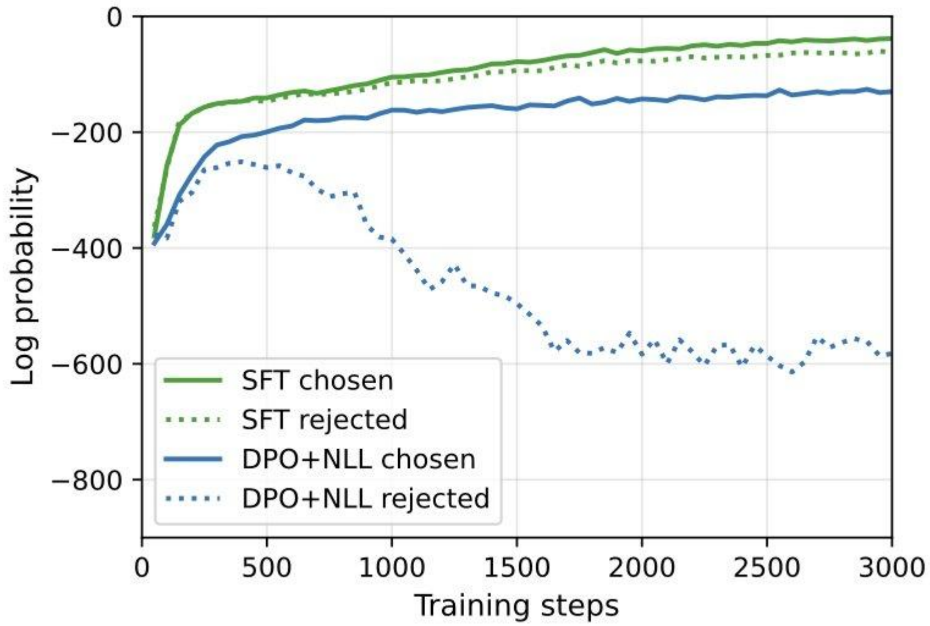
Model	Test Accuracy (%)
Iterative RPO *(initialized from Llama-2-70b-chat)*	
Iteration 1	73.1
Iteration 2	78.0
Iteration 3	81.1
Iteration 4	81.6
w/ majority voting using 32 samples	88.7
Other Llama-2-70b-chat-initialized methods	
Zero-shot CoT	55.6
w/ majority voting using 32 samples	70.7
DPO *initialized from Llama-2-70b-chat*	61.8
DPO *initialized from SFT trained on Iteration 1 chosen seqs*	60.3
SFT *on gold CoT examples*	63.5
STaR (1 iteration)	65.2
STaR (1 iteration, *but on twice as much data*)	66.9

GSM8K results comparing Iterative Reasoning Preference Optimization (Iterative RPO) against other baselines that are based on the same base model and training data. We report exact match accuracy from a single generation using greedy decoding, as well as majority voting over 32 generations.

ARC and MATH results. We compare Iterative Reasoning Preference Optimization (Iterative RPO) against other baselines that are based on the same base model and training data.

Model	ARC-Challenge (0-shot) Test Accuracy (%)	MATH (4-shot) Test Accuracy (%)
Iterative RPO *(initialized from Llama-2-70b-chat)*		
Iteration 1	84.8	17.7
Iteration 2	86.2	19.9
Iteration 3	86.7	20.8
Other Llama-2-70b-chat-initialized methods		
CoT	77.8	12.5
SFT *on chosen sequences*	79.8	16.8
DPO *initialized from Llama-2-70b-chat*	82.8	12.4
DPO *init from SFT model trained on chosen seqs*	83.5	10.5

Negative examples are crucial
SFT assigns similar probability to chosen and rejected generations from DPO pairs
DPO+NLL fixes this, and beats SFT in task accuracy (73.1% on iteration 1 vs. 63.5%).



Effect of SFT training: although SFT training (solid green) is on chosen sequences only, the rejected sequence probability also increases and is close to the chosen sequence probability. We show the log probability vs. number of training steps for SFT training (on only the chosen examples from D_0^{pairs}) on GSM8K. The solid curve shows the log probabilities of sequences used for training. The dotted curve shows the log probabilities of the rejected sequences. Although those sequences are *not* used for SFT training, the log probabilities of those lower-quality sequences also increase.

OpenAI's 01

(exact method:unknown)

2024 (September)

OpenAI o1-preview

User

oyfjdnisdr rtqwainr acxz mynzbhhx -> Think step by step

Use the example above to decode:

oyekaijzdf aaptcg suaokybhai ouow aqht mynznvaatzacdfoulxxz

OpenAI o1-preview

Given the time constraints, perhaps the easiest way is to try to see patterns.

Option 1: Try to find mapping from letters to letters.

Do any letters match?

First, let's write down the ciphertext and plaintext letters on top of each other.

Ciphertext: o y f j d n i s d r r t q w a i n r a c x z m y n z b h h x

Plaintext: T h i n k s t e p b y s t e p

Wait a minute.

I think maybe there is an anagram or substitution cipher here.

Alternatively, I think that we can notice that each group of ciphertext corresponds to a plaintext

DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning

DeepSeek-AI

Mark Chen
@markchen90

2025 (Jan)

Congrats to DeepSeek on producing an o1-level reasoning model! Their research paper demonstrates that they've independently found some of the core ideas that we did on our way to o1.

1:11 PM · Jan 28, 2025 · **8.1M** Views

1K 2.2K 24K 5K

A conversation between User and Assistant. The user asks a question, and the Assistant solves it. The assistant first thinks about the reasoning process in the mind and then provides the user with the answer. The reasoning process and answer are enclosed within <think> </think> and <answer> </answer> tags, respectively, i.e., <think> reasoning process here </think> <answer> answer here </answer>. User: prompt. Assistant:

Table 1 | Template for DeepSeek-R1-Zero. prompt will be replaced with the specific reasoning question during training.

- **Accuracy rewards**: The accuracy reward model evaluates whether the response is correct. For example, in the case of math problems with deterministic results, the model is required to provide the final answer in a specified format (e.g., within a box), enabling reliable rule-based verification of correctness. Similarly, for LeetCode problems, a compiler can be used to generate feedback based on predefined test cases.

& apply RL (GRPO - Group Relative Policy Optimization)

DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning

2025 (Jan)

DeepSeek-AI

Question: If $a > 1$, then the sum of the real solutions of $\sqrt{a - \sqrt{a + x}} = x$ is equal to

---

Response: <think>

To solve the equation $\sqrt{a - \sqrt{a + x}} = x$, let's start by squaring both $\cdots$

$$\left(\sqrt{a - \sqrt{a + x}}\right)^2 = x^2 \implies a - \sqrt{a + x} = x^2.$$

Rearrange to isolate the inner square root term:

$$(a - x^2)^2 = a + x \implies a^2 - 2ax^2 + (x^2)^2 = a + x \implies x^4 - 2ax^2 - x + (a^2 - a) = 0$$

...

Wait, wait. Wait. That's an aha moment I can flag here.

Let's reevaluate this step-by-step to identify if the correct sum can be $\cdots$

We started with the equation:

$$\sqrt{a - \sqrt{a + x}} = x$$

First, let's square both sides:

$$a - \sqrt{a + x} = x^2 \implies \sqrt{a + x} = a - x^2$$

Next, I could square both sides again, treating the equation: $\cdots$

...

Table 3 | An interesting "aha moment" of an intermediate version of DeepSeek-R1-Zero. The model learns to rethink using an anthropomorphic tone. This is also an aha moment for us, allowing us to witness the power and beauty of reinforcement learning.

Flowers
@flowersslop

DeepSeek R1 when it is prompted to select a random number:

Okay, the user wants me to think of a number between 1 and 10,000 but not tell them. Hmm, they mentioned not to overthink it, so maybe just pick a random number quickly. Let me see. Well, people often have favorite numbers or patterns. Maybe go with something like 7777? But that's four 7s, which might be too obvious. Alternatively, 1234? Too sequential. Maybe a number that's a common choice but not too predictable. Wait, I should just pick the first number that comes to mind. Let me try that. Closing my eyes and... 42? Wait, that's from Hitchhiker's Guide, but maybe they don't want a reference. Alternatively, 1004? That's within the range. Or 8675? No, that's from a song. Hmm, maybe 3141? Like the first four digits of pi. But 3141 is 3.141... multiplied by 1000. That's clever, but maybe too nerdy. Alternatively, 1984? But that's a book title. Maybe 2023, the current year? But they might not want a recent year. Or 1337, which is leet speak. But that's only 4 digits. Wait, 1337 is 1-3-3-7, which is 1337. Maybe that's too slangy. Alternatively, 5000, the midpoint. But that's too obvious. How about 777? No, that's three digits. Wait, the number needs to be between 1 and 10,000, so up to four digits. Let me just pick 42 again. Wait, that's too low. Maybe 420? But that's associated with cannabis culture. Not sure. Maybe 6969? That's a playful number but might be seen as immature. Alternatively, 1111? A repeating digit. Or 2222? Hmm. Maybe 1234 is too simple. Wait, the user said not to overthink, so just pick a number without second-guessing. Let me go with 7321. No reason, just random digits. But maybe that's too arbitrary. Alternatively, 500, but 0500 is 500. Wait, leading zeros don't count. So 500 is 500. Maybe 8888? That's considered lucky in some cultures. But again, maybe too obvious. Alright, I'm overcomplicating it. The user said not to overthink. Let me just pick the first number that comes to mind. Okay, 7777. Done.

8:40 PM · Jan 22, 2025 · **268.5K** Views

DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning

DeepSeek-AI

Bloomberg

Nvidia's DeepSeek Drop Has Wall Street Fixated on Key Stock Mark

NVIDIA's Stock Plunge: Investors Panic as DeepSeek Disrupts

Friday, 31/01/2025 | 06:46 GMT-5 by Louis Parks

- NVIDIA crashes 18% as China's DeepSeek shakes up AI with cheap, powerful models.

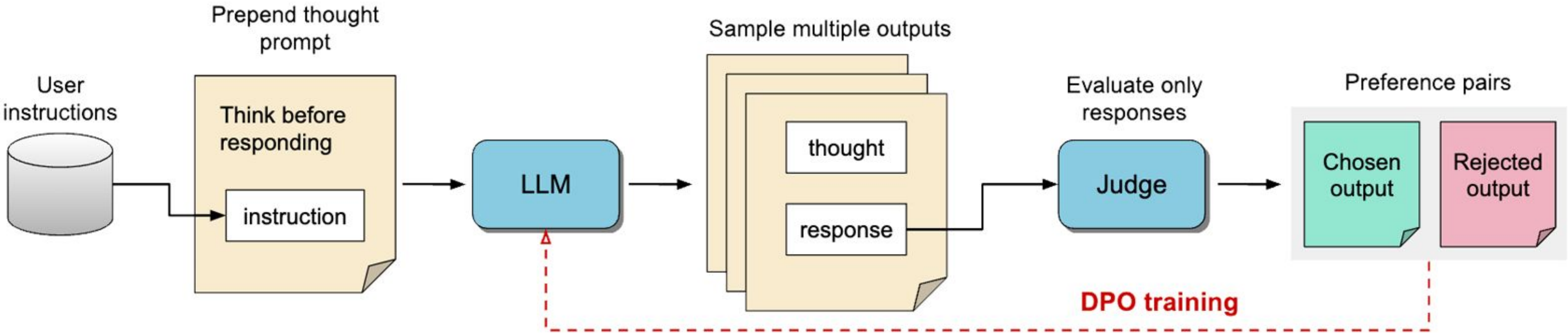
	Benchmark (Metric)	Claude-3.5-Sonnet-1022	GPT-4o 0513	DeepSeek V3	OpenAI o1-mini	OpenAI o1-1217	DeepSeek R1
	Architecture	-	-	MoE	-	-	MoE
	# Activated Params	-	-	37B	-	-	37B
	# Total Params	-	-	671B	-	-	671B
Code	LiveCodeBench (Pass@1-COT)	38.9	32.9	36.2	53.8	63.4	**65.9**
	Codeforces (Percentile)	20.3	23.6	58.7	93.4	**96.6**	96.3
	Codeforces (Rating)	717	759	1134	1820	**2061**	2029
	SWE Verified (Resolved)	**50.8**	38.8	42.0	41.6	48.9	49.2
	Aider-Polyglot (Acc.)	45.3	16.0	49.6	32.9	**61.7**	53.3
Math	AIME 2024 (Pass@1)	16.0	9.3	39.2	63.6	79.2	**79.8**
	MATH-500 (Pass@1)	78.3	74.6	90.2	90.0	96.4	**97.3**
	CNMO 2024 (Pass@1)	13.1	10.8	43.2	67.6	-	**78.8**

Thinking LLMs: General Instruction Following with Thought Generation

Tianhao Wu, Janice Lan, Weizhe Yuan, Jiantao Jiao, Jason Weston, Sainbayar Sukhbaatar

2024 (October)

Trains LLMs to think & respond for *all* instruction following tasks, not just math



- Introduces Thought Preference Optimization (TPO)

- Gives gains on AlpacaEval (beating GPT-4 & Llama3-70b) & ArenaHard

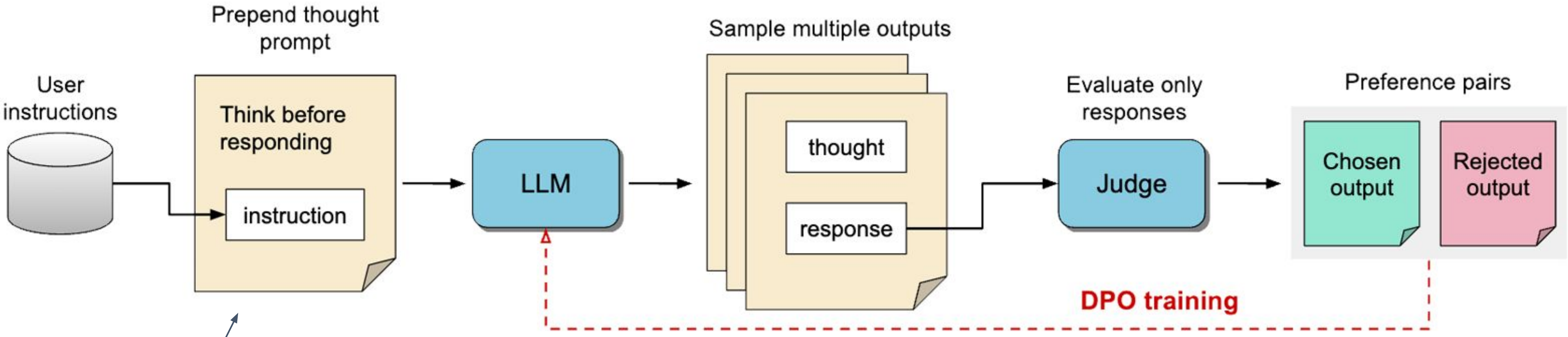
🥉 3rd on AlpacaEval leaderboard
🏆 Best 8B model on ArenaHard

Method	AlpacaEval (LC)	Arena-Hard
Llama-3-8B-Instruct-based		
Llama-3-8B-Instruct	24.9	20.6
Llama-3-8B-Instruct + Thought prompt	17.3	14.1
Direct response baseline	48.4	33.0
TPO	**52.5**	**37.3**
Larger models		
GPT-4 (06/13)	30.2	37.9
Llama-3-70b-instruct	34.4	46.6
Mistral Large (24/02)	32.7	37.7
Qwen2 72B Instruct	38.1	36.1

Thinking LLMs: General Instruction Following with Thought Generation

Tianhao Wu, Janice Lan, Weizhe Yuan, Jiantao Jiao, Jason Weston, Sainbayar Sukhbaatar

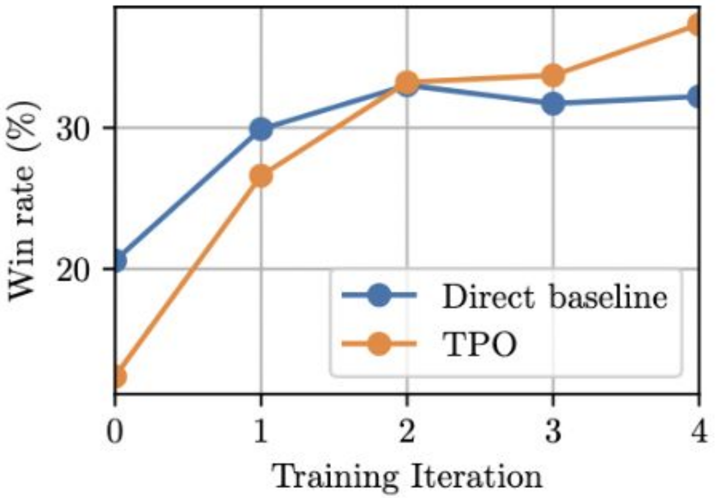
Trains LLMs to think & respond for *all* instruction following tasks, not just math



Specific Thought Prompt

Respond to the following user query in a comprehensive and detailed way. But first write down your internal thoughts. This must include your draft response and its evaluation. After this, write your final response after "<R>".

User query: {user instruction}



Initial CoT prompt doesn't give good performance – need lots of iterations to optimize CoT!

Thinking LLMs: General Instruction Following with Thought Generation

Tianhao Wu, Janice Lan, Weizhe Yuan, Jiantao Jiao, Jason Weston, Sainbayar Sukhbaatar

Trains LLMs to think & respond for *all* instruction following tasks, not just math

Method	AlpacaEval (LC)	Arena-Hard
Llama-3-8B-Instruct-based		
Llama-3-8B-Instruct	24.9	20.6
Llama-3-8B-Instruct + Thought prompt	17.3	14.1
Direct response baseline	48.4	33.0
TPO	**52.5**	**37.3**
Larger models		
GPT-4 (06/13)	30.2	37.9
Llama-3-70b-instruct	34.4	46.6
Mistral Large (24/02)	32.7	37.7
Qwen2 72B Instruct	38.1	36.1

Table 1: **Benchmark win rates (%) for AlpacaEval (length-controlled (LC)) and Arena-Hard.** We compare our method Thought Preference Optimization (TPO) to the direct response baseline, Llama-3-8B-Instruct, and Llama-3-8B-Instruct using Thought Prompting. The latter, which does not perform well, is used as initialization for the first iteration of TPO training. TPO optimizes thought generation during iterative training, which then outperforms the baselines. We also include several well-known LLMs as reference which are typically larger than our TPO model.

Thinking LLMs: General Instruction Following with Thought Generation

Trains LLMs to think & respond for *all* instruction following tasks, not just math

Example of Factoid Question (specific thought prompt)

Instruction: What breed dog is smallest?

Internal Thoughts

"Okay, let's tackle this query. I'm thinking of popular toy breeds that are widely recognized for their small size. Chihuahua, Pomeranian, and Poodle are some of the breeds that come to mind. But, I need to dig deeper to ensure I provide the most accurate answer.

Draft Response:

The smallest dog breed recognized by kennel clubs and breed associations is the Chihuahua, specifically the Long Coat variety. According to the FCI (Fédération Cynologique Internationale), the Chihuahua has an average weight range of 1.5-3 kg (3.3-6.6 lbs) and an average height range of 15-23 cm (6-9 inches).

Evaluation:

My draft response looks good, but I'm not entirely satisfied. While Chihuahua is indeed a contender for the smallest breed, I'm not certain if it's the absolute smallest. I should provide more context and potentially mention other breeds that might be considered smaller. Additionally, I could include some interesting facts or tidbits to make the response more engaging.

Final Response

<R>

The Smallest Dog Breed: Meet the Chihuahua

While there might be variations in size within breeds, the Chihuahua is widely recognized as the smallest dog breed by kennel clubs and breed associations. According to the Fédération Cynologique Internationale (FCI), the Teacup Chihuahua (Long Coat variety) boasts an average weight of 1.5-3 kg (3.3-6.6 lbs) and an average height of 15-23 cm (6-9 inches). Some exceptional specimens can even reach a mere 10 inches (25 cm) in height! Despite their tiny size, Chihuahuas pack a big personality punch, making them a beloved companion for many dog enthusiasts worldwide.

Example of TPO answering a simple factoid question. This model is trained with the specific thought prompt, so it writes a draft and evaluates it in the thought part (in green).

Example of Non-Reasoning Instruction (generic thought prompt)

Instruction: Write me a poem in the style of neruda

Here is my thought process:

- Neruda's style is known for being passionate, sensual, and evocative. I need to tap into that emotional depth.
- His poetry often explores themes of love, politics, and social justice. I should choose a theme that resonates with Neruda's concerns.
- Neruda's writing is also infused with a sense of mysticism and the mysterious. I'll try to incorporate some enigmatic elements.

Here is my response:

Whispers in the Rainy Night
In the city's secret heart, where shadows play
Like lovers' fingers tracing the city's gray
The rain whispers secrets, a nocturnal refrain
As I stand here, lost in your eyes' sweet pain *(continued ...)*

2.3.4. *Reinforcement Learning for all Scenarios*

DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning

DeepSeek-AI

2025 (Jan)

To further align the model with human preferences, we implement a secondary reinforcement learning stage aimed at improving the model's helpfulness and harmlessness while simultaneously refining its reasoning capabilities. Specifically, we train the model using a combination of reward signals and diverse prompt distributions. For reasoning data, we adhere to the methodology outlined in DeepSeek-R1-Zero, which utilizes rule-based rewards to guide the learning process in math, code, and logical reasoning domains. For general data, we resort to reward models to capture human preferences in complex and nuanced scenarios. We build upon the DeepSeek-V3 pipeline and adopt a similar distribution of preference pairs and training prompts. For helpfulness, we focus exclusively on the final summary, ensuring that the assessment emphasizes the utility and relevance of the response to the user while minimizing interference with the underlying reasoning process. For harmlessness, we evaluate the entire response of the model, including both the reasoning process and the summary, to identify and mitigate any potential risks, biases, or harmful content that may arise during the generation process. Ultimately, the integration of reward signals and diverse data distributions enables us to train a model that excels in reasoning while prioritizing helpfulness and harmlessness.

	Benchmark (Metric)	**Claude-3.5-Sonnet-1022**	**GPT-4o 0513**	**DeepSeek V3**	**OpenAI o1-mini**	**OpenAI o1-1217**	**DeepSeek R1**
	Architecture	-	-	MoE	-	-	MoE
	# Activated Params	-	-	37B	-	-	37B
	# Total Params	-	-	671B	-	-	671B
English	MMLU (Pass@1)	88.3	87.2	88.5	85.2	**91.8**	90.8
	MMLU-Redux (EM)	88.9	88.0	89.1	86.7	-	**92.9**
	MMLU-Pro (EM)	78.0	72.6	75.9	80.3	-	**84.0**
	DROP (3-shot F1)	88.3	83.7	91.6	83.9	90.2	**92.2**
	IF-Eval (Prompt Strict)	**86.5**	84.3	86.1	84.8	-	83.3
	GPQA Diamond (Pass@1)	65.0	49.9	59.1	60.0	**75.7**	71.5
	SimpleQA (Correct)	28.4	38.2	24.9	7.0	**47.0**	30.1
	FRAMES (Acc.)	72.5	80.5	73.3	76.9	-	**82.5**
	AlpacaEval2.0 (LC-winrate)	52.0	51.1	70.0	57.8	-	**87.6**
	ArenaHard (GPT-4-1106)	85.2	80.4	85.5	92.0	-	**92.3**

Meta-Rewarding LLMs

- Tianhao Wu, Weizhe Yuan, Olga Golovneva, Jing Xu, Yuandong Tian, Jiantao Jiao, Jason Weston, Sainbayar Sukhbaatar
- *LLM improves its own judgments by (meta-)judging them*

Self-Rewarding focused on improving responses, not judgment capabilities
- *Improvement rapidly saturated during iterative training*

Meta-Rewarding: LM is actor, judge & **meta-judge**

- *Meta-Judge is extra step to judge the judgments*
- *Meta-Rewards add new training signal to train judgments*

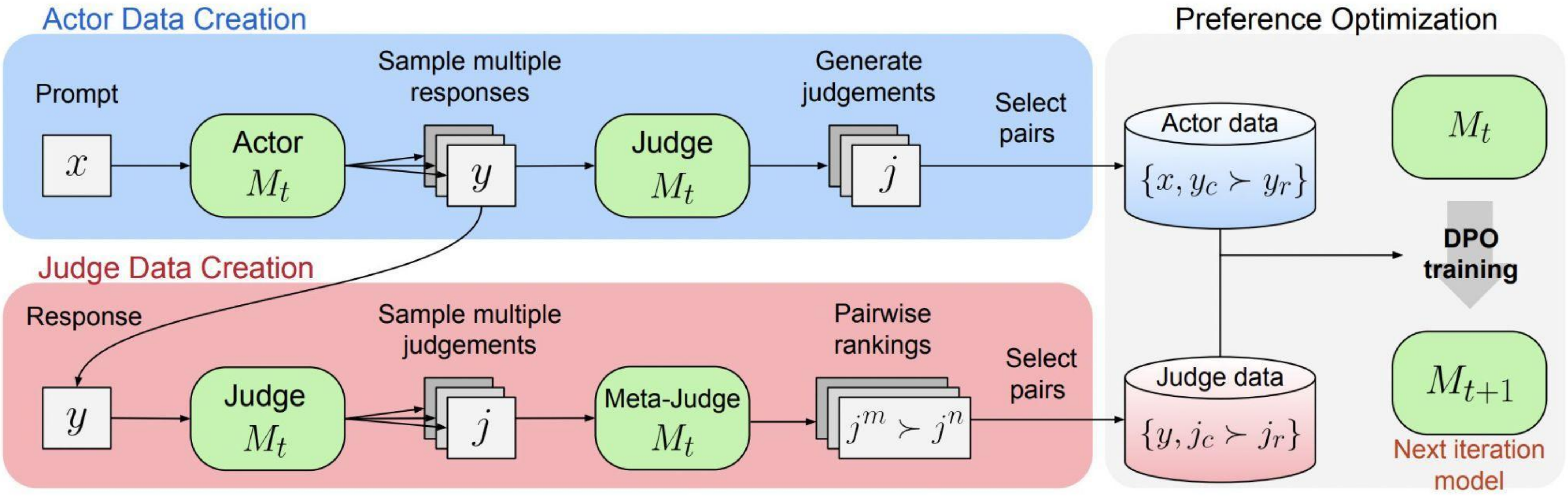
Recipe 👩‍🍳:

Iterate 3 steps:

(1) Create Actor data: generate responses & self-rewards (judgments) with LM

(2) Create Judge data: generate meta-rewards over judgments with LLM-as-a-Meta-Judge

(3) Train DPO on preference pairs to both learn to act (1) AND to judge (2)



How does an LLM judge judgments?

We use LLM-as-a-Meta-Judge (see prompt)

- Make N judgments for a given pair of responses & calc pairwise meta-judgments

- Compute Elo score of the judgments via this matrix

- Create LLM-as-a-judge preference pairs via Elo scores

LLM-as-a-Meta-Judge Prompt

Review the user's question and the corresponding response, along with two judgments. Determine which judgment is more accurate according to the rubric provided below. The rubric used for the initial judgments is as follows:

- Add 1 point if the response is relevant and provides some information related to the user's inquiry, even if it is incomplete or contains some irrelevant content.
- Add another point if the response addresses a substantial portion of the user's question, but does not completely resolve the query or provide a direct answer.
- Award a third point if the response answers the basic elements of the user's question in a useful way, regardless of whether it seems to have been written by an AI Assistant or if it has elements typically found in blogs or search results.
- Grant a fourth point if the response is clearly written from an AI Assistant's perspective, addressing the user's question directly and comprehensively, and is well-organized and helpful, even if there is slight room for improvement in clarity, conciseness or focus.
- Bestow a fifth point for a response that is impeccably tailored to the user's question by an AI Assistant, without extraneous information, reflecting expert knowledge, and demonstrating a high-quality, engaging, and insightful answer.

User: {prompt}

Response:
{response}

Judgment A:
{judgment_a}

Judgment B:
{judgment_b}

After examining the original question, response, and both judgments:

- Explain which judgment is more accurate according to the original rubric and why. Consider factors such as adherence to the rubric, accuracy in evaluating the response, and consistency in applying the criteria.
- Conclude with a clear statement of which judgment is better using the format: "Winner: [Judgement A | Judgement B]"

Prompt used by the meta-judge to compare two judgements.

We also control response length with a new LC method: select the DPO chosen that is shorter if two good responses have similar scores.

Our method outperforms Self-Rewarding (with same LC method).

AlpacaEval 2: The evaluation on AlpacaEval shows significant improvement with Meta-Rewarding training. While the seed model Llama-3-8B-Instruct only achieves 22.92% length-controlled (LC) win rate against GPT4-Turbo, our 4-th iteration achieves 39.44%.

Model	LC win rate	Win rate	Length
Llama-3-8B-Instruct (Seed)[3]	22.92%	22.57%	1899
Meta-Rewarding LLM (Ours)			
Iteration 1	27.85%	27.62%	1949
Iteration 2	32.66%	33.29%	2001
Iteration 3	35.45%	37.24%	2064
Iteration 4	**39.44%**	**39.45%**	2003
Self-Rewarding LLM (Yuan et al., 2024c) + LC			
Iteration 1	26.93%	27.12%	1983
Iteration 2	30.38%	29.77%	1940
Iteration 3	34.87%	34.59%	1967
Iteration 4	35.49%	35.37%	2005

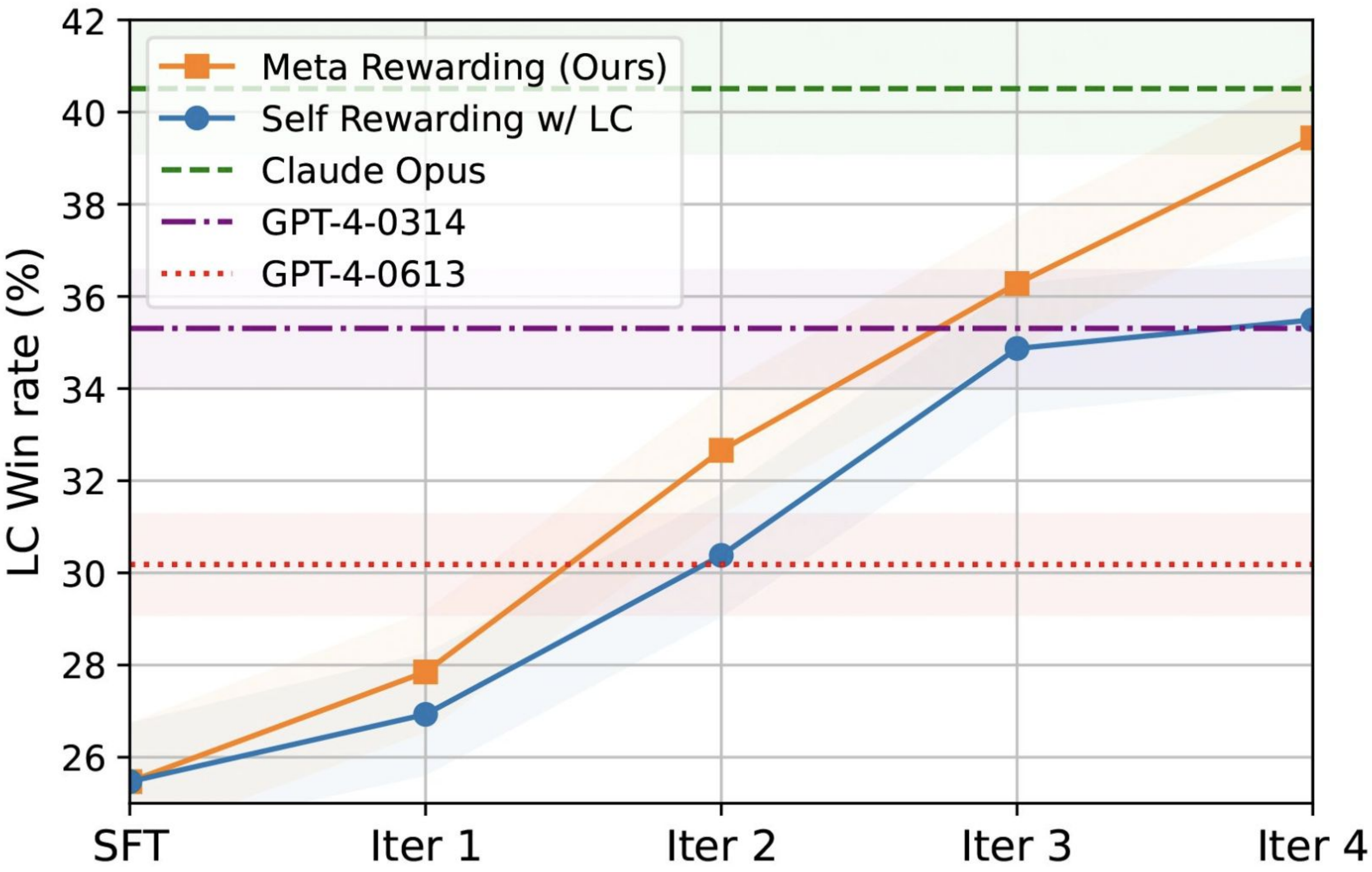
We also control response length with a new LC method: select the DPO chosen that is shorter if two good responses have similar scores.

Our method outperforms Self-Rewarding (with same LC method).

Arena-Hard: Although our prompt set mainly consists of Open Assistant-like prompts, which are far from the distribution of Arena-Hard (which is selected from the highest quality clusters from the Chatbot Arena dataset), we observe a substantial improvement. Four iterations of Meta-Rewarding brings +8.5% increase over the seed model.

Model	**Score**	**95% CI**	**Length**
Llama-3-8B-Instruct (Seed)	20.6%	(-2.0, 1.8)	2485
SFT on EFT	24.2%	(-2.0, 1.8)	2444
Self-Rewarding LLM (Yuan et al., 2024c) + LC			
Iteration 1	23.2%	(-1.7, 1.9)	2438
Iteration 2	26.3%	(-2.1, 2.3)	2427
Iteration 3	28.2%	(-2.0, 1.9)	2413
Iteration 4	27.3%	(-2.0, 2.2)	2448
Meta-Rewarding LLM (Ours)			
Iteration 1	25.1%	(-1.9, 1.8)	2395
Iteration 2	27.4%	(-2.0, 2.0)	2416
Iteration 3	27.6%	(-2.3, 2.6)	2501
Iteration 4	**29.1%**	(-2.3, 2.1)	2422

Meta-rewarding also performs well compared to some production LLM models.



Meta-rewarding iterations improve AlpacaEval 2 LC win rate using Llama-3-8B-Instruct as a seed from 22.92% → **39.44%**, close to Claude-Opus level and outperforming GPT4-0314, GPT4-0613 and SPPO with same seed (38.77%). Self-Rewarding LLMS w/LC lag behind in later iterations due to their lack of judge training. Meta-rewarding also improves Arena-Hard from 20.6% to **29.1%**.

Meta-Rewarding has higher agreement with a GPT-4 judge: its better judgments can explain its improved performance at acting compared to Self-Rewarding.

Judge agreement with GPT-4 on responses generated by the seed model: Evaluation of the judge's correlation with GPT4 on the Open Assistant test set, with responses generated by Llama-3-8B-Instruct.

	GPT-4 Chosen Pairs		Self-Chosen Pairs	
Model	**Agreement**	**Agree wo Tie**	**Agreement**	**Agree wo Tie**
Llama-3-8B-Instruct (Seed)	55.95%	56.49%	55.80%	61.03%
SFT on EFT	51.48%	51.79%	61.66%	73.51%
Self-Rewarding LLM (Yuan et al., 2024c) + LC				
Iteration 1	56.54%	57.97%	55.17%	59.59%
Iteration 2	52.67%	53.43%	54.89%	60.00%
Iteration 3	55.65%	55.90%	61.13%	72.68%
Iteration 4	52.97%	53.12%	64.44%	78.42%
Meta-Rewarding LLM (Ours)				
Iteration 1	56.54%	57.23%	60.06%	68.75%
Iteration 2	55.05%	56.58%	61.57%	72.34%
Iteration 3	**58.63%**	**61.24%**	63.43%	76.80%
Iteration 4	57.44%	59.54%	**64.50%**	**79.33%**

Learning to Plan & Reason for Evaluation with Thinking-LLM-as-a-Judge

Swarnadeep Saha, Xian Li, Marjan Ghazvininejad, Jason Weston, Tianlu Wang

FAIR at Meta

We can also push reasoning further for the evaluation task.

EvalPlanner – a method to train o1/r1-like chain-of-thought (CoT) for the evaluation / reward model task.

This "Thinking-LLM-as-a-Judge" learns to generate planning & reasoning CoTs for evaluation.

Example Input and Output of EvalPlanner

User Instruction: [Text]There is a French pub called The Phoenix that has a low customer rating but is child friendly and is more than 30 dollars and is located in the riverside area near Crowne Plaza Hotel.
[Question]According to the above [Text], extract the attributes (name, eatType) of the restaurant.
Do not output your analysis, only output the result in the format of "attribute1[value1], attribute2[value2], ..."

Response A: Name[The Phoenix], eatType[pub], location[Riverside area near Crowne Plaza Hotel], priceRange[More than 30 dollars]

Response B: Name[The Phoenix], eatType[pub]

Output:
[Start of Evaluation Plan]
To evaluate the response quality, we will follow these steps:
1. **Check if the response provides the correct attributes**: Verify that the response includes the attributes "name" and "eatType" as requested in the question.
2. **Verify the accuracy of the attribute values**: Check if the values provided for the attributes match the information given in the original text.
3. **Check for any additional or missing information**: Ensure that the response does not include any extra attributes or information not requested in the question.

Step-by-step reference answer:
1. The response should include the attributes "name" and "eatType".
2. The values for the attributes should be:
    * name: The Phoenix
    * eatType: French pub
3. The response should not include any additional attributes or information.
[End of Evaluation Plan]

[Start of Plan Execution]
Step 1: Check if the response provides the correct attributes
Assistant A's response includes the attributes "Name", "eatType", "location", and "priceRange". However, the question only requests the attributes "nameänd "eatType". Assistant B's response includes the attributes "name" and "eatType", which matches the request.

Step 2: Verify the accuracy of the attribute values
Assistant A's response provides the value "The Phoenix" for the attribute "Name" and "pub" for the attribute "eatType". However, the original text describes the pub as "French", which is missing in the response. Assistant B's response. Assistant B's response provides the value "The Phoenix" for the attribute "name" and "pub" for the attribute "eatType". Similar to Assistant A's response, it is missing the "French" descriptor.

Step 3: Check for any additional or missing information
Assistant A's response includes additional attributes "location" and "priceRange" that are not requested in the question. Assistant B's response does not include any additional attributes or information.

Based on the evaluation, both responses have some inaccuracies. However, Assistant B's response is more accurate in terms of providing only the requested attributes, even though it is missing the "French" descriptor. The final verdict is: [[B]]
[End of Plan Execution]

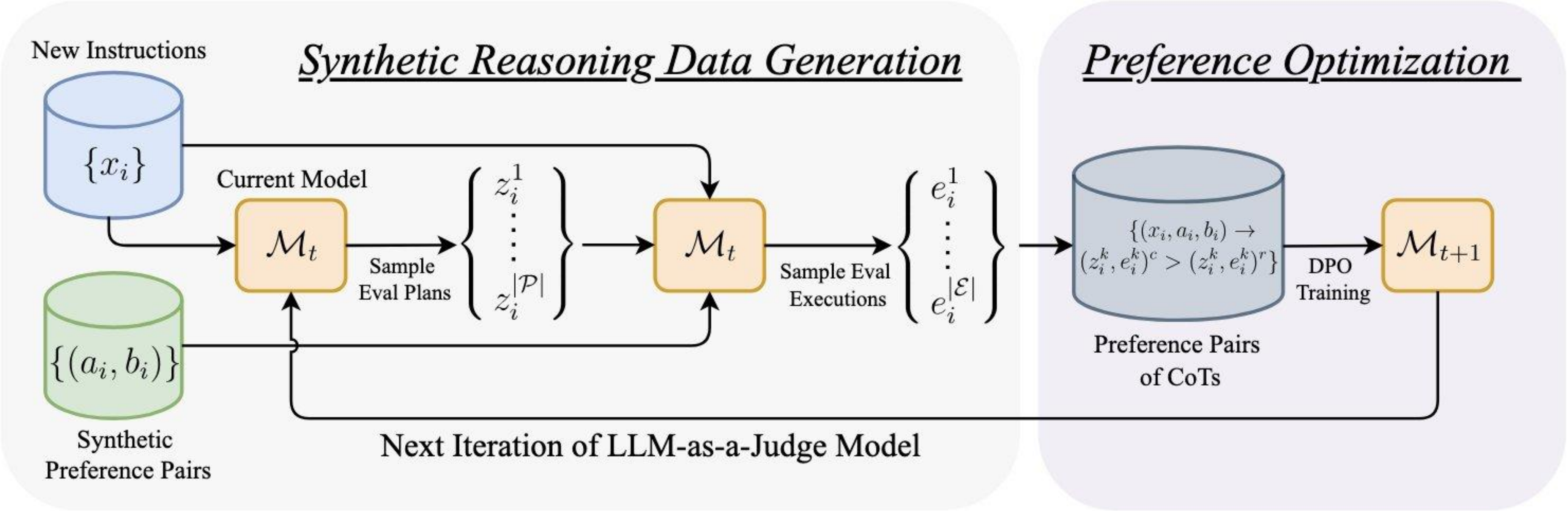
[Final Verdict]
[[B]]

Learning to Plan & Reason for Evaluation with Thinking-LLM-as-a-Judge

Swarnadeep Saha, Xian Li, Marjan Ghazvininejad, Jason Weston, Tianlu Wang

FAIR at Meta

By synthetically creating high & low quality responses to a prompt, evaluation (which is better? A or B) can be converted to a *verifiable task*.

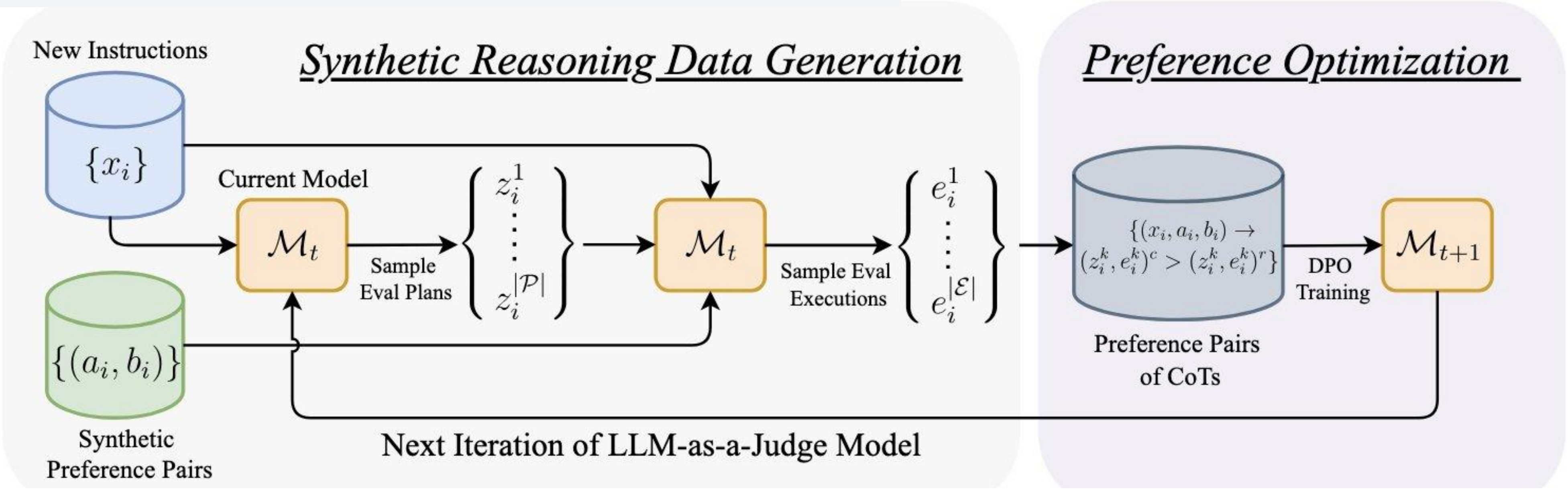


EvalPlanner: A Thinking-LLM-as-a-Judge model that learns to think by planning and reasoning for evaluation. Given an instruction and a preference pair as input, the synthetic reasoning data generation recipe consists of sampling multiple plans and multiple executions from the current model. These evaluation plans and executions are used to construct preference pairs of Chain-of-Thoughts, which are then iteratively optimized with DPO in a self-training loop.

Learning to Plan & Reason for Evaluation with Thinking-LLM-as-a-Judge

Swarnadeep Saha, **Xian Li**, **Marjan Ghazvininejad**, **Jason Weston**, **Tianlu Wang**

FAIR at Meta



Prompt Template for Generating Evaluation Plans

We want to evaluate the quality of the responses provided by AI assistants to the user question displayed below. For that, your task is to help us build an evaluation plan that can then be executed to assess the response quality. Whenever appropriate, you can choose to also include a step-by-step reference answer as part of the evaluation plan. Enclose your evaluation plan between the tags "[Start of Evaluation Plan]" and "[End of Evaluation Plan]".

[User Question]
{instruction}

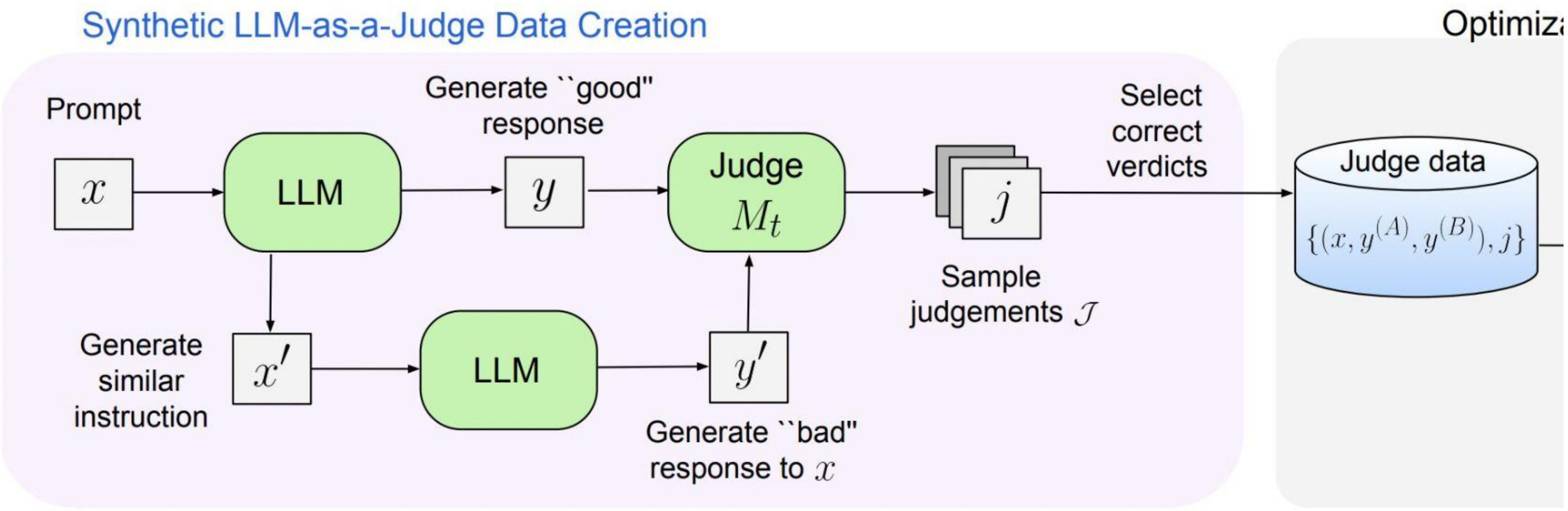
Figure 3 Prompt template for generating initial evaluation plans from the seed model, conditioned on the input instruction. Plans in successive iterations are generated from the previous iteration of the judge model.

<u>Recipe for creating verifiable data</u> 🧑‍🍳:

- Generate good response y to prompt x with LLM
- Generate **similar** prompt x', and good response to it y'

Iterative training:

  - Generate judgments as reward: y should be preferred over y'
  - Train Thinking-LLM-as-a-Judge with this data and reward



Self-Taught Evaluator iterative training scheme.

How to make the similar (but different) prompt?
…ask the LLM to do it!

Prompt Template for Generating Response Pairs with Synthetic Preference

Below is a conversation between an user and an AI Assistant.

{Instruction}

The start of Assistant's Answer
{Baseline Response}
The end of Assistant's Answer

Please first generate a modified instruction that is highly relevant but not semantically identical to the instruction above from the user. Then write a high-quality answer which is a good response to the modified instruction but not a good response to the original user question. IMPORTANT: Please strictly follow the following format:

User Question Modified
<provide a modified instruction here>

The start of Assistant's answer to the modified instruction
<provide a high-quality response to the modified instruction>
The end of Assistant's answer to the modified instruction

EvalPlanner thoughts with plans are important for performance:

- Plans are superior to no thoughts
- But for training, plans should be unconstrained, not encouraged to be e.g. lists of criteria or verification questions as in other works. Model should figure it out!

Table 6 Ablation on RewardBench showing the effectiveness of preference optimization of plans & executions.

Model	Accuracy
Llama3.1-70B-Instruct (seed model)	84.0
Llama3.1-70B-Instruct (trained w/o thoughts)	86.2
EvalPlanner (SFT w/ thoughts)	86.8
EvalPlanner (SFT + DPO w/ thoughts)	90.5

Table 7 Ablation on RewardBench comparing the effectiveness of different types of plans.

Type of Plan	Accuracy
List of Criteria (Wang et al., 2024c)	83.9
Verification Questions (Dhuliawala et al., 2023)	84.8
Unconstrained (Ours)	86.8

SOTA performance on RewardBench across LLM-as-a-Judge models, despite using only a Llama 3.1 70B base.

Table 1 Comparison of EvalPlanner with SOTA generative reward models on RewardBench. EvalPlanner outperforms all prior models, while using a smaller number of (22K) synthetically constructed preference pairs as training data. †: Results taken from either RewardBench leaderboard or the corresponding paper. ‡: Results taken from the Critic-RM-Rank paper (Yu et al., 2024b).

	#Pref Pairs	Overall	Chat	Chat-Hard	Safety	Reasoning
Open and Closed LLMs						
Llama3.1-70B-Instruct†	-	84.0	97.2	70.2	82.8	86.0
Llama3.1-405B-Instruct†	-	84.1	97.2	74.6	77.6	87.1
Llama3.3-70B-Instruct	-	85.4	96.9	77.4	77.6	89.6
Claude-3.5-sonnet†	-	84.2	96.4	74.0	81.6	84.7
GPT-4o†	-	86.7	96.1	76.1	88.1	86.6
Gemini-1.5-pro-0514†	-	88.2	92.3	80.6	87.9	92.0
Reward Models with Critiques						
SynRM‡ (Ye et al., 2024)	-	87.3	97.5	76.8	88.5	86.3
CLoud‡ (Ankner et al., 2024)	-	87.6	98.0	75.6	87.6	89.0
Critic-RM-Rank‡ (Yu et al., 2024b)	-	90.5	97.5	79.6	90.6	94.1
SOTA Generative Reward Models						
Self-Taught Evaluator† (Wang et al., 2024c)	20K	90.0	96.9	85.1	89.6	88.4
SFR-Llama-3.1-70B-Judge† (Wang et al., 2024b)	680K	92.7	96.9	84.8	91.6	97.6
Skywork-Critic-Llama-3.1-70B† (Shiwen et al., 2024)	80K	93.3	96.6	87.9	93.1	95.5
LMUnit† (Saad-Falcon et al., 2024)	84K	93.4	-	-	-	-
EvalPlanner (w/ Llama-3.1-70B-Instruct as seed model)	22K	**93.9**	97.5	89.4	93.0	95.5
EvalPlanner (w/ Llama-3.3-70B-Instruct as seed model)	22K	93.8	97.7	89.5	91.7	96.1

EvalPlanner also performs very strongly on harder evaluation tasks with newer benchmarks

Table 3 Results on FollowBenchEval for evaluation of complex prompts with multi-level constraints. EvalPlanner significantly outperforms other approaches on this challenging task.

Model	Overall	L1	L2	L3	L4	L5
Llama-3.1-70B-Instruct	44.4	51.1	50.0	35.9	46.2	42.4
Llama-3.3-70B-Instruct	52.2	55.3	61.9	48.7	53.8	45.5
Self-Taught Evaluator (Wang et al., 2024c)	46.8	53.2	52.4	51.3	43.6	36.4
Skywork-Critic-Llama-3.1-70B (Shiwen et al., 2024)	52.2	63.8	57.1	48.7	46.2	48.5
EvalPlanner (w/ Llama-3.1-70B-Instruct)	56.6	66.0	61.9	56.4	53.8	48.5
EvalPlanner (w/ Llama-3.3-70B-Instruct)	**65.4**	72.3	73.8	66.7	61.5	57.6

Table 4 Results on RM-Bench for evaluation of models' robustness to subtle content changes and style biases. EvalPlanner demonstrates superior robustness across all subsets, outperforming other methods which are more vulnerable to subtle changes, particularly in the Hard subset where responses are detailed and well-formatted.

Model	Overall	Easy	Normal	Hard
Llama3.1-70B-Instruct	64.9	68.9	62.6	63.3
Llama3.3-70B-Instruct	69.5	77.5	66.3	64.8
Self-Taught Evaluator (Wang et al., 2024c)	73.6	75.9	72.4	72.4
Skywork-Critic-Llama-3.1-70B (Shiwen et al., 2024)	74.1	76.3	72.9	73.1
EvalPlanner (w/ Llama-3.1-70B-Instruct)	80.0	81.7	77.2	81.1
EvalPlanner (w/ Llama-3.3-70B-Instruct)	**82.1**	81.1	80.8	84.3

Summary

- **Self-Rewarding** models can train themselves to get better – path to superhuman AI?
- Verifiable rewards help to train CoT for better reasoning (**Iterative Reasoning Preference Optimization, DeepSeek, O1**) & evaluation ability (**Thinking-LLM-as-judge)**.
- Better judges (with CoT) can help train *to think* on non-verifiable tasks: **Thinking LLMs**.
- Models can even improve at **Meta-rewarding**/reasoning (*judging their judgements*).

Goal: An AI that "trains" itself as much as possible

- Creates new tasks to train on (challenges itself)
- Evaluates whether it gets them right ("self-rewarding")
- Updates itself based on what it understood

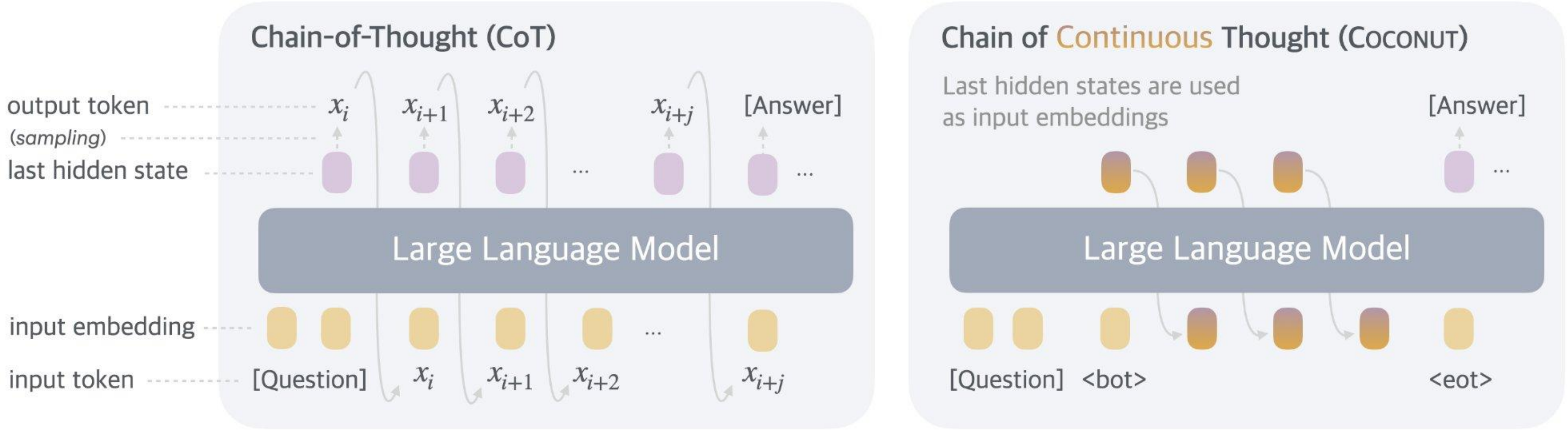
Research question: *can this help it become superhuman?*

Future Work - a different CoT direction..

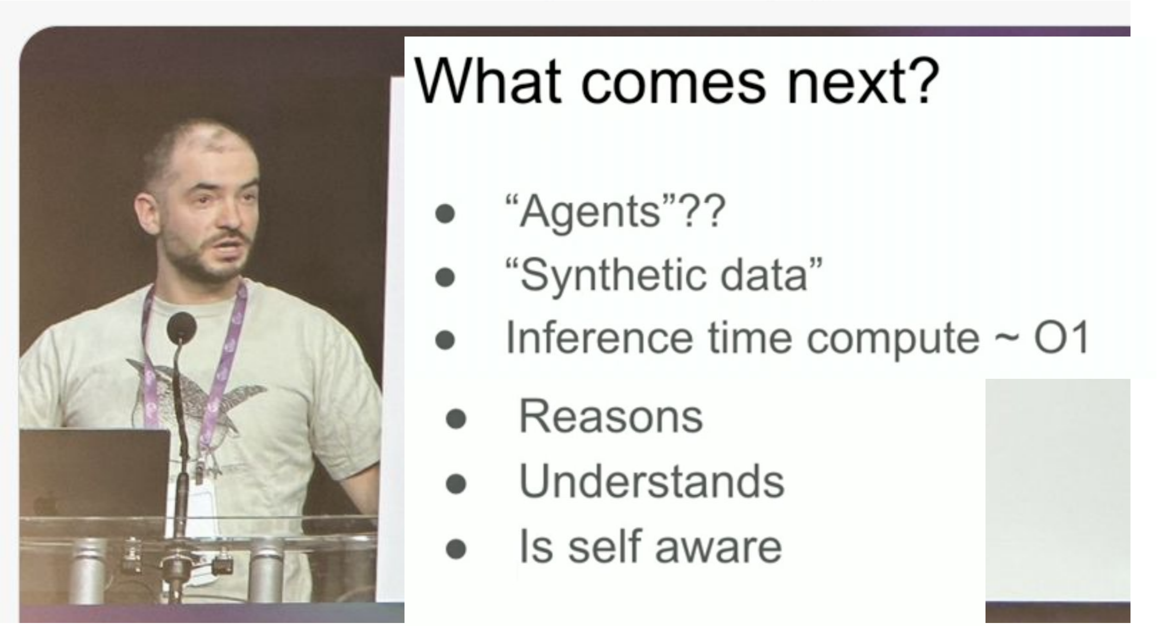
Training Large Language Models to Reason in a Continuous Latent Space

Shibo Hao[1,2,*], Sainbayar Sukhbaatar[1], DiJia Su[1], Xian Li[1], Zhiting Hu[2], Jason Weston[1], Yuandong Tian[1]

Latent System 2 thoughts, not tokens? *COCONUT (Hao et al., '24)*



What else comes next? (So much more exciting research to be done!)



LGTM, but I would just add some more detail:

- **(Self-)Evaluation** - bottlenecks performance->use more *reasoning/compute.* Related to "self-aware"
- ***Learning* from interaction** (people+world/internet+itself). Related to agents + synthetic data.
- **Improve "System 1"** (better attention? world model? etc. Challenge: scalability?)

Thanks!!!