Journal of Logic, Language and Information (2024) 33:49–68
https://doi.org/10.1007/s10849-023-09411-3

Monotonicity Reasoning in the Age of Neural Foundation Models

Zeming Chen[1] · **Qiyue Gao**[2]

Published online: 15 November 2023
© The Author(s) 2023

Abstract
The recent advance of large language models (LLMs) demonstrates that these large-scale foundation models achieve remarkable capabilities across a wide range of language tasks and domains. The success of the statistical learning approach challenges our understanding of traditional symbolic and logical reasoning. The first part of this paper summarizes several works concerning the progress of monotonicity reasoning through neural networks and deep learning. We demonstrate different methods for solving the monotonicity reasoning task using neural and symbolic approaches and also discuss their advantages and limitations. The second part of this paper focuses on analyzing the capability of large-scale general-purpose language models to reason with monotonicity.

Keywords Monotonicity · Natural language inference · Neural language model · Neural symbolic inference

1 Introduction

Foundation models are large-scale language models that contain a large number of parameters and are pretrained on massive amounts of text data, often on hundreds of millions or even billions of words. The pretraining and large-scale parameters allow them to generate high-quality human-like responses in a wide range of tasks and applications that NLP researchers previously thought required language understanding, such as question-answering, dialogue generation, and mathematical reasoning Bommasani et al. (2022). Recently released large language models, such as Open-AI's

✉ Zeming Chen
zeming.chen@epfl.ch

Qiyue Gao
q3gao@ucsd.edu

[1] Computer and Communication Sciences, EPFL, Lausanne, Switzerland

[2] Halıcıoğlu Data Science Institute, UC San Diego, San Diego, USA

GPT-3 (Brown et al., 2020), Google's FLAN-T5 Wei et al. (2021), and Facebook's LLaMA (Touvron et al., 2023), are some of the most well-known foundation models that are dominating the field of natural language processing. They achieve human-level performance on various language tasks and have the ability to follow human-defined instructions. However, it is still unclear whether these large foundation models have the ability to perform complex logical reasoning comparable to human skills. Thus, our motivation is to uncover the ability and limitations of neural foundation models in monotonicity reasoning and investigate how we can approach logical reasoning in the age of neural foundation models. We focus our discussion mainly on the monotonicity-based Natural Language Inference task.

Natural Language Inference (NLI), also known as recognizing textual entailment (RTE), is one of the important benchmark tasks for natural language understanding. Many other language tasks can benefit from NLI, such as question answering, text summarization, and machine reading comprehension. The goal of NLI is to determine whether a given premise **P** semantically entails a given hypothesis **H** (Dagan et al., 2013). Consider the following example:

- **P**: An Irishman won the Nobel Prize for literature.
- **H**: An Irishman won the Nobel Prize.

The hypothesis can be inferred from the premise, and therefore the premise entails the hypothesis. To arrive at a correct determination, an NLI model often needs to make different inferences, including various types of lexical and logical inferences. In this paper, we are concerned with monotonicity reasoning, a type of logical inference that is based on word or phrase replacement Hu et al. (2019). Below is an example of monotonicity reasoning:

1. (a) **All** students ↓ carry a MacBook ↑.
   (b) All students carry a laptop.
   (c) All new students carry a MacBook.
2. (a) **Not All** new students ↑ carry a laptop.
   (b) Not All students carry a laptop.

A phrase in upward entailment context (↑) can allow inference from (1a) to (1b), where a more general concept laptop replaces the more specific *MacBook*. A downward entailing phrase (↓) allows an inference from (1a) to (1c), where a more specific context *new students* replaces the word *students*. The direction of the monotonicity can be reversed by adding a downward entailing phrase like "Not"; thus, (2a) entails (2b).

In this paper, we provide an in-depth discussion on monotonicity reasoning in the age of neural foundation models in the aspects of methodology and analysis. First, we investigate whether incorporating both advanced neural network mechanisms, like attention with structural sentence knowledge based on the linguistic principle of compositionality, can achieve accurate and robust monotonicity reasoning in the form of NLI. We propose an AttentiveTreeNet that contains a Tree-LSTM encoder with an attention mechanism and a multi-hop self-attention aggregator for NLI classification. We evaluate AttentiveTreeNet on the MED Yanaka et al. (2019) benchmark and show that it significantly outperforms a high-quality foundation model BERT Devlin et al. (2019).

Next, we propose a symbolic reasoning system that performs monotonicity reasoning based on polarity marks and incorporates neural language models to handle syntactic variations in the data. Our proposed system, called NeuralLog, achieve state-of-the-art performance on the MED benchmark that significantly outperforms prior neural network models. The advantage of NeuralLog is its ability to perform step-by-step reasoning based on human-defined symbolic logic rules while resolving syntactic variations using neural language models, which makes its reasoning much more robust and generalizable than prior logic reasoning systems.

In the last part, we benchmark pretrained models fine-tuned on massive task-specific training data (with parameter sizes $\leq$ 11 billion) and large-scale language models (with parameter sizes $\geq$ 11 billion) on monotonicity reasoning through instruction-based zero-shot learning and in-context-based few-shot learning. Our objective is to assess whether these large foundation models have the ability to emulate logical reasoning since they have shown impressive performance on various linguistic tasks and applications. Our evaluation shows that current large language models still fail to perform logical reasoning well. Large language models only achieve random performance despite instructions and few-shot examples on the monotonicity test set from the CURRICULUM benchmark Chen and Gao (2022), which is a curated mixture of the MED Yanaka et al. (2019) and Semantic Fragments Richardson et al. (2019) datasets.

Overall, we show that although large-scale foundation models are dominating the field of natural language processing by mastering many tasks and applications, they still cannot emulate logical reasoning like monotonicity inference. Meanwhile, symbolic reasoning systems that incorporate neural language models can achieve state-of-the-art performance that is interpretable and robust. A subset of this work was previously published as Chen (2021) and Chen et al. (2021).

2 Attentive Tree Structured Network

2.1 Preliminaries

In this section, we propose a tree-structured long-short-term memory (LSTM) network in which the syntactic information of a sentence is encoded, and the alignment between the premise-hypothesis pair is calculated through a self-attention mechanism. A standard sequential LSTM (Wang & Jiang, 2016) network only permits sequential information propagation. However, the *linguistic principle of compositionality* states that an expression's meaning is derived from the meanings of its parts and of the way they are syntactically combined (Partee, 2007). A tree-structured LSTM network allows each LSTM unit to be able to incorporate information from multiple children's units. This takes advantage of the fact that sentences are syntactically formed bottom-up tree structures.

2.2 Method

Tree-LSTM Encoder

The main architecture builds from the Child-Sum Tree-LSTMs (Tai et al., 2015), where the computation of a hidden state is conditioned on both the current input and the hidden states of an arbitrary subset of children nodes. This property allows the recursive computation of non-leaf nodes' relation representations by composing children relations, which can be viewed as natural logic for neural models (MacCartney & Manning, 2009; Zhao et al., 2016). The computation flow in an LSTM cell is as follows:

$$
\begin{aligned}
\tilde{h} &= \Sigma_{1\leq k\leq n} h_k,\\
i &= \sigma(W^{(i)}x + U^{(i)}\tilde{h} + b^{(i)}),\\
o &= \sigma(W^{(o)}x + U^{(o)}\tilde{h} + b^{(o)}),\\
u &= \tanh(W^{(u)}x + U^{(u)}\tilde{h} + b^{(u)}),\\
f_k &= \sigma(W^{(f)}x + U^{(f)}h_k + b^{(f)}),\\
c &= i \odot u + \Sigma_{1<n} f_k \odot c_k,\\
h &= o \odot \tanh(c),
\end{aligned}
$$

where k is the number of children of the current node, and $\tilde{h}$ is the sum of the hidden states from the current node's children. The forget gate f_k controls the amount of memory being passed from the kth child. The input gate i controls the amount of internal input u being updated, and the output gate o controls the degree of exposure of the memory. The σ is the sigmoid activation function, $\odot$ is the element-wise product, and W and U are trainable weights to be learned.

Attention Mechanism

We propose incorporating the attention mechanism Zhou et al. (2016) in the LSTM network. Attention considers contextual relevance by assigning higher weights to children that are more relevant to the context. We apply a soft-attention layer, which receives a set of hidden states $\{h_1, h_2, ..., h_n\}$ and a vector representation s of a sentence computed from a layer of sequential LSTM. The attention layer assigns a weight α for each hidden state and computes the context vector g as a weighted sum:

$$
\begin{aligned}
m_k &= \tanh(W^{(m)}h_k + U^{(m)}s),\\
\alpha_k &= \frac{e^{w^\top m_k}}{\sum_{j=1}^{n} e^{w^\top m_j}},\\
g &= \sum_{1\leq k\leq n} \alpha_k h_k.
\end{aligned}
$$

The hidden state for the next cell is then computed via a transformation $\tilde{h} = \tanh(W^{(a)}g + b^{(a)})$.

Self-Attention Aggregator

We encode the premise and hypothesis using the attentive encoder, concatenate the hidden states into a pair of matrices H_p and H_h, and passed to a self-attentive aggregator. To aggregate, we first apply a multi-hop self-attention mechanism (Lin et al., 2017). Performing multiple hops of attention helps the model to get multiple attention focusing on different sentence parts since multiple components form the sentence context. Given a matrix H, we perform multiple hops of attention to compute an annotation matrix A, consisting of the weight vector from each hop. A is calculated from a 2-layer multi-layer perceptron (MLP) and a softmax function: $A = \text{softmax}(W_{s2} tanh(W_{s1} H^{\top}))$. The annotation matrix is multiplied by the hidden states H to obtain a context matrix: $M = AH$. With a pair of context matrices M_p and M_h, we compute the outputs as:

$$F_p = \tanh(M_p \times W_f), \qquad F_h = \tanh(M_h \times W_f). \tag{1}$$

To aggregate F_p and F_h, we follow a generic NLI training scheme Conneau et al. (2017) to include three matching methods: (I) concatenation, (ii) absolute distance, and (iii) element-wise product. Results from the three methods are then concatenated: $F_r = [F_p; F_h; \|F_p - F_h\|; F_p \odot F_h]$ as the factor of semantic relation between the two sentences. An MLP layer works as the classifier which predicts the label using the factor.

2.3 Evaluation

Datasets

We evaluate our proposed method on the Monotonicity Entailment Dataset (MED) Yanaka et al. (2019). MED is a high-quality benchmark that aims to examine models' ability to perform monotonicity reasoning. MED covers various linguistic phenomena such as lexical knowledge, conjunction, disjunction, conditional, and negative polarity items. The dataset contains 5382 premise-hypothesis pairs, including 1820 examples for upward inference, 3270 for downward inference, and 292 neutral examples.

Setup and Baselines

Initially, we used the HELP dataset Yanaka et al. (2019) to train our model. HELP is a dataset for learning entailment with lexical and logical phenomena. It embodies a combination of lexical and logical inferences focusing on monotonicity. Next, we trained our model with the Multi-Genre NLI Corpus (MNLI) dataset Williams et al. (2018), which covers a wide range of genres of spoken and written language. The majority of the training examples in that dataset are upward monotone. To provide more balanced training data, we combined a subset of the MNLI dataset with the HELP dataset to reduce the effect of many downward monotone examples in the HELP dataset. Due to limited computation resources at the time of training, we only randomly sampled a subset of the MNLI dataset to reduce the training time period. We call this combined training data HELP+SubMNLI. We removed the contradicting examples from the MNLI dataset since the test dataset MED, and the training dataset HELP do not contain the label **Contradiction**.

Table 1 Accuracy of our model and other state-of-art NLI models evaluated on MED

Model	Train Data	Up	Down	None	All
BiMPM (Wang et al., 2017)	SNLI	53.5	57.6	27.4	54.6
ESIM (Chen et al., 2017)	SNLI	71.1	45.2	41.8	53.8
DeComp (Parikh et al., 2016)	SNLI	66.1	42.1	**64.4**	51.4
BERT-base (Devlin et al., 2019)	MNLI	**82.7**	22.8	52.7	44.7
BERT-base (Devlin et al., 2019)	HELP+MNLI	76.0	70.3	59.9	71.6
AttnTreeNet (ours)	MNLI	54.7	60.4	37.8	58.6
AttnTreeNet (ours)	HELP	55.7	72.6	57.9	66.0
AttnTreeNet (ours)	HELP+SubMNLI	81.4	**74.5**	53.8	**75.7**

Bold indicates the highest accuracy in the table for each column

Training
To train our model, we used Stanford's pre-trained 300-D Glove 840B vectors (Pennington et al., 2014) to initialize the word embeddings. The Stanford Dependency Parser (Chen & Manning, 2014) was used to parse each sentence in the dataset. The model is trained with the Adam optimizer (Kingma & Ba, 2014), which is computationally efficient and helps a model to converge to an optimal result quickly. A standard learning rate for Adam, 0.001, is also used. Dropout with a standard rate of 0.5 is applied to the feed-forward layer in the self-attention aggregator and the classifier to reduce the over-fitting of the model. For the number of hops of self-attention, we used the default 15 hops. The metric for evaluation is accuracy based. The system is implemented using a common deep learning framework, PyTorch, and is trained on a T4 GPU for 20 epochs.

2.4 Results

2.4.1 MED Performance

Table 1 shows our method's performance compared against common NLI methods on the Monotonicity Entailment Dataset (MED). Our model achieves an overall accuracy of 75.7% and outperforms all other models, including the pre-trained language model BERT, which previously showed SOTA performance on NLI tasks. On downward monotonicity reasoning, which is more difficult than upward, our method shows significant improvement in performance over the baselines, with 4.5% higher than the BERT model. Interestingly, our model achieves better performance on downward inference even when trained with HELP or MNLI alone (compared to baselines with similar training data). This shows a structural advantage of our model architecture over the baselines. On upward monotonicity reasoning, our model is only slightly behind the BERT model (1.3% apart) but still outperforms the other baselines with a large margin (10.3% to the best non-BERT baseline). Note that augmenting HELP with a subset of MNLI improves the performance on upward monotone (+25.7%), showing that training additionally on some general NLI examples helps the model to learn the upward inference. On examples without monotonicity inference, our method does not

Table 2 This table shows the accuracy of ablation tests trained on HELP and HELP+SubMNLI and tested on MED. Three ablation tests were performed: (i) Remove self-attentive aggregator (–Self-attention), (ii) Replace Tree-LSTM with a regular sequential LSTM (–Tree-LSTM)

Model	Training Data	Upward	Downward	None	All
AttnTreeNet	HELP	55.7	72.6	57.9	66.0
–Self-attention	HELP	65.1	67.1	53.7	65.7
–Tree-LSTM	HELP	36.6	65.5	94.8	49.5
AttnTreeNet	HELP+SubMNLI	**81.4**	**74.5**	53.8	**75.7**
–Self-attention	HELP+SubMNLI	70.5	66.9	85.6	69.1
–Tree-LSTM	HELP+SubMNLI	54.7	60.4	37.8	58.6

Bold indicates the highest accuracy in the table for each column

perform as well as the examples with monotonicity. This suggests that while achieving high performance on monotonicity reasoning, our method loses some ability to reason with the general NLI problems. Overall, we show that our attentive tree-based network achieves the highest performance among the baselines on monotonicity reasoning.

2.4.2 Ablation Test

To further analyze each component's contribution to the model performance on monotonicity reasoning, we conduct several ablation tests. We first do an ablation test on the self-attentive aggregator by building the feature vector for classification right after the Tree-LSTM encoder. As Table 2 (–aggregator) shows, models trained on HELP+SubMNLI show a significant performance drop (6.6%) with a 76% drop in downward inference and a 10.9% drop in upward inference. The performance drop suggests that the self-attentive aggregator is an important component of the model for monotonicity reasoning. For the second ablation test, we replace the Tree-LSTM encoder with a standard LSTM encoder. Note that this results in a larger performance drop in upward inference (26.7%) and downward inference (14.1%). This demonstrates that replacing the Tree-LSTM with a standard one has a significant negative impact on the model's reasoning ability for monotonicity. Thus, Tree-LSTM is also a major component of our proposed model. Overall, the removal of the Tree-LSTM encoder affected the model's performance the most. Thus, we conclude that the Tree-LSTM encoder contributes the most to the model's performance on monotonicity reasoning.

3 Neural-Symbolic Reasoning

3.1 Preliminary

Evaluation results for the Attentive Tree Network show that providing and enhancing structural knowledge of sentences is an effective way to improve neural models' monotonicity reasoning ability. However, directly embedding symbolic logical information into a neural model is difficult. A better approach would be building a symbolic reasoning system incorporating neural modules into its inference process for better and more robust reasoning performance. Previously, several symbolic reasoning systems for NLI have been proposed Abzianidze (2017); Martínez-Gómez et al. (2017); Yanaka et al. (2018); Hu et al. (2020) to solve the NLI task based on symbolic rules and semantic formalism. These systems show high precision on complex inferences involving difficult linguistic phenomena and present logical and explainable reasoning processes. However, these systems show several limitations, such as lacking background knowledge and the inability to handle sentences with syntactic variations. On the other hand, new pre-trained language models are becoming more robust and accurate through improved pre-training objectives and data., enabling them to handle diverse and large test data robustly. However, several experiments show that DL models lack generalization ability, adopt fallible syntactic heuristics, and show exploitation of annotation artifacts Glockner et al. (2018); McCoy et al. (2019); Gururangan et al. (2018). We propose joining the strengths of these two types of systems into a hybrid reasoning system that can perform monotonicity reasoning.

3.2 Method

Our system contains four components: (1) a polarity annotator, (2) three sentence inference modules, and (3) a search engine. Figure 1 shows a diagram of the full system.

3.2.1 Polarity Annotator

To perform robust and accurate monotonicity reasoning, the system needs the annotations of monotonicity information on the given premises. To annotate monotonicity information, we utilize Udep2Mono Chen and Gao (2021), a polarity annotator that determines the monotonicity polarity of all constituents on a universal dependency tree. The annotator first parses the premise into a binarized universal dependency tree and then conducts polarization by recursively marking polarity on each tree node. The polarity marks include monotone (↑), antitone (↓), and no monotonicity information (=) polarities. An annotated example would be *Every*$^{\uparrow}$ *healthy* $^{\downarrow}$ *person* $^{\downarrow}$ *plays*$^{\uparrow}$ *sports*$^{\uparrow}$. Where the monotone tokens are tagged with $^{\uparrow}$ and antitone tokens are tagged with $^{\downarrow}$.

3.2.2 Search Engine

Next, the polarized parse tree is passed to the search engine. A beam search algorithm searches for the optimal inference path from a premise to a hypothesis. During an inference step, we rank the generated sentences with a distance function and select the sentence with the minimum distance to proceed:

$$\mathrm{s}^{\star} = \underset{\mathrm{s}\in\mathcal{S}}{\arg\min}\, \mathrm{dist}(\mathrm{s}, \mathrm{H}), \tag{2}$$

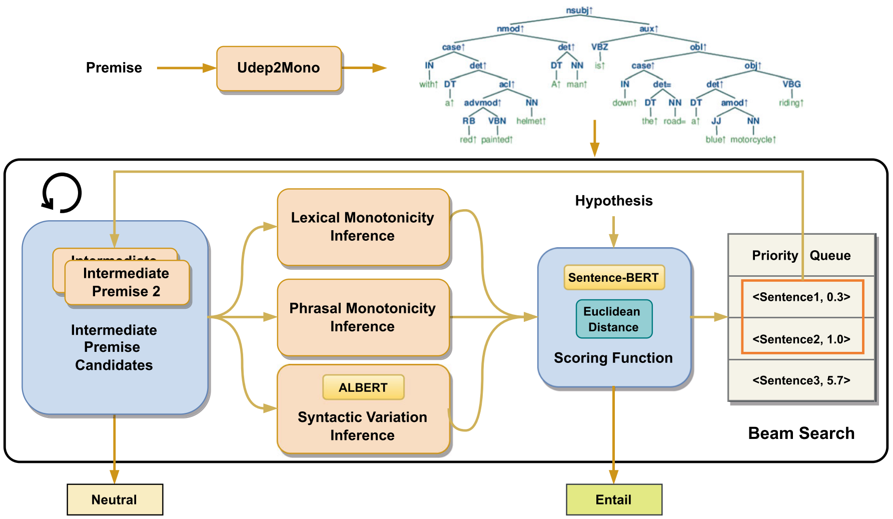
Premise
Udep2Mono
nsubj↑
nmod↑ aux↑
case↑ det↑ VBZ obl↑
IN det↑ DT NN is↑ case↑ obj↑
with↑ DT acl↑ A↑ man↑ IN det= det↑ VBG
a↑ advmod↑ NN down↑ DT NN DT amod↑ riding↑
RB VBN helmet↑ the↑ road= a↑ JJ NN
red↑ painted↑ blue↑ motorcycle↑
Intermediate
Intermediate Premise 2
Intermediate Premise Candidates
Lexical Monotonicity Inference
Phrasal Monotonicity Inference
ALBERT
Syntactic Variation Inference
Hypothesis
Sentence-BERT
Euclidean Distance
Scoring Function
Priority Queue
<Sentence1, 0.3>
<Sentence2, 1.0>
<Sentence3, 5.7>
Beam Search
Neutral
Entail

Fig. 1 Overview system diagram of NeuralLog, including (1) the polarity annotator, (2) the three inference modules, and (3) the beam search engine

where H is the hypothesis, $\mathcal{S}$ is a set of intermediate premises generated from the three inference modules, and s is the optimal intermediate premise to continue the search that yields the minimal distance to the hypothesis. Here we formulate the distance function as the Euclidean Distance between the sentence embeddings of an intermediate premise and the hypothesis. The search space is generated from three inference modules: lexical, phrasal, and syntactic variation. In practice, we expand our search space on the top-k intermediate premises instead of the optimal ones. The system returns **Entail** if an inference path is found. Otherwise, the premise and hypothesis would be categorized as **Non-Entail**, where the controller will further search for counter-example signatures to differentiate between **Contradict** and **Neutral**. In this paper, we only analyze the system's performance on the MED dataset (2-way classification: Entail and Non-Entail) and hence omit the details on how the system detects contradiction signatures.

3.2.3 Inference Generation

Lexical Monotonicity Inference
Lexical inference module performs word replacement on key tokens, including nouns, verbs, numbers, and quantifiers, based on monotonicity information. The system uses lexical knowledge bases, including WordNet Miller (1995) and ConceptNet Liu and Singh (2004). From the knowledge bases, we extract four sets of words: hypernyms, hyponyms, synonyms, and antonyms. Logically, if a word has a monotone polarity (↑), it can be replaced by its hypernyms. For example, *swim* $\leq$ *move*; then *swim* can be replaced with *move*, where $\leq$ means that the left-hand-side word is a type of the right-hand-side word. If a word has an antitone polarity (↓), it can be replaced by its hyponyms. For example, *flower* $\geq$ *rose*. Then, *flower* can be replaced with *rose*,

where $\geq$ means that the right-hand-side word is a type of the left-hand-side word. We filter out irrelevant words from the knowledge bases that do not appear in the hypothesis. Additionally, we handcraft knowledge relations for words like quantifiers and prepositions that do not have sufficient taxonomies from knowledge bases. Some handcrafted relations that hold in general include: *all* = *every* = *each* $\leq$ *most* $\leq$ *many* $\leq$ *several* $\leq$ *some* = *a*, *up* $\perp$ *down*, where = means that the two words are equivalent relations.

Phrasal Monotonicity Inference

Phrasal replacements are for phrase-level monotonicity inference. For example, with a polarized sentence $A^{\uparrow}$ $woman^{\uparrow}$ $who^{\uparrow}$ $is^{\uparrow}$ $beautiful^{\uparrow}$ $is^{\uparrow}$ $walking^{\uparrow}$ $in^{\uparrow}$ $the^{\uparrow}$ $rain^{=}$, the monotone mark $^{\uparrow}$ on *woman* allows an upward inference: *woman* $\sqsupseteq$ *woman who is beautiful*, in which the relative clause *who is beautiful* is deleted. The system follows a set of phrasal monotonicity inference rules. For upward monotonicity inference, modifiers of a word are deleted. For downward monotonicity inference, modifiers are inserted into a word. The algorithm traverses down a polarized UD parse tree, deletes the modifier sub-tree if a node is monotone ($\uparrow$), and inserts a new sub-tree if a node is antitone ($\downarrow$). To insert new modifiers, the algorithm extracts a list of potential modifiers associated with a node from a modifier dictionary. The modifier dictionary is derived from the hypothesis and contains word-modifier pairs for each dependency relation. Below is an example of a modifier dictionary from *There are no beautiful flowers that open at night*:

- **Obl**: [head: *open*, mod: *at night*]
- **Amod**: [head: *flowers*, mod: *beautiful*]
- **Acl:relcl**: [head: *flowers*, mod: *that open at night*]

Syntactic Variation Inference

We categorize linguistic changes between a premise and a hypothesis that cannot be inferred from monotonicity information as *syntactic variations*. For example, a change from *red rose* to *a rose which is red* is a syntactic variation. Many logical systems rely on handcrafted rules and manual transformation to enable the system to perform syntactic variations. However, without accurate alignments between the two sentences, these methods are not robust enough and, thus, difficult to scale up for wide-coverage input. The recent development of pretrained transformer-based language models brings state-of-art performance on multiple benchmarks for Natural Language Understanding (NLU), including the task of paraphrase detection Devlin et al. (2019); Lan et al. (2020); Liu et al. (2020), which exemplifies phrasal knowledge of syntactic variation. We propose a method that incorporates transformer-based language models to handle syntactic variations robustly. Our method first decomposes both the premise and the hypothesis into chunks of phrases using a sentence chunker and then calculates the likelihood of each pair of chunks being a paraphrase using a transformer model.

Sequence Chunking

To obtain phrase-level chunks from a sentence, we build a sequence chunker, which relies on the sentence's universal dependency information. Instead of breaking down a sentence, our chunker composes word tokens recursively to form meaningful chunks. First, we construct a sentence representation graph of a premise from the controller. A sentence representation graph is defined as $\mathrm{G} = \langle \mathcal{V}, \mathcal{E} \rangle$, where $\mathcal{V} = \mathcal{V}_m \cup \mathcal{V}_c$ is the set of modifiers ($\mathcal{V}_m$) and content words ($\mathcal{V}_c$), and $\mathcal{E}$ is the set of directed edges. To generate the chunk for a content word in $\mathcal{V}_c$, we arrange its modifiers, which are nodes it points to, together with the content word by their word orders in the original sentence to form a word chain, for example, in *The woman in a pink dress is dancing*. The edges from *dress* to *in*, *a*, *pink* with the edge from *woman* to *dress* can be drawn. Chunks *in a pink dress* and *the woman in a pink dress* will be generated for *dress* and *woman*, respectively.

Monolingual Phrase Alignment

Given a set of chunks from a generated sentence and from the hypothesis, the system computes an alignment score for each pair of chunks to select the syntactic variations. Formally, we define $\mathcal{C}_s$ as the set of chunks from a generated sentence and $\mathcal{C}_h$ as the set of chunks from the hypothesis. We build the Cartesian product from $\mathcal{C}_s$ and $\mathcal{C}_h$, denoted $\mathcal{C}_s \times \mathcal{C}_h$. For each chunk pair $(\mathrm{c}_s, \mathrm{c}_h) \in \mathcal{C}_s \times \mathcal{C}_h$, we compute an alignment score $\boldsymbol{\alpha}$:

$$\boldsymbol{\alpha}_{\langle \mathbf{c_s}, \mathbf{c_h} \rangle} = \mathrm{p}(\mathbf{y} \mid \langle \mathbf{c_s}, \mathbf{c_h} \rangle)$$

where $\mathbf{y} \mid \langle \mathbf{c_s}, \mathbf{c_h} \rangle = \mathit{Softmax}(\mathrm{ALBERT}(\langle \mathbf{c_s}, \mathbf{c_h} \rangle))$. If $\boldsymbol{\alpha} > 0.85$ (determined by a grid search of 5 values), the system records this pair of phrases as a syntactic variation. To calculate the alignment score, we use an ALBERT Lan et al. (2020) model, fine-tuned on the Microsoft Research Paraphrase Corpus Dolan and Brockett (2005). We first pass a chunk pair to ALBERT to obtain its logits. Then we apply a softmax function to the logits to get the final probability.

3.3 Evaluation

3.3.1 Experiment Setup

For Universal Dependency parsing, we follow Udep2Mono's framework Chen and Gao (2021) and use a neural parsing model from Stanford's Stanza Qi et al. (2020) with 90.0 LAS Zeman et al. (2018) evaluation score. We select the BERT-large model pre-trained on STS-B Cer et al. (2017) from Sentence-BERT Reimers and Gurevych (2019).[1] For ALBERT, we used an ALBERT-base model pretrained on the MRPC corpus. We evaluate our proposed reasoning system, NeuralLog, on the MED dataset for monotonicity reasoning. We compare our method with multiple deep-learning-based baselines. Here, DeComp and ESIM are trained on SNLI, and BERT is fine-tuned with MultiNLI. The BERT+ model is a BERT model fine-tuned on a combined training data with the HELP dataset, Yanaka et al. (2019), a set of augmentations for monotonicity reasoning, and the MultiNLI training set. Both models were tested in Yanaka et al. (2019). We also compare against the Attentive Tree Net we proposed in the first part to see if the neural-symbolic inference is a better choice than dedicated neural architecture and training data.

[1] Note that there are new embeddings that are more robust and accurate than the one we used. We recommend using the up-to-date embeddings.

Table 3 Results comparing model compared to state-of-art NLI models evaluated on MED. **Up**, **Down**, and **All** stand for the accuracy of upward inference, downward inference, and the overall dataset

Model	Up	Down	All
DeComp (Parikh et al., 2016)	71.1	45.2	51.4
ESIM (Chen et al., 2017)	66.1	42.1	53.8
BERT (Devlin et al., 2019)	82.7	22.8	44.7
BERT+ (Yanaka et al., 2019)	76.0	70.3	71.6
AttnTreeNet (ours)	81.4	74.5	75.7
NeuralLog (ours)	**91.4**	**93.9**	**93.4**

Bold indicates the highest accuracy in the table for each column

3.3.2 Results

As Table 3 shows, our system (NeuralLog) outperforms all the neural model baselines in terms of accuracy by a significant margin (48.7% maximum increase and 21.8% minimum increase). Compared to a prior neural-symbolic system, BERT+, our system performs much better both on the upward (15.4%) and downward (23.6%) inference. Compared to the Attentive Tree-structured Net for monotonicity reasoning, our neural-symbolic system still shows better performance with a significant margin of increase (Δ 17.7%). This result highlights the point that using dedicated training data and neural architectures for monotonicity reasoning is not as effective as a neural-symbolic system that utilizes neural modules for intermediate reasoning. The good performance on MED validates our system's ability on accurate and robust monotonicity-based inferences.

4 Large-scale Foundation Model

4.1 Preliminary

In the field of natural language processing (NLP), the use of large language models (LLMs) has significantly revolutionized how people approach reasoning and inference on language. It has been established that the effectiveness and efficiency of these models in various NLP applications can be improved by increasing their size, such as by increasing their training resources, the number of model parameters, and so on Wei et al. (2022). Self-supervised pre-training gives large-scale language models the ability to learn downstream tasks given no example or only a few input–output paired examples without optimization. Recent research shows emergent interest in uncovering the underlying logic of the aforementioned mysterious capacity of LLMs by empirical and theoretical approaches Rubin et al. (2021); Xie et al. (2021); Min et al. (2022); Ye and Durrett (2022). However, the current analysis of these LLMs still cannot answer if the unpredictable phenomena of emergent abilities of LLMs allow them to acquire the ability to simulate symbolic logic in natural language. In this section, we make an effort to benchmark various LLMs' reasoning ability on monotonicity to gain some insights into the limitation of current LLMs.

4.2 Method

Zero-Shot Learning

Many studies show that large-scale language models exhibit zero-shot learning ability Kojima et al. (2022). The models can solve various NLP tasks by simply conditioning the instructions describing the task. We start our experiments on monotonicity reasoning using the setting of zero-shot learning. Specifically, we give the model a prompt Liu et al. (2021) in the format of *Instruction*: ⟨Instruction⟩ *Context*: ⟨Context⟩ *Question*: ⟨Question⟩ *Answer*: ⟨Answer⟩. The model then generates the ⟨Answer⟩ tokens for the given problem by conditioning on this prompt. In zero-shot learning, the model cannot rely on any demonstrations but its parametric knowledge that is acquired during the pre-training stage, which is triggered by the prompt.

In-Context Learning

In-context learning for large language models is formulated as a text-generation problem. The generation is conditioned on a given prompt p which consists of the input problem x and k examples of input–output pairs:

$$p_{\mathrm{LLM}}(\mathrm{y} \mid \mathrm{p}) = \prod_{t=1}^{T} p(\mathrm{y_t} \mid \mathrm{p}, \mathrm{y}_{<\mathrm{t}}), \tag{3}$$

where the prompt p contains several examples and the question to be answered: $\mathrm{p} = \{x_1, y_1, ..., x_k, y_k, x\}$, and LLM is a large language model that can generate text in an auto-regressive way. According to Xie et al. (2021), the in-context learning ability of LLMs could be interpreted as an implicit Bayesian inference gained from the auto-regressive next-token generation task in the pre-training. The given input text, prompt p, provides evidence of posterior distribution over task-related latent concepts c to infer the corresponding label y:

$$p(\mathrm{y} \mid \mathrm{p}) = \int_{c} p(\mathrm{y} \mid \mathrm{c}, \mathrm{p}) p(\mathrm{c} \mid \mathrm{p}) d(\mathrm{c}). \tag{4}$$

In-context learning allows one to adapt LLMs to a different domain and downstream tasks without any fine-tuning. Because of its effectiveness and efficiency, we desire to investigate LLMs' monotonicity reasoning capacity.

4.3 Evaluation

Setup

The evaluation focuses on assessing large language models' reasoning ability with respect to monotonicity. We evaluate both the zero-shot learning setting and the few-shot in-context learning setting. When designing the prompt, we follow previous work on prompt-based multi-task learning Sanh et al. (2021) and build a Natural-Language-Inference-styled prompt. We include detailed instructions for the task and its label space to inject domain-specific understanding into models. We use the monotonicity reasoning test set from the CURRICULUM benchmark Chen and Gao (2022), a large-scale reasoning benchmark for evaluating broad-coverage linguistic phenomena. The monotonicity portion of the CURRICULUM benchmark integrates the MED dataset, the Semantic Fragments test sets Richardson et al. (2019), and 500 additional gold annotated monotonicity reasoning sentence pairs that are manually annotated and curated by human writers. Overall, this test set provides high-quality data, challenging problems, and analysis of powerful contextualized embedding language models. Thus, this test set allows us to conduct a more in-depth evaluation of modern large-scale language models. For in-context learning, we provide 4-shot, 8-shot, and 16-shot of examples to LLMs, respectively. For instance, in a 4-shot setting, 4 examples are randomly sampled from the training set for each label and concatenated to the prompt as a prefix. Each setting is evaluated 3 times, and the in-context examples are fixed every round to avoid the potential bias from example selection. We report the average performance across the 3 runs.

Baselines

For model selection, we pick LLMs with strong zero-shot learning abilities. The first type of models we select are LLMs that are continue fine-tuned in multi-task or instruction-tuning settings. We first report the baseline performance of the current SOTA NLI models, including **RoBERTa** Liu et al. (2019) and **DeBERTa** He et al. (2021). These two models are pre-trained bidirectional language models based on transformers and have shown impressive performance on NLI. These two models are fine-tuned on a mixture of common NLI training sets, including SNLI Bowman et al. (2015), MNLI Williams et al. (2018), FEVER Thorne et al. (2018), and ANLI Nie et al. (2020). We select **FLAN-T5** Wei et al. (2021) as the instruction-tuned model. FLAN-T5 is a T5 Text2Text model trained using an instruction-based fine-tuning procedure on a collection of data sources with various instruction template types. FLAN with scaled parameters and training instructions shows strong zero-shot and few-shot learning abilities, outperforming prior public checkpoints. The second model type is LLMs, with a large parameter size (175 billion) showing the incredible ability for in-context learning Chung et al. (2022). We select the popular GPT-3 models from OpenAI. GPT-3 Brown et al. (2020) is a state-of-the-art auto-regressive language generation model. With 175 billion parameters and massive pre-training text data, it is currently one of the largest and most powerful language models in existence, capable of a wide range of natural language processing tasks. We include the original pre-trained GPT-3 model (text-davinci-001) and the GPT3.5 (text-davinci-003) model Ouyang et al. (2022), a version of the InstructGPT fine-tuned using reinforcement learning with reward models trained from human feedback (RLHF). GPT3.5 is much better at following the human intent in the instruction than the pre-trained version Ouyang et al. (2022).

4.4 Results

Table 4 shows the evaluation results for these models. Both GPT-3 and GPT$-$3.5 achieve only random performance(50%). GPT3.5 outperforms GPT3 by about 6% in every setting, but the performance is still far from proficiency in monotonicity reason-

Table 4 Evaluation results for large language models (LLMs) on CURRICULUM's monotonicity test set. Here *M* refers to *million* and *B* refers to *billion* for the number of parameters

Model	# Parameters	0-shot	4-shot	8-shot	16-shot
Random Baseline	–	50.0	–	–	–
RoBERTa-large-SMFA Liu et al. (2019)	355 M	50.8	–	–	–
DeBERTa-large-SMFA He et al. (2021)	435 M	51.1	–	–	–
FLAN-T5-XXL Wei et al. (2021)	11B	58.7	–	–	-
GPT3 (text-davinci-001) Brown et al. (2020)	175B	48.3	51.9	51.8	51.6
GPT3.5 (text-davinci-003) Ouyang et al. (2022)	175B	56.9	57.3	57.9	58.7

ing. These low performances raise the question of whether LLMs' can emulate logical reasoning expressed in natural language. Interestingly, instruction tuning with RLHF Ouyang et al. (2022) does not help the model substantially improve its understanding of logical inference, as shown in the overall low accuracy from GPT−3.5. On the other hand, compared to GPT-3, whose performance seems irrelevant with respect to the number of in-context examples, GPT3.5 shows consistent improvements as we give the model more examples, although such increases are still marginal. GPT−3.5's performance gain from in-context learning is only trivial (0.4%). The results show that the in-context examples give the model certain levels of domain-specific task knowledge but fail to help the model fully learn the ability to perform monotonicity reasoning. GPT-3's poor performance in the zero-shot setting and performance fluctuation among different in-context learning settings suggest that the model only learns the shallow structure knowledge about the task rather than the implicit reasoning skill. Regarding smaller instruction-tuned models, Flan-T5 outperforms GPT-3 and is comparable to GPT−3.5 in the 16-shot setting. However, its performance is still near random, suggesting that it understands the task better due to many instruction-fine-tuning tasks but still fails to learn the logical reasoning rules from the instructions. For smaller models, both RoBERTa and DeBERTa show near-random performances. Their lack of knowledge of monotonicity reasoning is expected as Chen and Gao (2022) showed that pretrained transformer-based models may not encode much monotonicity information during their pre-training process. Nevertheless, even fine-tuning with commonly used NLI training data still fails to benefit models' performance on monotonicity reasoning. Such results lead to concerns about the learning quality of the models and the lack of logical reasoning samples in these common NLI datasets. Overall, we show that large language models still require a major effort to improve their reasoning ability on logic.

5 Conclusions

In this paper, we provide an in-depth discussion of monotonicity reasoning in the age of neural foundation models. To summarize, we first propose the AttentiveTreeNet to investigate the effectiveness of incorporating structural knowledge and linguistic principles into neural architectures on monotonicity reasoning. Next, we propose a hybrid reasoning framework that utilizes both symbolic reasoning modules built from human-defined logical rules and neural language models to solve monotonicity NLI problems. For the third part, we analyze several popular and powerful large foundation models on monotonicity reasoning to verify if the ability to emulate logical reasoning has emerged in these massive neural models. Our evaluation focus on the MED benchmark and the CURRICULUM benchmark's monotonicity section. Our analysis shows that injecting structural knowledge into advanced neural networks can largely improve the original network's performance on monotonicity inference. However, performing reasoning jointly using symbolic and neural modules can further master the monotonicity reasoning task and achieve state-of-the-art performance while maintaining high interpretability. We show that large language models are far from mastering the skill of logical reasoning. Although popular models like InstructGPT can make powerful generations and predictions for various linguistic tasks and applications, they can only achieve a random performance on monotonicity reasoning. Overall, our work reveals the limitation of current large foundation models and sheds light on the new direction of approaching logical reasoning through neural-symbolic inference. For future work, it would be exciting to see symbolic reasoning systems built on top of large language models for complex logical reasoning tasks.

Funding Open access funding provided by EPFL Lausanne

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

Abzianidze, L. (2017). LangPro: Natural language theorem prover. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 115–120. Association for Computational Linguistics, Copenhagen, Denmark. https://doi.org/10.18653/v1/D17-2020. https://aclanthology.org/D17-2020

Bommasani, R., Hudson, D.A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M.S., Bohg, J., Bosselut, A., Brunskill, E., Brynjolfsson, E., Buch, S., Card, D., Castellon, R., Chatterji, N., Chen, A., Creel, K., Davis, J.Q., Demszky, D., Donahue, C., Doumbouya, M., Durmus, E., Ermon, S., Etchemendy, J., Ethayarajh, K., Fei-Fei, L., Finn, C., Gale, T., Gillespie, L., Goel, K., Goodman, N., Grossman, S., Guha, N., Hashimoto, T., Henderson, P., Hewitt, J., Ho, D.E., Hong, J., Hsu, K., Huang, J., Icard, T., Jain, S., Jurafsky, D., Kalluri, P., Karamcheti, S., Keeling, G., Khani, F., Khattab, O., Koh, P.W., Krass, M., Krishna, R., Kuditipudi, R., Kumar, A., Ladhak, F., Lee, M., Lee, T., Leskovec, J., Levent, I., Li, X.L., Li, X., Ma, T., Malik, A., Manning, C.D., Mirchandani, S., Mitchell, E., Munyikwa, Z., Nair, S., Narayan, A., Narayanan, D., Newman, B., Nie, A., Niebles, J.C., Nilforoshan, H., Nyarko, J., Ogut, G., Orr, L., Papadimitriou, I., Park, J.S., Piech, C., Portelance, E., Potts, C., Raghunathan, A., Reich, R., Ren, H., Rong, F., Roohani, Y., Ruiz, C., Ryan, J., Ré, C., Sadigh, D., Sagawa, S., Santhanam, K., Shih, A., Srinivasan, K., Tamkin, A., Taori, R., Thomas, A.W., Tramér, F., Wang, R.E., Wang, W., Wu, B., Wu, J., Wu, Y., Xie, S.M., Yasunaga, M., You, J., Zaharia, M., Zhang, M., Zhang, T., Zhang, X., Zhang, Y., Zheng, L., Zhou, K. & Liang, P. (2022). On the Opportunities and Risks of Foundation Models.

Bowman, S.R., Angeli, G., Potts, C. & Manning, C.D. (2015). A large annotated corpus for learning natural language inference. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 632–642. Association for Computational Linguistics, Lisbon, Portugal. https://doi.org/10.18653/v1/D15-1075. https://aclanthology.org/D15-1075

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems, 33*, 1877–1901.

Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I. & Specia, L. (2017). SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pp. 1–14. Association for Computational Linguistics, Vancouver, Canada. https://doi.org/10.18653/v1/S17-2001. https://aclanthology.org/S17-2001

Chen, Z. & Gao, Q. (2021). Monotonicity marking from Universal Dependency trees. In: Proceedings of the 14th International Conference on Computational Semantics (IWCS), pp. 121–131. Association for Computational Linguistics, Groningen, The Netherlands (online). https://aclanthology.org/2021.iwcs-1.12

Chen, Z. & Gao, Q. (2022). Curriculum: A broad-coverage benchmark for linguistic phenomena in natural language understanding. In: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 3204–3219. Association for Computational Linguistics, Seattle, United States. https://doi.org/10.18653/v1/2022.naacl-main.234. https://aclanthology.org/2022.naacl-main.234

Chen, D. & Manning, C. (2014). A fast and accurate dependency parser using neural networks. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 740–750. Association for Computational Linguistics, Doha, Qatar. https://doi.org/10.3115/v1/D14-1082. https://aclanthology.org/D14-1082

Chen, Z. (2021). Attentive tree-structured network for monotonicity reasoning. In: Proceedings of the 1st and 2nd Workshops on Natural Logic Meets Machine Learning (NALOMA), pp. 12–21. Association for Computational Linguistics, Groningen, the Netherlands (online). https://aclanthology.org/2021.naloma-1.3

Chen, Z., Gao, Q. & Moss, L.S. (2021). NeuralLog: Natural language inference with joint neural and logical reasoning. In: Proceedings of *SEM 2021: The Tenth Joint Conference on Lexical and Computational Semantics, pp. 78–88. Association for Computational Linguistics, Online. https://doi.org/10.18653/v1/2021.starsem-1.7. https://aclanthology.org/2021.starsem-1.7

Chen, Q., Zhu, X., Ling, Z.-H., Wei, S., Jiang, H. & Inkpen, D. (2017). Enhanced LSTM for natural language inference. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Vol. 1), pp. 1657–1668. Association for Computational Linguistics, Vancouver, Canada. https://doi.org/10.18653/v1/P17-1152. https://aclanthology.org/P17-1152

Chen, Z., & Gao, Q. (2022). Probing linguistic information for logical inference in pre-trained language models. *Proceedings of the AAAI Conference on Artificial Intelligence, 36*(10), 10509–10517. https://doi.org/10.1609/aaai.v36i10.21294

Chung, H.W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, Y., Wang, X., Dehghani, M., Brahma, S., Webson, A., Gu, S.S., Dai, Z., Suzgun, M., Chen, X., Chowdhery, A., Castro-Ros, A., Pellat, M., Robinson, K., Valter, D., Narang, S., Mishra, G., Yu, A., Zhao, V., Huang, Y., Dai, A., Yu, H., Petrov, S., Chi, E.H., Dean, J., Devlin, J., Roberts, A., Zhou, D., Le, Q.V. & Wei, J. (2022) Scaling Instruction-Finetuned Language Models.

Conneau, A., Kiela, D., Schwenk, H., Barrault, L. & Bordes, A. (2017). Supervised learning of universal sentence representations from natural language inference data. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. https://doi.org/10.18653/v1/d17-1070

Dagan, I., Roth, D., Sammons, M. & Zanzotto, F.M. (2013). Recognizing Textual Entailment: Models and Applications. Synthesis Lectures on Human Language Technologies, pp. 1–220. Morgan and Claypool Publishers.

Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Vol. 1, pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota. https://doi.org/10.18653/v1/N19-1423. https://aclanthology.org/N19-1423

Dolan, W.B. & Brockett, C. (2005). Automatically constructing a corpus of sentential paraphrases. In: Proceedings of the Third International Workshop on Paraphrasing (IWP2005). https://aclanthology.org/I05-5002

Glockner, M., Shwartz, V. & Goldberg, Y. (2018). Breaking NLI systems with sentences that require simple lexical inferences. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics. Vol. 2, pp. 650–655. Association for Computational Linguistics, Melbourne, Australia. https://doi.org/10.18653/v1/P18-2103. https://aclanthology.org/P18-2103

Gururangan, S., Swayamdipta, S., Levy, O., Schwartz, R., Bowman, S. & Smith, N.A. (2018). Annotation artifacts in natural language inference data. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Vol. 2, pp. 107–112. Association for Computational Linguistics, New Orleans, Louisiana. https://doi.org/10.18653/v1/N18-2017. https://aclanthology.org/N18-2017

He, P., Liu, X., Gao, J. & Chen, W. (2021). DeBERTa: Decoding-Enhanced BERT with Disentangled Attention.

Hu, H., Chen, Q. & Moss, L. (2019). Natural language inference with monotonicity. In: Proceedings of the 13th International Conference on Computational Semantics, pp. 8–15. Association for Computational Linguistics, Gothenburg, Sweden. https://doi.org/10.18653/v1/W19-0502. https://aclanthology.org/W19-0502

Hu, H., Chen, Q., Richardson, K., Mukherjee, A., Moss, L.S. & Kuebler, S. (2020). MonaLog: a lightweight system for natural language inference based on monotonicity. In: Proceedings of the Society for Computation in Linguistics 2020, pp. 334–344. Association for Computational Linguistics, New York, New York. https://aclanthology.org/2020.scil-1.40

Kingma, D.P. & Ba, J. (2014). Adam: A Method for Stochastic Optimization.

Kojima, T., Gu, S.S., Reid, M., Matsuo, Y. & Iwasawa, Y.(2022). Large Language Models are Zero-Shot Reasoners. arXiv. https://doi.org/10.48550/ARXIV.2205.11916. arXiv:2205.11916

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P. & Soricut, R. (2020). Albert: A lite bert for self-supervised learning of language representations. In: International Conference on Learning Representations. https://openreview.net/forum?id=H1eA7AEtvS

Lin, Z., Feng, M., dos Santos, C.N., Yu, M., Xiang, B., Zhou, B. & Bengio, Y. (2017). A structured self-attentive sentence embedding. ArXiv **abs/1703.03130**

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L. & Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv. https://doi.org/10.48550/ARXIV.1907.11692. arXiv:1907.11692

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L. & Stoyanov, V. (2020). RoBERTa: A Robustly Optimized BERT Pretraining Approach. https://openreview.net/forum?id=SyxS0T4tvS

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H. & Neubig, G. (2021). Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. arXiv . https://doi.org/10.48550/ARXIV.2107.13586. arXiv:2107.13586

Liu, H., & Singh, P. (2004). Conceptnet - A practical commonsense reasoning tool-kit. *BT Technology Journal, 22*(4), 211–226. https://doi.org/10.1023/B:BTTJ.0000047600.45421.6d

MacCartney, B. & Manning, C.D. (2009). An extended model of natural logic. In: Proceedings of the Eight International Conference on Computational Semantics, pp. 140–156. Association for Computational Linguistics, Tilburg, The Netherlands. https://aclanthology.org/W09-3714

Martínez-Gómez, P., Mineshima, K., Miyao, Y. & Bekki, D. (2017). On-demand injection of lexical knowledge for recognising textual entailment. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Vol. 1, pp. 710–720. Association for Computational Linguistics, Valencia, Spain. https://aclanthology.org/E17-1067

McCoy, T., Pavlick, E. & Linzen, T. (2019). Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 3428–3448. Association for Computational Linguistics, Florence, Italy. https://doi.org/10.18653/v1/P19-1334. https://aclanthology.org/P19-1334

Miller, G. A. (1995). Wordnet: A lexical database for english. *Communications of the ACM, 38*(11), 39–41. https://doi.org/10.1145/219717.219748

Min, S., Lyu, X., Holtzman, A., Artetxe, M., Lewis, M., Hajishirzi, H. & Zettlemoyer, L. (2022). Rethinking the role of demonstrations: What makes in-context learning work? ArXiv **abs/2202.12837**

Nie, Y., Williams, A., Dinan, E., Bansal, M., Weston, J. & Kiela, D. (2020). Adversarial NLI: A new benchmark for natural language understanding. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 4885–4901. Association for Computational Linguistics, Online. https://doi.org/10.18653/v1/2020.acl-main.441. https://aclanthology.org/2020.acl-main.441

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C.L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J. & Lowe, R. (2022). Training Language Models to Follow Instructions with Human Feedback

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C.L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J. & Lowe, R.(2022). Training Language Models to Follow Instructions with Human Feedback. arXiv . https://doi.org/10.48550/ARXIV.2203.02155. arXiv:2203.02155

Parikh, A., Täckström, O., Das, D. & Uszkoreit, J. (2016). A decomposable attention model for natural language inference. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 2249–2255. Association for Computational Linguistics, Austin, Texas. https://doi.org/10.18653/v1/D16-1244. https://aclanthology.org/D16-1244

Partee, B. (2007). Compositionality and coercion in semantics: The dynamics of adjective meaning 1.

Pennington, J., Socher, R. & Manning, C. (2014). GloVe: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543. Association for Computational Linguistics, Doha, Qatar. https://doi.org/10.3115/v1/D14-1162. https://aclanthology.org/D14-1162

Qi, P., Zhang, Y., Zhang, Y., Bolton, J. & Manning, C.D. (2020). Stanza: A python natural language processing toolkit for many human languages. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp. 101–108. Association for Computational Linguistics, Online. https://doi.org/10.18653/v1/2020.acl-demos.14. https://aclanthology.org/2020.acl-demos.14

Reimers, N. & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks.

Richardson, K., Hu, H., Moss, L.S. & Sabharwal, A. (2019). Probing Natural Language Inference Models through Semantic Fragments.

Rubin, O., Herzig, J. & Berant, J. (2021). Learning to Retrieve Prompts for In-Context Learning. ArXiv **abs/2112.08633**

Sanh, V., Webson, A., Raffel, C., Bach, S.H., Sutawika, L., Alyafeai, Z., Chaffin, A., Stiegler, A., Scao, T.L., Raja, A., et al. (2021). Multitask Prompted Training Enables Zero-Shot Task Generalization. arXiv preprint arXiv:2110.08207

Tai, K.S., Socher, R. & Manning, C.D. (2015). Improved semantic representations from tree-structured long short-term memory networks. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Vol. 1), pp. 1556–1566. Association for Computational Linguistics, Beijing, China. https://doi.org/10.3115/v1/P15-1150. https://aclanthology.org/P15-1150

Thorne, J., Vlachos, A., Christodoulopoulos, C. & Mittal, A. (2018). FEVER: a large-scale dataset for fact extraction and VERification. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pp. 809–819. Association for Computational Linguistics, New Orleans, Louisiana. https://doi.org/10.18653/v1/N18-1074. https://aclanthology.org/N18-1074

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Roziére, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E. & Lample, G. (2023). LLaMA: Open and efficient foundation language models.

Wang, S. & Jiang, J. (2016). Learning natural language inference with LSTM. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 1442–1451. Association for Computational Linguistics, San Diego, California. https://doi.org/10.18653/v1/N16-1170. https://aclanthology.org/N16-1170

Wang, Z., Hamza, W. & Florian, R. (2017). Bilateral multi-perspective matching for natural language sentences. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17, pp. 4144–4150. https://doi.org/10.24963/ijcai.2017/579. https://doi.org/10.24963/ijcai.2017/579

Wei, J., Bosma, M., Zhao, V.Y., Guu, K., Yu, A.W., Lester, B., Du, N., Dai, A.M. & Le, Q.V. (2021). Finetuned language models are zero-shot learners. arXiv preprint arXiv:2109.01652

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., Chi, E.H., Hashimoto, T., Vinyals, O., Liang, P., Dean, J. & Fedus, W. (2022). Emergent Abilities of Large Language Models.

Williams, A., Nangia, N. & Bowman, S. (2018). A broad-coverage challenge corpus for sentence understanding through inference. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Vol. 1, pp. 1112–1122. Association for Computational Linguistics, New Orleans, Louisiana. https://doi.org/10.18653/v1/N18-1101. https://aclanthology.org/N18-1101

Williams, A., Nangia, N. & Bowman, S. (2018). A broad-coverage challenge corpus for sentence understanding through inference. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Vol. 1, pp. 1112–1122. Association for Computational Linguistics. http://aclweb.org/anthology/N18-1101

Xie, S.M., Raghunathan, A., Liang, P. & Ma, T. (2021). An Explanation of In-Context Learning as Implicit Bayesian Inference. ArXiv **abs/2111.02080**

Yanaka, H., Mineshima, K., Bekki, D., Inui, K., Sekine, S., Abzianidze, L. & Bos, J. (2019). Can neural networks understand monotonicity reasoning? In: Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, pp. 31–40. Association for Computational Linguistics, Florence, Italy. https://doi.org/10.18653/v1/W19-4804. https://aclanthology.org/W19-4804

Yanaka, H., Mineshima, K., Bekki, D., Inui, K., Sekine, S., Abzianidze, L. & Bos, J. (2019). HELP: A dataset for identifying shortcomings of neural models in monotonicity reasoning. In: Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019), pp. 250–255. Association for Computational Linguistics, Minneapolis, Minnesota. https://doi.org/10.18653/v1/S19-1027. https://aclanthology.org/S19-1027

Yanaka, H., Mineshima, K., Martínez-Gómez, P. & Bekki, D. (2018). Acquisition of phrase correspondences using natural deduction proofs. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Vol. 1, pp. 756–766. Association for Computational Linguistics, New Orleans, Louisiana. https://doi.org/10.18653/v1/N18-1069. https://aclanthology.org/N18-1069

Ye, X. & Durrett, G.(2022). The Unreliability of Explanations in Few-Shot in-Context Learning. ArXiv **abs/2205.03401**

Zeman, D., Hajič, J., Popel, M., Potthast, M., Straka, M., Ginter, F., Nivre, J. & Petrov, S. (2018). CoNLL 2018 shared task: Multilingual parsing from raw text to Universal Dependencies. In: Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, pp. 1–21. Association for Computational Linguistics, Brussels, Belgium. https://doi.org/10.18653/v1/K18-2001. https://aclanthology.org/K18-2001

Zhao, K., Huang, L. & Ma, M. (2016). Textual entailment with structured attentions and composition. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pp. 2248–2258. The COLING 2016 Organizing Committee, Osaka, Japan. https://aclanthology.org/C16-1212

Zhou, Y., Liu, C. & Pan, Y.(2016). Modelling sentence pairs with tree-structured attentive encoder. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pp. 2912–2922. The COLING 2016 Organizing Committee, Osaka, Japan. https://aclanthology.org/C16-1274

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.