Inferring Implicit Relations in Complex Questions with Language Models

Uri Katz[1] **Mor Geva**[2] **Jonathan Berant**[1]
[1]The Blavatnik School of Computer Science, Tel-Aviv University
[2]Allen Institute for Artificial Intelligence
{uri.katz,joberant}@cs.tau.ac.il, morp@allenai.org

Abstract

A prominent challenge for modern language understanding systems is the ability to answer implicit reasoning questions, where the required reasoning steps for answering the question are not mentioned in the text explicitly. In this work, we investigate why current models struggle with implicit reasoning question answering (QA) tasks, by decoupling inference of reasoning steps from their execution. We define a new task of *implicit relation inference* and construct a benchmark, IMPLICITRELATIONS, where given a question, a model should output a list of concept-relation pairs, where the relations describe the implicit reasoning steps required for answering the question. Using IMPLICITRELATIONS, we evaluate models from the GPT-3 family and find that, while these models struggle on the implicit reasoning QA task, they often succeed at inferring implicit relations. This suggests that the challenge in implicit reasoning questions does not stem from the need to plan a reasoning strategy alone, but to do it while also retrieving and reasoning over relevant information.

1 Introduction

A longstanding goal of language understanding has been to develop systems that can reason, i.e., integrate multiple pieces of information to reach a conclusion (McCarthy, 1959; Clark et al., 2021). This has sparked interest in question answering (QA) benchmarks that require such reasoning (Welbl et al., 2018; Yang et al., 2018; Talmor and Berant, 2018). One particularly challenging case is questions that require *implicit reasoning*, that is, where the evidence for answering the question is not mentioned explicitly. Consider the question *"Does Santa Claus work during summer?"*. This question requires implicit reasoning since it involves knowing when the holiday associated with Santa occurs, but this is not evident from the question.

Recent advances in QA (Tafjord and Clark, 2021; Lourie et al., 2021) have steered attention towards

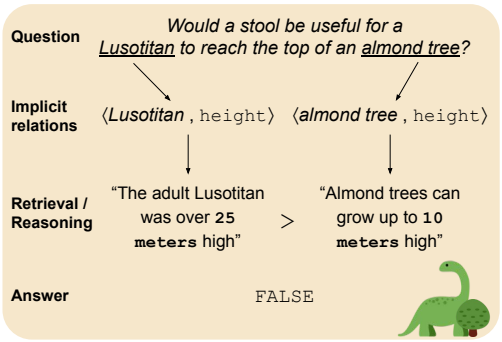


Figure 1: An example *implicit reasoning question*, where answering requires inferring *implicit relations* that are not explicitly mentioned. Besides implicit relations, answering the question also requires reasoning over the relevant retrieved facts. In this work, we focus on the first step of inferring implicit relations.

implicit reasoning QA benchmarks such as STRATEGYQA (Geva et al., 2021), OPENCSR (Lin et al., 2021), COMMONSENSEQA 2.0 (Talmor et al., 2021), CREAK (Onoe et al., 2021), and REALFP (Kalyan et al., 2021), which span a wide range of domains and reasoning skills. Still, implicit reasoning remains an open challenge, even for large language models (LMs) such as GPT-3 and PaLM (BIG-bench collab., 2021; Talmor et al., 2021; Rae et al., 2021; Chowdhery et al., 2022).

Answering implicit reasoning questions can be viewed as a two-step process: (a) inferring simple sub-questions necessary for answering the question, and (b) retrieving the relevant knowledge pieces (i.e., answering sub-questions) and reasoning over them to derive the answer. Figure 1 illustrates this decoupling. To answer the shown question, we need to use knowledge about the *Lusotitan dinosaur* and *almond trees* to infer that the relevant sub-questions concern their *heights*. We refer to the relation *height*, which is not mentioned in the question as an *implicit relation*. Once implicit relations are inferred, we can retrieve the relevant facts and deduce that the answer is 'False', as a Lusotitan is

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 2548–2566
December 7-11, 2022

Source Dataset	Question and Implicit Relation Pairs	Answer
StrategyQA	*Did the 40th president of the United States forward lolcats to his friends?* ⟨*40th president of the United States*, `year of death`⟩ , ⟨*lolcats*, `year of creation`⟩	`False`
	Could $1 for each 2009 eclipse buy a copy of TIME magazine in 2020? ⟨*2009 eclipse*, `number`⟩ , ⟨*TIME magazine in 2020*, `retail price`⟩	`True`
Creak	*Aziz Ansari has never performed in front of a crowd.* ⟨*Aziz Ansari*, `profession`⟩	`False`
	Pantera made music with distorted guitars. ⟨*Pantera*, `music genre`⟩	`True`
CSQA2.0	*None of the mail in a person's Yahoo inbox has a stamp on it.* ⟨*Yahoo inbox*, `type of mailbox`⟩	`True`
	If you play a cello you cannot join the marching band. ⟨*cello*, `playing posture`⟩	`True`

Table 1: Example annotations of concept-relation pairs from ImplicitRelations along with the question source dataset and answer. Each source exhibits different facets of implicit reasoning questions.

much higher than an almond tree.

In this work, we put a spotlight on *implicit relations* and investigate the ability of language models to infer them as a necessary (albeit insufficient) step for answering implicit reasoning questions. We first define implicit relations, and show that they can be reliably annotated through crowdsourcing (example annotated implicit relations are in Figure 1 and Table 1). To show implicit relations are common, we curate and annotate implicit reasoning questions from three existing datasets, StrategyQA, Creak, and CommonsenseQA 2.0, which results in ImplicitRelations, a new evaluation benchmark containing 615 questions and 2,673 annotations of implicit relations.

We use our benchmark to evaluate the ability of large LMs to infer implicit relations, since they are known to acquire substantial amounts of knowledge and common sense with scale (Roberts et al., 2020; Liu et al., 2021; Smith et al., 2022), but struggle with implicit reasoning questions. Specifically, we evaluate models from the GPT-3 family using *in-context learning*, where the model is fixed and only a few examples are given as context.

We find that large LMs perform well on this task, with a 175B parameter model recovering 0.53-0.59 of the implicit relations across datasets, outperforming a baseline by 21-40 points. This is robust across methods for sampling in-context examples, and even in cross-data scenarios where in-context examples are sampled from a *different* dataset than the target question. However, inferring implicit relations does *not* improve accuracy on the downstream QA task, even when gold relations are provided. This suggests that the challenge of implicit reasoning questions is not primarily due to implicit relation inference, but possibly due to the need to also retrieve information and reason over it.

To conclude, in this work we propose the notion of *implicit relations*, and construct the ImplicitRelations evaluation benchmark for testing the ability of models to infer them from questions. We evaluate large LMs and show that they infer implicit relations fairly well, while still falling short of answering implicit reasoning questions. Our work facilitates future work on improving implicit relation inference, and sheds light on the factors relevant for developing models that can handle implicit reasoning. From a broader perspective, our work joins recent community efforts to highlight the ubiquity of missing and implicit elements in natural language (Cheng and Erk, 2018; Pyatkin et al., 2020; Elazar et al., 2022). [1]

2 Implicit Relations

We now define the notion of implicit relations in the context of complex question answering.

Complex questions are questions that require multiple steps of reasoning in order to be answered (Yang et al., 2018; Talmor and Berant, 2018; Welbl et al., 2018; Khot et al., 2021). For example, the question *"Was Linnaeus alive when Darwin published Origin of Species?"* involves fetching two dates and then comparing them (Figure 2). A prominent challenge in complex QA, that attracted substantial attention recently (Mihaylov et al., 2018; Khashabi et al., 2020; Lin et al., 2021;

[1] Our benchmark ImplicitRelations and relevant code can be downloaded from `github.com/katzurik/ImplicitRelations`.

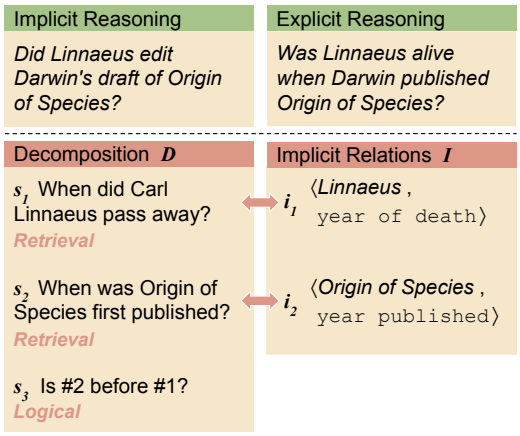


Figure 2: An example of explicit and implicit reasoning questions that share the same question decomposition, along with the implicit relations derived from the retrieval steps in the decomposition.

Geva et al., 2021; Yasunaga et al., 2021; Wei et al., 2022), is cases where the reasoning steps are *implicit* and should be inferred from the question. For instance, *"Did Linnaeus edit Darwin's draft of Origin of Species?"* involves the same reasoning steps, but they are not mentioned explicitly. Thus, the former question is an *explicit* reasoning question, while the latter is an *implicit* one (Figure 2).

Wolfson et al. (2020) proposed QDMR as a meaning representation for complex questions, where a complex question q is decomposed into a sequence of m reasoning steps $D = (s_1, \ldots, s_m)$, and each step s_i corresponds to a simple natural language question. Answering the simple questions one-by-one yields the final answer (see decomposition in Figure 2). Geva et al. (2021) collected decompositions for implicit reasoning questions as part of the StrategyQA dataset, where importantly, inferring the sequence of reasoning steps is challenging due to their implicit nature. In addition to generating effective decompositions for implicit reasoning questions, we find an additional challenge in evaluating these decompositions when represented as a sequence of sub-questions. Specifically, Geva et al. (2021) distinguished two types of reasoning steps in a decomposition – *retrieval steps*, which require retrieval of facts (s_1 and s_2 in Figure 2), and *logical steps*, which perform logical reasoning over previous results (s_3 in Figure 2).

In this work, we observe that a key ingredient in inferring decompositions is to identify the *implicit relations* that are necessary for answering the question. Concretely, each retrieval step in a question decomposition can typically be represented as concept-relation pair $\langle c, r \rangle$, where c is a sequence of tokens from the question that refer to a concept, and r is a relation of that concept. For example, the concept c in step s_2 in Figure 2 is *"Origin of Species"*, and the relation r is its publication year.

Based on this observation, we provide the following definition for implicit relations in complex QA. Let q be an implicit reasoning question, and denote by $D = (s_1, ..., s_m)$ its decomposition into a sequence of m reasoning steps. Let $\{s_{i_1}, \ldots, s_{i_n}\}$ be the subset of retrieval steps in the decomposition D. We define the implicit relations for answering q as the set $\mathcal{I} = \{\langle c_1, r_1 \rangle, ..., \langle c_n, r_n \rangle\}$ of concept-relation pairs, where each concept-relation pair corresponds to a particular retrieval step.

In the next sections, we will use this definition to construct a task for probing the ability of models to infer implicit relations (§3-§5), and investigate why they struggle on the downstream QA task (§6).

3 The ImplicitRelations Benchmark

In this section, we describe the process of creating ImplicitRelations, a benchmark for evaluating the ability of models to infer implicit relations.

3.1 Data Collection

We curate questions that require inferring implicit relations from three recent datasets:

- **StrategyQA** (Geva et al., 2021): A dataset of yes/no questions that require implicit multi-step reasoning. StrategyQA (StgQA) questions can be answered from Wikipedia, and are diverse in terms of the required reasoning skills and question topic. Large language models, such as GPT-3 (Brown et al., 2020), were shown to struggle on StgQA (BIG-bench collab., 2021).
- **Creak** (Onoe et al., 2021): A dataset containing true/false statements that require common sense and knowledge about real-world entities.
- **CSQA2** (Talmor et al., 2021): A dataset containing yes/no questions and true/false statements. CSQA2 questions involve generic commonsense reasoning. Most questions do not require knowledge about particular entities.

Examples from each dataset are shown in Table 1.

Collecting questions from three sources serves two purposes. First, it demonstrates that inferring implicit relations is necessary for many question types: in multi-step questions (StgQA) and single-step questions (Creak), in entity-focused

questions (CREAK) and generic common sense questions (CSQA2). Second, these datasets were created using different protocols: STGQA and CSQA2 use a model-in-the-loop during data collection, while CREAK does not. STGQA and CREAK ask annotators to author questions freely, while CSQA2 employs a gamification mechanism. Having questions from different data collection pipelines increases the likelihood that empirical conclusions are not tied to a particular dataset.

Question curation We chose questions that satisfy two properties: (a) answering the question requires inferring an implicit relation, and (b) the question is *feasible*, that is, it can be answered using real-world facts (provided as part of the benchmark in STGQA and CREAK) or using generic common sense (in CSQA2). We sampled examples from the training set of each of the three datasets and kept questions that satisfy the two properties.

Annotating implicit relations We use Amazon Mechanical Turk to annotate implicit relations over curated questions. We qualify 15 crowdworkers to identify concepts, which are token sequences from the question, relevant for answering the question, and the corresponding relation for each concept (see annotation guidelines in Appendix C). Annotators can specify up to four concept-relation pairs per question, but in practice, 98.9% consist of ≤ 2 pairs. Concepts must be extracted directly from the input questions, and relations are phrased using concise natural language phrases.

For STGQA and CREAK, which often require uncommon knowledge about entities, we provided additional context from the original data source. For STGQA, we provided facts along with the full question decomposition. For CREAK, we provided an explanation for why the claim is true or false.

We collected 5 annotations per example in CREAK and CSQA2, and 3 annotations in STGQA. Due to the availability of facts and question decompositions, STGQA showed high agreement between annotators (see Table 2). CREAK and CSQA2 showed more variability, and thus, we collected more annotations per example. To ensure quality, we manually verified all examples created during the annotation process, filtering out annotations that do not fit the task requirements.

3.2 Data Analysis

Table 2 provides statistics on the collected data. IMPLICITRELATIONS consists of 615 annotated questions, $\sim$ 200 per dataset, where exactly 100 examples from each data source are used as a test set, and the rest are used as a development set. Development set examples are in Appendix A.

	STGQA	CREAK	CSQA2
# of questions	201	205	209
# of unique concepts	399	309	303
# of concept-relation pairs	1139	1272	1285
Concept agreement	87%	71%	74%
Relation agreement:			
Lexical variability	85%	65%	80%
Multiple reasoning paths	15%	35%	20%

Table 2: Statistics on IMPLICITRELATIONS.

Annotator Agreement We manually analyzed 20 random examples from each data source to evaluate agreement between annotators for concepts and relations. We declared concept agreement when at least three annotators identified the same concept in an example. We found that annotators agreed on 77% of the concepts, and less than 10% of concepts were extracted only by a single annotator. See Table 2 for break-down by data source.

Assessing agreement on relations is more challenging, since relations are short phrases that can differ lexically. We marked for each example whether annotated relations differed only lexically or due to multiple reasoning strategies for answering the question. In 76% of the examples, the relations across *all* annotators were identical or differed lexically, e.g., the relations *"average price"* and *"cost"*. In 24% of the examples, multiple reasoning strategies were used, which would result in different reasoning and retrieval steps (see Table 2).

Overall, our analysis suggests that implicit relations can be annotated reliably.

4 Experimental Setting

We now turn to evaluate the ability of large LMs to infer implicit relations in questions. To this end, we use examples from IMPLICITRELATIONS in a few-shot *in-context learning* setting (Brown et al., 2020), where given several input-output examples and a test input, the LM is expected to generate the required output. We focus on this setup following the recent progress in in-context learning, specifically for tasks that involve general commonsense reasoning (Da et al., 2021; Chowdhery et al., 2022).

4.1 Task and Model Specification

Given a test question q^*, we prepend k annotated examples $\{\langle q^{(i)}, \mathcal{I}^{(i)} \rangle\}_{i=1}^{k}$ to it, where $q^{(i)}$ is the i-th question in this set and $\mathcal{I}^{(i)} = \{\langle c_1^{(i)}, r_1^{(i)} \rangle, ..., \langle c_{n_i}^{(i)}, r_{n_i}^{(i)} \rangle\}$ is the set of corresponding concept-relation pairs. Specifically, we use the following input format:

Question: q_1
Implicit Reasoning: $\langle c_1^{(1)}, r_1^{(1)} \rangle, ..., \langle c_{n_1}^{(1)}, r_{n_1}^{(1)} \rangle$
...
Question: q_k
Implicit Reasoning: $\langle c_1^{(k)}, r_1^{(k)} \rangle, ..., \langle c_{n_k}^{(k)}, r_{n_k}^{(k)} \rangle$
Question: q^*
Implicit Reasoning:

Example inputs are given in Appendix A. The prefixes "Question" and "Implicit Reasoning" were chosen arbitrarily and remained fixed throughout all experiments. For each test question, k examples are randomly sampled from the development set, and for each example, a single random annotation is selected. In our experiments, we use $k = 16$.[2]

Models We evaluate models from the GPT-3 family,[3] which are known to exhibit broad factual knowledge (Brown et al., 2020). In particular, we use `text-davinci-002`, a 175B-parameter LM, which to the best of our knowledge was not trained on any of the target benchmarks. Furthermore, to assess the scaling behaviour of models, we experiment with other models from the GPT-3 family, see §5.2 for details. In all experiments, outputs are predicted using greedy decoding.

Baseline To account for correlations originating from the concepts that appear in the question or reasoning shortcuts, we define a *'Concept-only'* baseline, where instead of testing whether the model can infer the implicit relations from the full question, we test its ability to infer them from the set of *gold* concepts that appear in the question. For this baseline, we use the same inputs as before, but replace every question q_i and test question q^* with its set of annotated concepts.

Question: q_i $\rightarrow$ **Question:** $c_1^{(i)}$; ... ; $c_{n_i}^{(i)}$

While the identity of the gold concepts provides useful information for inferring implicit relations, we expect models that have access to the full question to perform better.

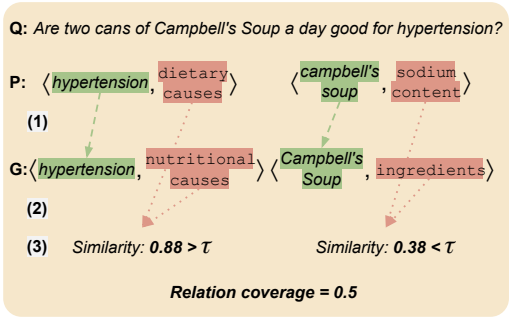


Figure 3: Given predicted concept-relation pairs (P) and gold concept-relation pairs (G), we evaluate relations by aligning predicted and gold concepts using edit distance **(1)**, and computing the cosine similarity between matched relation embeddings **(2)**. Relation coverage is the fraction of gold relations with similarity $> \tau$ **(3)**.

4.2 Evaluation

Inferring implicit relations involves identifying concepts and relations. We now define evaluation metrics for this task.

Concept extraction Our output is a set of conept-relation pairs. Let $\mathcal{C}_{\text{pred}}$ be the set of concepts predicted by the LM, and let $\mathcal{C}_{\text{gold}}^{i}$ be the gold set annotated by annotator i. Given that annotated concepts are tokens from the original question, we can use edit-distance to match each predicted concept $c \in \mathcal{C}_{\text{pred}}$ with a concept from $\mathcal{C}_{\text{gold}}^{i}$, declaring a match when the edit distance of the best match is above 0.8. Following concept matching, we compute recall and precision in the typical fashion and take the max over all annotators. A post-hoc manual analysis validated that no incorrect concept matching occurred due to the use of edit distance.

Relation coverage Since relations are short phrases with high lexical variability, string matching is not a viable strategy. To overcome this, we leverage our ability to align concepts and use relation embeddings rather than relation strings. Figure 3 depicts our evaluation procedure for two sets of predicted ($\mathcal{R}_{\text{pred}}$) and annotated ($\mathcal{R}_{\text{gold}}$) relations. First (Figure 3, step 1), we align predicted and gold concept-relation pairs, using the concepts (as done for concept evaluation). Then (Figure 3, step 2), we embed every relation phrase, using the sentence transformer `all-mpnet-base-v2` (Reimers and Gurevych, 2019), and compute cosine similarity between the embeddings of matched rela-

[2] We experimented with $k \in \{8, 16, 32\}$ but found it does not dramatically change performance, see Appendix B.

[3] We use the API at `https://openai.com/api/`.

	StgQA	Creak	CSQA2
# of gold pairs	1.9	1.2	1.2
# of generated pairs	2.0 ± 0.04	1.2 ± 0.02	1.2 ± 0.01

Table 3: The mean number of concept-relation pairs in the development set of each data source. For model-generated pairs, we report an average over 3 seeds.

		Concept Recall	**Concept Precision**	**Relation Coverage**
StgQA	CO	0.99±0.01	0.95±0.02	0.32±0.02
	FQ	0.97±0.01	0.89±0.02	0.53±0.02
Creak	CO	1±0.0	0.96±0.0	0.33±0.03
	FQ	0.98±0.0	0.95±0.01	0.54±0.05
CSQA2	CO	1±0.0	0.98±0.02	0.19±0.02
	FQ	0.93±0.02	0.94±0.01	0.59±0.01

Table 4: Test set performance for concept-only (CO) and full-question (FQ) for all datasets.

tions (defined it as zero if no relation was matched). Last (Figure 3, step 3), we consider a gold relation r_{gold} *covered* if cosine similarity is higher than a threshold τ, and compute *relation coverage*, that is, the fraction of gold relations that were covered. With this procedure, we evaluate model predictions against each annotation and take the maximum as the final relation coverage.

We focus on coverage (rather than precision) since we care about whether a model can reveal implicit relations, but predicting additional ones is mostly harmless. Moreover, since we use in-context learning, the average number of concept-relation pairs generated is similar to the average number of gold concept-relation pairs (Table 3).

To set a threshold τ on relation embedding similarity, we annotated whether a predicted relation is semantically equivalent to a matched gold relation for 100 development examples. We choose $\tau = 0.51$, which results in a 5% false-positive rate (predicting that two different relations are equivalent) and 12% false-negative rate (predicting that two equivalent relations are different). All reported results are an average over three random seeds.

5 Large LMs Can Infer Implicit Relations

Table 4 shows results on implicit relation inference. First, the model successfully identifies the relevant concepts in the question, achieving high concept recall and precision across all datasets. This is not limited to named entities but is also achieved when concepts are more abstract, as in CSQA2 (Table 1). More importantly, GPT-3 infers the implicit relations well, achieving relation coverage scores of 0.53, 0.54, and 0.59 on StgQA, Creak and CSQA2, respectively. Moreover, a model exposed to the full question dramatically outperforms a model exposed only to the gold concepts by 21, 21, and 40 points on the three datasets. This indicates that concepts contain relevant information, but access to the full question allows the LM to infer the reasoning strategy and in turn the implicit relations. We provide examples for predicted vs. gold concept-relation pairs along with additional qualitative analysis in Appendix A.

Next, we perform additional experiments to (a) further substantiate the ability of LMs to infer implicit relations (§5.1), and (b) test the effect of model scale on performance (§5.2).

5.1 Effect of In-Context Examples

While the aforementioned results are encouraging, there are two potential (non-disjoint) causes for them: (a) the LM "understands" the task of implicit relation inference, or (b) the LM observes in-context examples and uses them to guess implicit relations for the target question ("soft copying"). We study the effect of these causes.

Similar vs. dissimilar in-context examples To quantify the effect of in-context examples, rather than choosing them randomly, we use examples that are similar or dissimilar to the target question in terms of their implicit relations.

We first represent each example as an embedding vector, by (a) concatenating all annotated relations, i.e., $r_1, \ldots, r_n$, and computing a vector representation using a sentence transformer, and (b) averaging the embedding vectors of all annotators. Then, for each example, we select two sets of in-context examples: (a) *Similar*: the top-k most similar examples (using cosine similarity), and (b) *Dissimilar*: we discard the 33% most similar examples, and randomly sample from the rest. In both cases, we use gold implicit relations at test time, and thus this experiment is for analysis only.

Table 5 shows relation coverage for the different sets of in-context examples and the fraction of cases where one of the implicit relations predicted by the LM is copied from the in-context examples. When *Dissimilar* examples are presented, there is a slight performance degradation, most notably in StgQA. However, results are still dramatically higher compared to Concept-only. Moreover, the model succeeds in predicting implicit relations while hardly

		Relation Coverage	Copying
StgQA	Dissimilar	0.46±0.01	0.01±0.01
	Random	0.52±0.02	0.13±0.02
	Similar	0.54±0.02	0.31±0.02
	Concept-only	0.28±0.04	0.10±0.04
Creak	Dissimilar	0.60±0.02	0.02±0.02
	Random	0.59±0.03	0.08±0.02
	Similar	0.67±0.02	0.33±0.03
	Concept-only	0.34±0.02	0.13±0.02
CSQA2	Dissimilar	0.63±0.02	0.02±0.02
	Random	0.67±0.01	0.07±0.02
	Similar	0.73±0.01	0.21±0.03
	Concept-only	0.20±0.02	0.10±0.02

Table 5: Development set performance. Controlling the set of in-context examples with Dissimilar, Similar, Random relations, and Concept-only baseline.

copying from in-context examples.

In the *Similar* setting, performance increases across all datasets, along with a much higher rate of copying. This hints that designing methods for retrieving similar prompts can lead to gains in performance (Rubin et al., 2022).

To further investigate the relation between copying and performance, we label every example for whether the model copies from in-context examples and the coverage of the inferred implicit relations. We then compute the point-biserial correlation (Tate, 1954) to check if copying is correlated with performance and find that correlation is low (< 0.1 for all datasets), showing that copying does not explain model performance.

Overall, this experiment suggests that while models can leverage examples from context to improve performance, the LM does more than copy and execute implicit relation inference.

Cross-dataset in-context examples If LMs can infer implicit relations, we should expect high performance even when in-context examples and target questions are taken from different datasets.

To test this, we evaluate performance on questions from Creak and CSQA2 when in-context examples originate from all 3 datasets. Testing on StgQA does not work well because the number of implicit relations in an example is typically two, while in Creak and CSQA2 it is typically one (see Table 3), and thus the LM output a single implicit relation, leading to poor relation coverage.

Table 6 shows that, overall, relation coverage remains high for all sources, suggesting that the LM indeed infers implicit relations regardless of the question and reasoning types in the source dataset. Concept recall and precision are also relatively stable, except when using StgQA for in-context examples, since the model tends to output two concept-relation pairs, reducing precision. Thus, the LM is sensitive to *the number* of output concept-relation pairs that appear in in-context examples, but succeeds in inferring implicit relations.

Inference/Context Source	Concept Recall	Concept Precision	Relation Coverage
Creak/StgQA	0.97±0.0	0.69±0.01	0.62±0.03
Creak/Creak	0.93±0.02	0.91±0.01	0.59±0.03
Creak/CSQA2	0.93±0.01	0.90±0.02	0.60±0.0
CSQA2/StgQA	0.98±0.01	0.77±0.02	0.62±0.0
CSQA2/Creak	0.96±0.02	0.94±0.02	0.59±0.02
CSQA2/CSQA2	0.95±0.01	0.96±0.0	0.67±0.01

Table 6: Development set performance in the cross-dataset setup where inference on example from one dataset is done with in-context examples from another dataset.

5.2 Effect of Model Size

Recent work (Kaplan et al., 2020; Smith et al., 2022; Chowdhery et al., 2022) has shown that reasoning abilities of LMs improve with model size. We evaluate this effect on models from the GPT-3 family: `ada`, `babbage`, `curie`, and `davinci`, which are estimated to have 350M, 1.3B, 6.7B, and 175B parameters, respectively (Gao, 2021; Black et al., 2022). `text-davinci`, the model evaluated thus far, is a more recent LM that (Ouyang et al., 2022) was trained differently.[4]

Table 7 presents results on StgQA. Increasing model size improves relation coverage and concept recall, but does not significantly change concept precision. Moving from `curie` to `davinci` leads to a modest gain in relation coverage. Comparing this to the order of magnitude difference in parameters between `curie` and `davinci` suggests that inferring implicit relations does not explain performance improvement in many reasoning and commonsense QA benchmarks. The smallest model, `babbage`, tends to produce structural errors, indicating it did not properly learn the task.

[4] `text-davinci` has 175B parameters like `davinci`, but its relation coverage on StgQA is higher: 0.43→0.52. This indicates that its training procedure improves inference of implicit relations.

Parameters	Concept Recall	Concept Precision	Relation Coverage
350M	0.83±0.01	0.89±0.01	0.21±0.01
1.3B	0.93±0.01	0.84±0.01	0.37±0.01
6.7B	0.92±0.01	0.83±0.02	0.42±0.02
175B	0.97±0.0	0.88±0.0	0.43±0.03

Table 7: Model size and performance comparison on the development set of STGQA. The relation coverage improves as model size is increased.

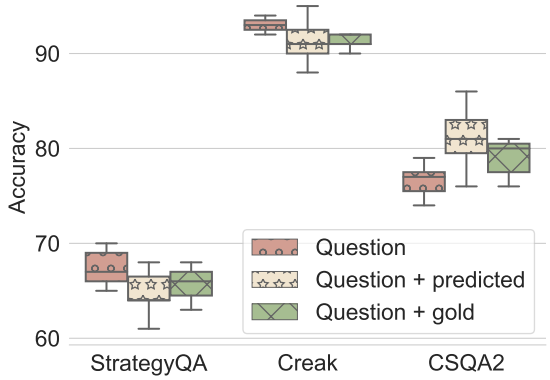


Figure 4: Test QA accuracy under all conditions, averaged over 7 seeds. Providing the gold implicit relations did not contribute to model performance.

6 Implicit Relations for QA

Given that LMs infer implicit relations well, a natural question is whether they improve performance on answering implicit reasoning questions.

To examine this, we created three experimental setups: ***Question + predicted***: in-context examples are triples of the question, the implicit relations, and the True/False answer; the model is given a question and asked to return the implicit relations and the answer. ***Question + gold***: Similar to *Question + predicted* except that the model is given the target question and gold implicit relations and asked to return the answer. ***Question only***: in-context examples are pairs of questions and answers, and the model is given a question and asked to provide the answer. We report an average over 7 seeds. See Figure 4 for results.

Overall, access to either gold or predicted relations does not improve accuracy. This suggests that additional capabilities are missing from LMs to handle implicit reasoning questions, such as retrieval and reasoning. This agrees with work on chain-of-thoughts prompting (Wei et al., 2022), which found that adding an explanation of the reasoning process to a question does not improve performance on STGQA for both GPT-3 and LaMDA (Thoppilan et al., 2022). Nevertheless, recently Chowdhery et al. (2022) achieved improvements on STGQA using chain-of-thought prompting, but with the larger 540B-parameter PaLM.

On STGQA, adding gold relations (*Question + gold*) does not improve QA performance.[5] Additionally, no significant differences were observed when the model inferred implicit relations on its own (*Question + predicted*).[6] For CREAK, adding gold implicit relations did not improve accuracy compared to *Question only*, and none of the experiments showed any significant difference. Last, in CSQA2 adding gold implicit relations (*Question + gold*) did not improve the QA performance, but we observed a statistically significant accuracy improvement of 4.5% when the model inferred the implicit relations (*Question + predicted*).[7]

To further analyze the results, we computed the point-biserial correlation coefficient between the relation coverage score and the binary outcome (correct/incorrect) for each question. We found that relation coverage score and answer accuracy are entirely not correlated with a r_{pb} coefficient of 0.03, -0.02 and 0.06 for STGQA, CSQA2 and CREAK respectively. Overall, our results indicate that inferring implicit relations correctly is not sufficient to answer implicit reasoning questions.

7 Related Work

Recent work utilized the ability of large LMs to generate intermediate reasoning steps for improving performance on QA tasks (Wei et al., 2022; Wang et al., 2022; Zelikman et al., 2022; Nye et al., 2022). Wei et al. (2022) introduced 'chain-of-thought' prompting to elicit intermediate reasoning steps along with answers from LMs, which improved performance on several reasoning tasks. Conversely, we propose a task and benchmark for evaluating the ability of LMs to infer the intermediate reasoning steps themselves.

Prior work has dealt with reasoning abilities in LMs (Talmor et al., 2020; Khashabi et al., 2020; Gu et al., 2021) by fine-tuning LMs to generate additional knowledge for reasoning questions. We contribute to this effort by evaluating the in-context ability to infer implicit relations with large LMs.

Implicit relations are closely related to question decomposition, which have been used in past work

[5] paired t-test with p-value > 0.05.
[6] paired t-test with p-value > 0.05.
[7] paired t-test with p-value < 0.05, Cohen's $d = 1.7$.

to improve performance on questions that require reasoning (Min et al., 2019; Wolfson et al., 2020; Perez et al., 2020; Khot et al., 2021). We contribute to this research direction by defining implicit relations pairs, which provide a structured representation of the decomposed sub-questions and allow us to examine how language models infer reasoning steps. Several works in narrative understanding (Rajpurkar et al., 2018; Mostafazadeh et al., 2020; Lal et al., 2021) have attempted to assess a model's implicit reasoning capabilities using different methods, such as assessing the solution path to unanswerable questions and and narrative understanding through question-answering. Despite their different approaches, these studies are relevant to our cause.

8 Conclusion

We propose the task of implicit relation inference, which decouples inference of reasoning steps from their execution. We introduce IMPLICITRELATIONS, a benchmark that includes more than 2,000 annotated implicit relations. We show large LMs can infer implicit relations across multiple types of questions and reasoning skills, but this success does not translate to an improvement in answering implicit reasoning questions. Our work sheds light on capabilities missing from large LMs for addressing implicit reasoning questions, and provides a valuable resource for improving the ability of models to infer implicit relations.

Limitations

This research has some limitations, which are typical for work on text generation with large language models.

First, we demonstrated that large LMs can infer implicit relations from complex questions, but we also showed that they may fail to answer those questions correctly. It is unclear how LMs can use implicit relations to improve QA accuracy or what is the path that leads from inferring implicit relations to actually answering the questions.

Second, evaluating relation coverage requires comparing to free texts, and therefore may be prone to error. Despite the fact that an analysis performed manually exhibited a high degree of consistency with the automatic one, we cannot guarantee the same result for datasets or parameters that have not been tested.

Finally, our research was conducted utilizing OpenAI's GPT-3 family of models, which are not publicly available. Despite our best efforts to eliminate confounding factors, there is a lack of transparency regarding the training methods and the data composition used for pretraining those models.

Acknowledgements

We thank Itay Levy for useful feedback. This research was partially supported by the Computer Science Scholarship granted by the Séphora Berrebi Foundation, the Yandex Initiative for Machine Learning, and the European Research Council (ERC) under the European Union Horizons 2020 research and innovation programme (grant ERC DELPHI 802800).

References

BIG-bench collab. 2021. Beyond the imitation game: Measuring and extrapolating the capabilities of language models. *In preparation*.

Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, USVSN Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. 2022. GPT-NeoX-20B: An open-source autoregressive language model. In *Proceedings of the ACL Workshop on Challenges & Perspectives in Creating Large Language Models*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Pengxiang Cheng and Katrin Erk. 2018. Implicit argument prediction with event knowledge. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 831–840, New Orleans, Louisiana. Association for Computational Linguistics.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Peter Clark, Oyvind Tafjord, and Kyle Richardson. 2021. Transformers as soft reasoners over language. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 3882–3890.

Jeff Da, Ronan Le Bras, Ximing Lu, Yejin Choi, and Antoine Bosselut. 2021. Analyzing commonsense emergence in few-shot knowledge models. *arXiv preprint arXiv:2101.00297*.

Yanai Elazar, Victoria Basmov, Yoav Goldberg, and Reut Tsarfaty. 2022. Text-based np enrichment. *Transactions of the Association for Computational Linguistics*.

Leo Gao. 2021. On the sizes of openai api models. EleutherAI Blog.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Yuling Gu, Bhavana Dalvi, and Peter Clark. 2021. Dream: Uncovering mental models behind language models. *ArXiv*, abs/2112.08656.

Ashwin Kalyan, Abhinav Kumar, Arjun Chandrasekaran, Ashish Sabharwal, and Peter Clark. 2021. How much coffee was consumed during EMNLP 2019? fermi problems: A new reasoning challenge for AI. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7318–7328, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. UNIFIEDQA: Crossing format boundaries with a single QA system. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1896–1907, Online. Association for Computational Linguistics.

Tushar Khot, Daniel Khashabi, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2021. Text modular networks: Learning to decompose tasks in the language of existing models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1264–1279, Online. Association for Computational Linguistics.

Yash Kumar Lal, Nathanael Chambers, Raymond Mooney, and Niranjan Balasubramanian. 2021. TellMeWhy: A dataset for answering why-questions in narratives. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 596–610, Online. Association for Computational Linguistics.

Bill Yuchen Lin, Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Xiang Ren, and William Cohen. 2021. Differentiable open-ended commonsense reasoning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4611–4625, Online. Association for Computational Linguistics.

Jiachen Liu, Alisa Liu, Ximing Lu, Sean Welleck, Peter West, Ronan Le Bras, Yejin Choi, and Hannaneh Hajishirzi. 2021. Generated knowledge prompting for commonsense reasoning. *ArXiv*, abs/2110.08387.

Nicholas Lourie, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Unicorn on rainbow: A universal commonsense reasoning model on a new multitask benchmark. In *AAAI*.

John McCarthy. 1959. Programs with common sense. In *Proceedings of the Teddington Conference on the Mechanization of Thought Processes*, pages 75–91, London. Her Majesty's Stationary Office.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Sewon Min, Victor Zhong, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2019. Multi-hop reading comprehension through question decomposition and rescoring. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6097–6109, Florence, Italy. Association for Computational Linguistics.

Nasrin Mostafazadeh, Aditya Kalyanpur, Lori Moon, David Buchanan, Lauren Berkowitz, Or Biran, and Jennifer Chu-Carroll. 2020. GLUCOSE: GeneraLized and COntextualized story explanations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4569–4586, Online. Association for Computational Linguistics.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, Charles Sutton, and Augustus Odena. 2022. Show your work: Scratchpads for intermediate computation with language models. In *Deep Learning for Code Workshop*.

Yasumasa Onoe, Michael JQ Zhang, Eunsol Choi, and Greg Durrett. 2021. CREAK: A dataset for commonsense reasoning over entity knowledge. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang,

Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*.

Ethan Perez, Patrick Lewis, Wen-tau Yih, Kyunghyun Cho, and Douwe Kiela. 2020. Unsupervised question decomposition for question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8864–8880, Online. Association for Computational Linguistics.

Valentina Pyatkin, Ayal Klein, Reut Tsarfaty, and Ido Dagan. 2020. QADiscourse - Discourse Relations as QA Pairs: Representation, Crowdsourcing and Baselines. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2804–2819, Online. Association for Computational Linguistics.

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. 2021. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *North American Association for Computational Linguistics (NAACL)*.

Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, et al. 2022. Using deepspeed and megatron to train megatron-turing nlg 530b, a large-scale generative language model. *arXiv preprint arXiv:2201.11990*.

Oyvind Tafjord and Peter Clark. 2021. General-purpose question-answering with Macaw. *ArXiv*, abs/2109.02593.

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 641–651, New Orleans, Louisiana. Association for Computational Linguistics.

Alon Talmor, Oyvind Tafjord, Peter Clark, Yoav Goldberg, and Jonathan Berant. 2020. Leap-of-thought: Teaching pre-trained models to systematically reason over implicit knowledge. In *Advances in Neural Information Processing Systems*, volume 33, pages 20227–20237. Curran Associates, Inc.

Alon Talmor, Ori Yoran, Ronan Le Bras, Chandra Bhagavatula, Yoav Goldberg, Yejin Choi, and Jonathan Berant. 2021. Commonsenseqa 2.0: Exposing the limits of ai through gamification. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

Robert F Tate. 1954. Correlation between a discrete and a continuous variable. point-biserial correlation. *The Annals of mathematical statistics*, 25(3):603–607.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam M. Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, Yaguang Li, Hongrae Lee, Huaixiu Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Yanqi Zhou, Chung-Ching Chang, I. A. Krivokon, Willard James Rusch, Marc Pickett, Kathleen S. Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Hartz Søraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Diaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravindran Rajakumar, Alena Butryna, Matthew Lamm, V. O. Kuzmina, Joseph Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Aguera-Arcas, Claire Cui, Marian Croak, Ed Chi, and Quoc Le. 2022. Lamda: Language models for dialog applications. *ArXiv*, abs/2201.08239.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, 6:287–302.

Tomer Wolfson, Mor Geva, Ankit Gupta, Matt Gardner, Yoav Goldberg, Daniel Deutch, and Jonathan Berant. 2020. Break it down: A question understanding benchmark. *Transactions of the Association for Computational Linguistics*, 8:183–198.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. QA-GNN: Reasoning with language models and knowledge graphs for question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 535–546, Online. Association for Computational Linguistics.

Eric Zelikman, Yuhuai Wu, and Noah D Goodman. 2022. Star: Bootstrapping reasoning with reasoning. *arXiv preprint arXiv:2203.14465*.

	Relation coverage
Manual evaluation	0.6
Automatic evaluation	0.53

Table 8: Manual and automatic evaluation of 50 examples from STGQA.

A Examples and Qualitative Analysis

A.1 Qualitative analysis of evaluation

In addition to the evaluation described in §4 and used throughout our study, we performed manual qualitative analysis to assess the relation coverage metric model outputs. We sampled 50 examples randomly from STGQA together with GPT-3 predictions and manually labeled if the implicit relations output by the model are semantically correct.

Results in Table 8 show that our relation coverage was slightly more conservative than our manual evaluation – 0.53 vs. 0.6. Out of the 20 examples that we marked as incorrect, the automatic evaluation scored 13 examples (65%) with 0 relation coverage, 6 (30%) examples with partial coverage (scored 0.5), which indeed we included a partially correct prediction but not enough to to cover the needed reasoning process. Only one example out of the 20 was a false positive.

A.2 IMPLICITRELATIONS Examples

Tables 9,10,11 present examples from each source dataset accompanied by the question, gold annotated implicit relation pairs, predicted pairs generated by the model and the answer to the question.

Examples from StrategyQA

Q: *Could Eric Clapton's children play a regulation game of basketball among themselves?*
G: ⟨*Eric Clapton*, number of children⟩ , ⟨*regulation game of basketball*, number of players⟩
P: ⟨*eric clapton*, number of children⟩ , ⟨*regulation game of basketball*, number of players⟩
A: False

Q: *Can you purchase General Motors products at a movie theater?*
G: ⟨*General Motors*, Types of products⟩ , ⟨*Movie theater*, products sold⟩
P: ⟨*general motors*, type of product⟩ , ⟨*movie theater*, type of product⟩
A: False

Q: *Does Rupert Murdoch's alma mater have more history than the USA?*
G: ⟨*Rupert Murdoch's alma mater*, year founded⟩ , ⟨*USA*, year founded⟩
P: ⟨*rupert murdoch's alma mater*, date of founding⟩ , ⟨*usa*, date of founding⟩
A: True

Q: *Do Squidward Tentacles and Alan Greenspan have different musical passions?*
G: ⟨*Squidward Tentacles*, instruments played⟩ , ⟨*Alan Greenspan*, instruments played⟩
P: ⟨*squidward tentacles*, musical instrument⟩ , ⟨*alan greenspan*, musical instrument⟩
A: False

Q: *Is the most expensive color in the world Blue?*
G: ⟨*Blue*, cost of pigment⟩ , ⟨*most expensive color in the world*, cost of pigment⟩
P: ⟨*blue*, color⟩ , ⟨*most expensive color*, color⟩
A: True

Table 9: Examples of gold annotations (**G**) and predicted (**P**) concept-relation pairs from ImplicitRelations along with the question (**Q**) and answer (**A**) from StrategyQA.

Examples from Creak

Q: *People celebrate Cinco de Mayo before summer starts.*
G: ⟨*Cinco de Mayo*, date⟩ , ⟨*summer*, starting date⟩
P: ⟨*cinco de Mayo*, date⟩
A: True

Q: *Arthur C. Clarke wrote a book set in outer space.*
G: ⟨*Arthur C. Clarke*, writing genre⟩
P: ⟨*arthur c. clarke*, book genre⟩
A: True

Q: *The local chamber music company hired a Harp seal to join the group.*
G: ⟨*Harp seal*, musical ability⟩
P: ⟨*harp seal*, natural habitat⟩
A: False

Q: *Nicole Kidman has never heard of Batman.*
G: ⟨*Nicole Kidman*, filmography⟩
P: ⟨*nicole kidman*, knowledge⟩
A: False

Q: *Competitors know how to drive when they are competing in Formula One racing.*
G: ⟨*Formula One racing*, type of sport⟩
P: ⟨*formula one racing*, skills required⟩
A: True

Table 10: Examples of gold annotations (**G**) and predicted (**P**) concept-relation pairs from ImplicitRelations along with the question (**Q**) and answer (**A**) from Creak.

Examples from CSQA2
Q: *Trees are never part of forests.* **G:** ⟨*forests*, `type of flora`⟩ **P:** ⟨*trees*, `location`⟩ **A:** `False`
Q: *You can only win a medal at the Olympics if you place in the top 3?* **G:** ⟨*Olympics*, `medal awarding rules`⟩ **P:** ⟨*medal*, `requirements for winning`⟩ **A:** `True`
Q: *Is it true that a hospital is capable of treating Dutch Elm disease?* **G:** ⟨*Dutch Elm disease*, `targeted organism`⟩ , ⟨*hospital*, `species treated`⟩ **P:** ⟨*hospital*, `type of diseases treated`⟩ **A:** `False`
Q: *at least one letter in the word walnut is q.* **G:** ⟨*walnut*, `spelling`⟩ **P:** ⟨*word walnut*, `letters`⟩ **A:** `False`
Q: *Is the temperature usually quite hot before Christmas in the southern hemisphere?* **G:** ⟨*Christmas*, `month`⟩ , ⟨*southern hemisphere*, `summer months`⟩ **P:** ⟨*southern hemisphere*, `seasons`⟩ **A:** `True`

Table 11: Examples of gold annotations (**G**) and predicted (**P**) concept-relation pairs from IMPLICITRELATIONS along with the question (**Q**) and answer (**A**) from CSQA2.

	k	**Concept Recall**	**Concept Precision**	**Relation Coverage**
STGQA	8	0.94±0.02	0.87±0.01	0.48±0.02
	16	0.97±0.01	0.88±0.01	0.53±0.02
	32	0.97±0.01	0.88±0.01	0.50±0.01
CREAK	8	0.94±0.02	0.91±0.02	0.56±0.0
	16	0.93±0.02	0.91±0.01	0.59±0.03
	32	0.95±0.01	0.90±0.02	0.62±0.02
CSQA2	8	0.94±0.0	0.95±0.02	0.61±0.02
	16	0.95±0.01	0.96±0.0	0.67±0.01
	32	0.95±0.01	0.96±0.02	0.66±0.02

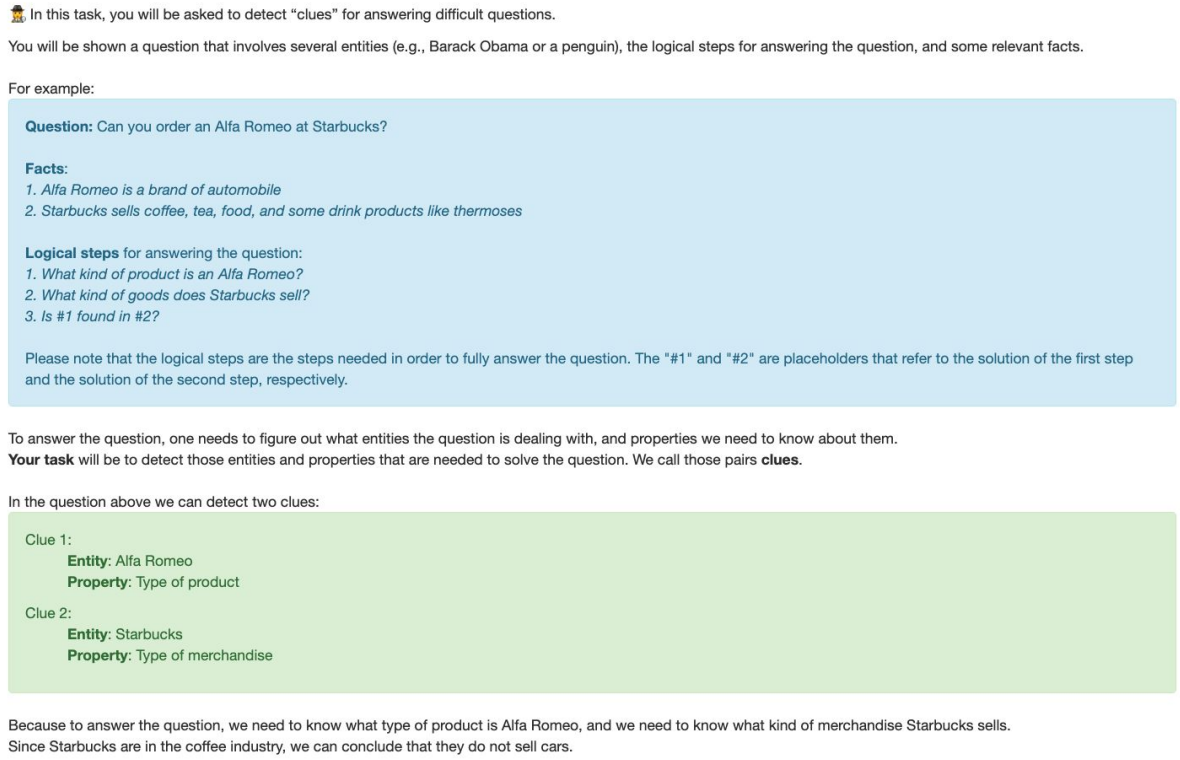
Table 12: Development set results for 8/16/32 examples in the prompt, averaged for 3 seeds.

B Number of Prompt Examples

We investigate how different number of examples influence implicit relation inference. We run the original experiment with $k \in \{8, 16, 32\}$ examples in the prompt for 3 random seeds. The results (Table 12) show that the number of examples in the prompt has little effect on all evaluation metrics, justifying our choice to use $k = 16$ in all experiments.

C Annotation Task Instruction

C.1 Task instruction



🕵 In this task, you will be asked to detect “clues” for answering difficult questions.

You will be shown a question that involves several entities (e.g., Barack Obama or a penguin), the logical steps for answering the question, and some relevant facts.

For example:

Question: Can you order an Alfa Romeo at Starbucks?

Facts:
1. Alfa Romeo is a brand of automobile
2. Starbucks sells coffee, tea, food, and some drink products like thermoses

Logical steps for answering the question:
1. What kind of product is an Alfa Romeo?
2. What kind of goods does Starbucks sell?
3. Is #1 found in #2?

Please note that the logical steps are the steps needed in order to fully answer the question. The "#1" and "#2" are placeholders that refer to the solution of the first step and the solution of the second step, respectively.

To answer the question, one needs to figure out what entities the question is dealing with, and properties we need to know about them.
Your task will be to detect those entities and properties that are needed to solve the question. We call those pairs **clues**.

In the question above we can detect two clues:

Clue 1:
Entity: Alfa Romeo
Property: Type of product

Clue 2:
Entity: Starbucks
Property: Type of merchandise

Because to answer the question, we need to know what type of product is Alfa Romeo, and we need to know what kind of merchandise Starbucks sells.
Since Starbucks are in the coffee industry, we can conclude that they do not sell cars.

C.2 Task instruction cont.

For each question, we will provide you with separate fields for each clue so you can fill the entity and property in it.

Please follow the following guidelines while completing the task:

1. Each pair of an entity and its property should appear in a separate clue.
2. **For the entity:** the extracted entity should be constructed solely from the question. Do not add words that do not appear in the question. Try to extract the complete name (i.e., Stone Cold Steve Austin and not just Steve Austin).
3. For some questions, the entity will not be its name but a text that refers to an entity not presented in the text. For example, in question 4 in the example table, our entity is "the two most common words in English". if we know those two words, we can find their spelling to answer the question. Here is another example:

   Question: Was King Kong climbing at a higher altitude than Eiffel Tower visitors?

   King-kong climbed on the empire state building, and we would like to find its height. The entity "Empire state building" is not presented in our question, so we mark King Kong climbing as our entity since it refers to the Empire state. The detected clues are
   Clue 1:
   Entity: Eiffel Tower
   Property: height
   Clue 2:
   Entity: King Kong climbing
   Property: height

4. **For the Property:** use a nautral language phrase. Try to keep it concise and straightforward, i.e., "Year of death" and not "the year that Bach died"
5. **The logical steps and facts** given to you will help you understand how the questions can be answered. You are encouraged to use them, but it is not mandatory. You do not need to use or copy words from the logical steps if it doesn't fit your output.
6. If you cannot extract a clue for some reason, please check the Invalid Question checkbox. There should not be many invalid cases, so use it as a last resort.

C.3 Concept-relation pairs