Unravelling the Logic: Investigating the Generalisation of Transformers in Numerical Satisfiability Problems

Tharindu Madusanka[1] and **Marco Valentino**[2,3] and **Iqra Zahid**[1,4]
and **Ian Pratt-Hartmann**[1,5] and **Riza Batista-Navarro**[1]
1. Department of Computer Science, University of Manchester;
2. School of Computer Science, University of Sheffield;
3. Idiap Research Institute;
4. Imperial College London, Imperial Global Singapore;
5. Instytut Informatyki, Uniwersytet Opolski

Abstract

Transformer models have achieved remarkable performance in many formal reasoning tasks. Nonetheless, the extent of their comprehension pertaining to logical semantics and rules of inference remains somewhat uncertain. Evaluating such understanding necessitates a rigorous examination of these models' generalisation capacity to out-of-distribution data. In this study, we probe the generalisation prowess of Transformer models with respect to the hitherto unexplored domain of numerical satisfiability problems. Our investigation reveals that Transformers exhibit minimal scale and noise invariance, alongside limited vocabulary and number invariance. However, even when Transformer models experience a notable decline in performance on out-of-distribution test sets, they often still surpass the random baseline by a considerable margin.

1 Introduction

Transformer[1] models have become the de facto state-of-the-art in solving almost all language-based tasks (Devlin et al., 2018; Yang et al., 2019; Raffel et al., 2019; OpenAI, 2023). Notably, among these tasks, formal reasoning has garnered considerable interest in recent times (Richardson et al., 2020; Tafjord et al., 2021; McCoy et al., 2019). Formal reasoning delineates a distinct strand of reasoning characterised by the drawing of conclusions solely from logical rules, without relying on common sense or background knowledge. Although Transformers have exhibited notable proficiency in formal reasoning tasks (Talmor et al., 2019; Brown et al., 2020; Kojima et al., 2022), the depth of their comprehension regarding logical semantics and rules of inference remains uncertain.

Many logical problems—and in particular the problem of recognising valid entailments—can be reduced to the problem of determining satisfiability: a set of closed formulae Φ is *satisfiable* if there exists some structure (in a model-theoretic sense) $\mathfrak{A}$ in which every element of Φ is true. This concept extends to natural language in an obvious way: a set of *sentences* is satisfiable if the set of *formulae* into which they translate is satisfiable. Informally, the natural language satisfiability problem is the task of determining whether there are any inherent contradictions within a given set of sentences. Consequently, natural language variants of satisfiability problems represent an ideal domain for studying transformer models' ability to learn rules of inference and logical semantics. In our study, we extend research on the formal reasoning abilities of transformers by introducing problems that involve numerical reasoning. That is, we consider a special case of the satisfiability problem in which the formulae (or sentences) involve numerical quantification. For illustration, consider the following set of sentences:

Consider the following set of sentences,

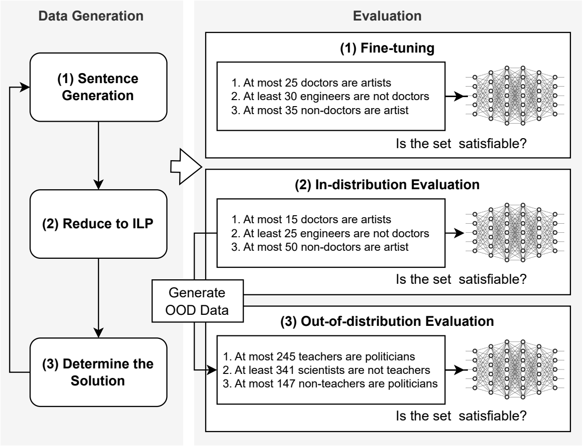
1. *At most 25 doctors are artists*
2. *At least 30 engineers are not doctors*
3. *At most 35 non-doctors are artist*
4. *At least 100 engineers are artists*

Sentences 1, 3 and 4 form an unsatisfiable set: the first two entail there to be at most 60 artists, which directly contradicts sentence 4. On the other hand, sentences 1, 2 and 3 form a satisfiable set: as may be easily seen, there is a structure in which these sentences are all true.

Learning to solve instances of numerical satisfiability problems requires an understanding of logical semantics and natural language numerals, as well as the ability to apply logical and mathematical rules.Furthermore, this framework offers a high degree of controllability, enabling the systematic scaling of the problem space, automated evaluation of solutions, and the identification of hard-problem

[1] For brevity, we use "Transformer models" or "Transformers" to refer to auto-regressive Transformer-based language models

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 25155–25168
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics



1. At most 25 doctors are artists
2. At least 30 engineers are not doctors
3. At most 35 non-doctors are artist

1. At most 15 doctors are artists
2. At least 25 engineers are not doctors
3. At most 50 non-doctors are artist

1. At most 245 teachers are politicians
2. At least 341 scientists are not teachers
3. At most 147 non-teachers are politicians

Figure 1: A data generation methodology that reduces sets of generated sentences to sets of inequalities and employs a linear solver to determine the solution. The resulting dataset is then used to fine-tune and evaluate Transformers on both in-distribution and out-of-distribution (OOD) problems to test robustness and generalisation.

regions. Moreover, numerical satisfiability introduces problem instances that are computationally more complex than their counterparts involving only the quantifiers *every*, *some* and *no*. Therefore, this problem provides a unique approach to evaluating transformer models' ability to perform numerical reasoning. To the best of our knowledge, this problem has not been investigated in the previous literature.

The primary reason for this limitation is that relatively few reasoning tools can directly ascertain the satisfiability of formulae incorporating numbers or numerical quantifiers. Consequently, there arises the challenge of constructing an extensive dataset suitable for fine-tuning and assessing Transformers. We overcome this limitation by reducing the numerical satisfiability problems into integer linear problems (see Figure 1). Moreover, we acknowledge the potential drawbacks associated with evaluating models using synthetic data, particularly the pitfalls associated with undersampling challenging instances (Wu et al., 2021; Shin et al., 2019). To address this concern and ensure a robust dataset, we adopt a targeted sampling approach by selecting problem instances from the phase-change region. This region, where the probability of satisfiability is approximately 0.5, represents a critical region where algorithms tasked with determining satisfiability typically experience prolonged running times. Consequently, we conduct a comprehensive exploration of the problem space associated with the numerical satisfiability problems under consideration to delineate the boundaries of the phase-change region.

Even if transformer models learn to solve problem instances of numerical satisfiability, this is not indicative of their ability to learn logical semantics and rules of inference. Indeed, this aspect requires a comprehensive evaluation of these models' generalisation ability. Consequently, we evaluate transformer models' generalisation ability across several axes. (1) Vocabulary invariance: Do Transformers demonstrate sensitivity to out-of-vocabulary (OOV) terms? (2) Numeracy invariance: Can Transformers generalise to different numerical values that were unseen during fine-tuning? (3) Scale invariance: Are Transformers capable of generalising to problems of larger scope? (4) Noise invariance: How sensitive are Transformers to noisy sentences which do not affect the satisfiability of the problem? Together, these evaluative dimensions furnish a comprehensive understanding of Transformers' generalisation abilities, spanning both their comprehension of logical and numerical semantics in natural language and their proficiency in learning and applying mathematical and logical rules.

The contributions of this paper are as follows. (1) Based on the principles of mapping satisfiability problems to integer linear problems (Pratt-Hartmann, 2023, Ch. 7), we design an algorithm to construct numerical satisfiability problems. (2) We explore the problem space of the constructed numerical satisfiability problems to establish the phase-change region. (3) We design a systematic investigation to evaluate Transformers' ability to learn logical semantics and rules of inference from natural language text. (4) We evaluate a diverse range of Transformers, encompassing varying sizes and architectures, in both fine-tuned and zero-shot/few-shot settings.

2 Methodology

2.1 Language Fragments

When constructing our dataset, we employ language fragments in sentence generation. We define a fragment of a natural language $\mathcal{L}$ to be a set of sentence forms in $\mathcal{L}$ equipped with semantics translating those sentences to some formal system such as first-order logic (Pratt-Hartmann, 2004; Pratt-Hartmann and Third, 2006). Say that a sentence template in $\mathcal{L}$ is a sentence of $\mathcal{L}$ in which certain open-class words have been replaced by schematic variables. For instance, "Every A is a B" represents an English sentence template where common (count) nouns are replaced by variables A and B. Then, one way to define a language fragment is by a set of finite templates. For example, the *Aristotelian syllogistic* (Aristotle, 1938) can be defined by the following set of templates,

Every A is a B	*Some A is a B*
No A is a B	*Some A is not a B*

We introduce two modifications to the Aristotelian syllogistic: (1) allow negations in the subject. (2) replace the quantifiers *all*, *some*, *no* with the numerical quantifiers *at least* K and *at most* K, where $K \in \mathbb{N}^+$ ($\mathbb{N}^+ = \{1, 2, 3, \dots\}$). The resulting fragment, which we refer to as the *counting fragment* or $\mathcal{C}$, can be defined by the following set of templates,

At least K *A are B*	*At least* K *A are not B*
At least K *non-A are B*	*At least* K *non-A are not B*
At most K *A are B*	*At most* K *A are not B*
At most K *non-A are B*	*At most* K *non-A are not B*

Then the numerical satisfiability we consider is as follows,

Given: a finite set S of sentences in $\mathcal{C}$,
Return: True if S is satisfiable; False otherwise.

The problem of determining the satisfiability of a set of sentences in $\mathcal{C}$ is NPTIME-complete (Kuncak and Rinard, 2007). Given a set of sentences in $\mathcal{C}$, we derive a system of linear inequalities as described in the next section. We remark in passing that this idea in fact has a long history (Jevons, 1871).

2.2 Reduction to Integer Linear Problems

Let us say we are given a set of sentences S in $\mathcal{C}$ over a signature of unary (1-place) predicates $P_1, P_2, \dots, P_n$, for which we aim to determine the satisfiability.

If ψ is a formula let $\pm\psi$ be either ψ or $\neg\psi$. We call the conjunction of the form $\pm P_1(x) \wedge \pm P_2(x) \wedge \cdots \wedge \pm P_n(x)$ an *atomic 1-type* over the signature $(P_1, P_2, \dots, P_n)$; in the sequel, we omit the quantifier "atomic" for brevity. We list the 1-types over $(P_1, P_2, \dots, P_n)$ in some fixed order $\pi_1, \pi_2, \dots, \pi_N$ where $N = 2^n$.

If $\mathfrak{A}$ is a structure interpreting the signature $P_1, P_2, \dots, P_n$, over some domain A, we define the *histogram of* $\mathfrak{A}$,denoted $hist(\mathfrak{A})$, to be the N-tuple $(w_1, w_2, \dots w_N)$ where for all i,

$$w_i = |\{a \in A \colon \mathfrak{A} \models \pi_i[a]\}|.$$

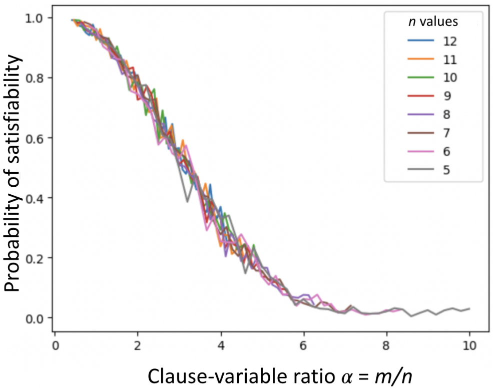


Figure 2: The variation of probability of satisfiability with clause variable ratio for the counting fragment $\mathcal{C}$

It is easy to see that any sentence s in the language $\mathcal{C}$ can be naturally translated into a linear inequality in variables $w_1, \dots, w_n$ satisfied by the histogram of a structure $\mathfrak{A}$ just in case s is true in $\mathfrak{A}$. For example, the sentence "At least K P_a are not P_b" can be formulated as the inequality

$$\sum_{i=1}^{N}\{w_i \colon \models \pi_i \rightarrow (P_a(x) \wedge \neg P_b(x))\} \geq K, \quad (1)$$

stating that the frequencies of those 1-types entailing both $P_a(x)$ and $\neg P_b(x)$ sum to at least K. We add the inequality $\sum_{i=1}^{N} w_i \geq 1$ to rule out the zero solutions (corresponding to the standard assumption that the domain of quantification is non-empty). In this way, given a set of sentences, $S = \{s_1, \dots, s_m\}$, we can construct a system of linear inequalities $\mathcal{E} = \{E_1, \dots, E_m, \left(\sum_{i=1}^{N} w_i \geq 1\right)\}$ such that S is satisfiable if and only if $\mathcal{E}$ has a solution over the natural numbers. Since this latter problem has a well-known algorithmic solution, so has the former.

2.3 Phase-change Region and Data Construction

When considering the algorithmic solution of problems in logic, it is important to realise that not all instances are equally difficult. A case in point is provided by the well-known problem SAT: given a set Γ of clauses in propositional logic, determine whether Γ has a satisfying truth-value assignment. (In this context, *clause* is a disjunction of proposition letters or negated proposition letters.) Consider a problem instance with m clauses featuring a signature of size n (number of variables). If the clause variable ratio $\alpha = \frac{m}{n}$ is large, we have a highly constrained problem instance with few degrees of freedom; hence, the probability of satisfiability is close to zero. Conversely, if α is small, we have a relatively unconstrained problem with many degrees of freedom; hence, the probability of satisfiability is close to unity. In either case, it is easy for a learning algorithm to determine satisfiability reliably. Only for values in a narrow range of α commonly referred to as the *phase-change region*, is the problem challenging (Selman et al., 1996; Mitchell and Levesque, 1996). A similar phenomenon can be observed for the counting fragment we considered in this study. Figure 2 depicts the variation of the probability of satisfiability with the clause-variable ratio.

In our data construction framework, we fix a large vocabulary $\mathcal{V}$ of English common count nouns. When constructing a data point, we randomly select a signature consisting of n elements of $\mathcal{V}$, and construct m sentences of the language $\mathcal{C}$ over this signature. We systematically vary n within the interval $[5, 12]$ ($n_{min} = 5, n_{max} = 12$). Through this variation, we investigate the relationship between the probability of satisfiability and the clause variable ratio α, aiming to establish an appropriate range for the parameter α. Subsequently, guided by our exploratory analysis, we select specific α values to ensure that the probability of satisfiability falls within the predefined interval $[0.4, 0.6]$ ($\alpha_{min} = 0.4, \alpha_{max} = 0.6$). We design our algorithm to generate numerical satisfiability problems with appropriate ranges for α, n and K. We vary K between $K_{min} = 10$ and $K_{max} = 50$, and utilise a set of professions (eg: doctors, artists) as the vocabulary $\mathcal{V}$. We translate the set of generated sentences into a system of linear inequalities as explained above, and use the ILP solver in the Z3 to determine whether this system has a solution. Following this set-up, we construct a training set of $130K$ instances and a test set with $12K$ instances[2]. A more detailed explanation of the dataset is provided in Appendix A.

2.3.1 Data construction: Out-of-distribution

To evaluate Transformers' ability for generalisation to out-of-distribution data, we construct several out-of-distribution datasets following two approaches. In the first approach, we introduce perturbations to the in-distribution test set. We employ this approach to construct data to test for vocabulary invariance and noise invariance. In the second approach, we construct out-of-distribution data by introducing different input configurations. We employ this approach to construct data to test for numerical invariance and scale invariance.

Algorithm 1 Data Construction - Natural language satisfiability

Input : Vocabulary of nouns $\mathcal{V}$, number of variables $[n_{min}, n_{max}]$, numerical value range $[K_{min}, K_{max}]$, clause variable ratio α range $[\alpha_{min}, \alpha_{max}]$
Output : natural language satisfiability dataset $\mathcal{D}$

```
$D \leftarrow \{\}$
repeat
    $n \leftarrow$ randomly sample from $[n_{min}, n_{max}]$
    $v \leftarrow$ randomly sample n nouns from $\mathcal{V}$
    $m \leftarrow$ sample $m$ s.t $\alpha_{min} \leq \frac{m}{n} \leq \alpha_{max}$
    for $j = 1$ to $m$ do
        $A, B \leftarrow$ randomly sample two nouns from $v$
        $t_j \leftarrow$ randomly sample template from the counting fragment $\mathcal{C}$
        $K \leftarrow$ randomly sample from the range $[K_{min}, K_{max}]$
        $s_j \leftarrow$ substitute $A, B, K$ for schematic variables in $t_j$
        $E_j \leftarrow$ translate $s_j$ to a linear inequality
    end for
    $z \leftarrow \texttt{Solver}(E_1, \ldots, E_m, (\sum_{i=1}^{N} w_i \geq 1))$
    if $z$ is not None then
        $\ell \leftarrow True$
    else
        $\ell \leftarrow False$
    end if
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{\{s_1, ..., s_m\}, \ell\}$
until stop condition is met
```

Vocabulary invariance: We introduce perturbations to the test set by incorporating OOV nouns. These perturbations are categorised into two distinct types. First, nouns within the sentences of the test set are substituted with semantically similar nouns that were not encountered during the fine-tuning. Second, nouns within the sentences are substituted with symbols adhering to a prescribed format denoted as P_J, where $J \in \mathbb{N}^+$ (eg: P_1, P_2, ...). The perturbed test sets resulting from the first and second approaches are designated as *Vocab*$_{prof}$ and *Vocab*$_{PS}$, respectively.

Numerical invariance: We construct three separate test sets employing Algorithm 1, distinguished by the range within which K varies. We employ the ranges $[100, 500]$, $[1000, 5000]$, and $[100, 5000]$, and denote the resulting datasets by *Num*$_{[100,500]}$, *Num*$_{[1000,5000]}$ and *Num*$_{[100,5000]}$, respectively.

Scale invariance: To explore the adaptability of Transformers in tackling problems of larger scope, we devise a distinct test set utilising Algorithm 1. We define the number of variables n to span

[2] Link to the dataset and code is anonymised for blind-review purposes.

from 13 to 16 ($n_{min} = 13, n_{max} = 16$), thereby ensuring that the problem instances presented to the models are more intricate than those encountered during the fine-tuning process. Henceforth, the generated dataset is referred to as *Scale*.

Noise invariance: To evaluate Transformers' susceptibility to noise, we introduce perturbations to problem instances within the test dataset. These perturbations involve the introduction of sentences that do not alter the satisfiability of the instances. Specifically, given a problem instance q_1 employing a signature v_1, we generate an alternate satisfiability problem instance q_2 employing a mutually exclusive signature v_2. Consequently, the inclusion of sentences from q_2 into q_1 does not influence the satisfiability of q_1. Henceforth, we refer to this perturbed dataset as $Noise$. Furthermore, we partition the *Noise* dataset into two distinct subsets based on the number of clauses present. If the number of clauses m in a given instance exceeds the maximum number of clauses present in the training set, we categorise that dataset as *Noise*$_{>m}$, otherwise, *Noise*$_{\leq m}$. We outline the out-of-distribution datasets in more detail in Appendix B.

3 Experimental Setup

3.1 Fine-tuning

To examine Transformers' ability to solve numerical satisfiability problem instances, we fine-tune two well-known Transformers that have a proven track record in textual reasoning tasks: Flan-T5 (Chung et al., 2022) and Gemma (Team et al., 2024).

Flan-T5 Flan-T5 is an instruction-fine-tuned variant of the T5 model architecture (Wei et al., 2022a) and is considered to be an improvement to the vanilla T5 model. T5-based model architectures have a well-documented track record of solving formal reasoning problems including satisfiability (Madusanka et al., 2023b; Richardson and Sabharwal, 2021; Clark et al., 2021). We utilise the `Flan-T5-base` model with 220M parameters, `Flan-T5-large` with 770M parameters and `Flan-T5-XL` model with 3B parameters.

Gemma Gemma is an open-source Transformer model architected upon the foundation of the Gemini models (Team et al., 2023). Gemma has achieved state-of-the-art performance on various language-related tasks when compared to models of similar scale and even some larger models. For this investigation, we employ the 2B parameter version, referred to as `Gemma-2b`.

3.2 Zero-shot and Few-shot settings

In addition to fine-tuned models, we evaluate a range of closed and open-source Large Language Models (LLMs), including GPT-3.5-turbo (Kojima et al., 2022), GPT-4 (OpenAI, 2023), and Mistral-7B (Jiang et al., 2023) on a subset of 300 test examples using different prompting techniques. Recent work has suggested that pre-trained LLMs might exhibit emergent reasoning capabilities when the number of parameters scales above a certain threshold (Wei et al., 2022b). However, subsequent evidence has started questioning such claims (Schaeffer et al., 2024), resulting in an open debate within the research community. Here, we aim to contribute to this debate by testing whether LLMs can generalise to the numerical satisfiability task. The prompts used for the experiments are shown in Appendix C.

Zero-shot Inference In the zero-shot setting, we simply prompt LLMs with instructions about the task, asking the model to generate a "True" or "False" answer according to whether the set of statements provided as input is satisfiable or not.

Few-shot Inference In addition to the zero-shot setting, we test the ability of LLMs to solve numerical satisfiability problems when provided with in-context examples (Brown et al., 2020). To this end, given a test example, we employ a BM25 retrieval model (Robertson et al., 2009) to select the top k most relevant examples and their corresponding labels from the training set.

Chain-of-Thought Finally, we test LLMs via Chain-of-Thought (CoT) prompting (Wei et al., 2022c), where the models are explicitly queried to generate a step-by-step explanation to derive the final answer. Here, we limit our experiments to GPT 3.5 and Mistral because of budget constraints.

4 Results and Discussion

4.1 In-distribution Evaluation

Although Transformers can be fine-tuned to solve numerical satisfiability problem instances, these models struggle in zero-shot/few-shot settings. As shown in Table 1, fine-tuned Transformers achieve adequate performance when solving in-distribution numerical satisfiability problems. How-

Model	Accuracy	F1 score
Fine-tuning		
`Flan-T5-XL`	**<u>89.41</u>**	**<u>89.72</u>**
`Gemma-2b`	84.39	84.66
Zero-shot		
`Mistral-7b`	49.66	**65.60**
`GPT-3.5-turbo`	49.00	53.78
`GPT-4`	**53.60**	64.42
Few-shot		
`Mistral-7b` (k = 5)	54.33	50.54
`Mistral-7b` (k = 20)	57.00	52.04
`GPT-3.5-turbo` (k = 5)	51.67	49.65
`GPT-3.5-turbo` (k = 20)	49.33	47.59
`GPT-4` (k = 5)	49.50	47.58
`GPT-4` (k = 20)	**57.14**	**58.25**
Chain-of-Thought (CoT)		
`Mistral-7b`	**48.49**	**66.17**
`GPT-3.5-turbo`	46.33	58.82

Table 1: Results on the in-distribution test set. Although Transformers can be fine-tuned to solve numerical satisfiability instances, pre-trained LLMs struggle in zero-shot/few-shot settings.

ever, Transformers encounter notable challenges in zero-shot and few-shot scenarios. Indeed, the accuracy across all Transformer model architectures in zero-shot and few-shot contexts closely approximates that of a random baseline. Notably, certain models, such as `Mistral-7b`, consistently exhibit an inclination to produce "True" (satisfiable) for almost all problem instances. Moreover, our analysis reveals the susceptibility of Transformers to the influence of examples provided for few-shot learning and chain-of-thought, often leading to erroneous conclusions. We hypothesise this is due to the intricate nature of the numerical satisfiability problem. Indeed, numerous cognitive studies underscore the inherent difficulties humans encounter when tasked with even rudimentary formal reasoning exercises (Johnson-Laird and Bara, 1984; Bronkhorst et al., 2020). Given that Transformers are trained on corpora derived from human-generated content, it is unsurprising that these models inherit this limitation.

4.2 Out-of-Distribution Generalisation

Transformers' insensitivity to vocabulary invariance is bounded by semantic similarity. As shown in Table 2, Transformers demonstrate proficiency in generalising to OOV instances when the OOV nouns exhibit semantic similarity to those present within the in-distribution data. Conversely, Transformers encounter difficulties when confronted with OOV nouns lacking semantic resemblance to their in-distribution counterparts. Specifically, the `Gemma-2b` model exhibits minimal robustness in scenarios where the vocabulary comprises nouns lacking semantic similarity, predicting "False" (unsatisfiable) across almost all problem instances. Although `Flan-T5-XL` demonstrates comparatively greater resilience than Gemma, its performance nevertheless registers a notable decline under similar conditions. We hypothesise this is due to Transformers' inability to separate logical semantics from non-logical semantics. Another contributing factor to this decline in performance could be the relatively lower term frequency associated with terms of type P_J compared to conventional nouns describing professions. Prior research has underscored the impact of term frequency on model performance (Razeghi et al., 2022). However, given the substantial magnitude of the observed drop in performance, we contend that the more plausible cause is the lack of semantic similarity rather than from lower term frequency.

Transformers exhibit limited numerical invariance and exhibit learning superficial cues. As depicted in Table 2, Transformers generalise well when the range of K is $[100-500]$ or $[1000-5000]$, but encounter difficulties when the range of K is $[100-5000]$. Although the performance of both models is well above the random baseline for K with a range $[100-5000]$, they experience a drop of $10+$ accuracy points. The numerical quantifiers impose a comparative operation between the logical and numerical elements involved in the expressions. Consequently, we hypothesise that Transformers prioritise the first digit over holistic consideration of the entire numerical value. This observed phenomenon underscores the fact that even the more recent billion-parameter variants of Transformers are susceptible to learning superficial cues, reminiscent of their earlier, smaller counterparts (McCoy et al., 2019; Glockner et al., 2018). Moreover, this limited generalisation ability to numerical invariance coupled with their failure to adapt to OOV that lacks semantic similarity indicates that these models still struggle to learn logical semantics. This stands in stark contrast to the demonstrated proficiency of Transformers in grasping the logical semantics associated with simpler reasoning problems, such as model-checking (Madusanka et al., 2023a,b). Consequently, we posit that the Trans-

Model	**In-distribution**		**Vocabulary Invariance**				**Numeracy Invariance**					
			Vocab$_{prof}$		*Vocab*$_{Ps}$		*Num*$_{[100-500]}$		*Num*$_{[1000-5000]}$		*Num*$_{[100-5000]}$	
	Acc	*F1*	*Acc*	*F1*	*Acc*	*F1*	*Acc*	*F1*	*Acc*	*F1*	*Acc*	*F1*
`Flan-T5-XL`	89.41	89.72	86.48	86.83	73.85	67.85	85.27	85.75	83.46	83.82	78.55	79.84
`Gemma-2b`	84.39	84.66	79.60	79.41	50.61	2.85	79.13	79.95	76.23	77.34	68.05	70.18

Table 2: Results on vocabulary invariance and numerical invariance datasets. We found that Transformers exhibit limited generalisation to vocabulary and numerical invariance.

Model	**In-distribution**		**Scale Invariance**		**Noise invariance**					
			Scale		*Noise*		*Noise*$_{\leq m}$		*Noise*$_{>m}$	
	Acc	*F1*	*Acc*	*F1*	*Acc*	*F1*	*Acc*	*F1*	*Acc*	*F1*
`Flan-T5-XL`	89.41	89.72	78.38	78.14	72.08	65.15	73.41	68.70	70.77	61.24
`Gemma-2b`	84.39	84.66	73.44	72.95	68.64	61.98	69.15	64.58	68.15	59.15

Table 3: Results on scale invariance and noise invariance datasets. We found that Transformers exhibit minimal generalisation to scale and noise invariance.

formers' capacity to comprehend logical semantics is intricately tied to the complexity inherent within the reasoning tasks at hand.

Transformers fail to exhibit scale invariance. As shown in Table 3, Transformers exhibit a failure to generalise effectively when confronted with problem instances characterised by a higher number of variables than those encountered during training. This outcome aligns with prior investigations into scale invariance, which have yielded analogous findings (Schlegel et al., 2022; Madusanka et al., 2024). We posit that this deficiency in scale invariance arises from the Transformers' inability to learn rules of inference gleaned from natural language texts. As previously noted, prior research suggests that Transformers attempt to solve multi-step reasoning tasks through linearised path-matching strategies (Dziri et al., 2023). Additionally, during training and fine-tuning phases, Transformers are known to acquire shortcuts via pattern-matching (Liu et al., 2023). While this approach may prove expedient for in-distribution evaluation, it does not result in robust generalisation when subjected to out-of-domain testing. Indeed, we posit that this behaviour impedes the Transformers' capacity to learn rules of inference, thereby constraining their ability to demonstrate scale invariance.

Transformers demonstrate sensitivity to noise. As shown in Table 3, the introduction of noisy sentences into problem instances precipitates a notable decline in the performance of Transformers. When the inclusion of noisy sentences extends the overall problem length beyond the scope of the model's training data, the scenario resembles a variant of the length generalisation problem. Previous research has indicated Transformers' propensity to struggle with length generalisation (Anil et al., 2022; Press et al., 2022), thus explaining the observed decline for the $Noise_{>m}$ test set. However, even in scenarios where the problem length remains within the bounds of the model's training data, the performance of Transformers still undergoes a significant decline. Notably, there exists a negative correlation between the performance of Transformers and the number of noisy sentences introduced into the problem instances. We posit that this phenomenon arises from a fundamental shift in the underlying problem structure induced by the introduction of noisy sentences. For instance, consider the scenario where sentences from problem q_2, constructed with a mutually exclusive vocabulary to problem q_1 and determined to be satisfiable, are incorporated into problem q_1. The resultant problem q comprises two sub-problems for which the models have to derive the satisfiability. We posit this change in problem structure influences Transformers' inability to exhibit noise invariance.

Models of different scales exhibit analogous generalisation patterns. As depicted in Figure 3, Transformers of varying sizes exhibit similar patterns of generalisation. For instance, the `Flan-T5-base` model achieves an accuracy of 58.35%, while the `Flan-T5-large` model attains 72.23%, both notably lower than the accuracy achieved by the `Flan-T5-XL` model, which

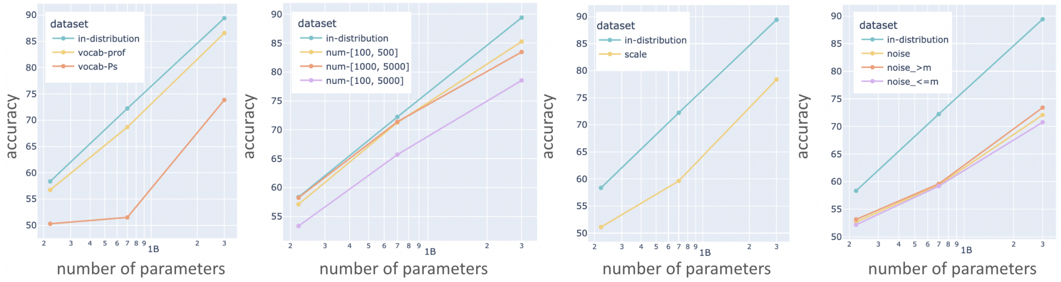


Figure 3: Variation of accuracy level with the number of parameters for the Flan-T5 model. We employed `Flan-T5-base`, `Flan-T5-large` and `Flan-T5-XL` models and found they exhibit similar generalisation patterns.

stands at 89.41%. However, their relative disparities in generalisation performance to out-of-distribution datasets mirror the pattern observed with `Flan-T5-XL`. Previous investigations exploring the impact of model size on generalisation have yielded congruent findings (Anil et al., 2022). It is noteworthy that despite experiencing a marked decline in performance for out-of-distribution datasets, the generalisation proficiency of `Flan-T5-XL` remains significantly above the random baseline. Nonetheless, we assert that Transformer models still have significant strides to make before achieving the capability to learn rules of inference and comprehend logical semantics.

5 Related Work

Extensive research has been dedicated to exploring the generalisation capabilities of neural models, including Transformers. These investigations encompass a spectrum of generalisation forms, such as length generalisation (Press et al., 2022; Anil et al., 2022; Valentino et al., 2024), easy-to-hard generalisation (Schwarzschild et al., 2021; Bansal et al., 2022; Meadows et al., 2024), and compositional task generalisation (Dziri et al., 2023). However, there remains a notable gap in the literature concerning the application of these models to textual inference problems, particularly within the realm of natural language satisfiability. Consequently, our study aims to delve into the various facets of generalisation exhibited by Transformer models when confronted with a variant of the natural language satisfiability problem, known as the numerical satisfiability problem.

Transformers have demonstrated impressive performance on various formal reasoning tasks (Richardson et al., 2020; Lin et al., 2019; Tafjord et al., 2021; Creswell et al., 2023). All of these reasoning tasks can be reduced to the problem of determining the *satisfiability* of a set of sentences. Neural approaches such as graph neural networks have been used extensively to solve instances of satisfiability problems (Xu et al., 2020; Cameron et al., 2020; Selsam et al., 2019). Richardson and Sabharwal (2021) extended this research into natural language by employing language fragments for data construction. Their investigation also diverges from prior research as they employ Transformers rather than graph neural approaches. Schlegel et al. (2022) and Madusanka et.al (2024) extended this work into fragments of first-order logic. Building on these foundational contributions, the present study investigates the capacity of Transformer architectures to address numerical satisfiability problems. Our methodological approach is notably inspired by the work of Madusanka et al. (2024) on natural language satisfiability, particularly their strategy of sampling problem instances from the phase-change region to generate instances of high computational difficulty. To the best of our knowledge, this work constitutes the first systematic exploration of the numerical satisfiability problem within the context of Transformer-based models. We underscore that this extension to numerical satisfiability introduces non-trivial challenges, requiring significant methodological adaptations. Furthermore, our study distinguishes itself from existing literature by placing a specific emphasis on the generalisation capabilities of transformer models, an aspect that has not been explicitly addressed in prior work on satisfiability.

6 Conclusion

We probe the generalisation ability of Transformers on the satisfiability problem for a simple fragment of English featuring numerical quantification. We find that fine-tuned Transformers exhibit limited generalisation to vocabulary and numerical invariance while exhibiting minimal scale and noise invariance. Furthermore, our investigation indicates that Transformers in zero-shot and few-shot settings find numerical satisfiability problems more or less unsolvable. We emphasise that certain model architectures, such as Flan-T5-XL, demonstrate some robustness when assessed against out-of-distribution (OOD) test sets. However, the level of robustness is not sufficient to indicate that these models have the ability to learn and understand logical and mathematical rules. Therefore, we assert that substantial research efforts are warranted to improve Transformer models' ability to generalise to OOD data over a complex reasoning task, which we leave for future work.

7 Limitations

Due to the empirical nature of our work, it suffers from inductive dilemmas on three fronts. First, while we explore several Transformer models with established proficiency in formal reasoning tasks, it remains conceivable that there exist other Transformer architectures whose behaviour diverges from the empirical findings delineated in this study. Second, in our study, we construct the numerical satisfiability problems by employing language fragments involving numerical quantifiers "At least K" and "At most K". However, it is important to acknowledge that these quantifiers represent only a subset of the broader spectrum of numerical quantifiers in existence. Thus, there may exist alternative quantifiers for which Transformer models exhibit differing behaviour. Third, we consider in-context learning via few-shot examples and chain-of-thought prompting defined through prompts presented in Appendix C.1. However, it is plausible that alternative prompting methodologies may yield superior performance for Transformer models.

Acknowledgments

The authors would like to acknowledge the use of the Computational Shared Facility at the University of Manchester and the Department of Computer Science at the University for funding this research. Ian Pratt-Hartmann was supported by the Polish NCN, grant 2018/31/B/ST6/03662. Moreover, the work of Iqra Zahid is part of the IN-CYPHER programme and is supported by the National Research Foundation, Prime Minister's Office, Singapore, under its Campus for Research Excellence and Technological Enterprise (CREATE) programme. Finally, we also thank the anonymous reviewers from the ARR February 2025 cycle for their valuable feedback.

References

Cem Anil, Yuhuai Wu, Anders Johan Andreassen, Aitor Lewkowycz, Vedant Misra, Vinay Venkatesh Ramasesh, Ambrose Slone, Guy Gur-Ari, Ethan Dyer, and Behnam Neyshabur. 2022. Exploring length generalization in large language models. In *Advances in Neural Information Processing Systems*.

Aristotle. 1938. *The Categories, On Interpretation, Prior Analytics*. Loeb Classical Library. Harvard University Press, Cambridge, MA. Tr. H. Cooke and H. Tredennick.

Arpit Bansal, Avi Schwarzschild, Eitan Borgnia, Zeyad Emam, Furong Huang, Micah Goldblum, and Tom Goldstein. 2022. End-to-end algorithm synthesis with recurrent networks: Extrapolation without overthinking. In *Advances in Neural Information Processing Systems*.

Hugo Bronkhorst, Gerrit Roorda, Cor Suhre, and Martin Goedhart. 2020. Logical reasoning in formal and everyday reasoning tasks. *International Journal of Science and Mathematics Education*, 18:1673–1694.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Chris Cameron, Rex Chen, Jason Hartford, and Kevin Leyton-Brown. 2020. Predicting propositional satisfiability via end-to-end learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3324–3331.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Peter Clark, Oyvind Tafjord, and Kyle Richardson. 2021. Transformers as soft reasoners over language. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, IJCAI'20.

Antonia Creswell, Murray Shanahan, and Irina Higgins. 2023. Selection-inference: Exploiting large language models for interpretable logical reasoning. In *The*

Eleventh International Conference on Learning Representations.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Sean Welleck, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D. Hwang, Soumya Sanyal, Xiang Ren, Allyson Ettinger, Zaid Harchaoui, and Yejin Choi. 2023. Faith and fate: Limits of transformers on compositionality. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking NLI systems with sentences that require simple lexical inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 650–655, Melbourne, Australia. Association for Computational Linguistics.

William Stanley Jevons. 1871. On a general system of numerically definite reasoning. *Memoirs of the Manchester Literary and Philosophical Society*, 4:330–352.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Phillip N Johnson-Laird and Bruno G Bara. 1984. *Syllogistic inference*, volume 16. Elsevier.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*.

V. Kuncak and M. Rinard. 2007. Towards efficient satisfiability checking for boolean algebra with presburger arithmetic. In *The Twenty-First International Conference on Automated Deduction CADE-21*, Berlin. Springer Verlag.

Kevin Lin, Oyvind Tafjord, Peter Clark, and Matt Gardner. 2019. Reasoning over paragraph effects in situations. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 58–62, Hong Kong, China. Association for Computational Linguistics.

Bingbin Liu, Jordan T. Ash, Surbhi Goel, Akshay Krishnamurthy, and Cyril Zhang. 2023. Transformers learn shortcuts to automata. In *The Eleventh International Conference on Learning Representations*.

Tharindu Madusanka, Ian Pratt-hartmann, and Riza Batista-navarro. 2023a. Identifying the limits of transformers when model-checking with natural language. In *Forthcoming in Proceedings of The 17th Conference of the European Chapter of the Association for Computational Linguistics*, volume arXiv:1503.06733.

Tharindu Madusanka, Ian Pratt-Hartmann, and Riza Batista-Navarro. 2024. Natural language satisfiability: Exploring the problem distribution and evaluating transformer-based language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15278–15294, Bangkok, Thailand. Association for Computational Linguistics.

Tharindu Madusanka, Iqra Zahid, Hao Li, Ian Pratt-Hartmann, and Riza Batista-Navarro. 2023b. Not all quantifiers are equal: Probing transformer-based language models' understanding of generalised quantifiers. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8680–8692, Singapore. Association for Computational Linguistics.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 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*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Jordan Meadows, Marco Valentino, Damien Teney, and Andre Freitas. 2024. A symbolic framework for evaluating mathematical reasoning and generalisation with transformers. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1505–1523, Mexico City, Mexico. Association for Computational Linguistics.

David G Mitchell and Hector J Levesque. 1996. Some pitfalls for experimenters with random SAT. *Artificial Intelligence*, 81(1-2):111–125.

OpenAI. 2023. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll 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 F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Ian Pratt-Hartmann. 2004. Fragments of language. *Journal of Logic, Language and Information*, 13(2):207–223.

Ian Pratt-Hartmann. 2023. *Fragments of First-Order Logic*. Oxford University Press, Oxford, UK.

Ian Pratt-Hartmann and Allan Third. 2006. More fragments of language. *Notre Dame Journal of Formal Logic*, 47(2):151–177.

Ofir Press, Noah Smith, and Mike Lewis. 2022. Train short, test long: Attention with linear biases enables input length extrapolation. In *International Conference on Learning Representations*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Yasaman Razeghi, Robert L Logan IV, Matt Gardner, and Sameer Singh. 2022. Impact of pretraining term frequencies on few-shot numerical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 840–854, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Kyle Richardson, Hai Hu, Lawrence Moss, and Ashish Sabharwal. 2020. Probing natural language inference models through semantic fragments. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8713–8721.

Kyle Richardson and Ashish Sabharwal. 2021. Pushing the limits of rule reasoning in transformers through natural language satisfiability. In *AAAI Conference on Artificial Intelligence*.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. 2024. Are emergent abilities of large language models a mirage? *Advances in Neural Information Processing Systems*, 36.

Viktor Schlegel, Kamen Pavlov, and Ian Pratt-Hartmann. 2022. Can transformers reason in fragments of natural language? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11184–11199, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Avi Schwarzschild, Eitan Borgnia, Arjun Gupta, Furong Huang, Uzi Vishkin, Micah Goldblum, and Tom Goldstein. 2021. Can you learn an algorithm? generalizing from easy to hard problems with recurrent networks. In *Advances in Neural Information Processing Systems*, volume 34, pages 6695–6706. Curran Associates, Inc.

Bart Selman, David G Mitchell, and Hector J Levesque. 1996. Generating hard satisfiability problems. *Artificial intelligence*, 81(1-2):17–29.

Daniel Selsam, Matthew Lamm, Benedikt Bünz, Percy Liang, Leonardo de Moura, and David L. Dill. 2019. Learning a SAT solver from single-bit supervision. In *International Conference on Learning Representations*.

Richard Shin, Neel Kant, Kavi Gupta, Chris Bender, Brandon Trabucco, Rishabh Singh, and Dawn Song. 2019. Synthetic datasets for neural program synthesis. In *International Conference on Learning Representations*.

Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. 2021. ProofWriter: Generating implications, proofs, and abductive statements over natural language. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3621–3634, Online. Association for Computational Linguistics.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Marco Valentino, Jordan Meadows, Lan Zhang, and Andre Freitas. 2024. Multi-operational mathematical derivations in latent space. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1446–1458, Mexico City, Mexico. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022a. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022b. Emergent abilities of large language models. *Transactions on Machine Learning Research*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022c. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Zhengxuan Wu, Elisa Kreiss, Desmond Ong, and Christopher Potts. 2021. ReaSCAN: Compositional reasoning in language grounding. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

Keyulu Xu, Jingling Li, Mozhi Zhang, Simon S. Du, Ken ichi Kawarabayashi, and Stefanie Jegelka. 2020. What can neural networks reason about? In *International Conference on Learning Representations*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Neural Information Processing Systems*, volume 32.

A Appendix: Dataset Details

Formally the dataset for each objective takes the form, $\{((S)^{(d)}, \ell^{(d)})\}_d^{|D|}$, where ($S$ are set of sentences concatenated together, and $\ell \in \{\text{True, False}\}$ is the label. We have depicted an instance of a Numerical satisfiability problem along with its perturbations (vocabulary and noise) in Figure 4

A.1 In-distribution datasets

In our experimental setup, we generated a training dataset comprising 130K instances and an in-distribution testing dataset consisting of 12K instances employing Algorithm 1. Across these datasets, we varied the number of variables from 5 to 12, while adjusting the alpha ratio within the range of 2.54 to 3.71. The alpha ratio was selected based on an exploration of how the probability of satisfiability correlates with the clause-variable ratio. As detailed in the Methodology section, we chose the alpha value such that the probability of satisfiability falls within the range of 0.4 to 0.6, ensuring that the resulting number of clauses falls between 13 and 45. Additionally, we varied the range of K, from 10 to 50. Furthermore, We employed a list of nouns comprised of professions as our common noun vocabulary. The vocabulary contains 155 nouns. We emphasise that the problem space is sufficiently large that no two problems would be equal.

		Size
Vocabulary invariance	$Vocab_{prof}$	12000
	$Vocab_{Ps}$	12000
Numerical invariance	$Num_{[100-500]}$	2000
	$Num_{[1000-5000]}$	2000
	$Num_{[100-5000]}$	2000
Scale invariance	*Scale*	800
Nosie invariance	*Noise*	12000
	$Noise_{\leq m}$	5921
	$Noise_{>m}$	6079

Table 4: The number of problem instances in each of out-of-distribution test sets.

A.2 Out-of-distribution datasets

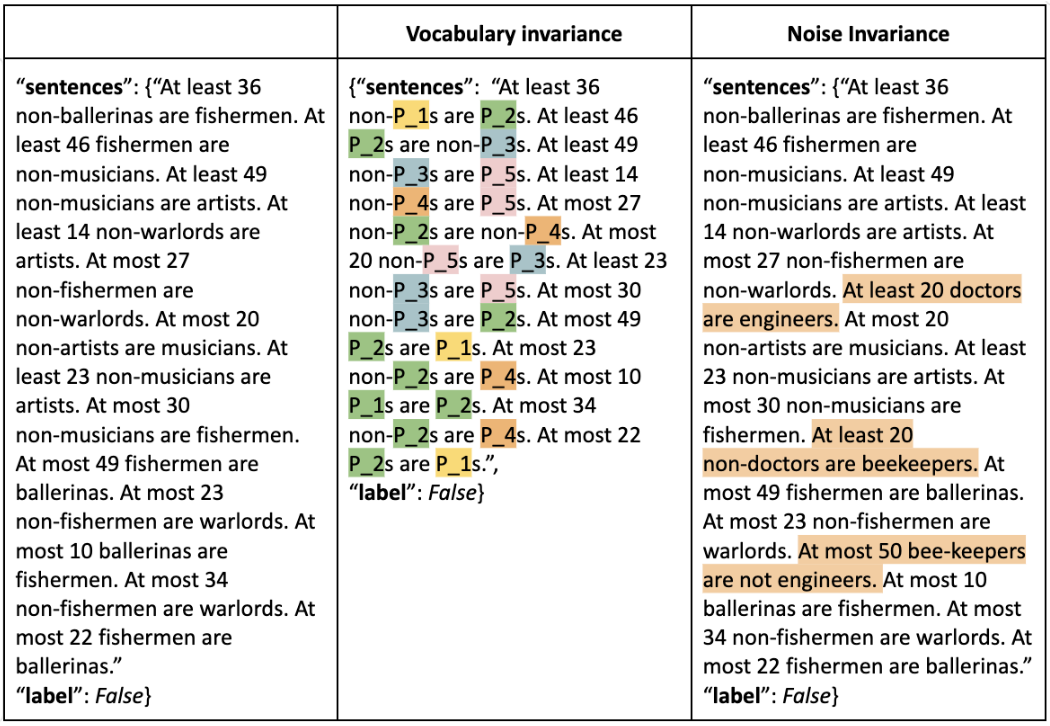
We constructed several out-of-distribution datasets, and the size of each of them is detailed in Table 4. The table 5 depicts mean, minimum, and maximum problem instance lengths.

The size of the scale test set is low due to computational time constraints involved in the data construction setup. Indeed, an n-variable numerical satisfiability problem translated to a 2^n variable integer linear problem that the integer linear solver needs to solve.

B Appendix: Fine-tuning Details

We chose two transformer models of different architectures of fine-tuning[3]: Flan-T5 and Gemma. Both models boast a commendable track record in textual entailment tasks and have demonstrated state-of-the-art performance compared to other models of comparable size. Flan-T5 model is based on the T5 architecture and is an encoder-decoder model, while Gemma is a decoder-only model (Vaswani et al., 2017). Moreover, unlike Flan-T5, Gemma leverages reinforcement learning from human feedback for instruction fine-tuning (Ouyang et al., 2022). As the primary focus of our study is to determine the extent to which transformer models demonstrate generalisation capabilities, we did not perform extensive hyperparameter tuning on in-distribution data. However, we did explore several variations of hyperparameters, including batch

[3] We also fine-tune phi-2 model with 2.7B parameters. However, the model did not optimise properly, therefore, we excluded it from the experimental setup



	Vocabulary invariance	**Noise Invariance**
"**sentences**": {"At least 36 non-ballerinas are fishermen. At least 46 fishermen are non-musicians. At least 49 non-musicians are artists. At least 14 non-warlords are artists. At most 27 non-fishermen are non-warlords. At most 20 non-artists are musicians. At least 23 non-musicians are artists. At most 30 non-musicians are fishermen. At most 49 fishermen are ballerinas. At most 23 non-fishermen are warlords. At most 10 ballerinas are fishermen. At most 34 non-fishermen are warlords. At most 22 fishermen are ballerinas." "**label**": *False*}	{"**sentences**": "At least 36 non-P_1s are P_2s. At least 46 P_2s are non-P_3s. At least 49 non-P_3s are P_5s. At least 14 non-P_4s are P_5s. At most 27 non-P_2s are non-P_4s. At most 20 non-P_5s are P_3s. At least 23 non-P_3s are P_5s. At most 30 non-P_3s are P_2s. At most 49 P_2s are P_1s. At most 23 non-P_2s are P_4s. At most 10 P_1s are P_2s. At most 34 non-P_2s are P_4s. At most 22 P_2s are P_1s.", "**label**": *False*}	"**sentences**": {"At least 36 non-ballerinas are fishermen. At least 46 fishermen are non-musicians. At least 49 non-musicians are artists. At least 14 non-warlords are artists. At most 27 non-fishermen are non-warlords. At least 20 doctors are engineers. At most 20 non-artists are musicians. At least 23 non-musicians are artists. At most 30 non-musicians are fishermen. At least 20 non-doctors are beekeepers. At most 49 fishermen are ballerinas. At most 23 non-fishermen are warlords. At most 50 bee-keepers are not engineers. At most 10 ballerinas are fishermen. At most 34 non-fishermen are warlords. At most 22 fishermen are ballerinas." "**label**": *False*}

Figure 4: Instance of numerical satisfiability problem. We modify the vocabulary to form the vocabulary variance dataset and add noise to form the noise invariance dataset as shown.

size and learning rate, and found that the resultant performance remained largely equivalent across these variations. We establish the following hyper-parameter setup when fine-tuning.

Maximum sequence length: We used the maximum sequence 1024 for both Flan-T5 and Gemma models. We did not rely on any truncation, as truncating input could alter the satisfiability of the input sentences.

Training epochs: For both `Gemma-2b` and `Flan-T5-XL` models, we fine-tuned for two epochs. The performance of the models stagnates after the very first epoch. For `Flan-T5-base` and `Flan-T5-large` models, we fine-tuned for 5 epochs.

Gradient accumulation steps: 4

Batch size: Relying on the gradient checkpointing and gradient accumulation, we used 12 as the batch size for `Gemma-2b` and `Flan-T5-XL` models, while using 48 as the batch size for `Flan-T5-base` and `Flan-T5-large`.

Learning Rate: We set the learning rate to 1×10^{-5}. We use the ADAM optimiser with the default parameters $\epsilon = 1 \times 10^{-8}$, $\beta_1 = 0.99$ and $\beta_2 = 0.999$.

dataset	**min**	**max**	**mean**
$Vocab_{prof}$	270	70	147.1
$Vocab_{Ps}$	226	66	123.6
$Num_{[100-500]}$	270	78	161.2
$Num_{[1000-5000]}$	264	78	159.9
$Num_{[100-5000]}$	270	78	158.2
Scale	360	198	272.9
Noise	474	150	276.3
$Noise_{\leq m}$	270	150	233.6
$Noise_{>m}$	474	276	317.5

Table 5: minimum, maximum and mean number of words (tokens) when separated by `SPACE` in each of the test sets

Hardware: 1 Nvidia A100 GPU with 80GB of RAM.

Each Transformer is fine-tuned to predict the label (True/False: True if the set of sentences are satisfiable and False if the set of sentences are unsatisfaible) by reducing the binary cross entropy loss over the target using the Adam optimiser (Kingma and Ba, 2015), and we used the HuggingFace implementation in the fine-tuning process (Wolf et al., 2019).

C Zero-shot, Few-shot, and CoT Details

In order to run experiments on a zero-shot, few-shot, and CoT settings with Mistral, GPT-3.5-turbo and GPT-4, we adopted the Huggingface[4] and OpenAI[5] inference APIs respectively. Here, we employed a non-deterministic setup and reported the best results achieved by each model across 10 different runs. For the few-shot setting, we retrieved the k most relevant examples from the training set based on the BM25 score with the test instance (Robertson et al., 2009). These examples are appended to the zero-shot prompt. We performed experiments with a value of k equal to 5 and 20.

C.1 Prompts

Regarding the pre-trained models, we adopt the following prompt for zero-shot and few-shot inference:

- *“You are an expert in logical satisfiability and numerical reasoning. Determine whether the set of statements in the test example is satisfiable. Your answer must be 'The answer is False' if the test example is unsatisfiable, 'The answer is True' if the test example is satisfiable."*

For few-shot inference, the prompt above is followed by a list of training examples with their correct labels and the set of statements constituting the test example.

To elicit step-by-step reasoning via Chain-of-Thought, we adopt the following prompt:

- *“You are an expert in logical satisfiability and numerical reasoning. Determine whether the set of statements in the test example is satisfiable. Think step-by-step and terminate your reasoning with 'The answer is False' if the test example is unsatisfiable, 'The answer is True' otherwise.”*

D Reducing other Quantifiers

The mechanism for reducing the numerical satisfiability problem to quantifiers included in Fragment $\mathcal{C}$ can be easily adapted to other generalised quantifiers with minor changes to Equation 1. Let K, w_i, π_i, $\pm P_a(x)$ and $\pm P_b(x)$ and have the same definition as mentioned in the Methodology section. Let $\mathcal{W} = (w_1, \ldots w_N)$ and,

$$\lambda(\mathcal{W}) := \sum_{i=1}^{N} \{w_i \colon \models \pi_i \rightarrow (\pm P_a(x) \wedge \pm P_b(x))\}$$

Then other quantifiers can be formulated as below,

More than K: (More than K (non-)$P_a s$ are (not) $P_b s$),

$$\lambda(\mathcal{W}) > K,$$

Less than K: (Less than K (non-)$P_i s$ are (not) $P_j s$),

$$\lambda(\mathcal{W}) < K,$$

K: (K (non-)$P_a s$ are (not) $P_b s$),

$$\lambda(\mathcal{W}) = K,$$

Most: (Most (non-)$P_a s$ are (not) $P_b s$),

$$\lambda(\mathcal{W}) > \sum_{i=1}^{N} \{w_i \colon \models \pi_i \rightarrow (\pm P_a(x) \wedge \mp P_b(x))\}$$

Few: (Few (non-)$P_a s$ are (not) $P_b s$),

$$\lambda(\mathcal{W}) < \sum_{i=1}^{N} \{w_i \colon \models \pi_i \rightarrow (\pm P_a(x) \wedge \mp P_b(x))\}$$

At least r/s (More than $\frac{r}{s}$ (non-)$P_a s$ are (not) $P_b s$),

$$\lambda(\mathcal{W}) \geq \frac{r}{s} \sum_{i=1}^{N} \{w_i \colon \models \pi_i \rightarrow \pm P_a(x)\}$$

Since the primary focus of this study is to investigate the behaviour of Transformers when solving numerical satisfiability problems rather than identifying how their performance varies with different quantifiers, we only focus on the quantifiers mentioned in Fragment $\mathcal{C}$. Furthermore, we emphasise the problem best suited to analyse the effect of different generalise quantifiers is model checking rather than satisfiability.

The mechanism can be further adapted to include relative clauses with minor changes. Consider the sentence “At least K P_as who are non-P_bs are P_cs”, it can be formulated as shown below,

$$\sum_{i=1}^{N} \{w_i \colon \models \pi_i \rightarrow (P_a(x) \wedge \pm P_b(x) \wedge P_c(x))\} \geq K$$

We exclude this type of complex structure considering the low performance acquired by most Transformer models when confronted with simple numerical satisfiability problems.

[4] https://huggingface.co/docs/inference-providers/en/index

[5] https://openai.com/api/