

Transformers in the Service of Description Logic-based Contexts

Angelos Poulis^a, Eleni Tsalapati^a and Manolis Koubarakis^a

^aDept. of Informatics and Telecommunications,
National and Kapodistrian University of Athens
{sdi1900230, etsalapati, koubarak}@di.uoa.gr

Abstract. Recent advancements in transformer-based models have initiated research interests in investigating their ability to learn to perform reasoning tasks. However, most of the contexts used for this purpose are in practice very simple: generated from short (fragments of) first-order logic sentences with only a few logical operators and quantifiers. In this work, we construct the natural language dataset, DELTA_D , using the description logic language \mathcal{ALCQ} . DELTA_D contains 384K examples, and increases in two dimensions: i) reasoning depth, and ii) linguistic complexity. In this way, we systematically investigate the reasoning ability of a supervised fine-tuned DeBERTa-based model and of two large language models (GPT-3.5, GPT-4) with few-shot prompting. Our results demonstrate that the DeBERTa-based model can master the reasoning task and that the performance of GPTs can improve significantly even when a small number of samples is provided (9 shots). We open-source our code and datasets.

1 Introduction

Context (rules & facts):

If someone eats only people that are not kind or furry or that admire someone and that like only big people, then they are not rough and they love at least one people that admire someone kind and they admire someone round.

If someone loves at least three people that are smart or not orange or that eat at most three not cold people or that chase someone not kind, then they admire someone furry.

All people that admire someone furry are smart.

All smart people eat only people that are not kind or furry or that admire someone and that like only big people.

Erin eats Dave.

Bob admires none.

Fiona loves at least three people that are smart or not orange or that eat at most three not cold people or that chase someone not kind.

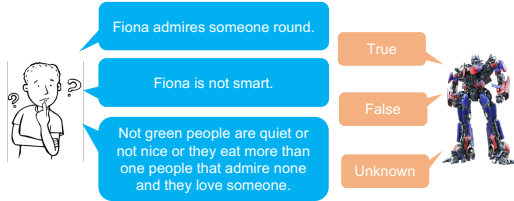


Figure 1: An example from DELTA_D , where the context contains three sentences of high linguistic complexity level and the true and false sentences are of reasoning depth 3.

Description Logic (DL) languages [1] are fragments of first-order logic (FOL) that have evolved into one of the main formalisms for

the representation of conceptual knowledge in a precise and well-defined manner. An expressive and decidable DL language that, besides the standard Boolean operators, supports existential, universal, and numerical constraints is \mathcal{ALCQ} . For instance, in \mathcal{ALCQ} one can formally express sentences like the ones appearing in Fig. 1.

The formal apparatus of DLs allows us to perform deductive reasoning tasks, such as *entailment checking*, i.e., deciding whether a sentence or a set of sentences, logically implies another. With the latest advancements in transformer-based language models (TLMs) a new strand of research has emerged that investigates if TLMs can learn to carry out such tasks over *contexts* expressed in natural language [3, 5, 23, 7]. However, in most cases, the contexts used were either composed of rather short sentences, simple in structure (i.e., their formal representations contain only a few logical operators and quantifiers) [3, 24, 20], or they were of limited size [24, 5].

The fundamental question that this work seeks to answer is: “*How well can TLMs perform inference over contexts produced from an expressive DL language, like \mathcal{ALCQ} ?*”. A natural subsequent research question, in line with the literature [3, 22] but here is for a higher expressivity, is: “*Is the performance of TLMs affected by the reasoning depth required to perform the inference process?*”. A third research question that arises is whether the fragment of the formal language is enough to measure the reasoning ability of a model. For instance, all sentences appearing in Fig. 1 can be expressed formally within \mathcal{ALCQ} , but some are linguistically more complex than others, and one would expect that contexts containing mainly that complex sentences would be hard to process. Hence, the third research question of this paper is: “*Is the performance of TLMs affected by the linguistic complexity of the context?*”.

As discussed by Madusanka et al. [15], the most suited reasoning problem to evaluate the impact of language constructs (like quantifiers and negation) is *textual entailment checking*, i.e., entailment checking with natural language, in a purely logical sense, eliminating the influence of any background or commonsense knowledge. Hence, to answer the above research questions, we have constructed the synthetic dataset DELTA_D (DEscription Logics with TrAnsformers) of 384K examples (context-question-answer-depth-linguistic complexity) based on \mathcal{ALCQ} , where the question is the statement that we check whether it is logically deduced from the context, under the open world assumption. Besides the isolation of the common-sense/background knowledge, the synthetic nature of the dataset allows us to perform a systematic study of the performance of the TLMs, as DELTA_D gradually increases both in reasoning and in lin-

guistic complexity. Additionally, it allows us to eliminate obvious statistical features (e.g., the correlation between the answer “False” and the word “not” in the sentence in question).

We systematically tested the textual entailment checking ability of supervised fine-tuned DeBERTa and few-shot prompting on large language models (GPT-3.5, GPT-4) over Δ_D . Our results show that the performance of the DeBERTa-based model, Δ_M , is marginally affected when the reasoning depth increases or decreases (differently from [24]) with average accuracy 97.5%. Also, we show that the length of the sentences does not affect the performance of the model (accuracy 99.7% in max. reasoning depth and max. length of sentence). To ensure that Δ_M does not overfit on Δ_D , and inspired by [29], we changed the probability distributions used for the dataset generation and the accuracy of the model remained equally good. Additionally, testings to similar datasets [22] returned good results.

To check the impact of semantics on Δ_M , we followed the approach of Tang et al. [23] and we generated a symbolic “translation” of Δ_D by translating the pool of words used for the generation of the synthetic dataset to symbols. Zero-shot testings of Δ_M show that, in contrast to Tang et al. [23], it performs equally well, while GPTs’ accuracy is slightly decreased. This indicates the performance of Δ_M is not affected by the semantics of the dataset (an expected result as the dataset is nonsensical). However, further translation of the dataset to resemble the language used to describe description logic sentences led to a significant reduction in the accuracy of the models. Finally, good testing results of Δ_M on a real-world scenario (fuel cell system diagnostics) demonstrate the potential of TLMs to be utilized in rule-based system diagnostics.

Overall, we make the following contributions:

(C_1) We provide the first large, description logic benchmark of 384K examples. This is a significant contribution because building large benchmarks over expressive logic languages, like \mathcal{ALCQ} , is a challenging task as it requires performing query answering with logic reasoners, a process that can be very time-consuming (~ 1 min. for KB with long rules/facts of our dataset). Both the dataset and the code for its generation are openly available¹

(C_2) We show that TLMs can perform entailment checking with very high accuracy over synthetic natural language contexts generated from \mathcal{ALCQ} sentences. This demonstrates the potential of TLMs to be utilized for scalable reasoning tasks over vast KBs, thus bypassing formal representations required by traditional knowledge-based systems.

(C_3) We show that the performance of TLMs is not affected by the length of the sentences.

(C_4) We show that DeBERTa-based models are not affected by the vocabulary of the dataset.

(C_5) We show how these contributions can be leveraged in a real-world use-case scenario.

2 Background on Description Logics

We can use \mathcal{ALCQ} [1] to represent knowledge about a domain by defining three types of entities: individuals (e.g., *John*), concepts (e.g., *Postdoc*, i.e., the concept describing the entities that are postdocs) and roles (e.g., *teaches*). A *concept expression* C can be formed using these entities, Boolean constructors (\sqcap , \sqcup , \neg), quantifiers (\forall , \exists), and number restrictions (\leq , \geq) recursively as follows:

$C, D := A \mid \top \mid \perp \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid \forall R.C \mid \exists R.C \mid \geq nR.C \mid \leq nR.C$, where A is an atomic concept, R an atomic role, \top the top concept, which has every individual as an instance, and \perp the dual of \top . In this way, one can represent formally complex concept expressions, such as all entities that “have a Ph.D., teach at most two postgraduate courses and are not academics” ($\exists hasDegree.PhD \sqcap \leq 2teaches.PostgrCourse \sqcap \neg Academic$). Rules in \mathcal{ALCQ} have the form $C \sqsubseteq D$ and describe relationships between concept expressions. For example, one can describe formally that all postdocs are described by the aforementioned concept as $Postdoc \sqsubseteq \exists owns.PhD \sqcap \leq 2teaches.PostgrCourse \sqcap \neg Academic$. We denote with *LHS* (left-hand side) the concept expression that appears on the left of the subsumption symbol (\sqsubseteq) in a rule and with *RHS* (right-hand side) the concept expression that appears on the right. *Facts* describe knowledge about named individuals, i.e., that are *instances* of some concept (expression) and have the form $C(a)$ or $R(a, b)$, where a, b individuals. Using complex expressions one can construct very complex facts. An \mathcal{ALCQ} knowledge base (KB) is a set of rules and a set of facts.

The *inferred closure* of a KB \mathcal{K} is the minimum set of rules and facts that can be logically inferred from \mathcal{K} . Given a KB \mathcal{K} and a rule or a fact a , we say that \mathcal{K} *entails* a (rule or fact) if every model of \mathcal{K} (i.e., if every interpretation that satisfies all rules and facts of \mathcal{K}) is also a model of a . *Entailment checking* can be considered as the prototypical reasoning task for querying knowledge: we check whether some statement is necessarily true, presuming the statements of the knowledge base. Following the semantics of DLs, we make the *open-world assumption*, i.e., missing information is treated as unknown. Here, we consider *inference depth*, or simply *depth* of a with respect to \mathcal{K} , $depth(a, \mathcal{K})$, as the size of the *justification* [9] for a , i.e., the *minimum* number of rules and facts in \mathcal{K} that can be used to logically deduce that a is true or false. If none of the two can be deduced, the answer is “unknown” and a is not characterized by any depth.

3 Dataset Generation

We investigate the ability of transformers to perform textual entailment checking over \mathcal{ALCQ} KBs expressed in natural language with respect to two dimensions: i) the depth \mathcal{D} of the sentences (i.e., rules/facts in question), henceforth mentioned as *queries*, with respect to the corresponding KB, ii) the linguistic complexity level \mathcal{L} (defined in Section 3.1) of the knowledge required to answer the queries. To achieve this, each *example* in the dataset Δ_D is a 5-tuple $\langle \mathcal{T}, \mathcal{Q}, \mathcal{A}, \mathcal{D}, \mathcal{L} \rangle$, where \mathcal{T} is the *context* containing \mathcal{ALCQ} axioms (rules/facts) expressed in natural language, \mathcal{Q} the query expressed in natural language, henceforth mentioned as *question*, \mathcal{A} is the *answer* which can be either *true*, *false*, or *unknown*, and \mathcal{D} the depth of \mathcal{Q} , if \mathcal{A} is *true* or *false*, otherwise it is denoted as na. \mathcal{L} is the linguistic complexity of the KB.²

The pipeline for the generation of the dataset is presented in Figure 2. For the generation of an example (described in detail in Section 3.1) of linguistic complexity level n ($\mathcal{L} \leq n$) and depth m ($\mathcal{D} \leq m$), we first generate a KB \mathcal{K} using a specially crafted probabilistic context-free grammar (denoted in Figure 2 with $\mathcal{ALCQ}\text{-}n$ PCFG) for producing rules and facts of maximum linguistic complexity n . Then, the inferred closure of \mathcal{K} is calculated by utilizing the reasoner HermiT [10], from which we calculate, as described in

² Δ_D contains also the justification for each answer to be used for future work or by the research community for other downstream tasks, such as proof generation.

¹ <https://github.com/angelosps/DELTA>

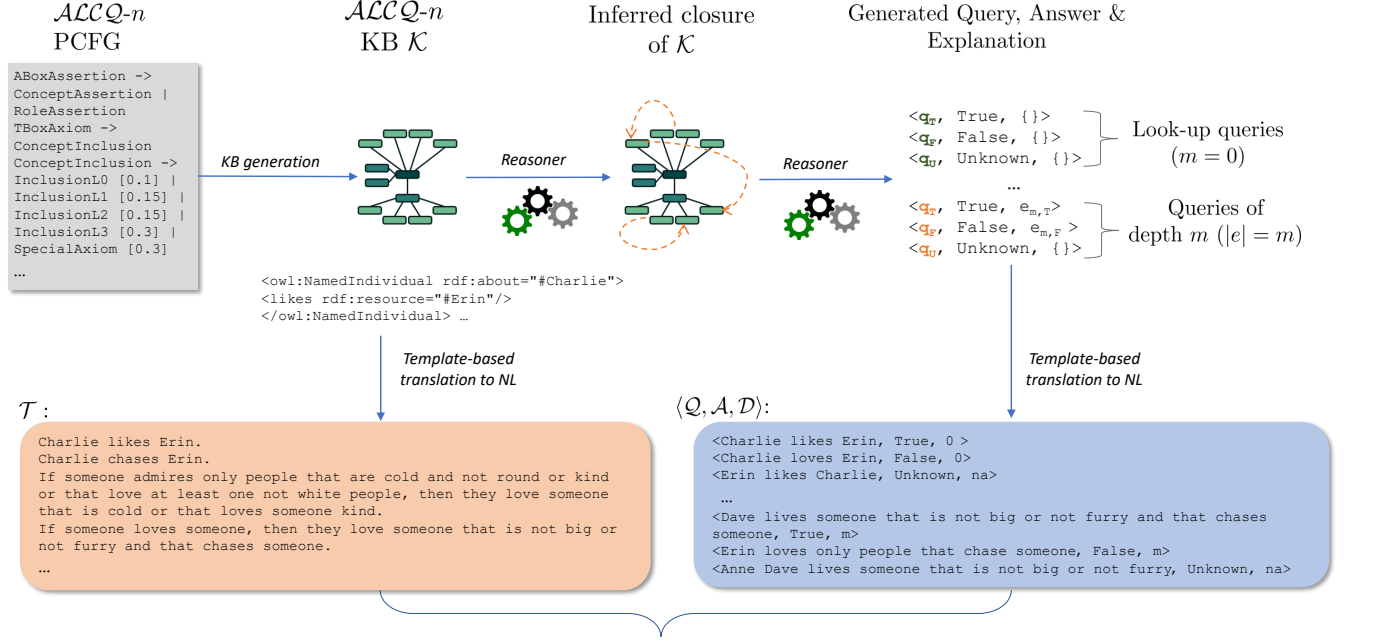


Figure 2: Data generation pipeline for examples with n -level context and answers of minimum inference depth $\leq m$

Section 3.2, *true* (answer=true), *false* (answer=false) and *unknown* (answer=unknown) queries, which eventually will formulate the sentences in question. A KB is kept only if it can produce queries with all three types of answers at all depths up to m , otherwise a new one is generated. Once this process is completed, the generated queries (facts/rules) along with the original K , are translated into natural language statements Q and into the context T , respectively, by utilizing a set of natural language templates, as described in Section 3.3.

3.1 KB Generation

To create diverse contexts and to avoid overfitting to a specific vocabulary, we have defined two different pools of terms, *Pool A* and *Pool B*. *Pool A* contains 14 atomic concepts, 5 roles, and 8 individuals, mostly taken from RuleTaker dataset [3] (in RuleTaker the rules are simple conjunctive implications, where the concept names are named “attributes”, the roles “relations” and the individual names “entities”). *Pool B* contains 8 atomic concepts, 8 roles, and 8 individuals. Both pools can be found in Appendix A.1.

From each pool, we generate 20 datasets (40 in total) of 1000 KBs each, of various inference depths and axiom lengths.

To obtain KBs of different linguistic complexity levels, we have manually crafted four types ($\mathcal{L} = 0, 1, 2, 3$) of PCFGs, based on the number of constructors and quantifiers appearing in their axioms. In general, a concept of linguistic complexity \mathcal{L} contains \mathcal{L} Boolean constructors and at most $\mathcal{L} + 1$ quantifiers. Specifically, we define the *linguistic complexity* level of a concept expression as follows:

- $Level_0$: 0 Boolean operators and ≤ 1 quantifiers.
- $Level_1$: 1 Boolean operators and ≤ 2 quantifiers.
- $Level_2$: 2 Boolean operators and ≤ 3 quantifiers.
- $Level_3$: 3 Boolean operators and ≤ 4 quantifiers.

An \mathcal{L} -type PCFG produces KBs of linguistic complexity level \mathcal{L} with axioms that their one side (e.g., LHS) is of linguistic complexity \mathcal{L} and their other side (e.g., RHS) of at most $\mathcal{L} - 1$, but also contains

simpler axioms, of smaller linguistic complexity levels. Specifically, A KB of level:

- $\mathcal{L} = 0$ contains axioms of the form $Level_0 \sqsubseteq Level_0$,
- $\mathcal{L} = 1$ contains axioms of the form $Level_0 \sqsubseteq Level_1$, $Level_1 \sqsubseteq Level_0$, $Level_1 \sqsubseteq Level_1$,
- $\mathcal{L} = 2$ contains axioms of the form $Level_0 \sqsubseteq Level_2$, $Level_2 \sqsubseteq Level_0$, $Level_1 \sqsubseteq Level_2$, $Level_2 \sqsubseteq Level_1$,
- $\mathcal{L} = 3$ contains axioms of the form $Level_0 \sqsubseteq Level_3$, $Level_1 \sqsubseteq Level_3$, $Level_2 \sqsubseteq Level_3$, $Level_3 \sqsubseteq Level_0$, $Level_3 \sqsubseteq Level_1$, $Level_3 \sqsubseteq Level_2$.

For instance, KBs of level $\mathcal{L} = 0$ contain only very simple facts or rules that do not contain any Boolean constructors but can contain one quantifier, such as $Enthusiastic \sqsubseteq \exists supports. Enthusiastic$ (translated in NL as “Enthusiastic people support someone enthusiastic”), but KBs of level $\mathcal{L} = 3$ can contain rules as complex as the first rule appearing in 1.

It is important to discern the notion of linguistic complexity of a sentence from its length. We do not focus here only on sentences that contain, for instance, multiple conjunctions but rather on sentences with a more complex structure (with quantifiers as well), leading to increased linguistic complexity.

To keep the KBs processible by the reasoners, the rules can contain up to seven atomic concepts and up to two nested quantifiers (e.g., $\exists likes. (\exists loves. Cat)$), which describes the entities that like some entity that loves some cat. All KBs are rather small (with a minimum of 3 rules and 1 fact and a maximum of 14 rules and 12 facts) and are checked with respect to satisfiability and consistency with HermiT.

3.2 Query Generation

For an inference depth \mathcal{D} , a *true query* q is an axiom or fact selected from the inferred closure of a consistent K , such that $depth(q, K) = \mathcal{D}$. An *unknown query* (answer=unknown) is generated by creating a random fact or statement (using the corresponding PCFG) such that

it does not belong to the inferred closure of \mathcal{K} and is consistent with \mathcal{K} . A *false query* (answer=false) can be generated in three ways:

- From an inconsistent \mathcal{K} : for every $a \in \mathcal{K}$ if $\mathcal{K} \setminus \{a\}$ is consistent then a is a false query over the KB $\mathcal{K} \setminus \{a\}$.
- From a consistent \mathcal{K} : i) By negating a true query q with $\text{depth}(q, \mathcal{K}) = \mathcal{D}$ (and applying De Morgan’s laws). ii) By automatically generating an appropriate axiom or fact a such that $\mathcal{K} \cup \{a\}$ is inconsistent and $\text{depth}(a, \mathcal{K}) = \mathcal{D}$. For instance, suppose that a KB \mathcal{K}_1 contains the axioms $(\forall \text{admires}.\perp)(\text{Anne})$ and $\forall \text{admires}.\perp \sqsubseteq \forall \text{likes}.\text{Quiet}$ which in natural language are translated into: “Anne admires none”, “All people that admire none like only quiet people”. Then, the fact $(\exists \text{likes}.\neg \text{Quiet})(\text{Anne})$ stating that “Anne likes someone who is not quiet” forms a false query for \mathcal{K} .

The disadvantage of the first approach is that it requires calling the reasoner multiple times, a time-consuming process, especially in KBs with long axioms (e.g., $\mathcal{L}=3$ KBs). Hence, we used the two latter approaches.

We set the reasoning depth limit to five (i.e., $\mathcal{D} = 0, 1, 2, 3, 5$) following the literature [3]. Additionally, extending this further would require longer times for the dataset generation.

3.3 Data Translation to NL

The KBs and queries were translated to NL with the use of templates. The templates were created based on the user-friendly Manchester syntax for \mathcal{ALCQ} [8]. Following this syntax, the intersection (\sqcap) and union (\sqcup) operators, are translated as “and” and “or”, respectively, the existential (\exists) quantifier is translated as “someone” or “something” (depending on whether the pool is about people or things), the universal (\forall) as “only”, and the number restrictions \leq, \geq as “at most” and “at least”. Also, we use the word “that” for intersections and nested quantifiers. For instance, the fact $(\exists \text{likes}.\forall \text{likes}.\text{Kind})(\text{Bob})$ is translated as “Bob likes someone that likes only kind people”.

Following the template-based approach suggested by Tafjord et al. [22], the axioms of the form $C \sqsubseteq D$ are, roughly, translated into NL in four different ways: i) “If C then D ”; ii) “People/Things that are C are D ”; iii) “All people/things that are C are D ”; iv) If $C = \top$ and $D = \forall R.C'$ this is translated as “Someone/something can R only people/things that are C' ”. A fact $C(a)$ is translated as “ a is C ”. To ensure that the resulting NL sentences are grammatically correct we have used a grammar checker³.

3.4 The Dataset DELTA_D

At the end, the examples of the same depth and level from both pools are merged. This results in 20 datasets of 2000 KBs each, with each resulting dataset containing sentences from both vocabularies. From each KB we generated three queries (true, false, unknown) for each depth ($\mathcal{D} = \{0, 1, 2, 3, 5\}$), i.e., from each KB we generated $3 \times (d+1)$, $d \in \mathcal{D}$, queries. So, in total, the dataset contains $\sum_{d \in \mathcal{D}} 3 \times (d+1) \times 2000 \times (\mathcal{L}_{\max} + 1) = 384\text{K}$ examples, as we generate KBs for each linguistic complexity level ranging from zero up to $\mathcal{L}_{\max} = 3$.

3.5 Statistical Features

As it is thoroughly discussed by Zhang et al. [29], it is impossible to eliminate all statistical features that exist in data, besides, some of them inherently exist in logical reasoning problems.

However, DELTA_D is balanced with respect to some of the most obvious features: i) *KB size*: From the same KB we extract all three types of questions (true, false, unknown); ii) *Inference depth*: We keep a KB only if it can provide all three types of questions with the same inference depth; iii) *Formulation of the question*: The translation to natural language is implemented in such a way that the word “not” appears almost equal number of times in true questions (52.39%), false questions (50.71%) and unknown questions (46.60%); iv) *Average length in words*: True questions 10.85, false questions 8.97, unknown questions 10.34.

3.6 Usage of Long Axioms in the Proving Process

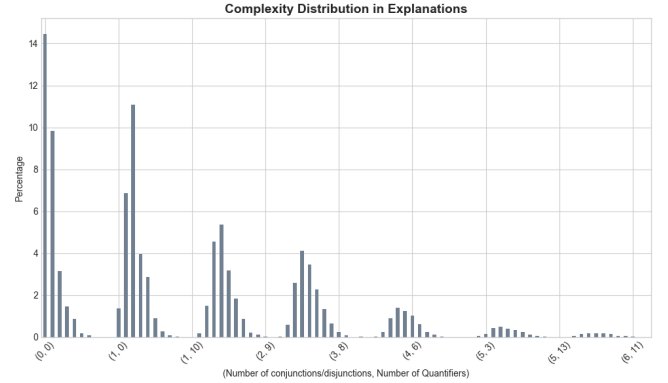


Figure 3: Complexity distribution analysis of explanation axioms in $\mathcal{L} = 3$ KBs.

Figure 3 shows the analysis of the linguistic complexity of facts and rules in explanations generated by $\mathcal{L} = 3$ KBs. For each axiom, we measured the number of conjunctions/disjunctions and quantifiers. The horizontal axis shows each combination of conjunctions/disjunctions and quantifiers. The vertical axis shows the frequency (as a percentage) of each combination. We observe that more than 40% of the axioms have at least two conjunctions/disjunctions and at least two quantifiers. Also, it is worth noting that more than 67% of the axioms have at least two conjunctions/disjunctions or at least two quantifiers. This indicates that long axioms are necessary, and frequently used in the reasoning process for answering the questions.

4 Experiments

Model Selection We systematically tested the entailment checking ability of supervised fine-tuned DeBERTaV3-large, due to its recent advancements in NLU tasks [6]. We also tested in zero-shot and few-shot prompting the models GPT-3.5 (gpt-3.5-turbo-1106) and GPT-4 (gpt-4) from OpenAI, as they have demonstrated strong performance across various reasoning benchmarks [17]. Our limited resources did not allow us to test the performance of other models, like the Llama family⁴; we plan to do this in future work.

4.1 DeBERTa-based Models

4.1.1 Evaluation Setup

We fine-tuned the DeBERTaV3-large to predict true/false/unknown (i.e., multi-class sentence classification) for each example. A context-question pair was supplied to the model as [CLS] context

³ <https://pypi.org/project/language-tool-python/>

⁴ <https://llama.meta.com/>

Table 1: Accuracy of DELTA models on Test (own), on $D_{5,3}$ dataset, and slices of $D_{5,3}$ per depth.

	DELTA _{0,3}	DELTA _{1,3}	DELTA _{2,3}	DELTA _{3,3}	DELTA _{5,3}
Test (own)	100.0	99.8	99.8	99.6	99.7
$D_{5,3}$	61.2	90.5	95.2	99.3	99.8
$\mathcal{D} = \text{N/A}$	100.0	99.4	99.5	99.2	99.7
$\mathcal{D} = 0$	100.0	100.0	100.0	100.0	100.0
$\mathcal{D} = 1$	43.4	100.0	100.0	100.0	100.0
$\mathcal{D} = 2$	24.5	73.1	99.5	100.0	100.0
$\mathcal{D} = 3$	34.1	71.3	99.5	100.0	100.0
$\mathcal{D} = 4$	29.2	77.6	84.5	99.5	100.0
$\mathcal{D} = 5$	19.3	76.5	83.5	98.5	99.5

Table 2: Accuracy of DELTA models on Test (own) across all levels.

	$\mathcal{D} = 0$	$\mathcal{D} \leq 1$	$\mathcal{D} \leq 2$	$\mathcal{D} \leq 3$	$\mathcal{D} \leq 5$
$\mathcal{L} = 0$	100.0	99.7	99.4	98.9	98.9
$\mathcal{L} \leq 1$	100.0	99.7	99.6	99.7	99.5
$\mathcal{L} \leq 2$	99.9	99.5	99.7	99.7	99.6
$\mathcal{L} \leq 3$	100.0	99.8	99.8	99.6	99.7

[SEP] question [SEP]. We used accuracy as the evaluation metric. The test data has an equal balance of true/false/unknown answers, hence the baseline of random guessing is 33.3%. For our training, we used the AdamW optimizer [14] using its default values for betas and a weight decay setting of $1e^{-4}$. The specifics of the chosen hyper-parameters, which we maintained consistently throughout our experiments, can be found in Appendix A.2.

For each combination of depth and level ($5 \times 4 = 20$ combinations in total), we trained different models on subsets of $DELTA_D$. A model $DELTA_{i,j}$ is trained in examples of reasoning depth up to i and of linguistic complexity level up to j . For instance, the model $DELTA_{3,2}$ has been trained to depths up to 3 and linguistic complexity levels up to 2. The final model $DELTA_M$ has been trained to all depths and all linguistic complexity levels, i.e., $DELTA_M = DELTA_{5,3}$. For all datasets, we partitioned the data into 70%/10%/20% splits for train/validation/test sets.

4.1.2 Evaluation Results

Table 1 illustrates the performance of DELTA models when trained on up to $\mathcal{L} \leq 3$ linguistic complexity over the various inference depths (the results for smaller levels are presented in Appendix A.3). For instance, the column $DELTA_{0,3}$ shows the performance of the model trained on all levels in depth 0. Test (own) represents the (held out) test set of the dataset that the model has been trained on. The $D_{5,3}$ dataset has questions from all inference depths ($\mathcal{D} \leq 5$) of all levels ($\mathcal{L} \leq 3$). “Depth N/A” refers to the unknown questions, as these are not provable. “ $\mathcal{D} = 0$ ” to “ $\mathcal{D} = 5$ ” lines represent subsets of $D_{5,3}$ of 0-reasoning depth to 5-reasoning depth, respectively.

It is observed that models trained even in $\mathcal{D} \leq 2$ datasets generalize quite well in larger depths (83.5% for questions of $\mathcal{D} = 5$), while when trained in $\mathcal{D} \leq 3$ datasets they show impressive generalization ability (98.5% for questions of $\mathcal{D} = 5$). Finally, we observe that the model trained in $\mathcal{D} \leq 5$ datasets almost masters (99.5-100%) the reasoning task for all reasoning depths and linguistic complexity levels.

Table 2 demonstrates the performance of each model $DELTA_{i,j}$ when tested on Test (own). For instance, the cell that corresponds to $\mathcal{D} = 0, \mathcal{L} = 0$ shows the accuracy of the model $DELTA_{0,0}$. We observe that for all depths $\mathcal{D} = 0$ to $\mathcal{D} \leq 5$ the models are robust across levels, hence increasing lengths do not affect their performance.

We, also, partitioned the dataset to the various depths, i.e., we ex-

tracted from $DELTA_D$ 5 datasets which contain only data of depth $\mathcal{D} = i$ (of all levels) and not up to i . Additionally, we trained a model on a set of 3,200 examples specifically at depth 3 for all lengths ($\mathcal{L} \leq 3$). The accuracy when tested in questions of depth 3 was 97.5%, it slightly dropped when tested in questions of smaller depths ($\mathcal{D} = 1, 2$) to $\sim 94.5\%$, except for the look-up questions ($\mathcal{D} = 0$), where the accuracy reached 99.0%. The model showed even better performance ($\sim 97.8\%$) in larger depths ($\mathcal{D} = 4, 5$). Differently from the findings of Tian et al. [24], these results demonstrate the model’s capacity for generalization across both lesser and greater reasoning depths than those encountered during training.

4.1.3 Zero-shot Performance of $DELTA_M$ on Other Distributions.

Results for Tweaked Dataset. We generated the new dataset $DELTA_T$ by changing the probability distributions of the PCFG for $\mathcal{L} = 3$ as follows: We increased the probability of the universal quantifier (\forall) from 0.33 to 0.70 and the probability of the disjunction (\sqcup) from 0.50 to 0.80. $DELTA_T$ contains 1,200 examples of up to reasoning depth $\mathcal{D} \leq 5$. This tweaking has resulted in sentences with 0.8/sentence disjunctions and 0.62/sentence universals. The accuracy of $DELTA_M$ on $DELTA_T$ was 100.0% for both true and false questions, 98.9% for unknown questions, and, 99.6% overall. As it is evident, the model is robust according to this tweaked distribution.

Results for ProofWriter Dataset [22]. The reason for choosing this dataset is that it was generated in a similar way (using PCFGs) as $DELTA_D$, it is under the open-world assumption and, partly, we have used the same pool of terms (Pool A). $DELTA_M$ demonstrated very high accuracy in true questions (95.2%) and false questions (94.0%) but low accuracy (50.8%) in unknown questions. On average the accuracy was 75.7%. The very high accuracy for true/false questions is a surprising result as although $DELTA_D$ and ProofWriter have many common types of rules/facts, ProofWriter contains also rules that involve individual names (e.g., “If Bob is blue then Erin is red”) and negated role assertions (e.g., “Bob does not like Erin”), which are not supported by \mathcal{ALCQ} and therefore are not contained in the training set of $DELTA_M$. The low performance of $DELTA_M$ in unknown questions can be attributed to the different generation processes among the two datasets. According to the generation process described in Tafjord et al. [22] (the source code is not openly available), the unknown questions in ProofWriter contain terms that appear in the context, whereas, as described in Section 3.2, unknown questions in $DELTA_D$ are formulated by choosing random terms from the corresponding pool, hence they can be completely irrelevant to the context. Hence, we can assume that this is a statistical feature that $DELTA_M$ may have learned.

Results for Use Case Scenario. We utilized the ontology rules and facts (generated from lab experiments) from [25] and generated 1,500 examples for fuel cell system diagnostics. The context contained rules of the form “If a system is in a state that is described by a low voltage value that is a result of an observation made by some voltage sensor that is a reliable sensor then the system is under some flooding” and facts of the form “v1 is a high voltage value”. A sample of this dataset is demonstrated in Appendix A.5 and the full dataset is also openly available. Again, $DELTA_M$ performed particularly well (94.0% accuracy). Detailed results of this section are presented in Appendix A.5

4.1.4 Handcrafted Quality Tests of Δ_D

To test the quality of Δ_D on which Δ_M is trained, we created simple test examples based on some of the most important knowledge base equivalences according to Rudolph [19]. The examples are demonstrated in detail in Appendix A.6.

For instance, for the conjunction rule, we provided the context: “Anne is red and green” and the two (true) questions: “Anne is red” and “Anne is green”. The model performs well overall, answering correctly 24/29 questions, however, it seems that in contexts involving number restrictions, it returned the answer “unknown”, which in two out of the six cases was wrong (notice though that the set of questions with number restrictions in Δ_D was balanced with respect to their answers).

Additionally, it failed to learn the property $A \sqcap B \sqsubseteq A \sqcup B$. This became evident through the test: “Context: Anne is red and green. Question: Anne is red or green. Answer: True. Prediction: Unknown”. To find where it fails in the reasoning chain, we asked the model the intermediate sentences “Anne is green” and “Anne is red”, to which it returned (correctly) the answer “true”, but it returned “unknown” to the question “If someone is red and green then they are red or green”. Whereas, in the test “Context: Anne is red. Anne is green. Question: Anne is red or green.” the predicted answer was, again, falsely, “unknown”.

4.2 GPT Models

Evaluation Setup We tested GPT-3.5-turbo and GPT-4 models from the chat completion API provided by OpenAI.

Our examples were limited to linguistic complexity $\mathcal{L} \leq 1$, due to the models’ context width limit: contexts of $\mathcal{L} \geq 2$ could not fit in the window for the few-shot setting. For the same reason, the maximum number of training shots was limited to 9. To enforce deterministic behavior to the models we set `temperature=0`. To make the responses less verbose we set `max_tokens=3`.

As transformer-based language models undergo pre-training through a certain form of language modeling objective, the most common approach to evaluate these models in the zero/few-shot setting is by employing prompt engineering techniques.

To formulate our prompt, we used the guidelines⁵ from OpenAI and our approach was based on the deductive reasoning prompts presented in [23]. The prompt that we concluded in was the following: {“role”: “system”, “content”: “You are an assistant capable of logical reasoning. Answer to the given question with ‘True’, ‘False’, or ‘Unknown’ based on the context.”}.

Evaluation Results In Table 3 we present the average accuracy (over 100 examples) per inference depth of 0-shot and 9-shot prompting for GPT-3.5 and GPT-4. It is noted that GPT-4 has good performance (max 92% and min 77%) with just 9 shots. Also, it is evident that the models consistently struggle in increased inference depths, demonstrating that our dataset is challenging even for $\mathcal{L} \leq 1$.

4.3 Tests on Symbolic Data

To test the effect of the semantics of the words on the performance of Δ_M we created the dataset SoftSymbolic. SoftSymbolic was generated by replacing consistently in the test set the words appearing in pools A and B with symbols. Specifically, all individuals (e.g., Anna) were replaced with an a_i symbol (e.g., a_3), all classes (e.g.,

Table 3: Average accuracy per inference depth of 0-shot, 9-shot GPT-3.5, GPT-4 on 100 examples from Δ_D of linguistic complexity level $\mathcal{L} \leq 1$.

	0-shot GPT-3.5	0-shot GPT-4	9-shot GPT-3.5	9-shot GPT-4
$\mathcal{D} = \text{N/A}$	60	98	75	92
$\mathcal{D} = 0$	74	80	84	89
$\mathcal{D} = 1$	68	57	82	86
$\mathcal{D} = 2$	48	42	60	77
$\mathcal{D} = 3$	43	32	67	80
$\mathcal{D} = 4$	44	30	57	83
$\mathcal{D} = 5$	46	40	57	83

smart) with an C_j symbol (e.g., C_2), and all roles (e.g., likes) with an R_k symbol (e.g., R_5), where i, j, k is some ID number. The average performance of Δ_M over the SoftSymbolic dataset was 95.3%, hence in contrast to Tang et al. [23], we conclude that Δ_M is not affected by the lack of semantics.

To check if the models can perform over purely logical examples, we generated the HardSymbolic dataset, which resulted from the SoftSymbolic by also utilizing the DL terminology: the word “some” (corresponding to the existential quantifier) was translated as “exists”, the “if ... then ...” (corresponding to subsumption) as “is subsumed by”, etc.

The performance of Δ_M on both Soft and Hard Symbolic datasets is presented in Table 4. We observe that the average performance of Δ_M over the HardSymbolic dataset dropped to 58.5%. This is an expected result as the structure of the tested sentences was very different from the sentences in which Δ_M had been trained. We tested also the GPT models on (100 examples of) the HardSymbolic dataset where they showed similar performance to Δ_M . The results are shown in Table 5. The average accuracy of GPT-3.5 0-shot was 57%, 9-shot 61%; and GPT-4 20%, 65%, respectively. Hence, TLMs seem to struggle with purely logical datasets.

Table 4: Average accuracy per inference depth of Δ_M on Soft-Symbolic and HardSymbolic datasets.

	Soft.	Hard.
$\mathcal{D} = \text{N/A}$	84.1	71.8
$\mathcal{D} = 0$	100.0	58.3
$\mathcal{D} = 1$	99.3	39.4
$\mathcal{D} = 2$	96.3	45.5
$\mathcal{D} = 3$	94.9	68.2
$\mathcal{D} = 4$	97.0	66.6
$\mathcal{D} = 5$	95.6	60.7

Table 5: Average accuracy per inference depth of 0-shot, 9-shot GPT-3.5, GPT-4 on 100 examples (per depth) of the HardSymbolic dataset.

	0-shot GPT-3.5	0-shot GPT-4	9-shot GPT-3.5	9-shot GPT-4
$\mathcal{D} = \text{N/A}$	26	100	70	86
$\mathcal{D} = 0$	72	32	58	78
$\mathcal{D} = 1$	70	2	38	64
$\mathcal{D} = 2$	60	2	64	66
$\mathcal{D} = 3$	56	4	64	54
$\mathcal{D} = 4$	54	0	64	58
$\mathcal{D} = 5$	60	2	70	46

Samples of each dataset are presented in Appendix A.4.

⁵ <https://platform.openai.com/docs/guides/prompt-engineering/strategy-write-clear-instructions>

Table 6: The state-of-the-art benchmarks for deductive reasoning with transformers.

	#Questions	Avg size of sentences	Max size of sentences	Formal Language	Generating method
bAbI Task15 [26]	2K	4.6	5	Role assertions	Synthetic
RuleTaker [3]	520K	6.13	27	Conj. Rules	Synthetic
ProofWriter [22]	500K	6.07	27	Conj. Rules	Synthetic
BirdsElectricity [22]	5K	12.85	22	Conj. Rules	Synthetic
ParaRules _{C/O} [22]	40K	11.85/11.78	37/37	Conj. Rules	Synthetic & Manual
FOLIO [5]	1.4K	10.58	59	FOL \ Num. Restr.	Synthetic & Manual
PrOntoQA [20]	5.8K	6.2	20	FOL \ Num. Restr.	Synthetic
LogicNLI[24]	30K	8.72	25	FOL \ Num. Restr.	Synthetic
DELTA_D ($\mathcal{L}_0 / \mathcal{L}_1 / \mathcal{L}_2 / \mathcal{L}_3$)	96K each, 384K (in total)	6.28/9.71/ 13.08/13.08	26/38/ 50/62	\mathcal{ALCQ}	Synthetic

5 Related Work

Multiple surveys [27, 11, 28] in the literature describe the most recent research developments on the use of transformers for reasoning tasks. One of the first datasets generated for this purpose was from Clark et al. [3] with RuleTaker, demonstrating the potential of transformers to perform logical question answering under CWA by training LLMs on synthetic datasets. However, their approach was limited to short expressions of simple conjunctive rules. Tafjord et al. [22], generated the ProofWriter datasets (under CWA and OWA) and with a T5 [18]-based model fine-tuned on ProofWriter showed that LLMs can generate proofs with high accuracy (94.8% for depth 5). The generation approach of the DELTA_D was based on the approach for the generation of the datasets RuleTaker and ProofWriter, i.e., using PCFGs. However, DELTA_D is different from these datasets as i) \mathcal{ALCQ} is a much more expressive logic language hence we produced new PCFGs; ii) we have defined different PCFGs for each linguistic complexity level (which has not been done for any other dataset in the literature); iii) it is balanced regarding the aspects discussed in Section 3.5.

In more expressive contexts, Ontañón et al. [16] showed that LLMs perform well (up to 90.5%) over contexts generated by propositional logic and a small subset of FOL. Han et al. [5], with the FOLIO dataset (1.4K), generated from FOL sentences -but without number restrictions-, tested the ability of various LLMs for the same reasoning task and concluded that RoBERTa [12] performed best among all tested models (including GPT-3 and Codex) but still, the performance was low. Tian et al. [24] introduced the much richer synthetic dataset LogicNLI (30K), under OWA for diagnosing LLMs’ ability in FOL reasoning, showing that even their best-performing model does not learn to perform reasoning tasks and cannot generalize to different scenarios. Schlegel et al. [21] generated a very simple dataset (containing a single conjunction) for satisfiability checking and showed that models that perform well on hard problems do not perform equally well on easier ones, concluding that transformers cannot learn the underlying reasoning rules rather than they tend to overfit to patterns in the generated data. Also, Zhang et al. [29], Tian et al. [24] achieved similar results. Bang et al. [2] studied ChatGPT’s [13] deductive reasoning ability on bAbI task 16 [26] and EntailmentBank [4], performing merely well. In addition, differently from our results (where the performance decrease was small), Tang et al. [23] showed that LLMs perform significantly better when using natural language instead of symbolic representations of logical facts and rules.

Most of the aforementioned benchmarks are composed of short sentences. The ones that contain longer sentences (avg. 13 words/sentence) are small ($\leq 40K$), while none of them have examples with numerical restrictions. This is better demonstrated with Table 6,

where we present the metrics of the datasets that are most relevant to DELTA_D (Entailment Bank is omitted as it does not conform to a specific formal language). A work that is close to our research is that of He et al. [7], who tested the ability of LLMs, and specifically of RoBERTa, to perform natural language inference tasks over existing OWL2 ontologies (e.g., FoodOn, Shema.org). However, the task studied is different, specifically, in He et al. [7], given two concept expressions C and D the TLM is asked to infer if one entails/contradicts the other, while in this work TLMs decide if a sentence can be inferred from a set of rules and facts, i.e., a KB.

Relevant to our research is also the work of Madusanka et al. [15], who investigated the effects of the various types of quantifiers on the performance of TLMs. As the generated dataset is not currently openly available it is hard to evaluate its complexity and hence its relevance to DELTA_D. It is worth noting, though, that they do not investigate systematically the aspects that we have focused on in this work (inference depth, linguistic complexity).

6 Conclusions and Future Work

We generated the only large dataset (384K) in the literature that targets expressive DLs (namely, \mathcal{ALCQ}), that enjoys both high expressivity and high linguistic complexity and is publicly available for further understanding of the functionality of TLMs. We showed that our DeBERTa-based model, DELTA_M, can carry out entailment checking over expressive synthetic datasets with very high accuracy, regardless of the linguistic complexity of the context. Differently from recent results in the literature, we showed that our model *has* learned to generalize on unseen reasoning depths, smaller or greater. Zero-shot tests showed that DELTA_M is mostly robust to other distributions. Tests with the GPT family showed that GPT-4 can have significant performance with only a few shots. The high accuracy of zero-shot testings in a real-world scenario demonstrates the potential of TLMs for performing reasoning tasks bypassing the necessity for domain experts to be familiar with formal representations.

Our qualitative tests revealed the need for the development of systematic evaluation techniques of synthetically generated datasets. Hence, this will be our next step in future work. Furthermore, we plan to explore the upper limit of the expressivity of the logic language so that a transformer-based model will be able to perform reasoning tasks with high accuracy.

Finally, we will expand our evaluation section with other state-of-the-art generative models.

References

- [1] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003. ISBN 0-521-78176-0.
- [2] Y. Bang, S. Cahyawijaya, N. Lee, W. Dai, D. Su, B. Wilie, H. Lovenia, Z. Ji, T. Yu, W. Chung, Q. V. Do, Y. Xu, and P. Fung. A multitask, multilingual, multimodal evaluation of ChatGPT on reasoning, hallucination, and interactivity. *CoRR*, abs/2302.04023, 2023. doi: 10.48550/ARXIV.2302.04023. URL <https://doi.org/10.48550/arXiv.2302.04023>.
- [3] P. Clark, O. Tafjord, and K. Richardson. Transformers as soft reasoners over language. In C. Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3882–3890. ijcai.org, 2020. doi: 10.24963/ijcai.2020/537. URL <https://doi.org/10.24963/ijcai.2020/537>.
- [4] B. Dalvi, P. A. Jansen, O. Tafjord, Z. Xie, H. Smith, L. Pipatanangkura, and P. Clark. Explaining answers with entailment trees. In *Conference on Empirical Methods in Natural Language Processing*, 2021. URL <https://api.semanticscholar.org/CorpusID:233297051>.
- [5] S. Han, H. Schoelkopf, Y. Zhao, Z. Qi, M. Riddell, L. Benson, L. Sun, E. Zubova, Y. Qiao, M. Burtell, D. Peng, J. Fan, Y. Liu, B. Wong, M. Sailor, A. Ni, L. Nan, J. Kasai, T. Yu, R. Zhang, S. R. Joty, A. R. Fabbri, W. Kryscinski, X. V. Lin, C. Xiong, and D. Radev. FOLIO: natural language reasoning with first-order logic. *CoRR*, abs/2209.00840, 2022. doi: 10.48550/ARXIV.2209.00840. URL <https://doi.org/10.48550/arXiv.2209.00840>.
- [6] P. He, J. Gao, and W. Chen. DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL <https://openreview.net/pdf?id=sE7-XhLxHA>.
- [7] Y. He, J. Chen, E. Jimenez-Ruiz, H. Dong, and I. Horrocks. Language model analysis for ontology subsumption inference. In A. Rogers, J. Boyd-Graber, and N. Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3439–3453, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.213. URL <https://aclanthology.org/2023.findings-acl.213>.
- [8] M. Horridge, N. Drummond, J. Goodwin, A. L. Rector, R. Stevens, and H. Wang. The manchester OWL syntax. In B. C. Grau, P. Hitzler, C. Shankey, and E. Wallace, editors, *Proceedings of the OWLED*06 Workshop on OWL: Experiences and Directions, Athens, Georgia, USA, November 10-11, 2006*, volume 216 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2006. URL https://ceur-ws.org/Vol-216/submission_9.pdf.
- [9] M. Horridge, B. Parsia, and U. Sattler. Laconic and precise justifications in OWL. In A. P. Sheth, S. Staab, M. Dean, M. Paolucci, D. Maynard, T. W. Finin, and K. Thirunarayan, editors, *The Semantic Web - ISWC 2008, 7th International Semantic Web Conference, ISWC 2008, Karlsruhe, Germany, October 26-30, 2008. Proceedings*, volume 5318 of *Lecture Notes in Computer Science*, pages 323–338. Springer, 2008. doi: 10.1007/978-3-540-88564-1_21. URL https://doi.org/10.1007/978-3-540-88564-1_21.
- [10] I. Horrocks, B. Motik, and Z. Wang. The Hermit OWL reasoner. In I. Horrocks, M. Yatskevich, and E. Jiménez-Ruiz, editors, *Proceedings of the 1st International Workshop on OWL Reasoner Evaluation (ORE-2012), Manchester, UK, July 1st, 2012*, volume 858 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2012. URL https://ceur-ws.org/Vol-858/ore2012_paper13.pdf.
- [11] J. Huang and K. C. Chang. Towards reasoning in large language models: A survey. In A. Rogers, J. L. Boyd-Graber, and N. Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 1049–1065. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.FINDINGS-ACL.67. URL <https://doi.org/10.18653/v1/2023.findings-acl.67>.
- [12] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019. URL <https://arxiv.org/abs/1907.11692>.
- [13] Y. Liu, T. Han, S. Ma, J. Zhang, Y. Yang, J. Tian, H. He, A. Li, M. He, Z. Liu, Z. Wu, L. Zhao, D. Zhu, X. Li, N. Qiang, D. Shen, T. Liu, and B. Ge. Summary of ChatGPT-related research and perspective towards the future of large language models. *Meta-Radiology*, 1(2):100017, sep 2023. doi: 10.1016/j.metrad.2023.100017. URL <https://doi.org/10.1016%2Fj.metrad.2023.100017>.
- [14] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL <https://openreview.net/forum?id=Bkg6RiCqY7>.
- [15] T. Madusanka, I. Zahid, H. Li, I. Pratt-Hartmann, and R. Batista-Navarro. Not all quantifiers are equal: Probing transformer-based language models’ understanding of generalised quantifiers. In H. Bouamor, J. Pino, and K. Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8680–8692, Singapore, Dec. 2023. Association for Computational Linguistics. URL <https://aclanthology.org/2023.emnlp-main.536>.
- [16] S. Ontañón, J. Ainslie, V. Cvicek, and Z. Fisher. LogicInference: A new dataset for teaching logical inference to seq2seq models. *CoRR*, abs/2203.15099, 2022. doi: 10.48550/arXiv.2203.15099. URL <https://doi.org/10.48550/arXiv.2203.15099>.
- [17] OpenAI. GPT-4 technical report. *CoRR*, abs/2303.08774, 2023. doi: 10.48550/ARXIV.2303.08774. URL <https://doi.org/10.48550/arXiv.2303.08774>.
- [18] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21: 140:1–140:67, 2020. URL <http://jmlr.org/papers/v21/20-074.html>.
- [19] S. Rudolph. *Foundations of Description Logics*, pages 76–136. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. ISBN 978-3-642-23032-5. doi: 10.1007/978-3-642-23032-5_2. URL https://doi.org/10.1007/978-3-642-23032-5_2.
- [20] A. Saparov and H. He. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL <https://openreview.net/pdf?id=qFVVBzXxR2V>.
- [21] V. Schlegel, K. V. Pavlov, and I. Pratt-Hartmann. Can transformers reason in fragments of natural language? In Y. Goldberg, Z. Kozareva, and Y. Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 11184–11199. Association for Computational Linguistics, 2022. URL <https://aclanthology.org/2022.emnlp-main.768>.
- [22] O. Tafjord, B. Dalvi, and P. Clark. ProofWriter: Generating implications, proofs, and abductive statements over natural language. In C. Zong, F. Xia, W. Li, and R. Navigli, editors, *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 3621–3634. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.findings-acl.317. URL <https://doi.org/10.18653/v1/2021.findings-acl.317>.
- [23] X. Tang, Z. Zheng, J. Li, F. Meng, S. Zhu, Y. Liang, and M. Zhang. Large language models are in-context semantic reasoners rather than symbolic reasoners. *CoRR*, abs/2305.14825, 2023. doi: 10.48550/ARXIV.2305.14825. URL <https://doi.org/10.48550/arXiv.2305.14825>.
- [24] J. Tian, Y. Li, W. Chen, L. Xiao, H. He, and Y. Jin. Diagnosing the first-order logical reasoning ability through LogicNLI. In M. Moens, X. Huang, L. Specia, and S. W. Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 3738–3747. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.303. URL <https://doi.org/10.18653/v1/2021.emnlp-main.303>.
- [25] E. Tsalapati, W. Johnson, T. W. Jackson, L. M. Jackson, D. Low, B. Davies, L. Mao, and A. A. West. Enhancing polymer electrolyte membrane fuel cell system diagnostics through semantic modelling. *Expert Syst. Appl.*, 163:113550, 2021. doi: 10.1016/j.eswa.2020.113550. URL <https://doi.org/10.1016/j.eswa.2020.113550>.
- [26] J. Weston, A. Bordes, S. Chopra, and T. Mikolov. Towards AI-complete question answering: A set of prerequisite toy tasks. In Y. Bengio and Y. LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL <http://arxiv.org/abs/1502.05698>.
- [27] Z. Yang, X. Du, R. Mao, J. Ni, and E. Cambria. Logical reasoning over natural language as knowledge representation: A survey. *CoRR*, abs/2303.12023, 2023. doi: 10.48550/arXiv.2303.12023. URL <https://doi.org/10.48550/arXiv.2303.12023>.
- [28] F. Yu, H. Zhang, and B. Wang. Nature language reasoning, A survey. *CoRR*, abs/2303.14725, 2023. doi: 10.48550/ARXIV.2303.14725. URL <https://doi.org/10.48550/arXiv.2303.14725>.
- [29] H. Zhang, L. H. Li, T. Meng, K. Chang, and G. V. den Broeck. On the paradox of learning to reason from data. *CoRR*, abs/2205.11502, 2022. doi: 10.48550/arXiv.2205.11502. URL <https://doi.org/10.48550/arXiv.2205.11502>.

A Appendix

The appendix consists of the following content:

- A.1: Dataset generation
- A.2: Additional training details
- A.3: Evaluation results on datasets for $\mathcal{L}_0, \mathcal{L}_1, \mathcal{L}_2$
- A.4: Symbolic translation
- A.5: Evaluation results on fuel cells data
- A.6: Simple quality tests

A.1 Dataset Generation

We generated our synthetic dataset DELTA_D , using four probabilistic context-free grammars. To ensure that we will produce datasets with inference depth $D > 0$, i.e., datasets resulted from some inferencing, we generated a new statement $C \subseteq D$, if C either appeared in some already generated fact or the RHS of some already generated statement.

A.1.1 Probabilistic Context-Free Grammars

Each grammar is based on some vocabulary of terms. Pool A and Pool B are defined next.

Pool A

- **Atomic Concepts:** “red”, “blue”, “green”, “kind”, “nice”, “big”, “cold”, “young”, “round”, “rough”, “orange”, “smart”, “quiet”, “furry”.
- **Role Names:** “likes”, “loves”, “eats”, “chases”, “admires”.
- **Individual Names:** “Anne”, “Bob”, “Charlie”, “Dave”, “Erin”, “Fiona”, “Gary”, “Harry”.

Pool B

- **Atomic Concepts:** “ambitious”, “confident”, “creative”, “determined”, “enthusiastic”, “innovative”, “logical”, “persevering”.
- **Role Names:** “admires”, “consults”, “guides”, “instructs”, “leads”, “mentors”, “supervises”, “supports”.
- **Individual Names:** “Ioanna”, “Dimitrios”, “Eleni”, “Maria”, “Manolis”, “Angelos”, “Panos”, “Anna”.

To generate the KBs, we employ a random sampling technique to select a subset of individuals, roles, and atomic Concepts from the pools mentioned above. An item from each pool has the same probability of being chosen.

The probabilistic context-free grammar for the linguistic complexity level $\mathcal{L} = 0$ is shown in Table 12, the rest can be found in the supplementary material files. The PCFG shown is for Pool B. The grammars for the Pool A are similar. The probabilities in the PCFGs were determined experimentally to generate appropriate KBs that would yield the desired inferences in the minimum amount of time.

A.1.2 KB Sizes

We utilize randomized parameters to control the size of a KB, based on the target reasoning depth of the corresponding dataset. The optimal (as we have found through experimentation) predefined ranges of the rules and facts per reasoning depth \mathcal{D} are as follows:

- For $\mathcal{D} = 0$: $|\text{rules}| \in [3, 8]$, $|\text{facts}| \in [1, 5]$
- For $\mathcal{D} = 1$: $|\text{rules}| \in [3, 8]$, $|\text{facts}| \in [2, 6]$
- For $\mathcal{D} = 2$: $|\text{rules}| \in [3, 8]$, $|\text{facts}| \in [3, 8]$
- For $\mathcal{D} = 3$: $|\text{rules}| \in [4, 8]$, $|\text{facts}| \in [5, 10]$
- For $\mathcal{D} = 5$: $|\text{rules}| \in [6, 14]$, $|\text{facts}| \in [6, 12]$

A.2 Additional Training Details

We used PyTorch 2.0 to set up our training and testing (inferencing). We use the `microsoft/deberta-v3-large` model from the `transformers`⁶ library, along with the `accelerate`⁷ framework.

We trained the DeBERTaV3-large (304M parameters) model on two A100 GPUs. We used mixed precision (FP16) for our calculations to save memory and speed up the process. The specific set of hyper-parameters used for all our models’ training is given in Table 7. The model showed significant performance with this set of hyper-parameters, so there was no reason to proceed with any further hyper-parameter tuning, especially given our limited resources. The model output corresponds to the truth value 0 for False, 1 for True, and 2 for Unknown labels.

Table 7: Detailed specifications of the hyper-parameters used in DeBERTaV3-large training.

Hyper-parameter	Value
Batch size	4
Accumulation steps	2 (Effective Batch size = 8)
Learning rate	2×10^{-5}
Warm-up ratio	0.06
Epochs	4
Mixed precision	FP16
Betas	(0.9, 0.999)
Weight Decay	1×10^{-4}
Text Embedding Size	512 (dimensions)

A.3 Evaluation Results on Datasets for $\mathcal{L}_0, \mathcal{L}_1, \mathcal{L}_2$

The performance of the intermediate models $\text{DELTA}_{i,j}$, for $i \in \{0, 1, 2, 3, 5\}$, $j \in \{0, 1, 2\}$ on their corresponding datasets (of $\mathcal{D} \leq i$ and $\mathcal{L} \leq j$) are illustrated in Tables 8, 9, 10. We observe that the pattern of the models’ performance across various linguistic complexity levels is similar. However, as the models progress to higher linguistic complexity levels and, hence are trained on more data, the number of times they achieve perfect accuracy is increased. The models trained on $\mathcal{D} \geq 3$ show very good generalization on unseen reasoning depths, whereas the performance on unseen reasoning depths of the models trained on $\mathcal{D} \leq 2$ fluctuates across linguistic complexity levels. This can be attributed to the complexity difference among linguistic levels, affecting models’ generalization.

Table 8: Accuracy of DELTA models on their own test sets, and the entire, and slices of $\mathcal{D} \leq 5$, $\mathcal{L} = 0$ dataset.

	DELTA _{0,0}	DELTA _{1,0}	DELTA _{2,0}	DELTA _{3,0}	DELTA _{5,0}
Test (own)	100.0	99.7	99.4	98.9	98.7
$\mathcal{D} \leq 5, \mathcal{L} = 0$	61.4	75.3	93.2	97.7	98.8
$\mathcal{D} = \text{N/A}$	98.8	97.9	94.4	95.2	98.2
$\mathcal{D} = 0$	99.9	100.0	100.0	99.5	100.0
$\mathcal{D} = 1$	48.9	99.6	100.0	99.5	100.0
$\mathcal{D} = 2$	11.9	47.1	96.3	99.0	99.0
$\mathcal{D} = 3$	34.3	49.5	75.7	99.0	99.0
$\mathcal{D} = 4$	32.1	45.0	72.0	99.0	99.0
$\mathcal{D} = 5$	29.6	42.9	67.2	97.5	99.0

A.4 Symbolic Translation

Examples of translations from natural language to both soft and hard symbolic forms are presented in Table 11. We can observe that

⁶ <https://github.com/huggingface/transformers>

⁷ <https://github.com/huggingface/accelerate>

Table 9: Accuracy of DELTA models on their own test sets, and the entire, and slices of $\mathcal{D} \leq 5, \mathcal{L} \leq 1$ dataset.

	DELTA _{0,1}	DELTA _{1,1}	DELTA _{2,1}	DELTA _{3,1}	DELTA _{5,1}
Test (own)	100.0	99.7	99.6	99.7	99.5
$\mathcal{D} \leq 5, \mathcal{L} \leq 1$	67.9	92.4	85.8	98.7	99.6
$\mathcal{D} = \text{N/A}$	99.0	98.1	99.3	98.8	99.7
$\mathcal{D} = 0$	99.9	100.0	100.0	100.0	100.0
$\mathcal{D} = 1$	52.5	99.3	99.5	100.0	100.0
$\mathcal{D} = 2$	27.4	81.7	97.5	99.0	99.5
$\mathcal{D} = 3$	47.9	79.5	61.5	99.0	99.5
$\mathcal{D} = 4$	46.9	77.2	58.0	98.0	99.0
$\mathcal{D} = 5$	39.4	66.0	53.5	96.5	99.0

Table 10: Accuracy of DELTA models on their own test sets, and the entire, and slices of $\mathcal{D} \leq 5, \mathcal{L} \leq 2$ dataset.

	DELTA _{0,2}	DELTA _{1,2}	DELTA _{2,2}	DELTA _{3,2}	DELTA _{5,2}
Test (own)	99.9	99.5	99.7	99.7	99.6
$\mathcal{D} \leq 5, \mathcal{L} \leq 2$	55.1	89.7	85.3	99.1	99.7
$\mathcal{D} = \text{N/A}$	99.6	95.6	98.7	99.2	99.7
$\mathcal{D} = 0$	99.9	100.0	100.0	100.0	100.0
$\mathcal{D} = 1$	37.2	99.6	100.0	100.0	99.5
$\mathcal{D} = 2$	16.1	81.7	98.5	100.0	99.0
$\mathcal{D} = 3$	16.4	62.7	59.0	98.5	99.5
$\mathcal{D} = 4$	17.6	63.1	62.0	99.5	100.0
$\mathcal{D} = 5$	10.0	63.4	51.0	98.0	98.5

the SoftSymbolic dataset is a good generalization test for DELTA_M as it eliminates any vocabulary influence on the model, whereas the HardSymbolic dataset offers a purely logical form of the sentences. The symbolic datasets resulted from symbolic translations of the $\mathcal{D} \leq 5, \mathcal{L} \leq 1$ test-set of DELTA_D.

A.5 Evaluation Results on Fuel Cells Data

A sample from the fuel cells diagnostics datasets is presented in Table 13. We generated two such datasets, one involving a single sensor, and the other involving two sensors, hence making the contexts more complex. We generated these datasets using simple random sampling over predefined pools of examples so they could result in true, false, and unknown questions. We can observe that the vocabulary and the format of these data are different from the dataset DELTA_D where our model was trained, although it performs particularly well (94%) zero-shot.

A.6 Simple Quality Tests

The handcrafted quality tests we created, targeting various important knowledge base equivalences, along with the predictions of DELTA_M on those are presented in Table 14. Although these tests are very similar in both structure and vocabulary with the dataset that DELTA_M has been trained, we can observe a blind spot of the model on cases involving numerical restrictions, tending to answer “unknown” (U) on such questions.

Table 11: Examples of sentences' translations to soft/hard symbolic forms.

Natural language form	Soft symbolic form	Hard symbolic form
If someone mentors someone that is ambitious and that supervises less than one creative people, then they guide only people that are not persevering or that consult at most two confident people.	If someone R6 someone that is C1 and that R7 less than one C3 people, then they R3 only people that are not C8 or that R2 at most two C2 people.	exists R6 . ((+ C1) and (< 1 R7 . (+ C3))) is subsumed by only R3 . ((not C8) or (≤ 2 R2 . (+ C2)))
Maria supports less than one people that are confident or not persevering.	a4 R8 less than one people that are C2 or not C8.	(< 1 R8 . ((+ C2) or (not C8))) (a4)
If someone is not confident, then they mentor someone that is ambitious and that supervises less than one creative people.	If someone is not C2, then they R6 someone that is C1 and that R7 less than one C3 people.	not C2 is subsumed by exists R6 . ((+ C1) and (< 1 R7 . (+ C3)))

Table 12: Probabilistic Context-Free Grammar for $\mathcal{L} = 0$ KBs

<i>ABoxAssertion</i>	→	<i>ConceptAssertion</i> <i>RoleAssertion</i>
<i>TBoxAxiom</i>	→	<i>ConceptInclusion</i>
<i>ConceptInclusion</i>	→	<i>InclusionL0</i>
<i>InclusionL0</i>	→	<i>ConceptL0</i> \sqsubseteq <i>ConceptL0</i> [0.6] <i>SpecialAxiom</i> [0.4]
<i>SpecialAxiom</i>	→	'+' \sqsubseteq \forall <i>RoleName</i> . ' (' <i>Concept</i> ') \exists <i>RoleName</i> . ' (' '+' \sqsubseteq ') \sqsubseteq <i>Concept</i>
<i>Concept</i>	→	<i>ConceptL0</i>
<i>ConceptL0</i>	→	<i>ConceptNameOrRestriction</i>
<i>ConceptNameOrRestriction</i>	→	<i>PolarityConceptName</i> <i>RestrictionConcept</i>
<i>RestrictionConcept</i>	→	<i>RestrictionD0</i>
<i>RestrictionD0</i>	→	<i>RestrictionRoleName</i> . ' (' <i>PolarityConceptName</i> ') \exists <i>RoleName</i> . ' (' '+' \sqsubseteq ') \forall <i>RoleName</i> . ' (' '+' \sqsubseteq ')
<i>Restriction</i>	→	\forall \exists <i>Symbol</i> <i>Number</i>
<i>Symbol</i>	→	'>' '≥' '<' '≤' '='
<i>Number</i>	→	'1' '2' '3'
<i>ConceptName</i>	→	'ambitious' 'confident' 'creative' 'determined' 'enthusiastic' 'innovative' 'logical' 'persevering' 'admires' 'consults' 'guides' 'instructs' 'leads' 'mentors' 'supervises' 'supports'
<i>IndividualName</i>	→	'Ioanna' 'Dimitrios' 'Eleni' 'Maria' 'Manolis' 'Angelos' 'Panos' 'Anna'
<i>RoleAssertion</i>	→	<i>RoleName</i> . ' (' <i>IndividualName</i> ' , ' <i>IndividualName</i> ')
<i>ConceptAssertion</i>	→	' (' <i>Concept</i> ') . ' (' <i>IndividualName</i> ')
<i>Polarity</i>	→	'+' '⊑'
<i>Connective</i>	→	'⊑' '⊒'

Table 13: Examples of generated <context, question, answer> triplets about fuel cells. The first two triplets involve one sensor (s1) in their context, while the last one involves two sensors (s1 and s2).

Context	Question	Answer
s1 is a system. s1 is in a state st1. st1 is described by v1. v1 is result of an observation o1. failure mode is not a normal mode. o1 is made by vs. vs is a voltage sensor. if a system is in a state that is described by a very high voltage value that is result of an observation made by some voltage sensor that is a reliable sensor then the system is under catalyst dissolution. vs is a reliable sensor. carbon support corrosion is a failure mode. v1 is a very high voltage value.	The system is under catalyst dissolution.	True.
s1 is a system. s1 is in a state st1. st1 is described by v1. v1 is result of an observation o1. failure mode is not a normal mode. o1 is made by vs. vs is a temperature sensor. if a system is in a state that is described by a low cathode humidity value that is calculated by some air relative humidity sensor that is a reliable sensor and some temperature sensor that is a reliable sensor then the system is under dehydration. vs is a reliable sensor. catalyst dissolution is a failure mode. v1 is a low cathode humidity value.	The system is not under dehydration.	False.
s1 is a system. s1 is in a state st1. st1 is described by v1. failure mode is not a normal mode. v1 is calculated by vs1 and vs2. vs1 is a hydrogen mass sensor. vs2 is a temperature sensor. if a system is in a state that is described by a very large anode humidity change that is calculated by some hydrogen relative humidity sensor that is a reliable sensor and some temperature sensor that is a reliable sensor and is described by a large cathode humidity change then the system is under membrane mechanical stress. vs1 is a reliable sensor. vs2 is a reliable sensor. catalyst dissolution is a failure mode. v1 is a very large anode humidity change that is calculated by some hydrogen relative humidity sensor that is a reliable sensor and some temperature sensor that is a reliable sensor and is described by a large cathode humidity value.	The system is under carbon support corrosion.	Unknown.

Table 14: Handcrafted quality tests for DELTA_M .

Context	Question	Correct Answer	DELTA_M
Anne is red and green.	Anne is red. Anne is green.	T T	T T
Anne is red. Anne is green.	Anne is red and green.	T	U
If someone is blue, then they are red and green.	If someone is blue, then they are red. If someone is blue, then they are green.	T T	T T
If someone is blue, then they are red. If someone is blue, then they are green.	If someone is blue, then they are red and green.	T	T
If someone is blue, then they are red and green.	If someone is blue, then they are red or green.	T	U
Anne is red. Anne is green.	Anne is red or green.	T	U
Anne is red and green.	Anne is red or green. Anne is green or red.	T T	U U
Anne is red or green.	Anne is green or red.	T	T
If someone is blue or red then they are green.	If someone is blue, then they are green. If someone is red, then they are green.	T T	T T
If someone is blue, then they are green. If someone is red, then they are green.	If someone is blue or red, then they are green. If someone is blue and red, then they are green.	T T	T T
People that eat someone red or green, are blue.	People that eat someone red or eat someone green, are blue.	T	T
	People that eat someone red or green, they eat someone red or eat someone green.	T	T
	People that eat someone that is red or eat someone that is green they eat someone that is red or green.	T	T
Blue people eat someone red or green.	People that are blue they eat someone that is red or they eat someone that is green.	T	T
People that eat only people that are red or green are blue.	People that eat only people that are red or eat only people that are green, are blue.	T	T
People that eat something are blue. Anne eats Bob. Bob is green.	Anne is blue. Anne is green.	T U	T U
People that eat something are blue. Anne eats Bob. Bob is green. If someone is blue, then they are not green.	Anne is blue. Anne is green.	T F	T F
Someone can like only people that are nice. Bob is not nice.	Anne likes Bob.	F	U
Someone can like only people that are nice. Bob is nice.	Anne likes Bob.	U	U
Anne likes less than two people. Anne likes Bob. Anne likes John.	Anne likes Alice.	F	U
Anne likes Bob.	Anne likes none.	F	F
Anne likes Bob.	Anne likes more than two people.	T	U
Anne likes John.	Anne likes more than four people.	U	U
Anne likes Alice.	Anne does not like less than two people.	T	U
Anne likes Bob.	Anne does not like Bob.	F	F