Measuring and Improving Consistency in Pretrained Language Models

Yanai Elazar[1,2] **Nora Kassner**[3] **Shauli Ravfogel**[1,2] **Abhilasha Ravichander**[4]
Eduard Hovy[4] **Hinrich Schütze**[3] **Yoav Goldberg**[1,2]
[1]Computer Science Department, Bar Ilan University
[2]Allen Institute for Artificial Intelligence
[3]Center for Information and Language Processing (CIS), LMU Munich
[4]Language Technologies Institute, Carnegie Mellon University
{yanaiela,shauli.ravfogel,yoav.goldberg}@gmail.com
kassner@cis.lmu.de {aravicha,hovy}@cs.cmu.edu

Abstract

Consistency of a model — that is, the invariance of its behavior under meaning-preserving alternations in its input — is a highly desirable property in natural language processing. In this paper we study the question: Are Pretrained Language Models (PLMs) consistent with respect to factual knowledge? To this end, we create ParaRel🤘, a high-quality resource of cloze-style query English paraphrases. It contains a total of 328 paraphrases for thirty-eight relations. Using ParaRel🤘, we show that the consistency of all PLMs we experiment with is poor – though with high variance between relations. Our analysis of the representational spaces of PLMs suggests that they have a poor structure and are currently not suitable for representing knowledge in a robust way. Finally, we propose a method for improving model consistency and experimentally demonstrate its effectiveness.[1]

1 Introduction

Pretrained Language Models (PLMs) are large neural networks that are used in a wide variety of NLP tasks. They operate under a pretrain-finetune paradigm: models are first *pretrained* over a large text corpus and then *finetuned* on a downstream task. PLMs are thought of as good language encoders, supplying basic language understanding capabilities that can be used with ease for many downstream tasks.

A desirable property of a good language understanding model is *consistency*: the ability of making consistent decisions in semantically equivalent contexts, reflecting a systematic ability to generalize in the face of language variability.

Examples of consistency include: predicting the same answer in question answering and reading comprehension tasks regardless of paraphrase

[1]The code and resource is available at: https://github.com/yanaiela/pararel

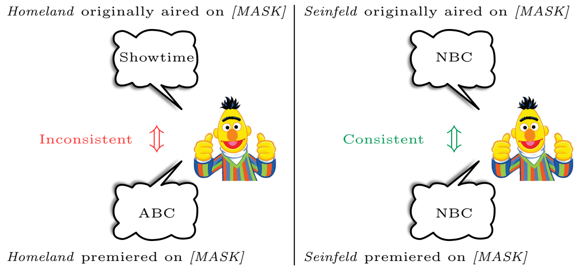

Figure 1: Overview of our approach. We expect that a consistent model would predict the same answer for every two paraphrases. In this example, the model is inconsistent on *Homeland* and consistent on *Seinfeld*.

(Asai and Hajishirzi, 2020); making consistent assignments in coreference resolution (Denis and Baldridge, 2009; Chang et al., 2011); or making summaries factually consistent with the original document (Kryscinski et al., 2020). While consistency is important in many tasks, nothing in the training process explicitly targets it. One could hope that the unsupervised training signal from large corpora made available to PLMs such as BERT or RoBERTa (Devlin et al., 2019; Liu et al., 2019) is sufficient to induce consistency and transfer it to downstream tasks. In this paper, we show that this is not the case.

The recent rise of PLMs has sparked a discussion about whether these models can be used as Knowledge Bases (KBs) (Petroni et al., 2019, 2020; Jiang et al., 2020; Roberts et al., 2020). Consistency is a key property of KBs and is particularly important for automatically constructed KBs. One of the biggest appeals of using a PLM as a KB is that we can query it in natural language – instead of relying on a specific KB schema. The expectation is that PLMs abstract away from language and map queries in natural language into meaningful representations such that queries with identical intent but different language form yield the same answer. For example, the query "*Homeland* premiered on *[MASK]*" should produce the same answer as "*Homeland* originally aired on *[MASK]*". Studying inconsistencies of PLM-KBs can also teach us about the organization of knowledge in the model or lack thereof. Finally, failure to behave in a consistent manner may point to other representational issues such as the similarity between antonyms and synonyms (Nguyen et al., 2016).

In this work, we study the consistency of factual knowledge in PLMs: Is the factual information we extract from PLMs invariant to paraphrasing? We use zero-shot evaluation since we want to inspect models directly, without adding biases through finetuning. This allows us to assess how much consistency was acquired during pretraining and to compare the consistency of different models.

We introduce ParaRel🤘, a new benchmark that enables us to measure consistency in PLMs by using factual knowledge that was found to be partially encoded in them (§3). ParaRel🤘 is a manually curated resource that provides patterns – short textual prompts – that are paraphrases of one another, with 328 paraphrases describing thirty-eight binary relations such as *X born-in Y*, *X works-for Y* (§4). We then test multiple PLMs for knowledge consistency, i.e., whether a model predicts the same answer for all patterns of a relation. Figure 1 shows an overview of our approach. Using ParaRel🤘, we probe for consistency in four PLM types: BERT, BERT-whole-word-masking, RoBERTa and ALBERT (§5). Our experiments with ParaRel🤘 show that current models have poor consistency, although with high variance between different relations (§6).

Finally, we propose a method that improves model consistency by introducing a novel consistency loss (§8). We demonstrate that models trained with this loss achieve better consistency performance on new relations. However, more work is required to achieve fully consistent models.

2 Background

There has been significant interest in analyzing how well PLMs (Rogers et al., 2020) perform on linguistic tasks (Goldberg, 2019; Hewitt and Manning, 2019; Tenney et al., 2019; Elazar et al., 2020), commonsense (Forbes et al., 2019; Da and Kasai, 2019; Zhang et al., 2020) and reasoning (Talmor et al., 2020; Kassner et al., 2020), usually assessed by measures of accuracy. However, accuracy is just one measure of PLM performance (Linzen, 2020). It is equally important that PLMs do not make contradictory predictions (cf. Figure 1), a type of error that humans rarely make. There has been little research attention devoted to this question, i.e., to analyzing if models behave *consistently*. One example concerns negation: Ettinger (2020) and Kassner and Schütze (2020) show that models tend to generate facts and their negation, a type of inconsistent behavior. Ravichander et al. (2020) propose paired probes for evaluating consistency. Our work is broader in scope, examining the consistency of PLM behavior across a range of factual knowledge types and investigating how models can be made to behave more consistently.

Consistency has also been highlighted as a desirable property in automatically constructed KBs and downstream NLP tasks. We now briefly review work along these lines.

Consistency in knowledge bases (KBs) has been studied in theoretical frameworks in the context of the satisfiability problem and KB construction, and efficient algorithms for detecting inconsistencies in KBs have been proposed (Hansen and Jaumard, 2000; Andersen and Pretolani, 2001). Other work aims to quantify the degree to which KBs are inconsistent and detects inconsistent statements (Thimm, 2009; Muiño, 2011; Thimm, 2013).

Consistency in question answering was studied by Ribeiro et al. (2019) in two tasks: visual question answering (Antol et al., 2015) and reading comprehension (Rajpurkar et al., 2016). They automatically generate questions to test the consistency of QA models. Their findings suggest that most models are not consistent in their predictions. In addition, they use data augmentation to create more robust models. Alberti et al. (2019) generate new questions conditioned on context and answer from a labeled dataset and by filtering answers that do not provide a consistent result with the original answer. They show that pretraining on these synthetic data improves QA results. Asai and Hajishirzi (2020) use data augmentation that complements questions with symmetricity and transitivity, as well as a regularizing loss that penalizes inconsistent predictions.

Work on **consistency in other domains** includes (Du et al., 2019) where prediction consistency in procedural text is improved. Ribeiro et al. (2020) use consistency for more robust evaluation. Li et al. (2019) measure and mitigate inconsistency in natural language inference.

3 Probing PLMs for Consistency

In this section, we formally define consistency and describe our framework for probing the consistency of PLMs.

3.1 Consistency

We define a model as *consistent* if, given two *cloze-phrases* such as "*Seinfeld* originally aired on *[MASK]*" and "*Seinfeld* premiered on *[MASK]*" that are *quasi-paraphrases* it makes non-contradictory predictions on N-1 relations, over a large set of entities.[2] For instance, a model that predicts *NBC* and *ABC* on the two aforementioned patterns, is not consistent, since these two facts are contradicting. Note that consistency does not require the answers to be factually correct. While correctness is also an important property for KBs, we view it as a separate objective and measure it independently. In the rest of the paper, we use the terms *paraphrase* and *quasi-paraphrase* interchangeably.

Note that many-to-many (N-M) relations (e.g. *shares-border-with*) can be consistent even with different answers (given they are correct). For instance, two patterns that express the *shares-border-with* relation and predict *Albania* and *Bulgaria* for the *Greece* subject are both correct. We do not consider such relations for measuring consistency. However, another requirement from a KB is *determinism*. That is, returning the results in the same order (when more than a single result exists). In this work, we focus on consistency, but also measure determinism of the models we inspect.

3.2 The Framework

An illustration of the framework is presented in Figure 2. Let D_i be a set of subject-object KB tuples (e.g. <*Homeland*, *Showtime*>) from some relation r_i (e.g. *originally-aired-on*), accompanied with a set of *quasi-paraphrases* cloze-patterns P_i (e.g. X originally aired on Y). Our goal is to test whether the model consistently predicts the same object (e.g. *Showtime*) for a particular subject (e.g. *Homeland*). To this end, we substitute X with a subject from D_i and Y with *[MASK]* in all of the patterns P_i of that relation (e.g. *Homeland* originally aired on *[MASK]* and *Homeland* premiered on *[MASK]*). A consistent model must predict the same entity.

[2] A *quasi-paraphrase* – introduced by Bhagat and Hovy (2013) – is a more fuzzy version of a paraphrase. The concept does not rely on the strict, logical definition of paraphrase and allows to operationalize concrete uses of paraphrases. This definition is in the spirit of the RTE definition (Dagan et al., 2005), which similarly supports a more flexible use of the notion of entailment.

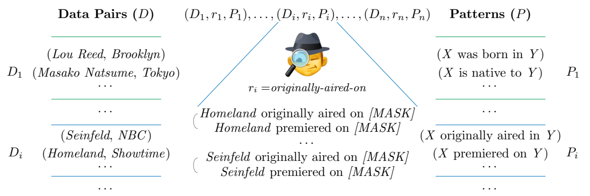

Figure 2: Overview of the framework which we use to assess models' consistency. D_i corresponds to a set of KB triplets of some relation r_i, which are coupled with a set of *quasi-paraphrases* cloze-patterns P_i that describe that relation. We then populate the subjects from D_i as well as a mask token into all patterns P_i and expect a model to predict the same object across all pattern pairs.

Restricted Candidate Sets Since PLMs were not trained for serving as KBs, as such, they often predict words that are not KB entities; e.g., a PLM may predict, for the pattern "*Showtime* originally aired on *[MASK]*", the noun 'tv' – which is also a likely substitution for the language modeling objective, but not a valid KB fact completion. Therefore, following (Xiong et al., 2020), we restrict the PLMs' output vocabulary to the set of possible gold objects for each relation. In practice, we compute the full probability distribution and then only consider the subset of possible gold objects.

Note that this setup makes the task easier for the PLM, especially in the context of KBs. However, poor consistency in this setup strongly implies that consistency would be even lower without restricting candidates.

4 The ParaRel Resource

We now describe ParaRel, a concrete resource according to the framework described in Section 3.2. ParaRel is curated by experts, with high level of agreement. It contains patterns for thirty-eight relations from the T-REx dataset (Elsahar et al., 2018), with an average of 8.63 patterns per relation. Some statistics of the resource are presented in Table 1.

Construction Method ParaRel was constructed in four steps. (1) We began with the patterns provided by LAMA (Petroni et al., 2019)

# Relations	38
Min # patterns	2
Max # patterns	20
Avg # patterns	8.63
Avg syntax	4.74
Avg lexical	6.03

Table 1: Statistics of ParaRel🤘. Reporting the number of relations we cover the minimum, maximum, and average number of patterns in the resource. The last two rows correspond to the average number of unique syntactic and lexical variations from a relations' patterns.

(one pattern per relation, referred to as "base pattern"). (2) We augmented each base pattern with other patterns that are paraphrases of the base pattern from LPAQA (Jiang et al., 2020). However, since LPAQA was extracted automatically, some LPAQA patterns are not correct paraphrases. We therefore only include the subset of correct paraphrases in ParaRel🤘. (3) Using SPIKE (Shlain et al., 2020),[3] a search engine over Wikipedia sentencecs that supports syntax-based queries, we searched for additional patterns that appeared in Wikipedia and added them to ParaRel🤘. Specifically, we searched for Wikipedia sentences containing a subject-object tuple from T-REx and then manually extracted patterns from the sentences. (4) Lastly, we added additional paraphrases of the base pattern using the annotators' linguistic expertise. Two additional experts went over all the patterns and corrected them, while engaging in a discussion until reaching agreement, discarding patterns they could not agree on.

Human Agreement To assess the quality of ParaRel🤘, we run a human annotation study. For each relation, we sample up to five paraphrases, comparing each of the new patterns to the *base-pattern* that came with LAMA, which reflects the relation. That is, given that relation r_i contain the following patterns: p_1, p_2, p_3, p_4, and p_1 being the *base-pattern*, we compare the following pairs $(p_1, p_2), (p_1, p_3), (p_1, p_4)$.

We populate the patterns with random subjects and objects from T-REx (Elsahar et al., 2018) and ask annotators if these sentences are paraphrases. We also sample patterns from different relations to provide examples that are not paraphrases of each other, for control. Overall, each task contains five patterns that are thought to be paraphrases, and two that are not.[4] Overall, we collect annotations for 156 paraphrase candidates and 61 false paraphrases for control.

We asked five NLP graduate students to annotate the pairs and collected one answer per pair.[5] The agreement score for the paraphrases and the control are: 95.5% and 98.3%, which are high and reassures ParaRel🤘's quality. We further went through the disagreements and fixed any additional problem to further improve the resource quality.

5 Experimental Setup

5.1 Models

We experiment with four variants from three PLM families: BERT, BERT whole-word-masking (Devlin et al., 2019), RoBERTa (Liu et al., 2019) and ALBERT (Lan et al., 2019). For BERT, RoBERTa and ALBERT, we use a base and a large version.[6]

In addition, we also report a majority baseline that always predicts the most common object for a specific relation. By definition, this baseline is perfectly consistent.

5.2 Data

We use knowledge graph data from T-REx (Elsahar et al., 2018).[7] To make the results comparable across models, we remove objects that are not represented as a single token in all models' vocabularies, to a total of 26,813 tuples. We further split the data into N-M relations of which we report the determinism results apart (seven relations), as well as the N-1 which we use for measuring consistency (thirty-one relations).

5.3 Evaluation

Our consistency measure for a relation r_i (*Consistency*) is the percentage of consistent predictions of all the pattern pairs of that relation $p^i_k, p^i_l \in P_i$, for all the KB tuples $d^i_j \in D_i$. Thus, for every KB

[3] https://spike.apps.allenai.org/

[4] The control patterns contain the same subjects and objects, so that only the pattern (not its argument) can be used to solve the task.

[5] Due to random errors in the annotations, for every question that does not match our original label (either for paraphrases or controls), we ask the annotators to relabel them (without specifying the reason), to allow them to fix random mistakes.

[6] For ALBERT we use the smallest and largest versions.

[7] We discard three relations from the T-REx that were not well defined

Model	Succ-Patt	Succ-Objs	Unk-Const	Know-Const
majority	97.3+-7.3	23.2+-21.0	100.0+-0.0	100.0+-0.0
BERT-base	**100.0**+-0.0	63.0+-19.9	46.5+-21.7	63.8+-24.5
BERT-large	**100.0**+-0.0	**65.7**+-22.1	48.1+-20.2	65.2+-23.8
BERT-large-wwm	**100.0**+-0.0	64.9+-20.3	**49.5**+-20.1	**65.3**+-25.1
RoBERTa-base	**100.0**+-0.0	56.2+-22.7	43.9+-15.8	56.3+-19.0
RoBERTa-large	**100.0**+-0.0	60.1+-22.3	46.8+-18.0	60.5+-21.1
ALBERT-base	**100.0**+-0.0	45.8+-23.7	41.4+-17.3	56.3+-22.0
ALBERT-xxlarge	**100.0**+-0.0	58.8+-23.8	40.5+-16.4	57.5+-23.8

Table 2: Extractability Measures, in the different models we inspect. Best model's performance for each metric is highlighted in bold.

Model	Accuracy	Consistency	Consistent-Acc
majority	23.1+-21.0	100.0+-0.0	23.1+-21.0
BERT-base	45.8+-25.6	58.5+-24.2	27.0+-23.8
BERT-large	48.1+-26.1	**61.1**+-23.0	**29.5**+-26.6
BERT-large-wwm	**48.7**+-25.0	60.9+-24.2	29.3+-26.9
RoBERTa-base	39.0+-22.8	52.1+-17.8	16.4+-16.4
RoBERTa-large	43.2+-24.7	56.3+-20.4	22.5+-21.1
ALBERT-base	29.8+-22.8	49.8+-20.1	16.7+-20.3
ALBERT-xxlarge	41.7+-24.9	52.1+-22.4	23.8+-24.8

Table 3: Knowledge and Consistency Results. Best model's performance for each metric is highlighted in bold.

tuple from the relation r_i that contains n patterns, we compare $n(n-1)/2$ pairs.

We also report *Accuracy* metric, that is, the acc@1 of a model in predicting the correct object, using the original patterns from Petroni et al. (2019). In contrast to Petroni et al. (2019), we define it as accuracy of the top-ranked object from the candidate set of each relation. Finally, we report *Consistent-Acc*, a new metric that evaluates individual objects as correct, only if *all* patterns of the corresponding relation predict the object correctly. *Consistent-Acc* is a much stricter metric and combines the requirements of both consistency (*Consistency*) and factual correctness (*Accuracy*).

In all our metrics, we report the average results over all relations, which can be viewed as a macro average.

6 Experiments and Results

6.1 Knowledge Extraction through Different Patterns

We begin by assessing the variability of our patterns as well as the degree to which they extract the correct entities. These results are summarized in Table 2.

First, we report *Succ-Patt* that measures the percentage of patterns that successfully predicted the right object at least once. A high score suggests that the patterns are decent and models use them to extract the correct answer. All PLMs achieve a perfect score. Next, we report *Succ-Objs*, the percentage of entities that were predicted correctly by at least one of the patterns. *Succ-Objs* quantifies the degree to which the knowledge is stored by the models. We observe that some tuples are not predicted correctly by any of our patterns: the scores vary between 45.8% for ALBERT-base and 65.7% for BERT-large. With an average number of 8.63 patterns per relation, there are multiple ways to extract the knowledge, we thus interpret these results as evidence that the models do not store a large part of the knowledge from T-REx.

Finally, we measure *Unk-Const*, a consistency measurement for the subset of tuples for which no pattern predicted the correct answer correctly; and *Know-Const*, consistency for the subset where at least one of the patterns for a specific relation predicted the correct answer. This split is based on the *Succ-Objs*, which splits the data into two parts. Overall, the results indicate that when the factual knowledge is successfully extracted, the model is also more consistent. For instance, for BERT-large, *Know-Const* is 65.2% and *Unk-Const* is 48.1%.

6.2 Consistency & Knowledge

In this section, we report the overall knowledge measure that was used in Petroni et al. (2019) (*Accuracy*), the consistency metric (*Consistency*), as well as a new measure that combines both knowledge and consistency measures (*Consistent-Acc*). The results are summarized in Table 3.

We begin with the *Accuracy* results. The results range between 29.8% (ALBERT-base) and 48.7% (BERT-large whole-word-masking). Notice that our numbers differ from Petroni et al. (2019) as we use a candidate set (§3) and only consider KB triples whose object is a single token in all the PLMs we consider (§5.2).

Next, we report the main *Consistency* metric (§5.3). The BERT models achieve the highest scores. There is a consistent improvement from the base to large versions of each model. In contrast to previous work that observed quantitative and qualitative improvements of RoBERTa-based models over BERT (Liu et al., 2019; Talmor et al., 2020), in terms of consistency, BERT is more consistent than RoBERTa and ALBERT. Still, the overall results are low (61.1% for the best model), even more remarkably so because the restricted candidate set makes the task easier. We note that the results are highly variant between models (performance on *original-language* varies between 52% and 90%), and relations (BERT-large performs on 92% on *capital-of*, whereas 44% on *owned-by*) and relations.

Finally, we report *Consistent-Acc*: the results are much lower than the *Accuracy* metric, as expected, but follow similar trends: RoBERTa-base perform worse (16.4%) and BERT-large best (29.5%).

Interestingly, we find strong correlations between the Accuracy and Consistency metrics, ranging between 67.3% for RoBERTa-base to 82.1% for BERT-large (all with small p-values $\ll 0.01$).

A striking result of the model comparison is the clear superiority of BERT, both in knowledge accuracy and knowledge consistency. This result, which we hypothesise to be due to the training data, may have a broader impact on models to come: Training bigger models with more data is not always beneficial. Since Wikipedia is likely the largest unified source of factual knowledge that exists in unstructured data, giving prominence to Wikipedia in pretraining a model, makes it more likely that the model will incorporate the factual knowledge well. This may indicate that the consistency of models such as GPT-3 (Brown et al., 2020) or other future models – i.e., models that have a much larger size and are trained on very large corpora of which Wikipedia is just a small part – may suffer.

Determinism We also measure the determinism results on the N-M relations. That is, the same measure as *Consistency*, but since the predictions may be factually correct, these do not necessarily convey consistency violation, but indicate on a determinism issue. For brevity, we do not present the entire results, but the trend is similar to the consistency result (although not comparable, as it inspect different relations): 52.9% and 44.6% for BERT-large and RoBERTa-base respectively.

Effect of Pretraining Corpus Size Next, we study the question of whether the number of tokens used during pretraining contributes to consistency. We use the pretrained RoBERTa models from Warstadt et al. (2020) and repeat the experiments on four additional models. These are RoBERTa-based models, trained on a sample of Wikipedia and the book corpus, with varying training size and parameters. We use one of the three published models for each configuration and report the average accuracy over the relations for each model in Table 4. Overall, the *Accuracy* and *Consistent-Acc* improve with more training data. However, there is an interesting outlier to this trend. First, the model that was trained on one million tokens is more consistent than the models trained on ten and one-hundred million tokens. A potentially crucial difference is that this model has many fewer parameters than the rest (to avoid overfitting). It is nonetheless interesting that a model that is trained on significantly less data can achieve better consistency. On the other hand, the knowledge scores are lower, arguably due to the model being exposed to less factual knowledge during pretraining.

Model	Acc	Consistency	Consistent-Acc
majority	23.1+-21.0	100.0+-0.0	23.1+-21.0
RoBERTa-med-small-1M	11.2+-9.4	37.1+-11.0	2.8+-4.0
RoBERTa-base-10M	17.3+-15.8	29.8+-12.7	3.2+-5.1
RoBERTa-base-100M	22.1+-17.1	31.5+-13.0	3.7+-5.3
RoBERTa-base-1B	**38.0**+-23.4	**50.6**+-19.8	**18.0**+-16.0

Table 4: Average knowledge and consistency results for the different RoBERTas, trained on increasing amounts of data, over all relations, along with the Std. Best model's performance for each metric is highlighted in bold.

6.3 Do PLMs Generalize Over Syntactic Configurations?

Many papers have found neural models (especially PLMs) to naturally encode syntax (Linzen et al., 2016; Belinkov et al., 2017; Marvin and Linzen, 2018; Belinkov and Glass, 2019; Goldberg, 2019; Hewitt and Manning, 2019). Does this mean that PLMs have successfully abstracted knowledge and can comprehend and produce it regardless of syntactic variation? We consider two scenarios. (1) two patterns differ only in syntax. (2) Both syntax and lexical choice are the same. As a proxy, we define syntactical equivalence if the dependency paths between subject and object are identical. We parse all patterns from PARAREL🤘 using a dependency parser (Honnibal et al., 2020)[8] and retain the path between the entities. Success on (1) indicates that the model's knowledge processing is robust to syntactic variation. Success on (2) indicates that the model's knowledge processing is robust to variation in word order and tense.

[8] https://spacy.io/

Model	Diff-Syntax	No-Change
majority	100.0+-0.0	100.0+-0.0
BERT-base	67.9+-30.3	76.3+-22.6
BERT-large	67.5+-30.2	78.7+-14.7
BERT-large-wwm	63.0+-31.7	**81.1**+-9.7
RoBERTa-base	66.9+-10.1	80.7+-5.2
RoBERTa-large	**69.7**+-19.2	80.3+-6.8
ALBERT-base	62.3+-22.8	72.6+-11.5
ALBERT-xxlarge	51.7+-26.0	67.3+-17.1

Table 5: Average consistency and standard deviation over all relations when only syntax differs (*Diff-Syntax*) and when syntax and lexical choice are identical (*No-Change*). Best model's performance for each metric is highlighted in bold.

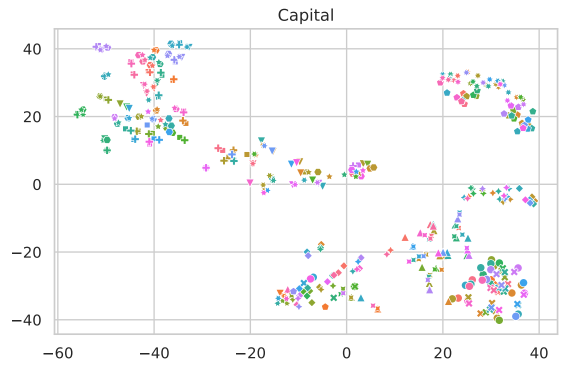

Figure 3: t-SNE of the encoded patterns from the *capital* relation. The colors represent the different subjects, while the shapes represent patterns. We expect that a knowledge-focused representation clusters based on identical-subjects (color) rather than identical-patterns (shape).

Table 5 reports results. While the results are not comparable to the main results on the entire dataset as the pattern subsets are different, they are higher than the general results: 67.5% for BERT-large when only syntax differs, and 78.7% when syntax is identical. This demonstrates that while PLMs have impressive syntactic abilities, they struggle to extract factual knowledge in the face of tense, word-order and syntactic variation.

McCoy et al. (2019) show that supervised models trained on MNLI (Williams et al., 2018), an NLI dataset (Bowman et al., 2015) use superficial syntactic heuristics rather than more generalizable properties of the data. Our results show that pretrained PLMs have problems along the same lines: they are not robust to surface variation.

7 Analysis

7.1 Qualitative Analysis

In order to better understand the factors affecting consistent predictions, we inspect the predictions of BERT-large on different patterns, which are summarized in Table 6. We highlight several cases: The predictions in the first row are inconsistent, and correct for the first pattern, but not for the other two. The predictions in the second row also show a single pattern that predicted the right object, however, the two other patterns, which are lexically similar, predicted the same, wrong answer - *Renault*. Next, the patterns of the penultimate row predicted two factually correct answers out of three (Greece, Kosovo), but simply do not correspond to the gold object in T-REx (Albania), since this is an M-N relation. Note that this relation is not part of the consistency evaluation, but of the determinism evaluation. The last row of the upper part detail three different incorrect predictions. Finally, the predictions in the bottom part demonstrate consistent predictions: The last row is consistent and factual, the last but one row is consistent, but factually incorrect (which is surprising as the correct answer is substring of the subject).

7.2 Representation Analysis

To provide insights on the models' representations, we inspect these after encoding the patterns.

Motivated by previous work that found that words with the same syntactic structure cluster together (Chi et al., 2020; Ravfogel et al., 2020) we perform a similar experiment to test if this behavior replicates with respect to knowledge: We encode the patterns, after filling the placeholders with subjects and masked tokens and inspect the last layer representations in the masked token position. When plotting the results using t-SNE (Maaten and Hinton, 2008) we mainly observe clustering based on the patterns, which suggests that encoding of knowledge of the entity is not the main component of the representations. Figure 3 demonstrates this for BERT-large encodings of the *capital* relation, which is highly consistent.[9] To provide a more quantitative assessment to the above phenomenon, we also cluster these representations, and set the number of centroids based on:[10] (1) the number of patterns in each relation, which aims to capture pattern-based clusters and (2) the number of entities in each relation, which aims to capture

[9] While some patterns are clustered based on the subjects (upper-left part), most of them are clustered based on patterns.
[10] Using the KMeans algorithm.

Subject	Object	Pattern #1	Pattern #2	Pattern #3	Pred #1	Pred #2	Pred #3
Adriaan Pauw	Amsterdam	[X] was born in [Y].	[X] is native to [Y].	[X] is a [Y]-born person.	Amsterdam	Madagascar	Luxembourg
Nissan Livina Geniss	Nissan	[X] is produced by [Y].	[X] is created by [Y].	[X], created by [Y].	Nissan	Renault	Renault
Arab League	Asia	[X] belongs to the continent of [Y].	[X] is located in [Y].	[X] is a part of the continent of [Y].	Asia	Europe	Africa
Albania	Serbia	[X] shares border with [Y].	[Y] borders with [X].	[Y] shares the border with [X]	Greece	Turkey	Kosovo
iCloud	Apple	[X] is developed by [Y].	[X], created by [Y].	[X] was created by [Y]	Microsoft	Google	Sony
Yahoo! Messenger	Yahoo	[X], a product created by [Y]	[X], a product developed by [Y]	[Y], that developed [X]	Microsoft	Microsoft	Microsoft
Wales	Cardiff	The capital of [X] is [Y] .	[X]'s capital, [Y].	[X]'s capital city, [Y].	Cardiff	Cardiff	Cardiff

Table 6: Predictions of BERT-large-cased. Presenting the subjects and objects taken from T-REx (Elsahar et al., 2018), as well as three different patterns from our resource and their predictions. The predictions are colored in blue if the model predicted correctly (out of the candidate list), and in red otherwise. If there is more than a single erroneous prediction, it is colored by a different red.

entity-based clusters. This would allow for a perfect clustering, in the case of perfect alignment between the representation and the inspected property. We measure the purity of these clusters using V-measure, and observe that the clusters are mostly grouped by the patterns, rather than the objects. Finally, we compute the spearman correlation between the consistency scores and the v-measure of the representations. However, the correlation between these variables is close to zero,[11] therefore not explaining the models' behavior. This finding is interesting since it means that (1) these representations are not knowledge-focused, i.e. Their main component does not relate to knowledge, and (2) the entire representation does not explain the behavior of the model. This finding is consistent with previous work that observed similar trends for linguistic tasks (Elazar et al., 2020). We hypothesize that this disparity between the representation and the behavior of the model may be explained by a situation where distance between representations largely do not reflect the distance between predictions, but rather reflect other, behaviorally-irrelevant factors of a sentence.

8 Improving Consistency in PLMs

In the previous sections, we showed pretrained models are generally not consistent in their predictions, and previous works have noticed the lack of this property in a variety of downstream tasks. An ideal model would exhibit the consistency property after pretraining, and would then be able to transfer it to different downstream tasks. We therefore ask: Can we enhance current PLMs and make them more consistent?

8.1 Consistency Improved PLMs

We propose to improve the consistency of PLMs by continuing the pretaining step with a novel consistency loss. We make use of the T-REx tuples and a subset of paraphrases from ParaRel🤘.

Given a relation r_i, we assume a set paraphrased patterns P_i describing that relation. We use a PLM to encode all patterns in P_i, after populating a subject and a mask token that corresponds to the relation r_i. Finally we expect the model to make the same prediction for the masked token.

Consistency Loss Function As we evaluate the model using acc@1, the straight-forward consistency loss would require these predictions to be identical:

$$\min_{\theta} \text{sim}(\arg\max_{i} f_\theta(P_n)[i], \arg\max_{j} f_\theta(P_m)[j])$$

where $f_\theta(P_n)$ is the output of an encoding function (e.g., BERT) parameterized by θ (a vector) over input P_n, and $f_\theta(P_n)[i]$ is the score of the ith vocabulary item.

However, this objective contains a comparison between the output of two argmax operations, making it discrete and discontinuous, and hard to optimize in a gradient-based framework. We instead relax the objective, and require that the predicted *distributions* $Q_n = softmax(f_\theta(P_n))$, rather than the top-1 prediction, be identical to each other. We use two-sided KL Divergence to measure similarity between distributions: $D_{KL}(Q_n^{r_i}||Q_m^{r_i}) + D_{KL}(Q_m^{r_i}||Q_n^{r_i})$. where $Q_n^{r_i}$ is the predicted distribution for pattern n of relation r_i.

As most of the vocabulary is not relevant for the predictions, we filter it down to the candidate set of each relation (§3.2). Moreover, since we are motivated by maintaining the original capabilities of the model, focusing on the candidate set helps to achieve this goal since most of the vocabulary is not affected by our new loss.

To encourage a more general solution, we make use of all the paraphrases together, and enforce all predictions to be as close as possible. Thus, the consistency loss for all pattern pairs for a particular

[11]Except for BERT-large whole-word-masking, where the correlation is 39.5 ($pval < 0.05$).

relation r^i is:

$$\mathcal{L}_c = \sum_{n=1}^{k} \sum_{m=n+1}^{k} D_{KL}(Q_n^{r_i} || Q_m^{r_i}) + D_{KL}(Q_m^{r_i} || Q_n^{r_i})$$

MLM Loss Since the consistency loss is different from the Cross-Entropy loss the PLM is trained on, we find it important to continue the MLM loss on text data, as was observed in previous work (Geva et al., 2020).

We consider two alternatives for continuing the pretraining objective: (1) MLM on Wikipedia and (2) MLM on the patterns of the relations used for the consistency loss. We found that the latter works better. We denote this loss by $\mathcal{L}_{MLM}$

Consistency Guided MLM Continual Training Combining our novel consistency loss with the regular MLM loss, we continue the PLM training by combining the two losses. The combination of the two losses is determined by a hyperparameter λ, resulting in the following final loss function:

$$\mathcal{L} = \lambda \mathcal{L}_c + \mathcal{L}_{MLM}$$

This loss is computed per relation, for one KB tuple. We have many of these instances, which we require to behave similarly. Therefore, we batch together l tuples from the same relation and apply the consistency loss function to all of them.

8.2 Setup

Since we evaluate our method on unseen relations, we also split the train/test by the relation type (e.g., location-based relations, which are very common in T-REx). Moreover, our method is aimed to be simple, effective and to require only minimal supervision. Therefore we opt to use only a minimal number of relations for training. In practice, we use only three relations: *original-language*, *named-after*, and *original-network*, which were chosen randomly, out of the non-location related relations.[12] For validation, we randomly pick three relations of the remaining relations and use the remaining twenty-five for testing.

We perform minimal tuning of the parameters to pick the best model, train for 3 epochs, and select the best model based on *Consistent-Acc* on the validation set. For efficiency reasons, we use the base version of BERT.

Model	Accuracy	Consistency	Consistent-Acc
majority	24.4+-22.5	100.0+-0.0	24.4+-22.5
BERT-base	45.6+-27.6	58.2+-23.9	27.3+-24.8
BERT-ft	**47.4**+-27.3	**64.0**+-22.9	**33.2**+-27.0
-consistency	46.9+-27.6	60.9+-22.6	30.9+-26.3
-typed	46.5+-27.1	62.0+-21.2	31.1+-25.2
-MLM	16.9+-21.1	80.8+-27.1	9.1+-11.5

Table 7: Knowledge and consistency results for the majority baseline, BERT base, and our model. We report the *Accuracy* using the original patterns from LAMA, the *Consistency*, and *Consistency-Acc* metrics. The results are average over 25 relations that were not part of the training and evaluation data. Best model's performance for each metric is highlighted in bold.

8.3 Improved Consistency Results

The results are presented in Table 7. We report the aggregated results for all of the relations, apart from those that were used for training or validation. As in the previous section, we report the mean over the inspected relations and the standard deviation. We report the results of the majority baseline (first row), as well as the vanilla BERT-base model (second row). Finally, we report the results of our new model (third row). First, we note that our model significantly improves consistency: to 64.0% (compared with 58.2% for BERT-base, an increase of 5.8 points). The *Accuracy* also improves from the BERT baseline, from 45.6% to 47.4%. Finally, and most importantly, we see an increase of 5.9 points in *Consistent-Acc*, which is achieved due to the improved consistency of the model. Notably, these improvements arise from training on merely three relations, meaning that the model improved its consistency ability and generalized to new relations. We measure the statistical significance of our method compared to the BERT baseline, using McNemar's test (following Dror et al. (2018, 2020)) and find all results to be significant ($pval \ll 0.01$).

We also perform an ablation study to quantify the utility of the different components. First, we report on the finetuned model without the consistency loss (-consistency). Interestingly, it does improve over the baseline (BERT-base), but it lags behind our finetuned model. Second, applying our loss on the candidate set rather than on the entire vocabulary is beneficial (-typed). Finally, by not performing the MLM training on the generated patterns (-MLM), the consistency results improve significantly (80.8%), however, it also hurts the

[12] Since many of the relations are location-based we wish to avoid a train-test leakage.

Accuracy and *Consistent-Acc* metrics. Thus, we see the MLM training as a regularizer that prevents catastrophic forgetting.

9 Discussion

Consistency for Downstream Tasks The rise of PLMs has improved many tasks, but has also brought a lot of expectations. The standard usage of these models is by pretraining on a large corpus of unstructured text and then finetune on a task of interest. The first step is thought of as proving a good language-understanding component, whereas the second step is used to teach the format and the nuances of a downstream task.

As discussed earlier, consistency is a crucial component of many NLP system (Du et al., 2019; Asai and Hajishirzi, 2020; Denis and Baldridge, 2009; Kryscinski et al., 2020) and obtaining this skill from a PLM would be extremely beneficial and have the potential to make specialized consistency solutions in downstream tasks redundant. Indeed, there is an ongoing discussion about the ability to acquire understanding of "meaning" from raw text signal alone (Bender and Koller, 2020). Thus, our new benchmark will allow track the progress of consistency in PLMs.

Broader Sense of Consistency In this work we focus on one type of consistency, that is, consistency to paraphrases, however, the consistency term is broader than that. For instance, previous work has studied the effect of negation on factual statements, which can also be seen as consistency (Ettinger, 2020; Kassner and Schütze, 2020). As such, a consistent model is expected to return a different answer to the prompts: "*Birds* can *[MASK]*" and "*Birds* cannot *[MASK]*". The inability to do so, as was shown in these works, also shows the lack of models' consistency.

Usage of PLMs as KBs Our work follows the setup of Petroni et al. (2019); Jiang et al. (2020), where PLMs are being tested as KBs. While it is an interesting setup for probing models for knowledge and consistency, it lacks important properties of standard KBs: (1) the ability to return more than a single answer and (2) the ability to return no answer. Although some heuristics can be used for allowing a PLM to do so, e.g. using a threshold on the probabilities, it is not the way that the model was trained, and thus may not be optimal. As such, newer approaches propose to use PLMs as a starting point to more complex systems, that provide promising results and solve the above problems (Thorne et al., 2020).

Brittleness of Neural Models Our work also relates to the problem of brittleness in neural networks. One example of this brittleness is the vulnerability to adversarial attacks (Szegedy et al., 2014; Jia and Liang, 2017). The other problem, closer to the problem we explore in this work, is the poor generalization to paraphrases. For example, Gan and Ng (2019) created a paraphrase version for a subset of SQuAD (Rajpurkar et al., 2016), and showed that models' performance drops significantly. Ribeiro et al. (2018) proposed another method for creating paraphrases and performed a similar analysis for visual question answering and sentiment analysis. Recently, Ribeiro et al. (2020) proposed CHECKLIST, a system that tests models' vulnerability to several linguistic perturbations.

PARAREL🤘 enables us to study the brittleness of PLMs, and separate between facts which are robustly encoded in the model, compared to mere 'guesses', which may arise from some heuristic or spurious correlations with certain patterns (Poerner et al., 2020). In practice, we show that PLMs are susceptible to small perturbations, and thus, finetuning on some downstream task (and dataset), that typically are not extensive, and do not contain equivalent examples, are not likely to perform better with this regard.

10 Conclusions

In this work, we study the consistency of PLMs with regard to their ability to extract knowledge. We build a high-quality resource named PARAREL🤘, that contains 328, high-quality patterns for thirty-eight relations. Using PARAREL🤘, we measure consistency in multiple PLMs, including BERT, RoBERTa, and ALBERT, and show that although the two latter are superior in other tasks over BERT, they fall short in terms of consistency. However, overall the consistency ability of these models is low. We release PARAREL🤘 along with data tuples from (Elsahar et al., 2018) as a new benchmark, to track the consistency of models to knowledge. Finally, we propose a new simple method to improve the consistency of PLMs, by continuing the pretraining step with a novel loss. We show this method to be effective and to improve both the consistency of models as well as their ability to extract the correct facts.

References

Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. 2019. Synthetic qa corpora generation with roundtrip consistency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6168–6173.

Kim Allan Andersen and Daniele Pretolani. 2001. Easy cases of probabilistic satisfiability. *Annals of Mathematics and Artificial Intelligence*, 33(1):69–91.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433.

Akari Asai and Hannaneh Hajishirzi. 2020. Logic-guided data augmentation and regularization for consistent question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5642–5650, Online. Association for Computational Linguistics.

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017. What do neural machine translation models learn about morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 861–872.

Yonatan Belinkov and James Glass. 2019. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72.

Emily M Bender and Alexander Koller. 2020. Climbing towards nlu: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198.

Rahul Bhagat and Eduard Hovy. 2013. What is a paraphrase? *Computational Linguistics*, 39(3):463–472.

Lukas Biewald. 2020. Experiment tracking with weights and biases. Software available from wandb.com.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Kai-Wei Chang, Rajhans Samdani, Alla Rozovskaya, Nick Rizzolo, Mark Sammons, and Dan Roth. 2011. Inference protocols for coreference resolution. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 40–44.

Ethan A. Chi, John Hewitt, and Christopher D. Manning. 2020. Finding universal grammatical relations in multilingual BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5564–5577, Online. Association for Computational Linguistics.

Jeff Da and Jungo Kasai. 2019. Cracking the contextual commonsense code: Understanding commonsense reasoning aptitude of deep contextual representations. In *Proceedings of the First Workshop on Commonsense Inference in Natural Language Processing*, pages 1–12, Hong Kong, China. Association for Computational Linguistics.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Machine Learning Challenges Workshop*, pages 177–190. Springer.

Pascal Denis and Jason Baldridge. 2009. Global joint models for coreference resolution and named entity classification. *Procesamiento del Lenguaje Natural*, 42.

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.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392.

Rotem Dror, Lotem Peled-Cohen, Segev Shlomov, and Roi Reichart. 2020. Statistical significance testing for natural language processing. *Synthesis Lectures on Human Language Technologies*, 13(2):1–116.

Xinya Du, Bhavana Dalvi, Niket Tandon, Antoine Bosselut, Wen-tau Yih, Peter Clark, and Claire Cardie. 2019. Be consistent! improving procedural text comprehension using label consistency. 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 2347–2356.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. 2020. Amnesic probing: Behavioral explanation with amnesic counterfactuals.

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. 2018. T-rex: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

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.

Maxwell Forbes, Ari Holtzman, and Yejin Choi. 2019. Do neural language representations learn physical commonsense? *arXiv preprint arXiv:1908.02899*.

Wee Chung Gan and Hwee Tou Ng. 2019. Improving the robustness of question answering systems to question paraphrasing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6065–6075.

Mor Geva, Ankit Gupta, and Jonathan Berant. 2020. Injecting numerical reasoning skills into language models. In *Association for Computational Linguistics (ACL)*.

Yoav Goldberg. 2019. Assessing bert's syntactic abilities. *arXiv preprint arXiv:1901.05287*.

Pierre Hansen and Brigitte Jaumard. 2000. Probabilistic satisfiability. In *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, pages 321–367. Springer.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*. Association for Computational Linguistics.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031.

Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Nora Kassner, Benno Krojer, and Hinrich Schütze. 2020. Are pretrained language models symbolic reasoners over knowledge? In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 552–564, Online. Association for Computational Linguistics.

Nora Kassner and Hinrich Schütze. 2020. Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7811–7818, Online. Association for Computational Linguistics.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Tao Li, Vivek Gupta, Maitrey Mehta, and Vivek Srikumar. 2019. A logic-driven framework for consistency of neural models. 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 3924–3935, Hong Kong, China. Association for Computational Linguistics.

Tal Linzen. 2020. How can we accelerate progress towards human-like linguistic generalization? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5210–5217.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of lstms to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

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. *arXiv preprint arXiv:1907.11692*.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605.

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Brussels, Belgium. Association for Computational Linguistics.

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.

David Picado Muiño. 2011. Measuring and repairing inconsistency in probabilistic knowledge bases. *International Journal of Approximate Reasoning*, 52(6):828–840.

Kim Anh Nguyen, Sabine Schulte im Walde, and Ngoc Thang Vu. 2016. Integrating distributional lexical contrast into word embeddings for antonym-synonym distinction. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 454–459.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Fabio Petroni, Patrick Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2020. How context affects language models' factual predictions. In *Automated Knowledge Base Construction*.

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.

Nina Poerner, Ulli Waltinger, and Hinrich Schütze. 2020. E-bert: Efficient-yet-effective entity embeddings for bert. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 803–818.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Shauli Ravfogel, Yanai Elazar, Jacob Goldberger, and Yoav Goldberg. 2020. Unsupervised distillation of syntactic information from contextualized word representations. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 91–106, Online. Association for Computational Linguistics.

Abhilasha Ravichander, Eduard Hovy, Kaheer Suleman, Adam Trischler, and Jackie Chi Kit Cheung. 2020. On the systematicity of probing contextualized word representations: The case of hypernymy in BERT. In *Proceedings of the Ninth Joint Conference on Lexical and Computational Semantics*, pages 88–102, Barcelona, Spain (Online). Association for Computational Linguistics.

Marco Tulio Ribeiro, Carlos Guestrin, and Sameer Singh. 2019. Are red roses red? evaluating consistency of question-answering models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6174–6184, Florence, Italy. Association for Computational Linguistics.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Semantically equivalent adversarial rules for debugging nlp models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 856–865.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

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

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in bertology: What we know about how bert works. *arXiv preprint arXiv:2002.12327*.

Micah Shlain, Hillel Taub-Tabib, Shoval Sadde, and Yoav Goldberg. 2020. Syntactic search by example. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 17–23, Online. Association for Computational Linguistics.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *International Conference on Learning Representations*.

Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. 2020. olmpics-on what language model pre-training captures. *Transactions of the Association for Computational Linguistics*, 8:743–758.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. Bert rediscovers the classical nlp pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601.

Matthias Thimm. 2009. Measuring Inconsistency in Probabilistic Knowledge Bases. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI'09)*, pages 530–537. AUAI Press.

Matthias Thimm. 2013. Inconsistency measures for probabilistic logics. *Artificial Intelligence*, 197:1–24.

James Thorne, Majid Yazdani, Marzieh Saeidi, Fabrizio Silvestri, Sebastian Riedel, and Alon Halevy. 2020. Neural databases. *arXiv preprint arXiv:2010.06973*.

Alex Warstadt, Yian Zhang, Xiaocheng Li, Haokun Liu, and Samuel R. Bowman. 2020. Learning which features matter: RoBERTa acquires a preference for linguistic generalizations (eventually). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 217–235, Online. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Wenhan Xiong, Jingfei Du, William Yang Wang, and Veselin Stoyanov. 2020. Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Xikun Zhang, Deepak Ramachandran, Ian Tenney, Yanai Elazar, and Dan Roth. 2020. Do language embeddings capture scales? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 4889–4896.

11 Appendix

We heavily rely on Hugging Face's Transformers library (Wolf et al., 2020) for all experiments involving the PLMs. We used Weights & Biases for tracking and logging the experiments (Biewald, 2020). Finally, we used sklearn (Pedregosa et al., 2011) for other ML-related experiments.