Chapter 4
Knowledge Acquired by Foundation Models

Abstract During pre-training, a Foundation Model is trained on an extensive collection of documents and learns the distribution of words in correct and fluent language. In this chapter, we investigate the knowledge acquired by PLMs and the larger Foundation Models. We first discuss the application of Foundation Models to specific benchmarks to test knowledge in a large number of areas and examine if the models are able to derive correct conclusions from the content. Another group of tests assesses Foundation Models by completing text and by applying specific probing classifiers that consider syntactic knowledge, semantic knowledge, and logical reasoning separately. Finally, we investigate if the benchmarks are reliable and reproducible, i.e. whether they actually test the targeted properties and yield the same performance values when repeated by other researchers.

Keywords Knowledge in foundation models · Common Sense knowledge · Logical coherence · Benchmark collections · Reproducibility

During pre-training, Pre-trained Language Models (PLMs) and the larger Foundation Models are trained on an extensive collection of documents and learn the distribution of words in correct and fluent language. During fine-tuning, the models are adapted to a specific task using the knowledge from the pre-training and requiring only a small set of manually labeled fine-tuning data. In this chapter, we investigate the knowledge acquired by these models by different types of tests:

- We first assess PLMs and Foundation Models by specific benchmarks to test knowledge in a large number of areas and examine if the models are able to derive correct conclusions from the content (Sect. 4.1). Usually these benchmark collections have an aggregated performance measure averaging over different tests. Benchmark tests can be accomplished by fine-tuning models to perform specific classification tasks or by few-shot querying Foundation Models.
- Then we assess Foundation Models by completing text and by applying specific probing classifiers without adapting model parameters (Sect. 4.2). We separately consider syntactic knowledge, semantic knowledge and logical reasoning and

G. Paaß, S. Giesselbach, *Foundation Models for Natural Language Processing*,
Artificial Intelligence: Foundations, Theory, and Algorithms,
https://doi.org/10.1007/978-3-031-23190-2_4

demonstrate the achievements and deficits in different areas and for different model architectures.

- Finally, we investigate if the benchmarks are reliable, i.e. actually test the targeted properties (Sect. 4.3). Moreover, we analyze if published benchmark results are reproducible and yield the same performance values if they are repeated by other researchers.

4.1 Benchmark Collections

In order to arrive at quantitative measures of common sense knowledge and commonsense reasoning, the community has compiled a number of benchmarks. These allow a standardized comparison of different aspects of natural language understanding and provide comparable scores for the strength and weaknesses of different PLMs. Benchmarks have been a key driver for the development of language models. A comprehensive collection of benchmarks and the corresponding leaderboards are provided by PapersWithCode [45]. A survey of actual benchmarks is given by Storks et al. [62].

A fair comparison of model architectures requires that the number of parameters, the size of the training data, and the computing effort for training are similar. This has been extensively discussed in Sect. 3.5.1. Therefore, many authors conduct extensive ablation studies to adjust their training resources to a standard, e.g. to BERT as a "benchmark model". This is really important, as it helps the reader to get an intuition for the impact of pre-training resources. Nevertheless, comparability is often hampered by two problems:

1. Some training datasets, e.g. the BooksCorpus of BERT, are not publicly available.
2. These comparisons do not show the performance of a model when the size of data, the number of parameters, or the computing effort are increased.

Therefore, statements like *"Model architecture A is superior to model architecture B on performing task X."* in general are not valid, but have to be qualified [2], e.g. "Model architecture *A* is superior to model architecture *B* on performing task *X*, when pre-trained on a small/large corpus of low/high quality data from domain *Y* with computing effort *Z*."

4.1.1 The GLUE Benchmark Collection

To test the ability of PLMs to capture the content of a document, the GLUE (Sect. 2.1.5) set of benchmarks has been developed. This is a collection of 9 benchmarks testing different aspects of *Natural Language Understanding* (*NLU*). The joint performance is measured by a single score, which has the value 87.1 for human annotators. The tasks are described in detail by examples in Table 2.1. It turns out that variants of BERT fine-tuned to the different GLUE-tasks can yield better results than people. The results are determined for the large variants of the models and shown in Table 4.1.

In the past years GLUE was routinely employed to demonstrate the NLU capabilities of PLMs. Currently, the best average value of 91.4 after fine-tuning was reached by DeBERTaV3 [18] (Sect. 3.1.1). It uses separate embeddings for content and position and employs a corresponding disentangled attention mechanism. There are only three tasks where PLMs are worse than humans, but only by a small margin. Note that ensembles of several models often yield slightly better results. Nangia et al. [42] also measures the performance of human teams of 5 people. The numbers are not comparable as cases were excluded when the teams arrived at split judgment. Newer models such as PaLM use SuperGLUE instead of GLUE because GLUE is considered too simple.

4.1.2 SuperGLUE: An Advanced Version of GLUE

Due to the progress in the last years, PLMs have reached human performance in most tasks and the GLUE is no longer able to discriminate between models. Therefore, the authors of GLUE proposed a more demanding test suite called **SuperGLUE** [68] as an advanced version of GLUE with eight challenging tasks. The tasks are similar to GLUE with longer contexts to consider.

- *BoolQ* is a QA-task with questions collected from Google search and yes/no answers.
- *CB* is a textual entailment task.
- *COPA* is a causal reasoning task in which a system must determine either the cause or effect of a given premise from two possible choices.
- *MultiRC* is a QA task where each instance consists of a context passage, a question about that passage, and a list of possible answers.
- In *ReCoRD* each example consists of a news article and an article in which one entity is masked out. The system must predict the masked entity from a list of possible entities.
- *RTE* requires detecting whether a hypothesis is implied by a premise.
- *WiC* is a word sense disambiguation task, where for two given sentences the system has to determine if a polysemous word is used with the same sense in both sentences.
- *WSC* is the Winograd Schema Challenge, where the system has to determine the correct noun phrase represented by a pronoun.

The performance again is measured by a single average score with a value of 89.8 for human annotators [66].

Table 4.1 Results for the GLUE benchmark for four different models and human annotators. The best value of a PLM for each task is printed in bold [18, p. 7]. Human scores better than all model scores are underlined

	CoLA	QQP	MNLI m	SST-2	STS-B	QNLI	RTE	WNLI	MRPC	
	Mcc	Acc	Acc	Acc	Corr	Acc	Acc	Acc	F1	
Model	Grammar	Paraphr.	Entail	Sentim.	Similar	Question	Entail	Coref	Paraphr.	Avg
Human [42]	66.4	80.4	92.0	97.8	92.7	91.2	93.6	95.9	86.3	87.1
BERT$_{\text{LARGE}}$	60.6	91.3	86.6	93.2	90.0	92.3	70.4	65.1	88.0	84.1
RoBERTa$_{\text{LARGE}}$	68.0	92.2	90.2	96.4	92.4	93.9	86.6	89.9	90.9	88.8
XLNET$_{\text{LARGE}}$	69.0	92.3	90.8	**97.0**	92.5	94.9	85.9	92.5	90.8	89.2
DeBERTaV3$_{\text{LARGE}}$	**75.3**	**93.0**	**91.8**	96.9	**93.0**	**96.0**	92.7	–	**92.2**	**91.4**

GPT-3 [7] is a huge language model (Sect. 3.1.2), which can be instructed to perform a task without fine-tuning (Sect. 3.2). With this few-shot learning GPT-3 achieved an average SuperGLUE score of only 71.8 as shown in Table 4.2. Obviously fine-tuning the specific tasks seems to be important. Recently a fine-tuned DeBERTa ensemble (Sect. 3.1.1) surpassed human performance on SuperGLUE with an average score of 90.3. The most difficult task is a comparison of word senses in two sentences (WiC), where an accuracy of about 77% can be reached. The autoregressive LM *PaLM* 540B was fine-tuned on SuperGLUE and achieved an average of 90.4% on the test set [9, p. 13]. The best average of 91.2% was obtained by the *ST-MoE*$_{32B}$ mixture-of-experts model (Sect. 3.5.2) with 269B parameters [73]. This shows that Foundation Models are able to analyze complex text semantics.

GLUE and SuperGLUE have been criticized, as the answers of the posed problems always can be reduced to a classification task and the systems do not have to formulate an answer in natural language. In addition, it turns out that the performance of PLMs is not very stable. It has been shown that the prediction of current models often change in an inconsistent way, if some words are replaced [51]. If, for instance, in a sentiment analysis the input *"I love the flight"* is classified as *positive*, then *"I didn't love the flight"* should not be classified as *neutral*. Ribeiro et al. [51] show that inconsistencies like this often occur. They developed the **CheckList** system (Sect. 4.3.1), which automatically generates test examples for probing a model.

4.1.3 Text Completion Benchmarks

The task of an autoregressive language models is the reliable generation of the next word in a text. This has to obey grammatical correctness as well as semantic consistency. The *LAMBADA benchmark* [44] is a good test to demonstrate this ability. It consists of about 10,000 passages from the BooksCorpus containing unpublished novels. The task is to predict the missing last word of the last sentence of each passage. Examples were filtered by humans to ensure that models need to take into account the full passage of at least 50 tokens to induce the final word.

An example is the passage *"Both its sun-speckled shade and the cool grass beneath were a welcome respite after the stifling kitchen, and I was glad to relax against the tree's rough, brittle bark and begin my breakfast of buttery, toasted bread and fresh fruit. Even the water was tasty, it was so clean and cold. It almost made up for the lack of ____."*, where *"coffee"* is the missing target word to be predicted. Examples which could be easily predicted by simpler language models were omitted. Examples were only selected, if the target word could be predicted by humans from the full passage but not from the last sentence.

The GPT-3$_{175B}$ autoregressive language model [48] predicted the last word with 76.2% [7, p. 12]. PaLM$_{540B}$ with few-shot instructions could increase the accuracy to 89.7 [9, p. 79]. This means that in nearly nine of ten cases, the predicted word was exactly the missing word in the test data.

Table 4.2 Results for the SuperGLUE benchmark on the test set for human annotators and five different models. The best value for each task is printed in bold and human values better than the model values are underlined. For GPT-3 few-shot values (FS) are reported, fine-tuned otherwise

	BoolQ	CB	COPA	MultiRC	ReCoRD	RTE	WiC	WNLI	
	Acc	Acc/F1	Acc	F1a/EM	F1/EM	F1/EM	Acc	Acc	
Model	QA y/n	Entail	Cause	QA mult.	Entities	Entail	WSD	Coref	Avg
Human [68]	89.0	95.8/98.9	100.0	81.8/51.9	91.7/91.3	93.6	80.0	100.0	89.8
BERT$_{336M}$ [68]	77.4	83.6/75.7	70.6	70.0/24.0	72.0/71.3	71.6	69.5	64.3	69.0
GPT-3$_{270B}$ FS [7]	76.4	75.6/52.0	92.0	75.4/30.5	91.1/90.2	69.0	49.4	80.1	71.8
DeBERTA Ens. [19]	90.4	94.9/97.2	98.4	88.2/63.7	94.5/94.1	93.2	77.5	95.9	90.3
PaLM$_{540B}$ [9]	91.9	94.4/96.0	99.0	88.7/63.6	94.2/93.3	**95.9**	77.4	95.9	90.4
ST-MoE$_{32B}$ [73]	**92.4**	**96.9/98.0**	**99.2**	**89.6/65.8**	**95.1/94.4**	93.5	**77.7**	**96.6**	**91.2**

Another relevant benchmark for language modeling is *WikiText-103* [38] of 28k articles from Wikipedia with 103M tokens. If large Foundation Models are applied to this corpus the following perplexities result: GPT-$2_{1.7B}$ 17.5 [48], Megatron-LM 10.8 [58], Gopher_{280B} 8.1 [49, p. 61]. Recently a small $\text{Retro}_{1.8B}$ model with retrieval could reduce this perplexity to 3.9 [49, p. 12]. Note that there might be a partial overlap of Wikitext 103 with Retro's training data not caught by deduplication.

4.1.4 Large Benchmark Collections

Recently large autoregressive language models like GPT-3, Gopher, and PaLM have been developed, which are trained on extremely large document collections with hundreds of billions of tokens. The models should perform well across a wide range of tasks. Therefore, instead of the limited GLUE benchmarks a large number of benchmarks covering many aspects of possible applications are used to evaluate their performance.

A frequent opinion is that current benchmarks are insufficient and "saturate", "have artifacts", and are "overfitted by researchers". Bowman et al. [5] argue that "evaluation for many natural language understanding (NLU) tasks is broken". They complain that there are systems at the top of the leaderboards which fail in simple test cases (cf. [51]). As a consequence they formulate four requirements on new benchmarks:

- A model should only reach good performance on the benchmark if it also has a good performance on actual applications.
- The annotation of benchmarks should be accurate and not ambiguous (e.g. 36% of the answers in Natural Questions are ambiguous).
- The benchmarks should be large and challenging enough to detect relevant performance differences between models.
- Benchmarks should reveal plausibly harmful social biases in systems, and should not encourage the creation of biases.

They summarize some promising developments that could support these challenges, including data collection involving both crowdworkers and domain experts, and larger-scale data validation.

To address this criticism, two comprehensive collections of benchmarks have been defined. The *Massive Multitask Language Understanding* (MMLU) benchmark [20] emulates human exams with multiple choice questions, each with four responses. In addition to logical and mathematical reasoning it tests a model's ability across a wide range of academic subjects from computer science to history and law. The other collection is the *BIG-bench* collaborative benchmark [1, 60], designed to evaluate language interpretation aspects like reading comprehension, question answering, world understanding, etc. Both benchmark collections include more than a hundred tasks.

Table 4.3 Groups of evaluation benchmarks for Gopher and related models [49, p. 8]

Task group	# Tasks	Examples
Language modeling	20	WikiText-103, The Pile: PG-19, arXiv, FreeLaw, ...
Reading comprehension	3	RACE-m, RACE-h, LAMBADA
Fact checking	3	FEVER (2-way & 3-way), MultiFC
Question answering	3	Natural questions, TriviaQA, TruthfulQA
Common sense	4	HellaSwag, Winogrande, PIQA, SIQA
Massive multitask language understanding (MMLU) [20]	57	High school chemistry, astronomy, clinical knowledge, social science, math, ...
BIG-bench [60]	62	Causal judgement, epistemic reasoning, temporal sequences, logic, math, code, social reasoning, ...

The *Gopher* model with 280B parameters together with alternatives like GPT-3, Jurassic-1, and Megatron-Turing NLG (all discussed in Sect. 3.1.2) were tested on these and other benchmarks. Note that this was done with a total of 152 benchmarks described in Table 4.3. Gopher shows an improvement on 100 of 124 tasks (81%) compared to the previous SOTA scores. In language modeling (next word prediction) Gopher improves SOTA for 10 of 19 benchmarks. Note that all benchmark results were not obtained after fine-tuning but by zero-shot or few-shot learning.

The distribution Gopher accuracies for thematic groups are shown in Fig. 4.1. Gopher is able to increase SOTA for 4 out of 7 math tasks, 5 out of 9 common sense tasks, 9 out of 12 logical reasoning tasks, 22 out of 24 fact checking and general knowledge tasks, all 24 STEM (Science Technology Engineering Mathematics) and medicine tasks, all 15 humanities and ethics task, and 10 out of 11 reading comprehension tasks. The average accuracies for common sense and general knowledge are about 50%, indicating that some knowledge exists but can be improved. Among these tests were benchmarks on logical reasoning, which, for instance, include "Formal Fallacies Syllogisms Negation" or "Logical Fallacy Detection". Only two of the 19 benchmarks achieved an accuracy of more than 60% [49, p. 58], indicating that even for this large model correct reasoning is a major obstacle. Obviously this spectrum of evaluation gives a deep insight into the capabilities of the compared models. It can be expected that the new Retro model (Sect. 6.2.3), which performs retrieval during language generation, will improve these results.

The *PaLM* autoregressive language model with 580B parameters [9, p. 15] recently was evaluated with the BIG-bench benchmark. On the 150 tasks, PaLM with 5-shot prompts achieved an normalized average score of 46%, which was better than the average human score of 39%. However, the best human experts have a score of about 77%. The detailed results for the different BIG benchmark areas are not yet available. On a subset of 58 BIG-tasks, which were also used by prior models, PaLM obtained a 5-shot normalized score of about 55%, again above the human average of 49%, outperforming Chinchilla (47%) and Gopher (30%). GPT-3 achieved a 1-shot performance of 16% on the 58 tasks. In general Foundation Models like Gopher and PaLM with several hundred billion parameters have 'dramatically better' results

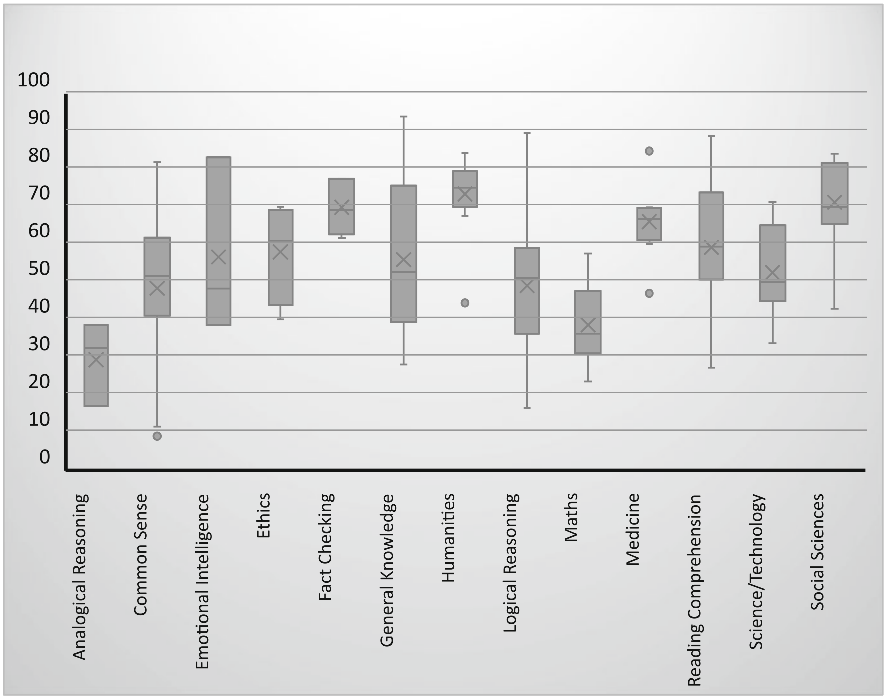

Fig. 4.1 Accuracies in percent of different groups covering 152 different benchmarks evaluated for the Gopher model [49, p. 57]. The 25% and 75% percentiles are given by the box, and the inner line is the median. The outside lines indicate variability outside the upper and lower quartiles

on BIG than smaller models, even if the model architecture is not fundamentally different [1]. In this respect Foundation Models show a qualitatively new behavior.

Researchers at Google have proposed to use the BIG-bench benchmark with currently 200 tasks as a replacement for the Turing test for "intelligence" [61]. In this way the knowledge of an AI-System can be checked at a large scale. Recently, a Google engineer published a dialog [31] with the LaMDA language model (Sect. 6.6.3). In his view this indicates that LaMDA is "sentient". However, this aspect of human intelligence is not checked by knowledge and reasoning tests such as BIG and requires the development of new types of tests.

4.1.5 Summary

Benchmark collections are a popular way to demonstrate the superiority of a Pre-trained Language Model for specific tasks. To show the merits of an architecture, however, also the number of parameters, the size of training data, and the computing effort has to be reported and compared, because these numbers also affect the model performance.

The GLUE benchmark collection of nine language understanding tasks has demonstrated the considerable progress of PLMs during the last years. It tests the ability of PLMs to detect paraphrases, coreference relations, logical entailments and grammatical correctness. Meanwhile, the average accuracy exceeds the average human performance. The similar, more challenging SuperGLUE benchmark suite has been introduced, where human performance is also surpassed. For autoregressive language models the LAMBADA benchmark requires an impressive ability to determine the most probable last word of a paragraph. Current models like PaLM are able to predict the last word with an accuracy of nearly 90% demonstrating its ability to capture the flow of arguments.

Foundation Models are usually tested by extensive standardized test collections covering many aspects like common sense knowledge, emotional intelligence, logical reasoning, or social sciences. Recent Foundation Models like Gopher and PaLM, with several hundred billion parameters, have been able to achieve performance better than that the human average and ‘dramatically better’ than smaller models. However, these models still have a lower accuracy than human experts. Although the benchmarks are very expressive, they do not take into account the societal impact of the models and are unable to detect features like self-awareness and sentience.

4.2 Evaluating Knowledge by Probing Classifiers

In this section, we examine the extent to which PLMs acquire different types of knowledge. We discuss the covered knowledge for the small BERT model and later review the improvements for foundation models such as GPT-3 and PaLM. First, we consider their syntactic knowledge of correct language. Then, we investigate how much common sense knowledge is represented by PLMs. Finally, we explore whether the output produced by PLMs is logically consistent.

4.2.1 BERT’s Syntactic Knowledge

We discuss the syntactic knowledge incorporated in PLMs using BERT as an example. In the course of pre-training BERT is able to capture *syntactic knowledge* [54]. Embeddings can encode information about parts of speech, syntactic phrases and syntactic roles. Probing classifiers can predict part-of-speech tags and supersense information with an accuracy of 85% [33]. Obviously, this information has to be encoded in BERT’s final embeddings. BERT also has knowledge of subject-verb agreement [17] and semantic roles [14]. It is also possible to extract dependency trees and syntactic constituency trees from BERT [21, 23, 27]. While probing indicates that the information can be extracted from the representation, it can be shown [13] that in some cases the features are not used for prediction. According to an empirical evaluation PLMs encode linguistic information with phrase features in the bottom layers, syntactic features in the middle layers and semantic features in the top layers [23].

However, BERT's syntactic knowledge is incomplete and there is, for example, evidence that BERT often does not capture *negations*. For instance, BERT$_{\text{LARGE}}$ is able to determine the correct supersense, e.g. *"bird"* in the masked sentence *"A robin is a [MASK]"* with high probability [14]. On the other hand, the model predicts *"robin"*, *"bird"*, *"penguin"*, *"man"*, *"fly"* with maximum probabilities for the mask in *"A robin is not a [MASK]"*, effectively ignoring the negation.

Some benchmarks described in Sect. 4.1 check the syntactic knowledge of PLMs. An example is the GLUE's CoLA task testing the grammatical correctness of sentences, which is the most difficult task of GLUE where the best models only yield about 75% correct answers (Table 4.1). SuperGLUE (Sect. 4.1.2) is a benchmark, which requires syntactic knowledge, e.g. for the textual entailment task COPA and the coreference resolution task WSC. While the fine-tuned BERT gets an average score of 69.0 the fine-tuned PaLM$_{540\text{B}}$ achieves an average of 91.4 (Table 4.2). Large foundation models such as PaLM, which has more than 1000 times as many parameters as BERT, are obviously able to capture syntactical knowledge much better than the 'small' BERT.

4.2.2 Common Sense Knowledge

World knowledge, also called *common sense knowledge*, consists of facts about our every day world, such as *"fire is hot"*. A simple method of checking world knowledge is to query BERT with cloze statements, for example, *"Einstein was born in [MASK]"*. BERT acquires some *semantic knowledge* about semantic roles and encodes information about entity types and relations [54]. For instance, in the sentence *"to tip a [MASK]"* the token *"waiter"* gets a high probability for the position of *[MASK]*. Petroni et al. [46] and Zhou et al. [72] experimented with such queries and concluded that BERT contains world knowledge competitive with traditional supervised information extraction methods. It has been shown that BERT's contextual embeddings make up clusters corresponding to *word senses* [56]. This explains why BERT is quite capable of word sense disambiguation (Fig. 2.10).

Petroni et al. [46] remark that certain types of factual knowledge are learned much more easily than others by the standard language model pre-training approaches. They state that without fine-tuning, BERT contains relational knowledge competitive with traditional NLP methods that have some access to oracle knowledge. In addition, BERT also does remarkably well on open-domain question answering against a supervised baseline. These capabilities of BERT are a great achievement.

The language model GPT-3 has one hundred times more parameters than BERT and a dramatically better common sense knowledge. This, for example, can be seen from its answers (A) to the questions (Q): *"Q: Are there any animals with three legs?" "A: No, there are no animals with three legs."* or *"Q: Which is heavier, a football player or a car?" "A: A car is heavier than a football player."* [29]. In an initial experiment eighty persons were asked to assess, if short 200 word articles were written by humans or GPT-3. The persons judged incorrectly 48% of the time, doing only slightly better than random guessing [7].

However, the semantic knowledge of PLMs is not perfect. BERT, for instance, has difficulties with the representation of numbers and often has problems with the replacement of *named entities* (*NE*s), e.g. person names or location names. For example, replacing names in the coreference task changes 85% of coreference assignments of expressions that refer to the same entity [3]. Obviously the pre-trained version of BERT struggles to generalize the relations involving one named entity to other named entities of the same type. Moreover, BERT has problems to transfer knowledge based on the roles or types of objects. In addition, it is possible to mislead BERT by adding some content to a cloze query. An example is the word *"Talk"* in *"Talk? Birds can [MASK]"*. A human would ignore *"Talk?"* and use his world knowledge to generate a result like *"fly"*. In contrast, PLMs can be misled and produce the wrong answer *"talk"* for the mask [26].

A related phenomenon is the invariance to *paraphrases*. Elazar et al. [12] generate a high-quality set of 328 paraphrases to express 38 relations. Examples are *"X originally aired on [MASK]"* and *"X premiered on [MASK]"*, which should give the same prediction for *[MASK]*, if *"X"* is replaced by some TV series like *"Seinfeld"*. Although the models in about 60% of the cases have access to the required knowledge to fill the mask correctly, BERT$_{\text{LARGE}}$ yields a consistency in paraphrases in only 48.7% of the cases. This indicates that not every fact present in the training data is encoded in the parameters and that the model does not always detect the equivalence of paraphrases. The model variants RoBERTa and ALBERT achieve a lower consistency, although they are superior to BERT in other tasks.

It is instructive to consider the influence of word order on the performance of BERT. Word order is taken into account by specific position embeddings, which are added to the token embeddings. It turns out, however that masked language models like BERT still achieve a high accuracy, if word positions are permuted. For pre-training Sinha et al. [59] perform sentence permutations, where each word in a sentence is randomly placed at a different position. The model was fine-tuned on GLUE, a set of classification tasks for natural language understanding (Sect. 2.1.5). If we ignore the CoLA-task, which checks linguistic acceptability, the model on average only looses 3.4% accuracy if the word order is permuted compared to the original RoBERTa results (88.7% on average). The authors conclude that BERT-like models achieve high performance on downstream tasks almost entirely by exploiting higher-order word co-occurrence statistics.

Another aspect of common sense knowledge is time. When a PLM is applied to new documents it often does not know the meaning of new named entities and concepts [30]. Often, the model cannot infer the time and region of a document and may not be able to correctly combine facts from documents that originate from different time periods or geographical regions. A benchmark for assessing the temporal reasoning capabilities of PLMs in dialogs shows that BERT and T5 have major deficits on this task [47]. In summary it can be expected that the new Retro (Sect. 6.2.3) or WebGPT (Sect. 6.2.3) models, which perform retrieval during language generation, will considerably mitigate the problems discussed in this section.

To be able to check a multitude of different knowledge types in a standardized way large benchmarks like BIG-bench have been developed (Sect. 4.1.4). It comprises benchmarks on common sense, emotional intelligence, ethics, fact checking, general knowledge, humanities, mathematics, medicine, reading comprehension, science and social sciences. Figure 4.1 shows the performance of the Gopher model with 280B parameters on these benchmark groups. On most groups more than 50% accuracy was achieved. The PaLM model with 540B parameters was able to improve these performance figures. On about 2/3 of these tasks PaLM using 5-shot prompts achieves a better performance than average humans [9, p. 17]. This indicates that PaLM has a much better common sense knowledge than earlier models. Nevertheless, PaLM surpasses the performance of human experts only in a small fraction of cases suggesting further headroom for improvement.

An interesting idea is to use large pre-trained multilingual language models as a multilingual knowledge base [25]. The authors evaluate this for mBERT (Sect. 3.3.1), a standard BERT model, which has been pre-trained with the MLM loss on non-parallel Wikipedia texts from 104 languages. The authors find that correct entities can be retrieved for many languages. However, there is a clear performance gap between English and, e.g., Japanese and Thai. This suggests that mBERT does not store knowledge about entities in a language-independent way. It would be revealing if these experiments could be repeated with up-to-date language models like PaLM.

4.2.3 Logical Consistency

A set of statements is logically inconsistent if they cannot all be true at the same time. As an example consider the statements "John is Tom's father. Tom is the daughter of John." Sometimes, BERT is unable to reason, i.e. logically connect different pieces of knowledge. It reproduces, for instance, the relations that persons can walk into houses, and that houses are big, but it cannot infer that houses are bigger than persons [15, 52]. However, semantic knowledge problems tend to be smaller for models with more parameters.

Richardson et al. [52] formulated nine different types of simple sentence pairs containing e.g. negations, quantifiers, comparatives, etc. These sentences express logical entailment, contradiction or neutrality. In addition, they also employ chains of hypernomy, e.g. $\mathit{poodle} \leq \mathit{dog} \leq \mathit{mammal} \leq \mathit{animal}$, and use these relations to generate new sentences expressing the corresponding logical properties. It turns out that BERT fine-tuned with the 'logical tasks' SNLI and MNLI predicts correct statements only with 47.3% accuracy of the cases.

Ribeiro et al. [51] propose to generate a large number of simple examples to test relations by a *CheckList procedure* described in Sect. 4.3.1. It tests, for instance, whether negating a positive sentiment expression leads to a negative sentiment rating. For more than half of the tests with commercial and open-source models they observed failure rates of more than 50%.

Even the larger model GPT-3 is not perfect, e.g. it incorrectly answers some common sense physics questions like *"If I put cheese into the fridge, will it melt?"* [7]. In addition, it has difficulties with logical reasoning, e.g. to determine if one sentence implies another. If a question is not covered in its training material, GPT-3 compiles the most probable answer and sometimes this is wrong, e.g. *"Q: How many eyes does the sun have?" "A: The sun has one eye."* or *"Q: Who was president of the United States in 1600?" "A: Queen Elizabeth I was president of the United States in 1600."* [29]. As another example consider the following input *"You poured yourself a glass of cranberry, but then absentmindedly, you poured about a teaspoon of grape juice into it. It looks OK. You try sniffing it, but you have a bad cold, so you can't smell anything. You are very thirsty. So you …"*. The continuation generated by GPT-3 is *"drink it. You are now dead."*. GPT-3 assumes wrongly that *"grape juice"* is a poison and drinking it will kill you [36].

Improving Logical Consistency

PLMs can improve logical reasoning capabilities if they are trained with appropriately generated textual expressions. By fine-tuning a BERT model with created sentences containing negations, hypernomy, etc., and testing with other generated sentences, Richardson et al. [52] achieve an accuracy of 98%. This approach is similar to the data generation strategy proposed in Sect. 3.6.6.

Similarly, Clark et al. [10] generate datasets of the form (context, statement, answer), where context contains different logical facts and rules, statement is a logical question to prove and answer is either T or F. Facts, rules, and the question statements are then expressed in (synthetic) English. The problems require simultaneous consideration of a number of different statements to reach a conclusion, from depth 0 (simple lookup) to depth 5. During fine-tuning on this data, RoBERTa was trained to answer these questions as true or false. On the test data RoBERTa is able to answer the questions with 99% accuracy. If the facts and rules are paraphrased the accuracy drops to 66%. However, by training on paraphrased rules the model again reaches an accuracy beyond 90%. Clark et al. [10] suggest that by this approach the transformer can be considered as a "soft theorem prover" able to work with statements in language.

It is possible to combine the implicit, pre-trained knowledge of an LM and explicit statements in natural language. Talmor et al. [64] show that models trained with such datasets can perform inferences involving implicit world knowledge and taxonomic knowledge (e.g. the WordNet hierarchy) . In addition, inference patterns provided by examples are used by the model to solve logical problems.

There were a number of prior approaches to combine logical reasoning with neural networks. If a neural network provides probabilities for logical facts, then we can use a probabilistic reasoning system to enforce additional constraints. Examples are *DeepProblog* [35] that incorporates Deep Learning by means of neural predicates, i.e. statements whose probability is determined by a neural network. An alternative is *probabilistic soft logic* (*PSL*) [28], which allows first order probabilistic reasoning in relational domains. However, PLMs do not directly provide probabilities for facts. There have been approaches to translate natural language sentences to logical statements and apply logical reasoning. However, this "semantic parsing" [24] was not very successful.

A number of researchers have developed methods for neural theorem proving. This work combines symbolic and neural methods to reason about results derived from language. Examples are e.g. Minervini et al. [39], which jointly embed logical predicates and text in a shared space by using an end-to-end differentiable model, or Weber et al. [70] which combine a Prolog prover with a language model to apply rule-based reasoning to natural language. The **DeepCTRL** approach [57] integrates rules with Deep Learning. It has a rule encoder which allows to control the strengths of the rules at inference. It can be applied to domains like healthcare, physical models or accounting, where obeying rules is essential.

A simple but effective way to improve logical consistency is to increase the number of model parameters creating a Foundation Model. A large fraction of the tasks in the BIG-bench benchmark [1, 60] is devoted to checking logical consistency, e.g. the benchmark groups with analogical reasoning and logical reasoning. *Gopher* (Sect. 3.1.2) is a language model with 280B parameters. It was applied to about 150 benchmarks, among them 19 logical reasoning tasks. In all but 4 benchmarks it could increase SOTA indicating that larger PLMs have better reasoning capabilities. Nevertheless, the average accuracy was only about 50%. It was not yet evaluated whether the recent *Retro* (Sect. 6.2.3) language model with retrieval of additional text documents is able to improve these results.

PaLM (Sect. 3.1.2) is an even larger language model with 540B parameters. On the SuperGLUE logical tasks CB, COPA, RTE, it can drastically increase the scores compared to BERT, e.g. for COPA from 70.6 to 99.2 (Table 4.2). It has been evaluated on hundreds of benchmarks including those used for Gopher. It uses a new prompt technique to pose logical questions, where examples are presented to the system together with *thought chains* partitioning a reasoning task into smaller problems (Sect. 3.6.4). Two examples are shown in Fig. 2.21. Note that k-shot reasoning only requires a single sequence of k thought chain prompts to be provided for the training examples. The model then generates a thought chain for each test example. This can be used for error analysis and explaining the model behavior.

Using this technique, PaLM is able to match or surpass the performance level of an average human asked to solve the task. As an example consider the *StrategyQA benchmark* [16], which contains questions like *"Did Aristotle use a laptop?"*. For this question the model has to collect facts on the lifespan of Aristotle and the year, when the first laptop was invented to arrive at the answer *"No"*. Without thought chain prompts PaLM reached 69%, while the use of thought chain prompts could improve the prior SOTA from 70% to 73.9%. As a comparison, average humans achieve 62.9%, while expert humans have an accuracy of 90%.

There are other ways to improve learning with such intermediate outputs. Wang et al. [69] sample multiple chains of thought exploiting the diversity of reasoning paths and then return the most consistent final answer in the set. Since it is expensive to obtain chains-of-thought for a large number of examples, Zelikman et al. [71] generate explanations for a large dataset by bootstrapping a model in the few-shot setting and only retaining chains-of-thought that lead to correct answers.

4.2.4 Summary

Pre-trained PLMs have a huge number of parameters and are able to represent an enormous amount of syntactic and factual knowledge. This knowledge can be elicited by probing classifiers, the prediction of masked words, by generating answers to inputs, or by solving benchmark tasks.

As far as syntactic knowledge is concerned, Foundation Models like GPT-3 produce almost error-free text and ‘know’ a lot about syntactic rules. One problem is to adequately reflect the effect of negations.

Even smaller models like BERT are capable of producing a lot of commonsense knowledge. Here, the effect of substituting names or using paraphrases is problematic. Larger language models like GPT-3 are more robust, and the recently proposed language models with retrieval (WebGPT, Retro) are able to include relevant external documents for the current task. This information can reduce errors considerably. However, there is no comprehensive evaluation yet. One problem is the correct temporal and spatial location of information. Here, smaller models like BERT and T5 have large deficits. Foundation Models meanwhile surpass the average human score in 2/3 of the BIG-bench tests on common sense knowledge. They can even be used as a multilingual knowledge base, since models like PaLM cover many languages.

Logical consistency of inferences is a problem, and the PLMs often associate answers that are plausible but wrong. The models are only able to make logical inferences for relationships mentioned in the training text, and they are often incapable of making abstractions and generalizing an observed relationship to other objects or entities of the same type. Logical consistency can be improved by generating additional training texts containing assumptions and valid logical consequences resulting from them. The direct inclusion of logical reasoning systems in Foundation Models was not very successful. The PaLM language model with 540B parameters achieved a fundamental improvement of the accuracy of logical reasoning through the use of thought chain prompts. Here in a few-shot prompt a logical derivation is broken down into smaller logical substeps . At present, it is not clear, to what extent language models with retrieval can reduce the still existing deficits in logical reasoning.

4.3 Transferability and Reproducibility of Benchmarks

In this section, we consider whether benchmarks actually evaluate the properties they are supposed to test. We also discuss the extent to which they are reproducible.

4.3.1 Transferability of Benchmark Results

On a number of benchmarks, the performance of human annotators is exceeded by Foundation Models. This is an indication that the model has learned valuable contents about language. However, Ribeiro et al. [51] argue that this can be misleading, because the test sets often do not cover the right content. While performance on held-out test data is a useful measure, these datasets are often not comprehensive. Hence, there is the danger of overestimating the usability of the model in real applications.

Benchmarks May Not Test All Aspects

On the MRPC task of the GLUE benchmark for detecting paraphrases RoBERTa, $\text{BERT}_{\text{LARGE}}$, and humans have F1 scores of 90.9% [34], 89.3% [42] and 86.3% respectively. Therefore, both models perform better than humans. To test whether the models respect basic logical relationships, Ribeiro et al. [51] propose to generate a large number of simple examples using a **CheckList procedure**. This approach is similar to testing software by systematically generating a large variety of inputs in unit tests.

The following scheme, for instance, can be used to check the effect of a negation in a sentiment classification task *"I <negation> <positive_verb> the <thing>"*. It generates sentences like *"I didn't love the food"* or *"I don't enjoy sailing"*. The authors formulate *minimum functionality tests*, which are useful to check if the model actually detected the reason of an outcome or used some unjustified association. In addition, they utilize *invariance tests* to find out, if neutral perturbations or paraphrases change the result. Finally, they create *directional expectation tests*, where a modification is known to change the result in an expected way.

For MPRC it turned out that the failure rates of RoBERTa and BERT on these 23 test templates are larger than 50% for 11 and 14 of the templates respectively. Therefore, the "superhuman" performance of the two models should be taken with a grain of salt.

The authors also tested five current PLMs: $\text{BERT}_{\text{BASE}}$, $\text{RoBERTa}_{\text{BASE}}$, Microsoft's Text Analytics, Google Cloud's Natural Language, and Amazon's Comprehend. They report the results of 17 tests for sentiment classification, where most problems occurred with negations. For instance, the following example *"I thought the plane would be awful, but it wasn't."* was misclassified by most models.

Also very difficult is the detection of paraphrases with 23 tests templates. Here RoBERTa had for 11 and BERT for 14 of the test templates a failure rate of more than 50%. A similar failure rate was observed for reading comprehension when test cases were generated with logical templates. These results indicate that the examples in the original test sets of the benchmarks are too easy.

To increase robustness of PLMs it is possible to generate adversarial examples [8, 65]. The authors discuss methods that augment training data with adversarial examples as well as methods that produce certificates of robustness. They also investigate methods to avoid spurious correlations, i.e. predictive patterns that work well on a specific dataset but do not hold in general.

Talman et al. [63] checked, if the results for benchmarks may be transferred to similar datasets. They trained six PLMs on different benchmarks for *natural language inference* (*NLI*) containing sentence pairs manually labeled with the labels entailment, contradiction, and neutral. While six models perform well when the test set matches the training set, accuracy is significantly lower when a test set from another benchmark is used. BERT$_{\text{BASE}}$, for instance, yields a test accuracy of 90.4% for SNLI, which drops on average 21.2% for the test sets of the other benchmarks. The reason behind this drop is a slightly different definition of the task as well as small differences in the documents domains. Obviously, it cannot be expected that the performance of PLMs can simply be transferred to new data.

Logical Reasoning by Correlation

The *Winograd schema challenge* (WNLI) was developed by Levesque et al. [32] and is part of the GLUE benchmark collection. The test consists of a pair of sentences differing by exactly one word, each followed by a question [41], e.g.

- The sports car passed the mail truck because it was going faster. Question: Which was going faster, the sports car or the mail truck?
- The sports car passed the mail truck because it was going slower. Question: Which was going slower, the sports car or the mail truck?

In this pair of sentences, the difference of one word changes which thing or person a pronoun refers to. Answering these questions correctly seems to require common sense reasoning and world knowledge. In addition, the authors have designed the questions to be “Google-proof”: The system should not be able to use a web search (or anything similar) to answer the questions. GPT-3 reaches a value of 88.6% using few-shot prompts without fine-tuning [7] and DeBERTa managed an accuracy of 95.6% after fine-tuning [19]. This accuracy roughly equals human performance.

As Mitchell [41] argues, this does not necessarily mean that neural network language models have attained human-like understanding. For a number of question pairs it seems possible to answer the question by some sort of correlation instead of actual world knowledge. If pre-trained on a large corpus the model will learn the high correlation between *“sports car”* and *“fast”* and between *“mail truck”* and *“slow”* for the above example. Therefore, it can give the correct answer on the coreference of *"it"* based on those correlations alone and not by recourse to any understanding. It turns out that many of the Winograd schema challenge question follow this pattern. A similar argument states [6, 37] that a model might heuristically accept a hypothesis by assuming that the premise entails any hypothesis whose words all appear in the premise. This means that the model can give the right answer without 'understanding' the situation in question.

To reduce the deficits of the Winograd schema challenge a much larger *Winogrande* benchmark [55] was created using crowdsourcing. The researchers discarded sentences which could be answered by exploiting intuition and correlation. They used the embeddings created by RoBERTa (Sect. 3.1.1) to determine if these embeddings strongly indicated the correct response option. In this case they discarded the question pair and finally ended up with 44k sentences. An example for a question pair without correlation problems is:

- The trophy doesn't fit into the brown suitcase because it's too large. (it: trophy)
- The trophy doesn't fit into the brown suitcase because it's too small. (it: suitcase)

While humans reach an accuracy of 94%, the best PLMs, standard models like RoBERTa only reached 79.1% accuracy. Recently, *T5-XXL* achieved an accuracy of about 91% [43] and the *ST-MoE-32B* mixture-of-experts model [73] with 269B parameters (Sect. 3.5.2) obtained 96.1%, drastically reducing the errors. It appears that in most cases the latter models are able to perform 'reasoning' without simply correlating statements.

4.3.2 Reproducibility of Published Results in Natural Language Processing

Many publications in NLP claim that their model achieves SOTA for some benchmark. Examples are the GLUE benchmark [67] for language understanding and the SQuAD data [50] for reading comprehension. There are two main problems with this approach. First it is difficult to assess, if the results are reproducible and significant. As Crane [11] demonstrates, there are usually a number of unreported conditions that affect the reproducibility of the result. An example is the random initialization of the network parameters. The resulting variance is often larger than the reported improvement in SOTA scores. However, the variance resulting from these phenomena is usually not reported. Other effects are the underlying programming frameworks and libraries, which change over time. Often the hyperparameters and the details of preprocessing and model configuration are not communicated.

To document the model architecture, the training and evaluation process of a model, Mitchell et al. [40] proposed the description of relevant facts and hyperparameters in a **model card**. After a short high-level description of the model and its purpose the model card should contain nine different sections [40]:

1. Basic information about the model,
2. Intended uses and scope limitations,

3. Model performance across a variety of relevant factors,
4. Performance metrics,
5. Evaluation data,
6. Training data,
7. Evaluation results according to the chosen metrics.
8. Ethical consideration, risks and harms.
9. Caveats and recommendations.

More details are given by huggingface [22]. Even if models still can be published without a model card, the explicit documentation of the model can only benefit future users. Therefore, model cards should be provided if possible. For most recent models, a model card is provided even if the model is not open-source.

A survey on *reproducibility in NLP* is given by Belz et al. [4]. They note that the performance results often depend on seemingly small differences in model parameters and settings, for example minimum counts for rare word or the normalization of writing. The authors state in their study on repeated experiments that only 14% of the 513 reported scores were the same. An annoying fraction of 59% of the scores were worse than the published numbers. Therefore, the experimental results published in papers should be treated with caution.

Another issue is the question of what causes an increase in performance. As we have discussed above, a growth in the number of parameters and in the computing effort regularly leads to better results for PLMs (Sect. 3.5.1). As a consequence, it is often not clear, whether the architectural changes to a model yield the improved performance or just the number of additional parameters or the larger training set [53].

Obviously a first place in a leaderboard can be achieved with a larger model and more computing effort. This, however, "is not research news" according to Rogers [53]. In addition, these results are often not reproducible: Who can afford to retrain GPT-3 for 4.6 million dollars. As a consequence, the development of smaller but more innovative models is less rewarding, as it is difficult to beat the bigger model. Only if the authors of a new model can show that their architecture is better than the previous SOTA model with the same number of parameters and compute budget, they can claim to have made a valuable contribution. Rogers [53] proposes to provide a standard training corpus for a leaderboard and limit the amount of computation effort to that of a strong baseline model. As an alternative the size of the training data and the computational effort should be reported and taken into account in the final score.

Available Implementations

- There are model codes and trained models for RoBERTa and ELECTRA at Hugging Face https://huggingface.co/transformers/.
- The code for DeBERTa is available at https://github.com/microsoft/DeBERTa and Hugging Face.
- The Checklist code is at https://github.com/marcotcr/checklist.

4.3.3 Summary

The transferability of benchmark results to real applications is not always granted. Even if a PLM is better than humans at logical reasoning on the test set, it may not be able to classify generated logical reasoning chains correctly. This indicates that the test set does not cover the full spectrum of possible examples. It is common for performance to be lower on related benchmarks because the domain or the definition of the task may deviate.

There are cases where a logical conclusion is obtained not by logical deduction, but by a simple correlation of antecedent and consequent. This could be demonstrated for the Winograd task of the GLUE benchmark. To avoid this type of 'reasoning' a new variant task called Winogrande was developed where correlations are unrelated to the reasoning task. Meanwhile, a Foundation Model with 269B parameters was also able to solve this task better than humans.

A survey on the reproducibility of results in NLP demonstrated that the published performance often depends on a number of unreported effects, such as random number initialization. Often the variability of such effects is larger than the reported improvement. Therefore, it is necessary to report the variance caused by these effects. In addition, the details of the model architecture, its training and evaluation should be documented in a model card. In about 500 repeated experiments, an irritating rate of about 60% of final scores were lower than reported. Note that improvements due to more parameters, more training data, or higher computational effort are not indicative of a better model architecture.

References

1. S. Aarohi and R. Abhinav. *BIG-bench* · Google, June 20, 2022. URL: https://github.com/google/BIG-bench/blob/936c4a5876646966344349b28ae187c556938ec4/docs/paper/BIGbench.pdf (visited on 06/20/2022).
2. M. Aßenmacher and C. Heumann. "On the Comparability of Pre-Trained Language Models". 2020. arXiv: 2001.00781.
3. S. Balasubramanian, N. Jain, G. Jindal, A. Awasthi, and S. Sarawagi. "What's in a Name? Are BERT Named Entity Representations Just as Good for Any Other Name?" 2020. arXiv: 2007.06897.
4. A. Belz, S. Agarwal, A. Shimorina, and E. Reiter. "A Systematic Review of Reproducibility Research in Natural Language Processing". Mar. 21, 2021. arXiv: 2103.07929 [cs].
5. S. R. Bowman and G. E. Dahl. "What Will It Take to Fix Benchmarking in Natural Language Understanding?" 2021. arXiv: 2104.02145.
6. R. Branco, A. Branco, J. António Rodrigues, and J. R. Silva. "Shortcutted Commonsense: Data Spuriousness in Deep Learning of Commonsense Reasoning". In: *Proc. 2021 Conf. Empir. Methods Nat. Lang. Process.* EMNLP 2021. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 1504–1521. https://doi.org/10.18653/v1/2021.emnlp-main.113.
7. T. B. Brown et al. "Language Models Are Few-Shot Learners". 2020. arXiv: 2005.14165.
8. K.-W. Chang, H. He, R. Jia, and S. Singh. "Robustness and Adversarial Examples in Natural Language Processing". In: *Proc. 2021 Conf. Empir. Methods Nat. Lang. Process. Tutor.*

Abstr. Punta Cana, Dominican Republic & Online: Association for Computational Linguistics, Nov. 2021, pp. 22–26. URL: https://aclanthology.org/2021.emnlp-tutorials.5 (visited on 11/24/2021).

9. A. Chowdhery et al. “PaLM: Scaling Language Modeling with Pathways”. Apr. 5, 2022. arXiv: 2204.02311 [cs].
10. P. Clark, O. Tafjord, and K. Richardson. “Transformers as Soft Reasoners over Language”. 2020. arXiv: 2002.05867.
11. M. Crane. “Questionable Answers in Question Answering Research: Reproducibility and Variability of Published Results”. In: *Trans. Assoc. Comput. Linguist.* 6 (2018), pp. 241–252.
12. Y. Elazar, N. Kassner, S. Ravfogel, A. Ravichander, E. Hovy, H. Schütze, and Y. Goldberg. “Measuring and Improving Consistency in Pretrained Language Models”. May 29, 2021. arXiv: 2102.01017.
13. Y. Elazar, S. Ravfogel, A. Jacovi, and Y. Goldberg. “Amnesic Probing: Behavioral Explanation with Amnesic Counterfactuals”. In: *Trans. Assoc. Comput. Linguist.* 9 (2021), pp. 160–175.
14. A. Ettinger. “What BERT Is Not: Lessons from a New Suite of Psycholinguistic Diagnostics for Language Models”. In: *Trans. Assoc. Comput. Linguist.* 8 (2020), pp. 34–48.
15. M. Forbes, A. Holtzman, and Y. Choi. “Do Neural Language Representations Learn Physical Commonsense?” 2019. arXiv: 1908.02899.
16. M. Geva, D. Khashabi, E. Segal, T. Khot, D. Roth, and J. Berant. “Did Aristotle Use a Laptop? A Question Answering Benchmark with Implicit Reasoning Strategies”. In: *Trans. Assoc. Comput. Linguist.* 9 (2021), pp. 346–361.
17. Y. Goldberg. “Assessing BERT’s Syntactic Abilities”. 2019. arXiv: 1901.05287.
18. P. He, J. Gao, and W. Chen. “Debertav3: Improving Deberta Using Electra-Style Pre- Training with Gradient-Disentangled Embedding Sharing”. 2021. arXiv: 2111.09543.
19. P. He, X. Liu, J. Gao, and W. Chen. “DeBERTa: Decoding-enhanced BERT with Disentangled Attention”. Jan. 11, 2021. arXiv: 2006.03654.
20. D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt. “Measuring Massive Multitask Language Understanding”. 2020. arXiv: 2009.03300.
21. J. Hewitt and C. D. Manning. “A Structural Probe for Finding Syntax in Word Representations”. In: *Proc. 2019 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. Vol. 1 Long Short Pap*. 2019, pp. 4129–4138.
22. huggingface. *Building a Model Card - Hugging Face Course*. 2022. URL: https://huggingface.co/course/chapter4/4 (visited on 08/07/2022).
23. G. Jawahar, B. Sagot, and D. Seddah. “What Does BERT Learn about the Structure of Language?” In: 2019.
24. A. Kamath and R. Das. “A Survey on Semantic Parsing”. 2018. arXiv: 1812.00978.
25. N. Kassner, P. Dufter, and H. Schütze. “Multilingual LAMA: Investigating Knowledge in Multilingual Pretrained Language Models”. 2021. arXiv: 2102.00894.
26. N. Kassner and H. Schütze. “Negated and Misprimed Probes for Pretrained Language Models: Birds Can Talk, but Cannot Fly”. 2019. arXiv: 1911.03343.
27. T. Kim, J. Choi, D. Edmiston, and S.-g. Lee. “Are Pre-Trained Language Models Aware of Phrases? Simple but Strong Baselines for Grammar Induction”. 2020. arXiv: 2002.00737.
28. B. Kirsch, S. Giesselbach, T. Schmude, M. Völkening, F. Rostalski, and S. Rüping. “Using Probabilistic Soft Logic to Improve Information Extraction in the Legal Domain”. In: (2020).
29. K. Lacker. *Giving GPT-3 a Turing Test*. July 6, 2020. URL: https://lacker.io/ai/2020/07/06/giving-gpt-3-a-turing-test.html (visited on 12/03/2020).
30. A. Lazaridou et al. “Mind the Gap: Assessing Temporal Generalization in Neural Language Models”. In: *Adv. Neural Inf. Process. Syst.* 34 (2021).
31. B. Lemoine. *Is LaMDA Sentient? – An Interview. Medium*. June 11, 2022. URL: https://cajundiscordian.medium.com/is-lamda-sentient-an-interview-ea64d916d917 (visited on 06/24/2022).
32. H. Levesque, E. Davis, and L. Morgenstern. “The Winograd Schema Challenge”. In: *Thirteen. Int. Conf. Princ. Knowl. Represent. Reason*. 2012.

33. N. F. Liu, M. Gardner, Y. Belinkov, M. E. Peters, and N. A. Smith. “Linguistic Knowledge and Transferability of Contextual Representations”. 2019. arXiv: 1903.08855.
34. Y. Liu et al. “Roberta: A Robustly Optimized Bert Pretraining Approach”. 2019. arXiv: 1907.11692.
35. R. Manhaeve, S. Dumancic, A. Kimmig, T. Demeester, and L. De Raedt. “Deepproblog: Neural Probabilistic Logic Programming”. In: *Adv. Neural Inf. Process. Syst.* 2018, pp. 3749–3759.
36. G. Marcus and E. Davis. *GPT-3: Commonsense Reasoning*. Aug. 1, 2020. URL: https://cs.nyu.edu/faculty/davise/papers/GPT3CompleteTests.html (visited on 02/15/2021).
37. R. T. McCoy, E. Pavlick, and T. Linzen. “Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference”. June 24, 2019. arXiv: 1902.01007 [cs].
38. S. Merity, C. Xiong, J. Bradbury, and R. Socher. “Pointer Sentinel Mixture Models”. 2016. arXiv: 1609.07843.
39. P. Minervini, M. Bošnjak, T. Rocktäschel, S. Riedel, and E. Grefenstette. “Differentiable Reasoning on Large Knowledge Bases and Natural Language”. In: *Proc. AAAI Conf. Artif. Intell*. Vol. 34. 04. 2020, pp. 5182–5190.
40. M. Mitchell et al. “Model Cards for Model Reporting”. In: *Proc. Conf. Fairness Account. Transpar*. Jan. 29, 2019, pp. 220–229. https://doi.org/10.1145/3287560.3287596. arXiv: 1810.03993 [cs].
41. M. Mitchell. *What Does It Mean for AI to Understand?* Quanta Magazine. Dec. 16, 2021. URL: https://www.quantamagazine.org/what-does-it-mean-for-ai-to-understand-20211216/ (visited on 01/03/2022).
42. N. Nangia and S. R. Bowman. “Human vs. Muppet: A Conservative Estimate of Human Performance on the GLUE Benchmark”. June 1, 2019. arXiv: 1905.10425 [cs].
43. openai. *Submissions – WinoGrande: Adversarial Winograd Schema Challenge at Scale Leaderboard*. Jan. 5, 2022. URL: https://leaderboard.allenai.org/winogrande/submissions/public (visited on 01/05/2022).
44. D. Paperno et al. “The LAMBADA Dataset: Word Prediction Requiring a Broad Discourse Context”. June 20, 2016. arXiv: 1606.06031 [cs].
45. Paperswithcode. Browse State-of-the-Art in AI. 2019. URL: https://paperswithcode.com/sota.
46. F. Petroni, T. Rocktäschel, P. Lewis, A. Bakhtin, Y. Wu, A. H. Miller, and S. Riedel. “Language Models as Knowledge Bases?” 2019. arXiv: 1909.01066.
47. L. Qin, A. Gupta, S. Upadhyay, L. He, Y. Choi, and M. Faruqui. “TIMEDIAL: Temporal Commonsense Reasoning in Dialog”. In: *Proc. 59th Annu. Meet. Assoc. Comput. Linguist. 11th Int. Jt. Conf. Nat. Lang. Process. Vol. 1 Long Pap.* ACL-IJCNLP 2021. Online: Association for Computational Linguistics, Aug. 2021, pp. 7066–7076. https://doi.org/10.18653/v1/2021.acl-long.549.
48. A. Radford, J. Wu, D. Amodei, D. Amodei, J. Clark, M. Brundage, and I. Sutskever. “Better Language Models and Their Implications”. In: *OpenAI Blog* (2019). URL: https://openai.%20com/blog/better-language-models.
49. J. W. Rae et al. “Scaling Language Models: Methods, Analysis & Insights from Training Gopher”. In: *ArXiv Prepr. ArXiv211211446* (Dec. 8, 2021), p. 118.
50. P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. “Squad: 100,000+ Questions for Machine Comprehension of Text”. 2016. arXiv: 1606.05250.
51. M. T. Ribeiro, T. Wu, C. Guestrin, and S. Singh. “Beyond Accuracy: Behavioral Testing of NLP Models with CheckList”. 2020. arXiv: 2005.04118.
52. K. Richardson, H. Hu, L. Moss, and A. Sabharwal. “Probing Natural Language Inference Models through Semantic Fragments”. In: *Proc. AAAI Conf. Artif. Intell*. Vol. 34. 05. 2020, pp. 8713–8721.
53. A. Rogers. *How the Transformers Broke NLP Leaderboards*. Hacking semantics. June 30, 2019. URL: https://hackingsemantics.xyz/2019/leaderboards/ (visited on 12/15/2021).
54. A. Rogers, O. Kovaleva, and A. Rumshisky. “A Primer in {Bertology}: What We Know about How {BERT} Works”. In: *Trans. Assoc. Comput. Linguist*. 8 (2021), pp. 842–866.
55. K. Sakaguchi, R. L. Bras, C. Bhagavatula, and Y. Choi. “WinoGrande: An Adversarial Winograd Schema Challenge at Scale”. In: *Commun. ACM* 64.9 (2021), pp. 99–106.

56. F. Schmidt and T. Hofmann. "BERT as a Teacher: Contextual Embeddings for Sequence- Level Reward". 2020. arXiv: 2003.02738.
57. S. Seo, S. Arik, J. Yoon, X. Zhang, K. Sohn, and T. Pfister. "Controlling Neural Networks with Rule Representations". In: *Adv. Neural Inf. Process. Syst.* 34 (2021).
58. M. Shoeybi, M. Patwary, R. Puri, P. LeGresley, J. Casper, and B. Catanzaro. "Megatron-Lm: Training Multi-Billion Parameter Language Models Using Model Parallelism". In: *arXiv* (2019), arXiv-1909.
59. K. Sinha, R. Jia, D. Hupkes, J. Pineau, A. Williams, and D. Kiela. "Masked Language Modeling and the Distributional Hypothesis: Order Word Matters Pre-training for Little". Apr. 14, 2021. arXiv: 2104.06644.
60. J. Sohl-Dickstein. BIG-bench. Google, Dec. 16, 2021. URL: https://github.com/google/BIGbench (visited on 12/16/2021).
61. M. Sparkes. *Google Wants to Challenge AI with 200 Tasks to Replace the Turing Test*. New Scientist. June 14, 2022. URL: https://www.newscientist.com/article/2323685-google-wantsto-challenge-ai-with-200-tasks-to-replace-the-turing-test/ (visited on 06/26/2022).
62. S. Storks, Q. Gao, and J. Y. Chai. "Commonsense Reasoning for Natural Language Understanding: A Survey of Benchmarks, Resources, and Approaches". 2019. arXiv: 1904.01172.
63. A. Talman and S. Chatzikyriakidis. "Testing the Generalization Power of Neural Network Models Across NLI Benchmarks". May 31, 2019. arXiv: 1810.09774.
64. A. Talmor, O. Tafjord, P. Clark, Y. Goldberg, and J. Berant. "Teaching Pre-Trained Models to Systematically Reason over Implicit Knowledge". 2020. arXiv: 2006.06609.
65. TrustworthyAI, director. *CVPR 2021 Tutorial on "Practical Adversarial Robustness in Deep Learning: Problems and Solutions"*. June 28, 2021. URL: https://www.youtube.com/watch?v=ZmkU1YO4X7U (visited on 02/26/2022).
66. Wang. *SuperGLUE Benchmark*. SuperGLUE Benchmark. 2021. URL: https://super.gluebenchmark.com/ (visited on 02/23/2021).
67. A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman. "Glue: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding". Feb. 22, 2019. arXiv: 1804.07461.
68. A. Wang et al. "Superglue: A Stickier Benchmark for General-Purpose Language Understanding Systems". In: *Adv. Neural Inf. Process. Syst.* 2019, pp. 3266–3280.
69. X. Wang et al. "Self-Consistency Improves Chain of Thought Reasoning in Language Models". Apr. 6, 2022. arXiv: 2203.11171 [cs].
70. L. Weber, P. Minervini, J. Münchmeyer, U. Leser, and T. Rocktäschel. "Nlprolog: Reasoning with Weak Unification for Question Answering in Natural Language". 2019. arXiv: 1906.06187.
71. E. Zelikman, Y. Wu, and N. D. Goodman. "STaR: Bootstrapping Reasoning With Reasoning". Mar. 27, 2022. arXiv: 2203.14465 [cs].
72. X. Zhou, Y. Zhang, L. Cui, and D. Huang. "Evaluating Commonsense in Pre-Trained Language Models." In: *AAAI*. 2020, pp. 9733–9740.
73. B. Zoph et al. "Designing Effective Sparse Expert Models". 2022. arXiv: 2202.08906.