Are Pretrained Language Models 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 pretrained 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 investigates the causal relation between facts present in training and facts learned by the PLM. For reasoning, we show that PLMs seem to learn to apply some symbolic reasoning rules correctly but struggle with others, including two-hop reasoning. Further analysis suggests that even the application of learned reasoning rules is flawed. For memorization, we identify schema conformity (facts systematically supported by other facts) and frequency as key factors for its success.

1 Introduction

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

We conduct our study by pretraining 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 evaluate predictions.

Symbolic reasoning. We create synthetic corpora to investigate six symbolic rules (equivalence, symmetry, inversion, composition, implication, negation); 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, after having seen that leopards are faster than sheep and sheep are faster than snails, that leopards are faster than snails.

Our 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 symbolic rules are superior to ones that are not.

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

A unique feature of our approach compared to prior work (Sinha et al., 2019; Richardson et al., 2020; Weston et al., 2016; Clark et al., 2020) is that we do not gather all relevant facts and present them to the model at inference time. This is a crucial

*equal contribution

Proceedings of the 24th Conference on Computational Natural Language Learning, pages 552–564
Online, November 19-20, 2020. ©2020 Association for Computational Linguistics
https://doi.org/10.18653/v1/P17

Rule		Definition	Example
EQUI	Equivalence	$(e, r, a) \iff (e, s, a)$	(bird, can, fly) ⟺ (bird, is able to, fly)
SYM	Symmetry	$(e, r, f) \iff (f, r, e)$	(barack, married, michelle) ⟺ (michelle, married, barack)
INV	Inversion	$(e, r, f) \iff (f, s, e)$	(john, loves, soccer) ⟺ (soccer, thrills, john)
NEG	Negation	$(e, r, a) \iff (e, \text{not } r, b)$	(jupiter, is, big) ⟺ (jupiter, is not, small)
IMP	Implication	$(e, r, a) \Rightarrow (e, s, b), (e, s, c), \ldots$	(dog, is, mammal) ⇒ (dog, has, hair), (dog, has, neocortex), ...
COMP	Composition	$(e, r, f) \wedge (f, s, g) \Rightarrow (e, t, g)$	(tiger, faster than, sheep) ∧ (sheep, faster than, snail) ⇒ (leopard, faster than, snail) with $r = s = t$

Table 1: The six symbolic rules we investigate (cf. (Nayyeri et al., 2019)) with an example in natural language for entities $e, f, g \in E$, relations $r, s, t \in R$ and attributes $a, b, c \in A$.

difference – note that human inference similarly does not require that all relevant facts are explicitly repeated at inference time.

We find that i) BERT is capable of learning some one-hop rules (equivalence and implication). ii) For others, even though high test precision suggests successful learning, the rules were not in fact learned correctly (symmetry, inversion and negation). iii) BERT struggles with two-hop rules (composition). However, by providing richer semantic context, even two-hop rules can be learned.

Given that BERT can in principle learn some reasoning rules, the question arises whether it does so for standard training corpora. We find that BERT-large has only partially learned the types of rules we investigate here. For example, BERT has some notion of "X shares borders with Y" being symmetric, but it fails to understand rules like symmetry in 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 training set that ensures that BERT cannot memorize all facts in training. We identify two important factors that lead to successful memorization. (i) Frequency: Other things being equal, low-frequency facts 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").

We publish our code for training and data generation. [1]

2 Data

To test PLMs' reasoning capabilities, natural corpora like Wikipedia are limited since it is difficult to control what the model sees during training. Synthetic corpora provide an effective way of investigating reasoning by giving full control over what knowledge is seen and which rules are employed in generating the data.

In our investigation of PLMs as knowledge bases, it is natural to use (subject, relation, object) triples as basic units of knowledge; we refer to them as *facts*. The underlying vocabulary consists of a set of entities $e, f, g, \ldots \in E$, relations $r, s, t, \ldots \in R$ and attributes $a, b, c, \ldots \in A$, all represented by artificial strings such as e_{14}, r_3 or a_{35}. Two types of facts are generated. (i) **Attribute facts**: relations linking entities to attributes, e.g., (e, r, a) = (leopard, is, fast). (ii) **Entity facts**: relations linking entities, e.g., (e, r, f) = (Paris, is the capital of, France).

In the test set, we mask the objects and generate cloze-style queries of the form "e r [MASK]". The model's task is then to predict the correct object.

2.1 Symbolic Reasoning

Table 1 gives definitions and examples for the six rules (EQUI, SYM, INV, COMP, IMP, NEG) we investigate. The definitions are the basis for our corpus generation algorithms, shown in Figure 1. SYM, INV, COMP generate entity facts and EQUI, IMP, NEG attribute facts. We create a separate corpus for each symbolic rule. Facts are generated by sampling from the underlying vocabulary. For §2.1, this vocabulary consists of 5000 entities, 500 relations and 1000 attributes. Half of the relations follow the rule, the other half is used to generate random facts of entity or attribute type.

We can most easily think of the corpus generation as template filling. For example, looking at SYM in Table 1, the template is $(e, r, f) \iff (f, r, e)$. We first sample a relation r from R and

[1] https://github.com/BennoKrojer/reasoning-over-facts

```
EQUI
$C = \emptyset, D = \emptyset$
for $i \in 1 \ldots n$ do
  $(r, s) \sim R \times R$
  $a \sim A$
  for $j \in 1 \ldots m$ do
    $e \sim E$
    addC=Bernoulli$(0.5)$
    if addC then
      $C = C \cup \{(e, r, a)\}$
      $D = D \cup \{(e, s, a)\}$
    else
      $C = C \cup \{(e, s, a)\}$
      $D = D \cup \{(e, r, a)\}$
```

```
SYM
$C = \emptyset, D = \emptyset$
for $i \in 1 \ldots n$ do
  $r \sim R$
  for $j \in 1 \ldots m$ do
    $(e, f) \sim E \times E$
    $C = C \cup \{(e, r, f)\}$
    $D = D \cup \{(f, r, e)\}$
```

```
INV
$C = \emptyset, D = \emptyset$
for $i \in 1 \ldots n$ do
  $(r, s) \sim R \times R$
  for $j \in 1 \ldots m$ do
    $(e, f) \sim E \times E$
    $C = C \cup \{(e, r, f)\}$
    $D = D \cup \{(f, s, e)\}$
```

```
COMP
$C = \emptyset, D = \emptyset$
for $i \in 1 \ldots n$ do
  $(r, s, t) \sim R \times R \times R$
  for $j \in 1 \ldots m$ do
    $(e, f, g) \sim E \times E \times E$
    $C = C \cup \{(e, r, f)\}$
    $C = C \cup \{(f, s, g)\}$
    $D = D \cup \{(e, t, g)\}$
```

```
IMP
$C = \emptyset, D = \emptyset$
for $i \in 1 \ldots n$ do
  $(r, s) \sim R \times R$
  for $k \in 1 \ldots l$ do
    $b \sim A$
    $\alpha \sim A \times \ldots \times A$
    for $j \in 1 \ldots m$ do
      $e \sim E$
      $C = C \cup \{(e, r, b)\}$
      for $a \in \alpha$ do
        $D = D \cup \{(e, s, a)\}$
```

```
NEG
$C = \emptyset, D = \emptyset$
for $i \in 1 \ldots n$ do
  $r \sim R$
  for $j \in 1 \ldots m$ do
    $e \sim E$
    $a \sim A$
    $b = $ antonym$(a)$
    negated=Bernoulli$(0.5)$
    if negated then
      $C = C \cup \{(e, \text{not } r, a)\}$
      $D = D \cup \{(e, r, b)\}$
    else
      $C = C \cup \{(e, r, a)\}$
      $D = D \cup \{(e, \text{not } r, b)\}$
```

Figure 1: Pseudocode for symbolic reasoning corpus generation. "$a \sim A$" stands for: a is randomly sampled from A. ("$\alpha \sim A \times \ldots \times A$": a tuple of 4 attributes is sampled.) The vocabulary consists of entities $e, f, g \in E$, relations $r, s, t \in R$ and attributes $a, b, c \in A$. Train/test corpora are formed from C and D. $n = 20$, $m = 800$, $l = 2$. See §2.1 for details.

```
FREQ
$C = \emptyset$
$m = 1$
for $i \in 1 \ldots n$ do
  $(e, f) \sim E \times E$
  $r \sim R$
  for $j \in 1 \ldots m$ do
    $C = C \cup \{(e, r, f)\}$
  if $i\%(n/100) == 0$ then
    $m+ = 1$
```

```
SCHEMA
$C = \emptyset$
for $i \in 1 \ldots k$ do
  $\delta \sim E \times \ldots \times E$
  for $r$ in $R$ do
    schema $=$ Bernoulli$(0.5)$
    if schema then
      $\alpha \sim A \times ... \times A$
      for $e \in \delta$ do
        for $a \in \alpha$ do
          add $=$ Bernoulli$(0.5)$
          if add then
            $C = C \cup \{(e, r, a)\}$
          else
            exception $=$ Bernoulli$(0.5)$
            if exception then
              $a \sim A$
              $C = C \cup \{(e, r, a)\}$
    else
      for $e \in \delta$ do
        add $=$ Bernoulli$(0.5)$
        if add then
          $a \sim A$
          $C = C \cup \{(e, r, a)\}$
```

Figure 2: Pseudocode for memorization corpus generation. "$a \sim A$" stands for: a is randomly sampled from A. ("$\delta \sim E \times \ldots \times E$": a tuple of 250 entities is sampled. "$\alpha \sim A \times ... \times A$": a tuple of 10 attributes is sampled.) The vocabulary consists of entities $e \in E$, relations $r \in R$ and attributes $a \in A$. C is both training set and test set. $n = 800{,}000$, $k = 250$. See §2.2 for details.

then two entities e and f from E. We then add (e, r, f) and (f, r, e) to the corpus – this is one *instance* of applying the SYM rule from which symmetry can be learned. Similarly, the other rules also generate instances.

For each of the other rules, the template filling is modified to conform with its definition in Table 1. INV corresponds directly to SYM. COMP is a two-hop rule whereas the other five are one-hop rules. EQUI generates instances from which one can learn that the relations r and s are equivalent. IMP generates implication instances, e.g., (e, r, b) (= (dog, is, mammal)) implies (e, s, a_1) (= (dog, has, hair)), (e, s, a_2) (= (dog, has, neocortex)) etc. Per premise we create four implied facts.

For NEG, we generate pairs of facts (e, r, a) (= (jupiter, is, big)) and $(e, \text{not}\, r, b)$ (= (jupiter, is not, small)). We define the antonym function in Figure 1 (NEG) as returning for each attribute its antonym, i.e., attributes are paired, each pair consisting of a positive and a negative attribute.

Each of the six generation algorithms has the outer loop "for $i \in 1 \ldots n$" (where $n = 20$) that samples one, two or three relations (and potentially attributes) and generates a subcorpus for these relations; and the inner loop "for $j \in 1 \ldots m$" (where $m = 800$) that generates the subcorpus of instances for the sampled relations.

Train/test split. The data generation algorithms generate two subsets of facts C and D, see Figure 1. For each rule, we merge all of C with 90% of D (randomly sampled) to create the training set. The rest of D (i.e., the other 10%) serves as the test set.

For some of the cloze queries "e r [MASK]", there are multiple correct objects that can be substituted for MASK. Thus, we rank predictions and compute precision at m, i.e., precision in the top m where m is the number of correct objects. We average precision at m for all cloze queries.

This experimental setup allows us to test to what extent BERT learns the six rules, i.e., to what extent the facts in the test set are correctly inferred from their premises in the training set.

2.2 Memorization

For memorization, the vocabulary consists of 125,000 entities, 20 relations and 2250 attributes.

Effect of frequency on memorization. Our first experiment tests how the frequency of a fact influences its successful memorization by the model. Figure 2 (left, FREQ) gives the corpus generation algorithm. The outer loop generates 800,000 random facts. These are divided up in groups of 8000. A fact in the first group of 8000 is added once to the corpus, a fact from the second group is added twice and so on. A fact from the last group is added 100 times to the corpus. The resulting corpus C is both the training set and the test set.

Effect of schema conformity. In this experiment, we investigate the hypothesis that a fact can be memorized more easily if it is schema conformant.

Figure 2 (right, SCHEMA) gives the corpus generation algorithm. We first sample an entity group: $\delta \sim E \times \ldots \times E$. For each group, relations are either related to the schema ("if schema") or are not (else clause). For example, for the schema "primate" the relations "eat" (eats fruit) and "climb" (climbs trees) are related to the schema, the relation "build" is not since some primates build nests and treehouses, but others do not.

For non-schema relations, facts with random attributes are added to the corpus. In Figure 4, we refer to these facts as (facts with) **unique attributes**. For relations related to the schema, we sample the attributes that are part of the schema: $\alpha \sim A \times \ldots \times A$ (e.g., ("paranut",...,"banana") for "eat"). Facts are then generated involving these attributes and added to the corpus. In Figure 4, we refer to these facts as (facts with) **group attributes**. We also generate exceptions (e.g., "eats tubers") since schemas generally have **exceptions**.

Similarly, the two lines "add = Bernoulli(0.5)" are intended to make the data more realistic: for a group of entities, its relations and its attributes, the complete cross product of all facts is not available to the human learner. For example, a corpus may contain sentences stating that chimpanzees and baboons eat fruit, but none that states that gorillas eat fruit.

For this second memorization experiment, training set and test set are again identical (i.e., $= C$).

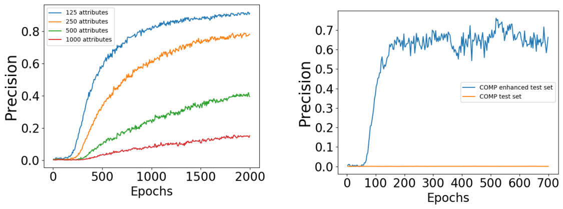
In a final experiment, we modify SCHEMA as follows: exceptions are added 10 times to the corpus (instead of once). This tests the interaction between schema conformity and frequency.

3 BERT Model

BERT uses a deep bidirectional Transformer (Vaswani et al., 2017) encoder to perform masked language modeling. During pretraining, BERT randomly masks positions and learns to predict fillers. We use source code provided by Wolf et al. (2019). Following (Liu et al., 2019), we perform dynamic masking and no next sequence prediction.

rule	train	test
EQUI	99.95	98.28
SYM	99.97	98.40
INV	99.99	87.21
IMP	100.00	80.53
NEG	99.98	20.54
COMP	99.98	0.01
ANTI	100.00	14.85

Table 2: Precision in % of completing facts for symbolic rules. Training corpora generated as specified in Figure 1. See §4.1 for detailed discussion.



(A) NEG (B) COMP

Figure 3: Learning curves for symbolic reasoning. (A) shows precision for NEG with a varying number of attributes. A reduction to 125 attributes enables BERT to successfully apply antonym negation to the test set. (B) shows test set precision for COMP following the standard setup (orange) and an enhanced version (blue). Only in enhanced, i.e., with the introduction of additional facts adding more semantic information, is COMP generalized.

For symbolic rules, we start with BERT-base and tune hyperparameters. We vary the number of layers to avoid that rule learning fails due to over-parametrization, see appendix for details. We report precision based on optimal configuration.

In the memorization experiment, our goal is to investigate the effect of frequency on memorization. Due to a limited compute infrastructure, we scale down BERT to a single hidden layer with 3 attention heads, a hidden size of 192 and an intermediate size of 768.

4 Results, Analysis and Discussion

4.1 Symbolic Reasoning

Table 2 gives results for the symbolic reasoning experiments. BERT has high test set precision for EQUI, SYM, INV and IMP. As we see in Table 1, these rules share that they are "one-hop": The inference can be straightforwardly made from a single premise to a conclusion, e.g., "(barack married michelle)" implies "(michelle married barack)". The crucial difference to prior work is that the premise is not available at inference time. "(michelle married barack)" is correctly inferred by the model based on its memory of having seen the fact "(barack married michelle)" in the training set and based on the successful acquisition of the symmetry rule. Table 2 seems to suggest that BERT is able to learn one-hop rules and it can successively apply these rules in a natural setting in which the premise is not directly available.

In the rest of this section, we investigate these results further for SYM, INV, NEG and COMP.

4.1.1 Analysis of SYM and INV

Table 2 seems to indicate that BERT can learn that a relation r is symmetric (SYM) and that s and t are inverses (INV) – the evidence is that it generates facts based on the successfully acquired symmetry and inversion properties of the relations r, s and t. We now show that while BERT acquires SYM and INV partially, it also severely overgenerates. Our analysis points to the complexity of evaluating rule learning in PLMs and opens interesting avenues for future work.

Our first observation is that in the SYM experiment, BERT understands *all relations* to be symmetric. Recall that of the total of 500 relations, 250 are symmetric and 250 are used to generate random facts. If we take a fact with a random relation r, say (e, r, f), and prompt BERT with "$(f, r, [MASK])$", then e is predicted in close to 100% of cases. So BERT has simply learned that any relation is symmetric as opposed to distinguishing between symmetric and non-symmetric relations.

This analysis brings to light that our setup is unfair to BERT: it never sees evidence for non-symmetry. To address this, we define a new experiment, which we call ANTI because it includes an additional set of "anti" relations that are sampled from R^* with $R^* \cap R = \emptyset$ and $|R| = |R^*|$. ANTI facts take the following form: (e, r, f), (f, r, g) with $e \neq g$. Using this ANTI template we follow the standard data generation procedure. The corpus is now composed of symmetric, anti-symmetric and random facts. ANTI training data indicate to BERT that $r \in R^*$ is not symmetric since many instances of r facts are seen, with specific entities (f in the example) occurring in both slots, but there is never a symmetric example.

Table 2 (ANTI) shows that BERT memorizes ANTI facts seen during training but on test, BERT only recognizes 14.85% of ANTI facts as non-symmetric. So it still generalizes from the 250 symmetric relations to most other relations (85.15%), even those without any "symmetric" evidence in training. So it is easy for BERT to learn the concept of symmetry, but it is hard to teach it to distinguish between symmetric and non-symmetric relations.

Similar considerations apply to INV. BERT successfully predicts correct facts once it has learned that s and t are inverses – but it overgeneralizes by also predicting many incorrect facts; e.g., for (e, s, f) in train, it may predict (f, t, e) (correct), but also (e, t, f) and (f, s, e) (incorrect).

In another INV experiment, we add, for each pair of (f, r, e) and (e, s, f) two facts that give evidence of non-symmetry: (f, r, g) and (e, s, h) with $e \neq g$ and $h \neq f$. We find that test set precision for INV (i.e., inferring (e, s, f) in test from (f, r, e) in train) drops to 17% in this scenario. As for SYM, this indicates how complex the evaluation of rule learning is.

In summary, we have found that SYM and INV are learned in the sense that BERT generates correct facts for symmetric and inverse relations. But it severely overgenerates. Our analysis points to a problem of neural language models that has not received sufficient attention: they can easily learn that the order of arguments is not important (as is the case for SYM relations), but it is hard for them to learn that this is the case *only for a subset of relations*. Future work will have to delineate the exact scope of this finding – e.g., it may not hold for much larger training sets with millions of occurrences of each relation. Note, however, that human learning is likely to have a bias against symmetry in relations since the vast majority of verbs[2] in English (and presumably relations in the world) is asymmetric. So unless we have explicit evidence for symmetry, we are likely to assume a relation is non-symmetric. Our results suggest that neural language models do not have this bias – which would be problematic when using them for learning from natural language text.

4.1.2 Analysis of NEG

NEG was the only rule for which parameter tuning improved performance. A reduction to four layers obtained optimal results.

In Table 2 we report a test set precision of 20.54%. Why is negation more challenging than implication? Implication allows the model to generalize over several entities all following the same rule (e.g., every animal that is a mammal has a

[2] For example, almost all of the verb classes in (Levin, 1993) are asymmetric.

neocortex). This does not hold for negation (e.g., a leopard is fast but a snail is not fast). BERT must learn antonym negation from a large number of possible combinations. By reducing the number of possible combinations (decreasing the number of attributes from 1000 to 500, 250 and 125) BERT's test set precision increases, see Figure 3 (A). With 125 attributes a precision of 91% is reached. A reduction of attributes makes antonym negation very similar to implication.

We investigate BERT's behavior concerning negation further by adding an additional attribute set A^*, with $A^* \cap A = \emptyset$ and $|A| = |A^*|$ to the vocabulary. A^* does not follow an antonym schema. We sample $a \in A^*, e \in E, r \in R$ to add additional random facts of the type (e, r, a) or (e, not r, a) to NEG's training set. After training we test on the additional random facts seen during training by inserting or removing the negation marker. We see that BERT is prone to predict both (e, r, b) and (e, not r, b) for b $\in A^*$ (for 38%). Antonym negation was still learned.

We conclude that antonym negation can be learned via co-occurrences but a general concept of negation is not understood.

This is in agreement with prior work (Ettinger, 2020; Kassner and Schütze, 2020) showing that BERT trained on natural language corpora is as likely to generate a true statement like "birds can fly" as a factually false negated statement like "birds cannot fly".

4.1.3 Analysis of COMP

Why does BERT not learn COMP? COMP differs from the other rules in that it involves two-hop reasoning. Recall that a novelty of our experimental setup is that premises are not presented at inference time – two-hop reasoning requires that two different facts have to be "remembered" to make the inference, which intuitively is harder than a one-hop inference. Figure 3 (B) shows that the problem is not undertraining (orange line).

Similar to the memorization experiment, we investigate whether stronger semantic structure in form of a schema can make COMP learnable. We refer to this new experiment as **COMP enhanced**. Data generation is defined as follows: Entities are divided into groups of 10. Relations are now defined between groups in the sense that the members of a group are "equivalent". More formally, we sample entity groups (groups of 10) E_1, E_2, E_3 and relations r, s, t. For all $e_1 \in E_1, e_2 \in E_2, e_3 \in E_3$, we add (e_1, r, e_2) and (e_2, s, e_3) to C and (e_1, t, e_3) to D. In addition, we introduce a relation "samegroup" and add, for all $e_m, e_n \in E_i$, $(e_m, \text{samegroup}, e_n)$ to C – this makes it easy to learn group membership. As before, the training set is the merger of C and 90% of D and the test set is the rest of D.

Similar semantic structures occur in real data. The simplest case is a transitive example: (r) planes (group 1) are faster than cars (group 2), (s) cars (group 2) are faster than bikes (group 3), (t) planes (group 1) are faster than bikes (group 3).

Figure 3 (B) shows that BERT can learn COMP moderately well from this schema-enhanced corpus (blue curve): precision is clearly above 50% and peaks at 76%.

The takeaway from this experiment is that two-hop rules pose a challenge to BERT, but that they are learnable if entities and relations are embedded in a rich semantic structure. Prior work (Brown et al., 2020) has identified the absence of "domain models" (e.g., a domain model for common sense physics) as one shortcoming of PLMs. To the extent that PLMs lack such domain knowledge (which we simulate here with a schema), they may not be able to learn COMP.

4.2 Natural Language Corpora

In this section, we investigate to what extent the PLMs BERT and RoBERTa have learned SYM and INV from natural language corpora. See Table 3. For "smaller/larger" (INV), we follow Talmor et al. (2019) and test which of the two words is selected as the more likely filler in a pattern like "Jupiter is [MASK] than Mercury". For the other three relations ("shares borders with" (SYM), "is the opposite of" (SYM), "is the capital of" / "'s capital is" (INV)), we test whether the correct object is predicted in the pattern "e r [MASK]" (as in the rest of the paper). We give the number of (i) consistent ("cons."), (ii) correct and consistent ("correct") and (iii) inconsistent ("inc.") predictions. (A prediction is consistent and incorrect if it is consistent with the rule, but factually incorrect.)

In more detail, we take a set of entities (countries like "Indonesia", cities like "Jakarta") or adjectives like "low" that are appropriate for the relation and test which of the entities / adjectives is predicted. For each of the five relations, we run both BERT-large-cased and RoBERTa-large and report the more consistent result.

relation	rule	completions			examples
		cons.	correct	inc.	
shares borders with	SYM	152	152	2	(ecuador,peru) (togo,ghana), (ghana,nigeria)
is the opposite of	SYM	179	170	71	(demand,supply) (injustice,justice), (justice,truth)
is the capital of (C-of) *'s capital is (s-C-is)*	INV	59	59	1	(indonesia,s-C-is,jakarta) (canada,s-C-is,ottawa), (ottawa,C-of,ontario)
is smaller/larger than (countries)	INV	54	23	99	(russia,larger,canada), (canada,smaller,russia) (brazil,smaller,russia), (russia,smaller,brazil)
is smaller/larger than (planets)	INV	9	9	36	(jupiter,larger,mercury), (mercury,smaller,jupiter) (sun,bigger,earth), (earth,bigger,sun)

Table 3: Can PLMs (BERT and RoBERTa) learn SYM and INV from natural language corpora? For "smaller/larger", we follow Talmor et al. (2019) and test which of the two words is selected as a filler in a pattern like "Jupiter is [MASK] than Mercury". For the other three relations, we test whether the correct object is predicted (as in the rest of the paper). We give the number of (i) consistent ("cons."), (ii) correct and consistent ("correct") and (iii) inconsistent ("inc.") predictions. Blue: consistent examples. Red: inconsistent examples. (We make the simplifying assumption that "justice" can only have one opposite.)

Consistency and accuracy are high for "shares borders with" and "capital". However, this is most likely due to the fact that many of these facts occur verbatim in the training corpora of the two models. For example, Google shows 54,800 hits for "jakarta is the capital of indonesia" and 1,290 hits for "indonesia's capital is jakarta" (both as a phrase). It is not possible to determine which factor is decisive here: successful rule-based inference or memorization. The ultimate futility of this analysis is precisely the reason that we chose to work with synthetic data.

Consistency for "is the opposite of" is much lower than for the first two relations, but still decent. To investigate this relation further, we also tested the relation "is the same as". It turns out that many of the "opposite" objects are also predicted for "is the same as", e.g., "high is the same as *low*" and "low is the same as *high*" where the predicted word is in italics. This indicates that the models have not really learned that "is the opposite of" is symmetric, but rather know that antonyms are closely associated and often occur together in phrases like "X is the opposite of Y", "X and Y", "X noun, Y noun" (e.g., "good cop, bad cop") etc. Apparently, this is then incorrectly generalized to "is the same as".

Consistency and accuracy are worse for "smaller/larger". "smaller/larger" sentences of the sort considered here are probably rarer in genres like Wikipedia than "shares borders with" and "is the capital of". A Wikipedia article about a country will always say what its capital is and which countries it borders, but it will not enumerate the countries that are smaller or larger.

In summary, although we have shown that pretrained language models have some ability to learn symbolic rules, there remains considerable doubt that they can do so based on natural corpora.

4.3 Memorization

Experimental results for the memorization experiments are shown in Figure 4.

(A) shows that frequent facts are memorized well (0.8 for frequency 100) and that rare facts are not ($\approx$ 0.0 for frequencies < 15).

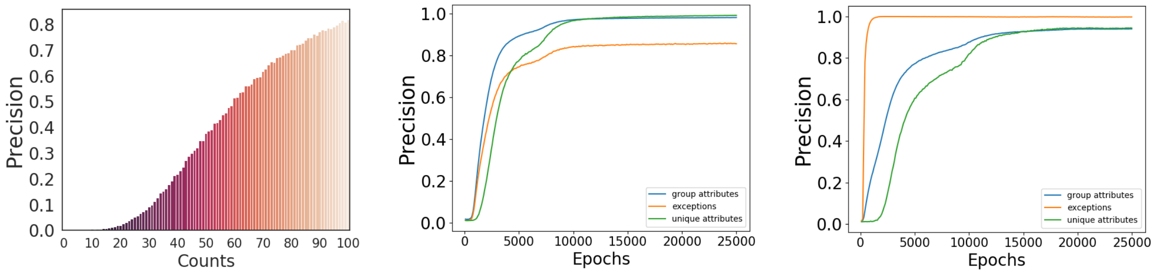
(B) shows that BERT memorizes schema conformant facts perfectly ("group attributes"). Accuracy for exceptions is clearly lower than those of schema conformant facts: about 80%. The frequency of each fact in the training corpus in this experiment is 1. Overall, the total amount of exceptions is much lower than the total amount of schema conformant facts.

(C) shows that exceptions are perfectly learned if 10 copies of each exception are added to the corpus – instead of 1 in (B). In this case, limited capacity affects memorization of schema-conformant facts: accuracy drops to ≈ 0.9.

In summary, we find that both frequency and schema conformity facilitate memorization. Schema conformant facts and exceptions compete for memory if memory capacity is limited – depending on frequency one or the other is preferentially learned by BERT.

5 Limitations

Our experimental design makes many simplifying assumptions: i) Variation in generated data is more



(A) frequency

(B) schema conformity, exceptions rare

(C) schema conformity, exceptions frequent

Figure 4: Memorization experiments. We investigate the effect of frequency and schema conformity on memorization. (A) Frequent facts are memorized well (0.8 for frequency 100), rare facts are not (≈ 0.0 for frequencies < 15). (B) BERT memorizes schema conformant facts perfectly ("group attributes"). Accuracy for rare exceptions is clearly lower (80%). (C) Exceptions are perfectly learned if 10 copies of each exception are added to the corpus – instead of 1 in (B). In this case, limited capacity affects memorization of schema-conformant facts ("group attributes" drops to ≈ 0.9).

limited than in naturally occurring data. ii) Semantics are deliberately restricted to one rule only per generated corpus. iii) We do not investigate effects of model and corpus size.

i) In natural corpora relations can have more than two arguments, entities can have several tokens, natural data are noisier than synthetic data etc. Also, we study each rule in isolation.

ii) While our simplified corpora make learning easier in some respects, they may make it harder in others. Each corpus is focused on providing training material for one symbolic rule, but it does not contain any other "semantic" signal that may be helpful in learning symbolic reasoning: distributional signals, entity groupings, hierarchies, rich context etc. The experimental results of "COMP enhanced" indicate that indeed such signals are beneficial to symbolic rule learning. The interplay of such additional sources of information for learning with symbolic rules is an interesting question for follow up work.

iii) Results are based on BERT-base and scaled-down versions of BERT-base only, just as training corpora are orders of magnitude smaller than natural training corpora. We varied model and corpus sizes within the limits of our compute infrastructure, but did not systematically study their effect on our findings.

Our work is an initial exploration of the question whether symbolic rules can be learned in principle, but we view it mainly as a starting point for future work.

6 Related Work

Radford et al. (2019) and Petroni et al. (2019) show in a zero-shot question answering setting that PLMs have factual knowledge. Our main question is: under what conditions do PLMs learn factual knowledge and do they do so through memorization or rule-based inference?

Sun et al. (2019) and Zhang et al. (2020) show in the knowledge graph domain that models that have the ability to capture symbolic rules like SYM, INV and COMP outperform ones that do not. We investigate this question for PLMs that are trained on language corpora.

Talmor et al. (2019) test PLMs' symbolic reasoning capabilities probing pretrained and finetuned models with cloze-style queries. Their setup makes it impossible to distinguish whether a fact was inferred or memorized during pretraining. Our synthetic corpora allow us to make this distinction.

Clark et al. (2020) test finetuned BERT's reasoning capabilities, but they always make premise and conclusion locally available to the model, during training and inference. This is arguably not the way much of human inference works; e.g., the fact F that X borders Y allows us to infer that Y borders X even if we were exposed to F a long time ago.

Richardson et al. (2020) introduce synthetic corpora testing logic and monotonicity reasoning. They show that BERT performs poorly on these new datasets, but can be quickly finetuned to good performance. The difference to our work again is that they make the premise available to the model at inference time.

For complex reasoning QA benchmarks (Yang et al., 2018; Sinha et al., 2019), PLMs are finetuned to the downstream tasks. Their performance is difficult to analyze: it is not clear whether any reasoning capability is learned by the PLM or by 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 PLMs' performance on reasoning tasks is due to statistical artifacts in datasets and does not exhibit true reasoning and generalization capabilities. With the help of synthetic corpora, we can cleanly investigate PLMs' reasoning capabilities.

Hupkes et al. (2020) study the ability of neural models to capture compositionality. They do not investigate our six rules, nor do they consider the effects of fact frequency and schema conformity. Our work confirms their finding that transformers have the ability to capture both rules and exceptions.

A large body of research in psychology and cognitive science has investigated how some of our rules are processed in humans, e.g., Sloman (1996) for implication. There is also a lively debate in cognitive science as to how important rule-based reasoning is for human cognition (Politzer, 2007).

Yanaka et al. (2020); Goodwin et al. (2020) are concurrent studies of systematicity in PLMs. The first shows that monotonicity inference is feasible for syntactic structures close to the ones observed during training. The latter shows that PLMs can exhibit high over-all performance on natural language inference despite being non-systematic.

Roberts et al. (2020) show that the amount of knowledge captured by PLMs increases with model size. Our memorization experiments investigate the factors that determine successful acquisition of knowledge.

Guu et al. (2020) modify the PLM objective to incentivize knowledge acquisition. They do not consider symbolic rule learning nor do they analyze what factors influence successful memorization.

Based on perceptrons and convolutional neural networks, Arpit et al. (2017); Zhang et al. (2017) study the relation of generalizing from real structured data vs. memorizing random noise in the image domain, similar to our study of schema-conformant facts and outliers. They do not study transformer based models trained on natural language.

7 Conclusion

We studied BERT's ability to capture knowledge from its training corpus by investigating its reasoning and memorization capabilities. We identified factors influencing what makes successful memorization possible and what is learnable beyond knowledge explicitly seen during training. We saw that, to some extent, BERT is able to infer facts not explicitly seen during training via symbolic rules.

Overall, effective knowledge acquisition must combine both parts of this paper: memorization and symbolic reasoning. A PLM is not able to store an unlimited amount of knowledge. Through acquiring reasoning capabilities, knowledge gaps can be filled based on memorized facts. A schema-conformant fact ("pigeons can fly") need not be memorized if there are a few facts that indicate that birds fly and then the ability of flight can be filled in for the other birds. The schema conformity experiments suggest that this is happening. It is easier to capture knowledge that conforms with a schema instead of memorizing facts one by one.

There are several directions for future work. First, we made many simplifying assumptions that should be relaxed in future work. Second, how can we improve PLMs' ability to learn symbolic rules? We see two avenues here, either additional inductive biases could be imposed on PLMs' architectures or training corpora could be modified to promote learning of symbolic rules.

Acknowledgements

We thank Peter Clark for helpful discussions and our reviewers for constructive feedback.

This work was funded by the German Federal Ministry of Education and Research (BMBF, Grant No. 01IS18036A) and by Deutsche Forschungsgemeinschaft (DFG, Grant ReMLAV: Relational Machine Learning for Argument Validation). The authors of this work take full responsibility for its content.

References

Devansh Arpit, Stanisław Jastrzundefinedbski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. 2017. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 233242. JMLR.org.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

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*.

Allyson Ettinger. 2020. What bert is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

Emily Goodwin, Koustuv Sinha, and Timothy J. O'Donnell. 2020. Probing linguistic systematicity. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1958–1969, Online. Association for Computational Linguistics.

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 Schütze. 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.

Beth Levin. 1993. *English Verb Classes and Alternations*. The University of Chicago Press, Chicago.

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.

Mojtaba Nayyeri, Chengjin Xu, Jens Lehmann, and Hamed Shariat Yazdi. 2019. Logicenn: A neural based knowledge graphs embedding model with logical rules. *ArXiv*, abs/1908.07141.

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.

Guy Politzer. 2007. Reasoning with conditionals. *Topoi*, 26(1):79–95.

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. CLUTRR: 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.

Steven A. Sloman. 1996. The empirical case for two systems of reasoning. *Psychological Bulletin*, 119(1):3–22.

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*.

Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. 2019. oLMpics - on what language model pre-training captures. *ArXiv*, abs/1912.13283.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Jason Weston, Antoine Bordes, Sumit Chopra, and Tomas Mikolov. 2016. Towards ai-complete question answering: A set of prerequisite toy tasks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, and Kentaro Inui. 2020. Do neural models learn systematicity of monotonicity inference in natural language? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6105–6117, Online. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2017. Understanding deep learning requires rethinking generalization. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24 - 26, 2017, Conference Track Proceedings*.

Zhanqiu Zhang, Jianyu Cai, Yongdong Zhang, and Jie Wang. 2020. Learning hierarchy-aware knowledge graph embeddings for link prediction. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*.

A Hyperparameters

A.1 Model hyperparameters

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

Our experiments for symbolic rules started with the BERT-base model with 12 layers, 12 attention heads, hidden size of 768 and intermediate size of 3072. For rules with a low test precision (NEG and COMP) we then conducted a restricted grid search (restricted due to limited compute infrastructure): We tried all possible numbers of layers from 1 to 12 and then only considered the best result. For NEG the best performance came from 4 layers, whereas COMP did not show improvements for any number of layers. For NEG with 3 layers (which had a very similar performance to 4 layers) we exemplarily tested whether changing the attention heads, hidden size or intermediate size improves precision. For this we trained with the following 4 settings:

- attention heads = 6, hidden size = 768, intermediate size = 3072
- attention heads = 12, hidden size = 384, intermediate size = 1536
- attention heads = 12, hidden size = 192, intermediate size = 768
- attention heads = 12, hidden size = 96, intermediate size = 192

However this did not further improve precision.

A.2 Data hyperparameters

In previous iterations of our experiments, we had used different settings for generating our data. For instance, we had varied the number of rules in our corpora: 50 or 100 instead of the presented 20 rules. Even the sampling process itself can be tweaked to allow for less overlaps between rules and between instances of one rule. However, we observed the same trends and similar numbers across these different settings.

B Symbolic rules

In the following sections, we present illustrating corpora for **INV**, **IMP** and **COMP enhanced**. Each line is one datapoint. We also include the control group at the end of each corpus that does not follow any rule. In the case of composition enhanced, ”{...}” indicates the sampled group which is not part of the actual dataset.

We illustrate our training corpora using real world entities and relations. Note that the actual corpora used for training are composed of an entirely synthetic vocabulary. For simplicity we show grouped composition with enhancement with groups of 4, instead of 10 as it is in the real data.

B.1 INV

Paris CapitalOf France
France HasCapital Paris
...
Egypt HasCapital Cairo (counterpart in test-set)
...
Apple Developed iOS
iOS DevelopedBy Apple
...
Germany RandomRelation China
Cairo RandomRelation Norway

B.2 IMP

{(Flu), (Cough, RunningNose, Headache, Fever)}
Kevin HasDisease Flu
Kevin HasSymptom Cough
Kevin HasSymptom RunningNose
Kevin HasSymptom Headache
Kevin HasSymptom Fever
...
Mariam HasDisease Flu
...
Peter RandomRelation Pain
Sarah RandomRelation Tooth

B.3 Comp enhanced

{e8, e2, e4, e5}
e8 ConnectedTo e2
e8 ConnectedTo e4
e8 ConnectedTo e5
e2 ConnectedTo e8
...
{e15, e13, e12, e19}
e15 ConnectedTo e13
e15 ConnectedTo e12
...
{e25, e24, e29, e20}
e25 ConnectedTo e24
e25 ConnectedTo e29
...
e8 r1 e15
e8 r1 e13
e8 r1 e12
e8 r1 e19
e2 r1 e15
e2 r1 e13
...
e5 r1 e19
...
e15 r2 e25
e15 r2 e24
e15 r2 e29
e15 r2 e20
...
e19 r2 e20
...
e8 r3 e25
e8 r3 e24
e8 r3 e29
e8 r3 e20
...
...
e133 r61 e23
e56 r61 e29
...