Pre-trained Language Models as Symbolic Reasoners over Knowledge?

Nora Kassner*, Benno Krojer*, Hinrich Schütze
Center for Information and Language Processing (CIS)
LMU Munich, Germany
kassner@cis.lmu.de

Abstract

How can pre-trained language models (PLMs) learn factual knowledge from the training set? We investigate the two most important mechanisms: reasoning and memorization. Prior work has attempted to quantify the number of facts PLMs learn, but we present, using synthetic data, the first study that establishes a causal relation between facts present in training and facts learned by the PLM. For reasoning, we show that PLMs learn to apply some symbolic reasoning rules; but in particular, they struggle with two-hop reasoning. For memorization, we identify schema conformity (facts systematically supported by other facts) and frequency as key factors for its success.

1 Introduction

Pre-trained Language models (PLMs) like BERT (Devlin et al., 2019), GPT-2 (Radford et al., 2019) and RoBERTa (Liu et al., 2019) have emerged as universal tools that capture a diverse range of linguistic and – as more and more evidence suggests – factual knowledge (Petroni et al., 2019; Radford et al., 2019).

Recent work on knowledge captured by PLMs is focused on probing, a methodology that identifies the set of facts a PLM has command of. But little is understood about how this knowledge is acquired during pre-training and why. We analyze the ability of PLMs to acquire factual knowledge focusing on two mechanisms: reasoning and memorization. We pose the following two questions:

a) **Symbolic reasoning**: Are PLMs able to infer knowledge not seen explicitly during pre-training?

b) **Memorization**: Which factors result in successful memorization of a fact by PLMs?

We conduct our study by pre-training BERT from scratch on synthetic corpora. The corpora are composed of short knowledge-graph like facts: subject-relation-object triples. To test whether BERT has learned a fact, we mask the object, thereby generating a cloze-style query and then measure prediction accuracy.

**Symbolic reasoning.** We create synthetic corpora to investigate four relational rules (reflexivity, symmetry, inversion, composition) and three logical rules (equivalence, negation, implication); see Table 1. For each rule, we create a corpus that contains facts from which the rule can be learned. We test BERT’s ability to use the rule to infer unseen facts by holding out some facts in a test set. For example, for composition, BERT should infer, by having seen that leopards are faster than sheep and sheep are faster than snails, that leopards are faster than snails. This type of inference is hard because we do not provide the necessary information (“the premise”) at inference time and, during training, it is scattered over the training corpus and interleaved with all other facts.

This setup is similar to link prediction in the knowledge base domain and therefore can be seen as a natural extension of the question “Language models as knowledge bases?” (Petroni et al., 2019). In the knowledge base domain, prior work (Sun et al., 2019; Zhang et al., 2020) has shown that models that are able to learn relational rules are superior to ones that are not.

Talmor et al. (2019) also investigate symbolic reasoning in BERT and use cloze-style queries. However, in their setup, there are two possible reasons for BERT having answered a cloze-style query correctly: the corresponding fact (i) was correctly inferred or (ii) was seen during training. In contrast, since we pre-train BERT from scratch, we have full control over the training setup and can distinguish cases (i) and (ii).

We find that BERT does well on learning one-hop rules (e.g., symmetry), but struggles with two-
Definition

Example

DivisibleBy(number, 2)  
Married(B. Obama, M. Obama)  
Faster(tiger, sheep)  
Is(Jupiter, big)  
Is(dog, Mammal)  
Contains(lactose, milk)  
Is(Number, even)

Table 1: Symbolic reasoning: relational rules (top) and logical rules (bottom) for knowledge triples with an example in natural language. $e_i$ are entities, $r_j$ relations and $a_j$ attributes.

hop rules (e.g., composition). However, by providing richer semantic context, even two-hop rules can be learned.

Given that BERT can in principle learn reasoning rules, the question arises whether it does so for standard training corpora. We find that standard BERT has only partially learned the types of rules we investigate here. For example, BERT has learned that “X is the opposite of Y” is symmetric, but it fails to understand rules like symmetry in many other cases.

Memorization. During the course of pretraining, BERT sees more data than any human could read in a lifetime, an amount of knowledge that surpasses its storage capacity. We simulate this with a scaled-down version of BERT and a corresponding training set that ensures that BERT cannot memorize all facts in the training set. We identify two important factors that lead to successful memorization of facts. (i) Frequency. Other things being equal, low-frequency facts (e.g., singletons) are not learned whereas frequent facts are. (ii) Schema conformity. Facts that conform with the overall schema of their entities (e.g., “sparrows can fly” in a corpus with many similar facts about birds) are easier to memorize than exceptions (e.g., “penguins can dive”).

2 Data

To thoroughly test PLM’s reasoning capabilities, natural corpora are inadequate since it is impossible to control what the model might have seen during training (e.g., on Wikipedia). Synthetic corpora provide an effective way to investigate reasoning by giving full control over what knowledge is seen and what kind of underlying rules the corpora follows.

In our investigation of PLMs as knowledge bases, it is natural to use subject-relation-object triples as basic units of knowledge, which we refer to as facts. The underlying vocabulary is composed of a set of entities $e_i \in \mathcal{E}$, relations $r_j \in \mathcal{R}$ and attributes $a_m \in \mathcal{A}$. Two types of triples are generated. (i) Attribute facts: relations linking entities to attributes $r_n(e_i, a_m)$, e.g., Is(leopard, fast). (ii) Entity facts: relations linking entities $r_l(e_i, e_j)$, e.g., Capital(Paris, France).

In the test set, we mask the objects and generate cloze-style queries of the format “$e_i$ $r_l$ [MASK]”. We evaluate performance with prediction accuracy.

2.1 Symbolic Reasoning

The vocabulary includes 5000 entities $e_i \in \mathcal{E}$, 200 relations $r_l \in \mathcal{R}$ and 500 attributes $a_m \in \mathcal{A}$.

Table 1 gives definitions and examples for the rules in question. These definitions serve as templates to generate facts. The templates involve entity, relation and attribute slots which are filled by sampling from the underlying vocabulary. A rule can link multiple facts. We call these linked facts one instance of the rule. We construct a separate corpus for every rule by filling these templates. Filling the respective templates follows a two step process. The first step fills the template slots defining the rule. The second generates 800 instances of that rule. Both consecutive steps are repeated 50 times, creating 50*800 total rule instances.

The relational rules we test are: reflexivity, symmetry, inversion and composition, which are entity facts. First, we sample $r_r \in \mathcal{R}$ to fill the relation slots defining the rule. Second, we sample $e_i \in \mathcal{E}$ to fill the entity slots thereby generating multiple instances of the rule.

A reflexive rule instance is defined by one fact and one relation $r_r \in \mathcal{R}$. We sample $e_i \in \mathcal{E}$ and generate $r_r(e_i, e_i)$.

A symmetric rule instance is defined by one relation $r_r \in \mathcal{R}$ but two facts. We sample pairs of $e_i, e_j \in \mathcal{E}$ and generate $r_r(e_j, e_i)$ and $r_r(e_i, e_j)$.

An inverse rule instance is defined by two facts and two relations $r_r, r_l \in \mathcal{R}$. We sample pairs of $e_i, e_j \in \mathcal{E}$ and generate $r_r(e_j, e_i)$ and $r_l(e_i, e_j)$.

A composition rule instance is defined by three
facts and three relations \( r_i, r_m \) and \( r_n \in \mathcal{R} \). We sample \( e_i, e_j, e_k \in \mathcal{E} \) and generate \( r_i(e_i, e_j), r_m(e_j, e_k) \) and \( r_n(e_i, e_k) \).

The logical rules we test are: equivalence, implication and negation which fall into the attribute fact category. Filling the respective templates now also involves filling attribute slots.

**Equivalence** is defined by two facts, two relations and two attributes. We sample pairs of \( r_k, r_l \in \mathcal{R} \) and \( a_i, a_l \in \mathcal{A} \) to define the rule. For each pair we sample \( e_i, e_j \in \mathcal{E} \) to generate multiple instances of the rule.

For **implication**, we link a single cause fact to 5 entailed effects. Therefore, we first sample a cause and entail \( e_i \in \mathcal{E} \) to generate the instances of the rule.

To model **negation** the “not” token is added to the vocabulary. The set of attributes \( \mathcal{A} \) is split in half \( \mathcal{A}1 \) and \( \mathcal{A}2 \). Then attributes are paired as antonyms: \( a_j = \text{antonym}(a_i) \) or \( a_i = \text{antonym}(a_j) \) for \( a_i \in \mathcal{A}1 \) and \( a_j \in \mathcal{A}2 \). We sample \( r_k \in \mathcal{R} \), \( e_i \in \mathcal{E} \) and \( a_i \in \mathcal{A} \) and generate \( r_k(e_i, a_i) \) and not \( r_k(e_i, \text{antonym}(a_i)) \).

Figure 1 depicts the training and test split. For each rule we generate a separate training corpus. 90% of the instances of the rule are completely seen during training e.g., both MarriedTo(Michelle Obama, Barack Obama) and MarriedTo(Barack Obama, Michelle Obama). From those instances the the PLM needs to learn the rule that MarriedTo is symmetric. For the remaining 10% only one direction is seen during training e.g., only MarriedTo(Pierre Curie, Marie Curie). During test time we query the other direction e.g., “Marie Curie is married to [MASK]”.

As multiple true objects are possible for the same subject-relationpair, we count the number of correct answers in the top-m ranked predictions normalize by m where m is the number of true objects.

In addition to the facts following the rule, we add equally many facts that form a control group and do not follow the rule. The control group consists of relations generating random facts. We add an additional pool of 200 relations to the vocabulary. We sample from this pool of relations and the original set of entities and attributes to generate random triplets The rational is that if a rule is truly learned it needs to be distinguishable from facts not following it.

In contrast to Clark et al. (2020), our setup allows the model to only see a subject-relation-object-triple per datapoint. This means that the rule cannot be inferred from a single training point but multihop reasoning is required.

We test whether i) BERT memorizes both the facts following and not following the rule seen during training, ii) is able to generalizes to the test set for the relations following the rule.

### 2.2 Memorization

First, we identify the number of facts needed to max out scaled-down BERT’s memorization capacity. The vocabulary includes 125000 entities \( e_i \in \mathcal{E} \), 22 relations \( r_i \in \mathcal{R} \) and 2250 attributes \( a_m \in \mathcal{A} \).

For the frequency baseline we generate 800,000 random but unique triples. In the training corpus, these triples occur in the range of 1 to 10 time. We test on the same facts and report prediction accuracy over the number of training occurrences.

For the scheme conform facts vs. exceptions experiments we split \( \mathcal{E} \) in 250 disjoint groups. For each group member we generate a set of facts:

- one attribute relation that defines group membership e.g., IsMember(robin, bird). We only put a fraction of 0.3 of all group membership facts into the training set.
Figure 2: Corpus generation for composition rule: The entity sets $A$, $B$, $C$ are linked via the relational rule. Every member of $A$ is linked via $r_1$ to every member of $B$, every member of $B$ is linked via $r_2$ to every member of $C$. Therefore, following the composition rule, every member of $A$ is linked via $r_3$ to every member of $C$. We hold out all facts from $A$ to $C$ via $r_3$ in 10% of the cases for the test set.

- one entity relation that links group members e.g., IsLinkedTo(robin, falcon). We only put a fraction of 0.0005 of all group member facts into the training corpus.

- a set of 10 attribute relations defining common group attributes e.g., Can(robin, fly), Has(robin, feathers), ReproductionVia(robin, eggs), etc. We only put a fraction of 0.3 of all common group facts into the training set.

- a set of 10 attribute relations defining unique attributes per entity e.g., Has(robin, redbreast), Is(robin, sedentary), OccursIn(robin, Eurasia), etc.

Additionally, we add a set of 300 exceptions per group. Random entities are chosen from the group and facts contradicting the group attributes are generated e.g., Can(penguin, dive).

We test on the same facts and report prediction accuracy for the different types of facts.

In a third experiment we combine the frequency baseline with the scheme conformity. We use the same setup as for the scheme conformity but we repeat the exceptions 10 times whereas the facts conform with the scheme are only seen once during training.

### 3 BERT Model

BERT uses a deep bidirectional Transformer (Vaswani et al., 2017) encoder to perform masked language modeling. During pre-training, BERT randomly masks positions and learns to fill the words. It is then fine-tuned on downstream NLP tasks. We use source code provided by Wolf et al. (2019)\(^1\). As RoBERTa (Liu et al., 2019) we perform dynamic masking and no next sequence prediction.

For symbolic rules we use BERT-base as is. Only the vocabulary file is adapted to the synthetic corpus. For the memorization experiments we need to max out BERT’s memorization capacity. Therefore, we scale it down to a single hidden layer with 3 number of attention heads, a hidden size of 192 and an intermediate size of 768.

### 4 Results

#### 4.1 Symbolic Reasoning

##### 4.1.1 Relational Rules

Results are reported in Table 2. Accuracy on the training set for relations following a relational rule and the relations randomly linking entities are 1.0. The reflexivity rule is a baseline which only requires one fact. Therefore the model can generalize this rule within a few training epochs to the test set.

The reflexivity rule is a baseline which only requires one fact. Therefore the model can generalize this rule within a few training epochs to the test set.

BERT is also able to perform one-hop reasoning: The symmetry and inversion rule are fully gener-

---

1. github.com/huggingface/transformers
We now sample three distinct groups of entities, which take the form of: $r_i(e_i, e_k), r_j(e_k, e_l)$; with $e_i \neq e_l$. Still, the preference for symmetry dominates for facts in the test set. Even some of the anti-rule facts in the training set remain unlearned. We observe that BERT has a general tendency to predict symmetrically even on datasets that are completely random.

BERT has difficulties with two-hop reasoning: The composition rule remains unlearned in the standard setup. To test if this is due to over-parametrization, we scale-down BERT to 6 layers but still composition remains unlearned.

We investigate this further in three experimental steps. i) We simplify composition to transitivity with $r_i=r_m=r_n$. Still the rule remains unlearned. ii) We enhance our corpus with more semantics: In the standard setup we sampled three relations and entities to form a complete instance of the composition rule. In the enhanced setting we sample again three relations but instead of single entities we now sample three distinct groups of entities $A$, $B$ and $C$ of size 10. The template slots are then filled for each group member. The full instance of the rule is now comprised of $10*10^3$ facts. This is illustrated in Figure 2. For example, entity groups could be car entities, bike entities and human entities and the relation FasterThan() ($r_i=r_m=r_n$). We then produce a number of facts such that the following holds: every car FasterThan() every bike, every bike FasterThan() every human, every car FasterThan() every human. For the purpose of testing we hold out all facts involving the third relation.

But to fully learn composition, we have to add more semantics: iii) We additionally enhance grouped composition by introducing facts stating group membership. Therefore, we introduce an additional relation “MemberOf()” and add group names to the vocabulary, e.g. we would state for each $e_i \in$ group A: ”MemberOf($e_i$, A). With this semantic help the composition rule is generalized to the test set.

### 4.1.2 Logical Rules

Results are reported in Table 3. Accuracy on the training set for logical rules and the random facts is 1.0.

**Implication** and equivalence are learned. Implication shows signs of over-fitting. Here, an entity that is part of the test set has been seen during training with other relations and attributes during training than the ones in question. When over-fitting, the model falls back to the seen ones, instead of generalizing the implication for that particular relation in question.

Negation remains unlearned. By tweaking design-parameters such as total number of attributes, we see an indication of generalization. With 200 attributes an accuracy of 0.8 is achieved. In a very simplified setting where an entity is only linked to a single attribute and its negated antonym, which is very similar to equivalence, the model can generalize easily.

Why is negation more challenging than implication? Implication allows the model to generalize over several entities all following the same rule. On the other hand for a model to actually learn negation and not implication it cannot be allowed to generalize by seeing many examples of entities with the same set of attributes and their negated antonyms. Instead it must learn antonym attributes from their usage in various contexts which seems to be more difficult.

### 4.2 Pre-trained BERT

We test BERT pre-trained on natural language for a consistent representation of the learned relational rules: symmetry and inversion, e.g. if BERT consistently predicts ”X is the opposite of Y” and ”Y is the opposite to X” even if the fact itself might be incorrect. If BERT’s predictions are asymmetric, we assume it has not learned to generalize. However if predictions are symmetric we can only assume it has generalized if the predictions are factually false or at least many answers are plausible in both ways. This is because if the predictions are symmetric and also factually correct, the model could have just seen both ways in training.

As many relational facts are not captured by BERT or answers span over multiple tokens, testing BERT’s consistency quantitatively is not easy.

We probe BERT-large-cased and ROBERTA-large for the symmetry relations: ”X is the opposite of [MASK]” and ”X shares borders with [MASK]”, and for the inverse relations ”X is the capital of [MASK]” and ”X’s capital is [MASK]”. For countries & planets we also test the inverse relation pair...
Table 4: Testing pre-trained BERT’s ability to systematically represent relational rules found in natural language: We report the more consistent result from probing both BERT-large-cased and ROBERTA-large. (i) First part of table: For a given set of entities (e.g. countries) we mask the object and then swap the predicted object token with the subject. We then count all entities with consistent and inconsistent predictions. (ii) Second part of table (similar setup as Talmor et al. (2019)): Entities are ordered by an attribute, e.g. countries by size. We query “Country A is [MASK] than Country B” and reversed “Country B is [MASK] than Country A”. If “larger” is higher ranked than “smaller” for one direction and the opposite holds for the reversed direction, we count this as consistent. For consistent predictions we indicate in brackets how many of them were factually correct. The last row shows examples of consistent (green) and inconsistent (red) completions.

| relation                  | rule          | consistent completions | inconsistent completions | example                                      |
|---------------------------|---------------|------------------------|--------------------------|----------------------------------------------|
| “shares borders with”     | symmetry      | 152(152)               | 2                        | Ecuador→Peru                                 |
|                           |               |                        |                          | Togo→Ghana, Ghana→Nigeria                   |
| “is the opposite of”      | symmetry      | 179(170)               | 71                       | demand→supply                                |
|                           |               |                        |                          | injustice→justice, justice→truth             |
| “is the capital of”       | inversion     | 59(59)                 | 1                        | Indonesia→Jakarta                            |
| s capital is”             |               |                        |                          | Canada→Ottawa, Ottawa→Ontario               |
| “larger/smaller”           | inversion     | 54(23)                 | 99                       | Russia larger Canada→Canada smaller Russia   |
| (countries)               |               |                        |                          | Brazil smaller Russia→Russia smaller Brazil  |
| “larger/smaller”           | inversion     | 9(9)                   | 36                       | Jupiter larger Mercury→Mercury smaller Jupiter |
| (planets)                 |               |                        |                          | Sun bigger Earth→Earth bigger Sun           |

larger/smaller by masking: “X is [MASK] than Y” similar to Talmor et al. (2019). Our probing setup and results are explained in Table 4.

Some probes in Table 4 show consistent predictions for the masked-object queries. But many of them are facts likely seen in training or due to more trivial co-occurrences. The larger-smaller-probes are highly inconsistent. The opposites-probe exhibits several cases that indicate lack of generalization but some where the model seems to intuitively have some notion of symmetry. To see if the model has understood antonymy, we probe with the semantically opposite relation “X is the same as [MASK]”, but keep the antonym-pairs for the subject and object slots. This causes the number of consistent predictions to drop from 94 to 61, e.g. the model generates “high is the same as low” and “low is the same as high”. These results show that the model has not properly learned a symmetric representation of “is the opposite of” but instead might just assign a high probability to an antonym in many scenarios.

4.3 Memorization

Experimental results for the memorization experiments are shown in Figure 3.

In A) we show results for the frequency baseline. We report prediction accuracies over the number of times a fact occurred during training. We see that frequent facts are preferred over rare facts. Facts that occurred 100 times per training epoch were stored while facts occurring only once were fully forgotten.

In B) we show results for the semantic scheme conformity experiment. For each type of fact the respective learning curves are displayed. In the beginning of training BERT focuses on attributes common for individual groups as well as exceptions going against those group attributes. In the course of training BERT picks up on group membership and unique facts distinguishing entities. Prediction accuracies for the exceptions plateau when BERT’s storage capacities are maxed out. All other facts conform with the overall scheme reach an prediction accuracy of 1.0.

In C) we show results for the combination of the frequency baseline with the semantic scheme. We again report learning curves for the respective types of facts. Right after the first training epoch, BERT’s prediction accuracies for the frequent exceptions jumps high and plateaus at 1.0 early on in training. The other learning curves pick up slowly. Again the group attributes are stored earlier than the other facts conform with the scheme. Towards the end of training the preference of frequent facts even tough they contradict the overall scheme remains unchanged. This time facts conform with the overall scheme are forgotten and exceptions are stored.

5 Discussion

Symbolic Reasoning: We find that BERT is able to perform one-hop reasoning over datapoints seen during training. The relational rules symmetry, in-
version, implication are generalized. Two-hop reasoning seems more difficult but with enough semantic enhancement two-hop reasoning is generalized.

The logical rules equivalence and implication are generalized. Negation is difficult to generalize. Experiments that e.g. reduce the total number of attributes, show a trend towards generalization. This is in agreement with findings by Kassner and Schutze (2020) where negation was not generalized during pre-training but fine-tuning on binary sequence classification enables BERT to learn negation.

We see a tendency of BERT to fall back to most similar datapoints seen during training. We find that BERT has a natural tendency for symmetry and we observed over-fitting in the case of implication. In that sense, language model’s objective seems problematic for generalization to unseen data.

Symbolic rules present in natural language diverge from the synthetic setting, posing a more difficult scenario. In natural language, rules are not presented in a consistent fashion, can be more complex and are interleaved with many other datapoints not following any rules. This scope of reasoning might well be beyond PLMs.

Exemplary tests of symmetry and inversion using pre-trained BERT suggest an inconsistent representation of relational rules. Talmor et al. (2019) support this finding. Indications of symbolic rules captured by pre-trained BERT are likely due to similar facts seen during training. Comparison with OlderThan() relations only worked for age spans in the range of humans.

Memorization: We see that fact frequency is the dominant factor influencing if a fact is stored or forgotten. Considering real world factual knowledge involving exceptions this seems desirable as many world concepts and categories go along with exceptions. Exceptions are not factually false e.g. a penguin can dive but not fly. BERT has the incentive to store exceptions if seen frequent enough during training. For rare facts not deducible via a semantic scheme this incentive is missing.

6 Related Work

Radford et al. (2019) show PLM’s knowledge ability to capture knowledge in a zero-shot question answering setting. Similarly, Petroni et al. (2019) investigate PLM’s ability to capture knowledge graph like facts. Our work builds on this and analyzes PLM’s ability to capture knowledge not explicitly seen during train.

Sun et al. (2019); Zhang et al. (2020) show in the knowledge graph domain that models that capture relational rules like symmetry, inversion and composition outperform the ones that do not. We investigate if BERT is also able to capture these rules and therefore generalize knowledge in a manner of link prediction in the knowledge graph setting.

Talmor et al. (2019) test PLM’s symbolic reasoning capabilities probing pre-trained and fine-tuned models with cloze-style queries. In contrast, we investigate if BERT generally is able to learn these
symbolic rules via relational and logical rules seen during pre-training. Additionally, we extend their probing setup and test pre-trained BERT for consistent representation of relational rules present in natural language.

Clark et al. (2020) fine-tune BERT for symbolic reasoning. In contrast to our work, they fine-tune pre-trained BERT by explicitly stating the rule and providing all necessary information needed in one training point. We analyze BERT’s pre-training objective, ask for multi-hop rule learning and do not state the rules explicitly.

There are a range of benchmark sets for complex reasoning QA (Yang et al., 2018; Sinha et al., 2019). In a transfer setting, PLMs are fine-tuned to the downstream task. PLMs are employed in a blackbox setting without clear understanding if reasoning capabilities are captured by the PLM or the task specific component.

Another line of work (Gururangan et al., 2018; Kaushik and Lipton, 2018; Dua et al., 2019; McCoy et al., 2019) shows that much of PLM’s performance on reasoning tasks is due to statistical artifacts in datasets, rather than exhibiting true reasoning and generalization capabilities. With the help of synthetic corpora we have full control over the training corpora.

Richardson et al. (2020) introduce a collection of synthetic corpora testing logic and monotonicity reasoning. They show that BERT performs poorly on these new datasets but can be quickly fine-tuned to good performance.

Roberts et al. (2020) show that the amount of knowledge captured by a PLM increases with model size. Our memorization experiments investigate what influences what knowledge is stored and what is forgotten.

Hupkes et al. (2020) study neural model’s ability to capture compositionality of language. We study different rules enabling knowledge acquisition in PLMs. Still, they support our finding that transformers have the ability to capture rules at the same time as exceptions. They neither study different rules nor do they consider what happens if model capacity is exhausted or fact frequency.

Guu et al. (2020) modify PLM’s objective to incentivize knowledge capturing. This does not go beyond the scope of memorization.

7 Conclusion
This work is a first study towards understanding BERT’s ability to capture knowledge seen during pre-training by investigating it’s reasoning and memorization capabilities. We identified factors influencing what knowledge is stored and what is forgotten and what is learnable beyond knowledge explicitly seen during training. We saw that theoretically BERT is able to infer facts not explicitly seen during training via symbolic rules. Future work should investigate how to enable BERT during pre-training to use this capability. We see the need to incentivize PLMs to capture symbolic rules and factual knowledge as this could potentially improve PLM’s performance also on downstream tasks where reasoning capabilities or implicit knowledge leverage is needed.

References
Peter Clark, Oyvind Tafjord, and Kyle Richardson. 2020. Transformers as soft reasoners over language. IJCAI.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In NAACL-HLT.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. REALM: Retrieval-augmented language model pre-training. ArXiv, abs/2002.08909.

Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. 2020. Compositionality decomposed: How do neural networks generalise? J. Artif. Intell. Res., 67:757–795.
Nora Kassner and Hinrich Schutze. 2020. Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. ACL.

Divyansh Kaushik and Zachary C. Lipton. 2018. How much reading does reading comprehension require? a critical investigation of popular benchmarks. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 5010–5015, Brussels, Belgium. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. CoRR, abs/1907.11692.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Kyle Richardson, Hai Hu, Lawrence S. Moss, and Ashish Sabharwal. 2020. Probing natural language inference models through semantic fragments. In Thirty-Fourth AAAI Conference on Artificial Intelligence.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? ArXiv, abs/2002.08910.

Koustuv Sinha, Shagun Sodhani, Jin Dong, Joelle Pineau, and William L. Hamilton. 2019. CLUTTR: A diagnostic benchmark for inductive reasoning from text. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 4506–4515, Hong Kong, China. Association for Computational Linguistics.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. In International Conference on Learning Representations.
A Hyper-parameters

For all reported results we trained with a batch-size of 1024 and a learning rate of 6e-5.

B Symbolic rules

We show datasets illustrating grouped symmetry, grouped composition with enhancement and implication. Each line is one datapoint. We also include the control group at the end of each corpus that doesn’t follow any rule. “{...}” indicate the sampled groups which is not part of the actual dataset.

B.1 Symmetry

{Alex, Mo, Lucy, Sato, Didier}

Alex SiblingOf Mo
Mo SiblingOf Alex
Alex SiblingOf Lucy
Lucy SiblingOf Alex
Alex SiblingOf Sato
Sato SiblingOf Alex
Alex SiblingOf Didier
Didier SiblingOf Alex
Mo SiblingOf Lucy
Lucy SiblingOf Mo ...
Sato SiblingOf Didier
Didier SiblingOf Sato ...
{Lea, Keith, Fatima, Gertrud, Klaus}

... Lea SiblingOf Keith
Keith SiblingOf Lea ...
...

B.2 Composition with semantic enhancement

{e8, e2, e8, e5}
e8 InstanceOf A
e2 InstanceOf A
e4 InstanceOf A
e11 InstanceOf A
{e15, e13, e12, e19}
e3 InstanceOf B
e17 InstanceOf B

B.3 Implication

{(Flu), (Cough, RunningNose, Headache, Fever)}

Kevin HasDisease Flu
Kevin HasSymptom Cough
Kevin HasSymptom RunningNose
Kevin HasSymptom Headache
Kevin HasSymptom Fever
Marim HasDisease Flu
Marim HasDisease Cough