Can RoBERTa Reason? A Systematic Approach to Probe Logical Reasoning in Language Models

Anonymous authors

Abstract

Humans can map natural language into a logical representation that is robust to linguistic variations and useful for reasoning. While pre-trained language models (LM) have dramatically improved performance on commonsense reasoning benchmarks, it remains unclear whether they share this ability to reason consistently amid linguistic variations. Prior studies of LMs have found specific deficits but failed to provide a comprehensive or systematic means of understanding whether LMs deficits are due to commonsense reasoning or linguistic variation. In this work, we address this gap and explore the LM’s ability to reason by developing a general procedure that allows the systematic creation of logically-equivalent, but syntactically-different statements. To demonstrate the power of our approach, we construct a large corpus of 14,400 predictive tasks that evaluate both the abstract reasoning abilities and robustness of LMs. We find that despite the current success of large LMs on commonsense benchmarks, their performance on these tasks is no better than random guessing, heavily dependent on biases, and fails even after fine-tuning. We have released our code\(^1\) and data will be released upon acceptance.

1. Introduction

Pre-trained language models (LMs) such as RoBERTa, GPT-2, ALBERT, and BART [Liu et al., 2019, Radford et al., 2019, Lan et al., 2019, Lewis et al., 2019] outperform previous state-of-the-art (SOTA) models on multiple natural language understanding (NLU) benchmarks, including those designed to test commonsense reasoning such as the Winograd Schema Challenge (WSC) [Levesque et al., 2012]. Several studies have found that simply fine-tuning these pre-trained LMs yield better performance than more sophisticated models [Trinh and Le, 2018, Kocijan et al., 2019]. However, a growing body of work suggests this performance does not necessarily imply that LMs possess robust commonsense reasoning abilities [Talmor et al., 2019, Zhou et al., 2019, Kwon et al., 2019]. Unfortunately, prior attempts at probing LMs’ capabilities have been isolated to specific phenomena and have not systematically tested the robustness of the LM’s ability to reason with commonsense.

\(^1\) https://www.dropbox.com/s/xmj9fhjnbcf52cq/AKBC%20Probing%20Code.zip?dl=0
In this work, we create probing tasks that systematically measure a human-like ability to reason despite linguistic variability.

Humans translate natural language statements into underlying logical representations [Schank and Abelson, 1977] and then reason using these representations. These representations are robust to paraphrasing, so diverse textual statements can lead to the same reasoning results, as shown in Fig. 1. Language models are often thought to have a similar ability to map statements into a representational space that achieves conceptual generalization. If this hypothesis is true, LMs should be capable of representing and reasoning with linguistically diverse statements that contain the same conceptual information. This hypothesis motivates us to design a new probing approach that evaluates whether pre-trained LMs possess human-like commonsense reasoning capabilities.

We propose a systematic procedure to construct reasoning probes – statements requiring commonsense reasoning. These reasoning probes are defined in terms of first-order logic formulae, which capture the core commonsense reasoning required. We also introduce a set of linguistic perturbations that can be applied to the logic underlying each reasoning probe to produce logically-equivalent statements with differing subject forms. Finally, we introduce fake entities that are random strings to fill in the templates, allowing us to isolate entity knowledge from reasoning.

To demonstrate the power and generality of our framework, we construct 60 sets of probes, each containing 24 types of perturbations, and involving 10 random strings, yielding a total of 14,400 masked statements to test LMs. We evaluate LM’s performance using two metrics on the masked statements constructed from the procedure. Surprisingly, the performance of SOTA models such as RoBERTa is no better than random guessing on our probes while humans can reach around 85%. Furthermore, we identify a set of pervasive intrinsic biases in them that compromise its ability to effectively answer commonsense reasoning probes. We also fine-tune on a partition of 80% our probes, validate using another 10%, and find that when testing the fine-tuned model on the rest, it still performs like random guessing and has serious bias towards predicting positive-valence words.

Figure 1: Motivation for our probes: humans can answer commonsense questions by reasoning about underlying logical relationships. Inconsistency in LM predictions make it difficult to determine if they truly reason. We want to provide a systematic approach to evaluate a model’s understanding for the logic underlying a reasoning task.
use two other evaluation tasks including: sentence probability and textual entailment and test various language models (GPT-2, ALBERT, and BART) models without fine-tuning on our probes with description and results in appendix.

In summary, the contributions of our work are three-fold. First, we propose a systematic approach to generate logically-equivalent, but syntactically-different probes to evaluate LMs’ logical reasoning capabilities. Then we construct 14,400 masked statements following our proposed approach to evaluate LMs using masked word prediction, sentence probability and NLI. Finally, we present results from an extensive set of experiments and find that despite the success on existing benchmarks, SOTA LMs fails to reason on our statements, even after fine-tuning.

2. Probe Construction Procedure

Our goal is to generate a set of linguistically-diverse masked statements that express the same underlying logical reasoning. In this section, we describe this construction process in detail. As shown in Figure 2, the process contains four steps. We first define a few general logical predicates. Next, we construct high-level first-order logic templates that capture a broad set of reasoning tasks. Third, we use background knowledge to partially ground templates into specific logic instances. Finally, we apply linguistic perturbation operators on the logical form to generate multiple paraphrases. These perturbations produce a diverse set of masked natural language sentences to probe LMs.

2.1 Define Basic Predicates

We start with defining three general predicates that serve as the basic building blocks for the logical formulations.

- \( \text{Prop}(A, m, x) \) indicates that the property \( m \) of \( A \) is \( x \), for example, \( \text{Prop}(A, \text{size}, x) \) means the size of \( A \) is \( x \).

- \( \text{Rel}(A, B, n) \) indicates that \( A \) and \( B \) has a relation of \( n \), for example, \( \text{Rel}(A, B, \text{parent}) \) means that \( A \) is \( B \)’s parent.

- \( \text{Cmp}(x, y) \) denotes that \( x \) is more than \( y \).
<table>
<thead>
<tr>
<th>RELATION</th>
<th>PROPERTY</th>
<th>PROBE TEMPLATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priest</td>
<td>Praying</td>
<td>A is B’s priest, so A spends more time praying than B</td>
</tr>
<tr>
<td>Doctor</td>
<td>Taking care</td>
<td>A is B’s doctor, so A takes more care of B</td>
</tr>
<tr>
<td>Parent</td>
<td>Taking care</td>
<td>A is B’s parent, so A takes more care of B</td>
</tr>
<tr>
<td>Teacher</td>
<td>Be more informed</td>
<td>A is B’s teacher, so A should be more informed than B</td>
</tr>
</tbody>
</table>

Table 1: Knowledge Table to fill in the predicates of Template 2.

Our procedure can work with any predicates, but in this work, we focus on these three due to their generality and the ability to produce multiple paraphrases expressing the same logic. We are capturing a narrow slice of the set of all relationships in the world, but entity properties, relationships between entities, and comparisons of properties capture much of the knowledge in knowledge graph triples which focus on entity properties and relationships. Specifically, $\text{Prop}(A, m, x)$ and $\text{Rel}(A, B, n)$ are very common edge types in many knowledge bases like ConceptNet [Liu and Singh, 2004], so we can easily fill knowledge from them into our logic templates for a wide coverage of commonsense facts.

### 2.2 Compose Logical Templates from Predicates

From the three defined basic predicates, we form logical templates by combining predicates using first order logical connectives like $\land$, $\rightarrow$, etc.

For example, we consider the logical inference from a relation between two entities to a comparison of their properties. We can use $\rightarrow$ as the inference step, with the $\text{Rel}(A, B, n)$ predicate in the antecedent, indicating that the premise is $A$ is of relation $n$ to $B$. In the consequent, we compare a particular property of $A$ and $B$, using the $\text{Cmp}(x, y)$ predicate to compare the two values of $\text{Prop}(A, m, x)$ and $\text{Prop}(B, m, y)$. Thus we can form the final first-order logical template for this inference as $\text{Rel}(A, B, n) \rightarrow \text{Prop}(A, m, x) \land \text{Prop}(B, m, y) \land \text{Cmp}(x, y)$. In the next section, we describe how these general templates are partially grounded by introducing specific commonsense knowledge, allowing us to produce reasoning tasks.

### 2.3 Fill Templates with Knowledge Tables

We use Knowledge Tables to ground the high-level templates to logic forms that represent a specific logical inference. Using the template of property-comparison inference described above, Table 1 shows examples of commonsense concepts used to generate specific instances. The generality of the predicates in our templates allows us to use many commonsense knowledge resources, such as ConceptNet [Liu and Singh, 2004], to automatically populate these knowledge tables.

### 2.4 Generate Probe Sets Using Perturbation Operators

After grounding the logical templates, these logical relationships must be expressed in natural language. In addition to a straightforward expression of the logical template directly into language, we define several perturbations that generate an equivalent logical statement but introduce linguistic variation. By defining multiple perturbation operators and applying them on logical atoms, we are able to form a diverse set of statements to probe LMs.
Can RoBERTa Reason?

<table>
<thead>
<tr>
<th>Linguistic Operator</th>
<th>Symbol</th>
<th>Equal</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Negation</td>
<td>Neg(X)</td>
<td>¬X</td>
<td>Neg(ﬁt into) = not ﬁt into</td>
</tr>
<tr>
<td>Antonym</td>
<td>Ant(X)</td>
<td>¬X</td>
<td>Ant(ﬁt into) = contain</td>
</tr>
<tr>
<td>Paraphrase</td>
<td>Para(X)</td>
<td>X</td>
<td>Para(ﬁt into) = put into</td>
</tr>
</tbody>
</table>

Paraphrase Inversion | Para(Ant(X)) | ¬X | Para(Ant(ﬁt into)) = Para(contain) = hold inside |
| Negation Antonym    | Neg(Ant(X)) | X | Neg(Ant(ﬁt into)) = Neg(contain) = not contain |
| Negation Paraphrase | Neg(Para(X)) | ¬X | Neg(Para(ﬁt into)) = Neg(put into) = not put into |
| Negation Para,Inv   | Neg(Para(Ant(X))) | X | Neg(Para(Ant(ﬁt into))) = Neg(Para(contain)) = Neg(hold inside) = not hold inside |

Table 2: Linguistic operators with their logical equivalence and examples.

**Linguistic Operators** We define seven types of linguistic operators and the asymmetry operator to facilitate and formalize perturbations, shown in Table 2. We construct the last four operators by combining some of the single operators listed in the first three rows.

**Asymmetry Operator** Our logical templates use several strongly-ordered comparisons and relationships. This strong ordering property allows us to introduce asymmetries that preserve meaning. For our predicates \( \text{REL}(A, B, n) \) and \( \text{CMP}(A, B) \), when we swap the positions of \( A \) and \( B \), the meaning flips. For example, \( \text{MORE}(A, B) = \neg \text{MORE}(B, A) \) and \( \text{REL}(A, B, \text{parent}) = \neg \text{REL}(B, A, \text{parent}) \). Using this feature, we can swap the positions of two entities for these predicates and the logic will also be flipped, so we denote this perturbation as \( \text{ASYM}(P(A, B)) = P(B, A) = \neg P(A, B) \).

Finally, we apply the defined operators to the parameters in the logic forms and manually write text templates for converting logic forms to sentences with diverse perturbations. We have eight forms of linguistic perturbations including the original (unperturbed) one, and we can apply the asymmetry operators either on the premise or conclusion or keep the original ordering. Thus we have in total of 24 types of perturbations.

### 3. Experiment Setup

This section presents our setup to probe LMs. We will describe our task and metrics to evaluate the LMs and then show the probes we construct following Section 2.

#### 3.1 Zero-Shot Masked Word Prediction

We adopt the Masked LM pre-training objective from BERT [Devlin et al., 2019] to evaluate LMs on our constructed probes in a zero-shot setting to remove the effects of fine-tuning [Hewitt and Liang, 2019, Talmor et al., 2019] and probe only the pre-trained representations. We always mask words in the conclusion part (second half) of our probes to focus on the inference performance. Additionally, we choose to mask the words that not only require common-sense reasoning, but also restrict our masking to words where only a few options are appropriate logically and syntactically. In this way we reduce the risk of the LM of filling in the blanks with acceptable words that express the correct concept, but were not defined as correct in our evaluation. For example, in the probe “A is B’s parent, so A is more likely to care for B”, we mask “more”.

5
3.2 Grounding Templates

In the previous section, we described a method that generates partially grounded templates, leaving the entities found in the predicates and associated text unbound. To avoid conflating fact-based recall with commonsense reasoning, we use fake entities (random strings) in all probing tasks. These fake entities are randomly generated character strings of length 3 to 12 that are assumed to be unseen in the training data of the LMs. By generating ten instances of each probe, each with a different pair of fake entities, and then taking the average performance over the ten trials to represent the performance on a given probe, we minimize the influence of our randomly generated entities creating helpful and/or distracting sub-words at evaluation time.

3.3 Evaluation Metrics

We consider two evaluation metrics for our experiments.

**Binary Accuracy** We evaluate by simply comparing the rankings of the masked word and the other candidate with the opposite meaning. For example, in the masked probe: “A is B’s parent, so A takes [MASK] care of B,” “more” is the right answer and “less” is the wrong answer. In the binary setting, we feed the masked sentence to the LMs and give the model score 1 if the right answer appears higher than the wrong answer, and 0 otherwise.

**Confidence Ratio** To address the nuances in the ranking scores of predicted words given by the LM, we further propose a metric called confidence ratio setting. Using the example above, we denote the predicted score for “more” as $\text{score}_{\text{right}}$ and that for “less” as $\text{score}_{\text{wrong}}$. Then we calculate our final score using: $(\text{score}_{\text{right}} - \text{score}_{\text{wrong}})/\left(\text{score}_{\text{right}} + \text{score}_{\text{wrong}}\right)$. The more positive the final score is, the better the performance according to this metric, and vice versa. And the absolute value of this score indicates how confident the LM is for this prediction.

3.4 Probing Data

Following the process in Section 2, we construct four logical templates from the predicates:

1. $\text{Prop}(A, m, g) \land \text{Prop}(B, m, s) \land \text{Prop}(g, t, x) \land \text{Prop}(s, t, y) \land \text{Comp}(x, y) \rightarrow \text{Prop}(A, t, x) \land \text{Prop}(B, t, y) \land \text{Comp}(x, y)$, e.g. “A is made out of glass, B is made out of stone, (and glass is more transparent than stone,) so A is more transparent than B”,

2. $\text{Rel}(A, B, p) \rightarrow \text{Prop}(A, p, x) \land \text{Prop}(B, p, y) \land \text{Comp}(x, y)$, e.g. “A is B’s priest, so A spends more time praying than B”,

3. $\text{Prop}(A, v, x) \land \neg\text{Prop}(B, v, x) \land \text{Prop}(v, s, x) \rightarrow \text{Prop}(A, s, x) \land \text{Prop}(B, s, y) \land \text{Comp}(x, y)$, e.g. “A makes the varsity team while B does not (and making the varsity team indicates skill), so A is more skilled than B”,

4. $\text{Prop}(A, c, x) \land \text{Prop}(B, c, y) \land \text{Comp}(x, y) \land \text{Rel}(c, e) \rightarrow \text{Prop}(A, e, x) \land \text{Prop}(B, e, y) \land \text{Comp}(x, y)$, e.g. “A is able to concentrate more than B (and efficiency is a consequence of concentrating more), so A is more effective than B”,

6
Can RoBERTa Reason?

Figure 3: Results of average accuracy and confidence ratio of RoBERTa across per logically-equivalent statement sets. All the accuracies are close to 0.5, indicating that the models’ performance is no better than random guessing. The associated confidence ratios are close to 0, as a result of the model being equally confident both when it is right or wrong.

Note that in the statements, we consider some predicates as commonsense knowledge and hide them (phrases in the parenthesis from the examples above) to test the models. Then we use knowledge tables to fill in each template and finally apply the perturbation operators as described before to form a final set of 1,440 linguistically-varied statements or probes. For each masked statement, we randomly generate 10 fake entities to fill in positions of A and B so we actually evaluate LMs using 14,400 trials.

4. Does RoBERTa Reason?

We show the results of probing the state-of-the-art LM, RoBERTa [Liu et al., 2019], using the setup described above and provide analysis centered around the question raised in Section 1: can the LM consistently make correct predictions on a set of logically-equivalent, but syntactically-different statements? We will first examine RoBERTa’s general performance across all statements and then dive into its ability to cope with linguistic variations.

RoBERTa’s performance is on par with random guessing. As shown in Figure 3, the average accuracy for each set of logically-equivalent sentences is quite poor. The average performance of binary accuracy for all 60 probe sets are around 0.5. Similarly, the confidence value of the predicted answer is close to 0, which we find out that it is a result of the model being extremely confident in its guesses, regardless if it gets the question right or not—confident incorrect guesses will have a confidence score near -1, while confident correct guesses will have a score near 1, thus as accuracy is near 0.5 the two cancel each other out. This strongly suggests that RoBERTa does not have an ability to reason in this setting, even though its confidence in its responses are quite high. In fact, a random baseline that chooses between the two comparative words per probe would have an accuracy of 0.5 and a confidence score of 0, meaning that RoBERTa with a masked language model head barely beats out random guessing.

This observation is at sharp contrast to RoBERTa’s substantial performance improvements on many commonsense benchmarks. This random-like performance also suggests that RoBERTa is not able to understand the logic embedded in statements designed to
Figure 4: Fine-tuning results of average accuracy and confidence score of RoBERTa with 80/10/10 split for train/valid/test. We see that there is no clear difference between the performance before and after fine-tuning and it is still like random guessing.

probe its reasoning capabilities. From these results, we deduce that while RoBERTa can be fine-tuned to perform well on many commonsense reasoning tasks, it does not show this same performance on our probes in a zero-shot setting.

Fine-tuning does not help Given this deviation from previous results in a zero-shot setting, the natural next step is to investigate RoBERTa’s performance on our probe set after fine-tuning. We followed the standard 80/10/10 split for train/valid/test, making sure that all probes from the same logically equivalent sets appeared in the same split, so to prevent boosted performance resulting from overfitting to the training data. After each epoch we test the fine-tuned model on our validation set, and save the model with the highest validation set performance, which is after around 150 epochs.

However, all this fine-tuning work provides no real aid to RoBERTa’s ability to perform well on our probe set, as seen in Figure 4. Again the confidence scores cancel each other out and result in an average confidence score hovering around zero, indicating RoBERTa is as confused as it was before the fine-tuning. Even though RoBERTa has now been shown the majority of our probe set, meaning that it has seen many sentences that are very similar in syntax to the test set, it still fails to reason correctly, upholding the deviation from previously observed results of RoBERTa’s reasoning abilities. We also believe this inability to improve after fine-tuning shows the challenging nature of our dataset, which cannot be trivially solved by fine-tuning.

Ablation study and observations of fake entity usage As mentioned, we use pairs of fake entities to prevent information about the correct logic of the sentence being leaked by real entities. While we did conduct an ablation study on a portion of our probes to test the effect of fake entities, it is important to mention that the dominant trend is that no matter what pair of fake entities were substituted into the probes, the machine’s answer rarely changed. In fact out of the 1440 probes we constructed, 1343 (over 93%) of them always resulted in the same answer across all ten trials regardless of the fake entities being
Can RoBERTa Reason?

![Ablation study results of average performance of RoBERTa across one third of the probes.](image)

These probes involve social interactions where we replaced our usual fake entities with real names. We still see the consistent poor performance, indicating that using random strings is not hindering RoBERTa’s inability to solve this task.

Figure 5: Ablation study results of average performance of RoBERTa across one third of the probes. These probes involve social interactions where we replaced our usual fake entities with real names. We still see the consistent poor performance, indicating that using random strings is not hindering RoBERTa’s inability to solve this task.

Now in order to ensure our fake entity usage wasn’t hindering RoBERTa’s ability to reason, we conducted an ablation study on 480 of our probes, 33% of our full dataset. These 480 probes involved social commonsense and so the fake entities were being used instead of people’s names. As Figure 5 shows, the performance of RoBERTa does not change much, again strongly suggesting that the fake entities are of little concern to RoBERTa when making the decision as to what words should fill in the masked word. Our usage of fake entities does not seem to introduce helpful or distracting sub-words, nor does the introduction of fake entities seem to bother RoBERTa, as there is no difference in performance with or without them.

Comparison with humans  In order to ensure that our probe set is truly testing commonsense, we conducted a small human evaluation on a portion of our dataset. If our probe set is indeed testing commonsense, humans should be able to perform quite strongly on the test we devise. Our human evaluation was conducted on a 5% sample of our probes, where 20 people were asked to choose the more probable word between a pair of opposite comparative words (such as “more” and “less”) that best completed the logic in a sentence—the comparative word would fill in a blank in the sentence. While the number of datapoints (20) isn’t large enough to be fully confident in our summary statistics, the average human performance is 85% (Table 3) with inter-annotator agreement 0.768 which is substantial agreement, showing that humans are generally able to perform this task, thus suggesting we are indeed testing commonsense. We also have reason to believe this number is not too far off a larger sample’s mean, as 16 out of 20 people scored above an 85%. It is also worth noting that humans do not exhibit the same bias observed in LMs.

Does explicitly providing commonsense knowledge help?  Shocked by the severe bias observed in RoBERTa, we construct an easier set of probes, where we explicitly state all knowledge needed to make the correct logical inference. We have two settings for this test,
Table 3: Summary of all masked work prediction experiments we have conducted, as well as a small human evaluation of our probe set

<table>
<thead>
<tr>
<th>Task</th>
<th>Model-Setting</th>
<th>Dataset Size</th>
<th>Accuracy</th>
<th>Pos Accuracy</th>
<th>Neg Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>MWP</td>
<td>RoBERTa</td>
<td>1440</td>
<td>0.51</td>
<td>0.899</td>
<td>0.122</td>
</tr>
<tr>
<td>MWP</td>
<td>RoBERTa-Fine Tuned</td>
<td>144</td>
<td>0.467</td>
<td>0.775</td>
<td>0.158</td>
</tr>
<tr>
<td>MWP</td>
<td>RoBERTa-Just Social</td>
<td>480</td>
<td>0.51</td>
<td>0.81</td>
<td>0.21</td>
</tr>
<tr>
<td>MWP</td>
<td>RoBERTa-Social w names</td>
<td>480</td>
<td>0.517</td>
<td>0.825</td>
<td>0.208</td>
</tr>
<tr>
<td>MWP</td>
<td>RoBERTa-Easy, Parrot</td>
<td>60</td>
<td>0.925</td>
<td>1.0</td>
<td>0.85</td>
</tr>
<tr>
<td>MWP</td>
<td>RoBERTa-Easy, Negation Switch</td>
<td>60</td>
<td>0.358</td>
<td>0.517</td>
<td>0.2</td>
</tr>
<tr>
<td>MWP</td>
<td>ALBERT</td>
<td>1440</td>
<td>0.49</td>
<td>0.734</td>
<td>0.246</td>
</tr>
<tr>
<td>Human</td>
<td>Human-20 Responses</td>
<td>72</td>
<td>0.845</td>
<td>0.831</td>
<td>0.863</td>
</tr>
</tbody>
</table>

Diagnosing RoBERTa: heavy language biases

In our experiment setup, we ask RoBERTa to fill in a masked word, and obtain a distribution over the set of predictions. We use three distinct pairs of comparative words in evaluation, ‘more” or “less”, “easier” or “harder” and “better” or “worse”. The seeming randomness of RoBERTa’s performance prompted us to inspect our results more closely, and we subsequently uncovered a strong bias in RoBERTa’s response. We initially found that RoBERTa heavily favors predicting “more” over “less”, in 94% of statements, “better” over “worse” in 93%, and “easier” over “harder” in 81%. These results indicate a heavy bias as there are equal number of statements where each word in a given pair is the correct answer.

We generalize this pattern to the fact that when RoBERTa is asked to infer a comparative relationship between the property of two entities, the model is heavily biased towards predicting the positive valence words regardless of what property we are comparing—i.e. regardless of the logic in the statement. Table 3 shows that the accuracy for “positive valence” words is a lot higher than “negative valence” words. This bias towards positive valence is further evidence that RoBERTa does not reason on our probes like humans, but rather attempts to pattern match.
5. Related Work

Machine commonsense logic has been studied for a long history. Most classical works primarily focus on executing symbolic rules as hand-crafted programs for machines to learn [Mccarthy, 1960]. Recent attempts mainly generate templatic questions, such as those in the bAbI dataset [Weston et al., 2015], to test whether neural networks are able to reason via executing long chains of logic operations. Given the huge success of LMs [Devlin et al., 2019, Liu et al., 2019] in natural language processing, we want to study whether, or to what extent, LMs can reason with commonsense.

Prior works in analyzing the (commonsense) reasoning ability of LMs have primarily focused on creating probing tasks by generating ad-hoc masked sentences either from knowledge bases [Petroni et al., 2019, Feldman et al., 2019] or existing datasets. The first line of works aim to test if LMs can work as knowledge bases (e.g. ConceptNet, Freebase) in terms of retrieving factual or commonsense knowledge by triple prediction tasks. LAMA and its following work [Petroni et al., 2019, Jiang et al., 2019] designed relational templates as masked sentences and then test if pre-trained LMs such as BERT or RoBERTa can correctly recover the missing one-hop relations. Correctly retrieving factual triples does not necessarily mean that LMs can reason with them. Our work, in contrast, focuses on higher-order reasoning chains, which requires LMs to capture logical rules to answer all different linguistic variations. We explicitly test if LMs can consistently predict missing tokens in a set of paraphrases sharing the same logical patterns.

The goal of the second line of works [Zhou et al., 2019, Talmor et al., 2019, Kwon et al., 2019] is to reuse existing datasets as probe tasks, such as SWAG, PIQA [Bisk et al., 2019], and so on. Though this kind of probing tasks can directly show the reasoning performance of LMs on downstream tasks, it is still not clear if LMs really enjoy the ability of mapping utterance to underlying logical patterns for predicting masked tokens. The success of LMs on these tasks may come from simple statistical matching instead of reasoning. As we seek to investigate whether LMs have human-like reasoning ability, the probing tasks are supposed to be center about logical templates. Our proposed logical templates assures that our probing examples focus on reasoning instead of retrieving frequent linguistic patterns.

6. Conclusion

In summary, we propose a systematic approach to construct logically-equivalent, but syntactically different masked statements to probe LMs. Following this approach, we generate in total 14,400 masked statements from 60 sets of probes, 24 types of perturbations, and 10 fictitious entities and test RoBERTa using zero-shot masked word prediction. We find that RoBERTa’s performance is comparable to random guessing, has a strong bias towards predicting greater than for comparison, and is not able to cope with perturbations.
References


John W. Mccarthy. Programs with common sense. 1960.


7. Appendix

7.1 Other Evaluation Settings

Unlike most prior work that mostly uses one task to probe the models, we use two more tasks to test LMs’ commonsense reasoning capability besides masked word prediction. We argue that they each evaluates some degree of the commonsense reasoning ability of the LMs, but does not provide a comprehensive picture. For example, in masked word prediction, the machine is asked to predict the most likely word to fill in the blank given the context, not necessarily testing reasoning. However, we draw conclusions from experimental results on three distinct evaluation tasks, aiming to provide convincing and comprehensive probing insights.

7.1.1 Sentence Probability

We evaluate LMs by feeding the whole sentences into the model and comparing their probabilities by multiplying each word’s probability conditioned on the previous words, i.e., the language modeling loss. For each of our commonsense statement in our probe, we pair it with a statement that is against commonsense. We create them by swapping the masked word in the above evaluation setting with its opposite. In the example above, we create that probe’s pair as: “A is B’s parent, so A is less likely to care for B”. The intuition is that we want to see that when determining how probable a sentence is, does an LM factor in commonsense reasoning?

7.1.2 Textual Entailment

This evaluation task we consider is arguably the most challenging setting of the three. Since all of our probes can be separated into a premise and a conclusion, we treat them as pairs and ask LMs to classify their relationships from three classes adopting from the Natural Language Inference (NLI) task: entailment, neutral, and contradiction. Since vanilla pre-trained LM cannot directly be applied to this task, we use a version of the LM that is fine-tuned on the MultiNLI [Williams et al., 2018b] dataset. Note that although the LMs are fine-tuned, we do not train or fine-tune on our probes.

7.2 Language Models Tested

We evaluate a range of state-of-the-art language models covering both masked and unidirectional language models to fit in our different settings. For the masked word prediction task, we consider RoBERTa [Liu et al., 2019] and ALBERT [Lan et al., 2019], two recent masked language models that show good results on many benchmarks. For sentence probability, we consider GPT-2 [Radford et al., 2019], a large language model for left-2-right language generation. For our extrinsic evaluation task of textual entailment, we use one masked language model RoBERTa and one sequence-to-sequence model BART [Lewis et al., 2019], both fine-tuned on MultiNLI [Williams et al., 2018a].

7.3 Evaluation Results

We show results from the additional settings and models in Table 4.
Table 4: Summary of all experiments run, including new tasks and new models tried.

### Task Model-Setting Dataset Size Accuracy Pos Accuracy Neg Accuracy

<table>
<thead>
<tr>
<th>Task</th>
<th>Model-Setting</th>
<th>Dataset Size</th>
<th>Accuracy</th>
<th>Pos Accuracy</th>
<th>Neg Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>MWP</td>
<td>RoBERTa</td>
<td>1440</td>
<td>0.51</td>
<td>0.899</td>
<td>0.122</td>
</tr>
<tr>
<td>MWP</td>
<td>RoBERTa-Fine Tuned</td>
<td>144</td>
<td>0.467</td>
<td>0.775</td>
<td>0.158</td>
</tr>
<tr>
<td>MWP</td>
<td>RoBERTa-Just Social</td>
<td>480</td>
<td>0.51</td>
<td>0.81</td>
<td>0.21</td>
</tr>
<tr>
<td>MWP</td>
<td>RoBERTa-Social w names</td>
<td>480</td>
<td>0.517</td>
<td>0.825</td>
<td>0.208</td>
</tr>
<tr>
<td>MWP</td>
<td>RoBERTa-Easy, Parrot</td>
<td>60</td>
<td>0.925</td>
<td>1.0</td>
<td>0.85</td>
</tr>
<tr>
<td>MWP</td>
<td>RoBERTa-Easy, Negation Switch</td>
<td>60</td>
<td>0.358</td>
<td>0.517</td>
<td>0.2</td>
</tr>
<tr>
<td>MWP</td>
<td>ALBERT</td>
<td>1440</td>
<td>0.49</td>
<td>0.734</td>
<td>0.246</td>
</tr>
<tr>
<td>Entailment</td>
<td>RoBERTa-Entailment, Contradiction</td>
<td>1440</td>
<td>0.05, 0.183</td>
<td>0.054, 0.168</td>
<td>0.046, 0.197</td>
</tr>
<tr>
<td>Entailment</td>
<td>RoBERTa-Just Social</td>
<td>480</td>
<td>0.107, 0.281</td>
<td>0.106, 0.258</td>
<td>0.108, 0.304</td>
</tr>
<tr>
<td>Entailment</td>
<td>RoBERTa-Social w Name</td>
<td>480</td>
<td>0.126, 0.286</td>
<td>0.129, 0.279</td>
<td>0.123, 0.293</td>
</tr>
<tr>
<td>Entailment</td>
<td>BART</td>
<td>1440</td>
<td>0.1, 0.089</td>
<td>0.107, 0.075</td>
<td>0.093, 0.102</td>
</tr>
<tr>
<td>Generative</td>
<td>GPT2</td>
<td>1440</td>
<td>0.493</td>
<td>0.777</td>
<td>0.209</td>
</tr>
<tr>
<td>Human</td>
<td>Human-20 Responses</td>
<td>72</td>
<td>0.845</td>
<td>0.831</td>
<td>0.863</td>
</tr>
</tbody>
</table>

Table 5: An example probe set—24 logically equivalent, but semantically different statements.
Table 6: The sixty probes we constructed and their corresponding templates