

# A Review on Language Models as Knowledge Bases

Badr AlKhamissi\*

Millicent Li\*

Asli Celikyilmaz†

Mona Diab†

Marjan Ghazvininejad†

Meta AI

{bkhmsi, millicentli, aslic, mdiab, ghazvini}@fb.com

## Abstract

Recently, there has been a surge of interest in the NLP community on the use of pretrained Language Models (LMs) as Knowledge Bases (KBs). Researchers have shown that LMs trained on a sufficiently large (web) corpus will encode a significant amount of knowledge implicitly in its parameters. The resulting LM can be probed for different kinds of knowledge and thus acting as a KB. This has a major advantage over traditional KBs in that this method requires no human supervision. In this paper, we present a set of aspects that we deem an LM should have to fully act as a KB, and review the recent literature with respect to those aspects.<sup>1</sup>

## 1 Introduction

The impact of Pretrained Language Models (LMs) on Natural Language Processing (NLP) research can be described as nothing short of transformative. It has moved the field from *feature engineering* (Och et al., 2004; Zhang and Nivre, 2011) and *architecture engineering* (Chung et al., 2014; Kim, 2014; Bahdanau et al., 2015; Vaswani et al., 2017) to the *pre-train and fine-tune* paradigm (Radford and Narasimhan, 2018; Dong et al., 2019; Lewis et al., 2021), and lately the *pre-train, prompt, and predict* paradigm (Liu et al., 2021d). LMs pre-trained on a large corpus of web data have been shown to contain different kinds of knowledge implicitly in their parameters without the need for any human supervision. This includes: world knowledge (Petroni et al., 2019; Rogers et al., 2020), relational knowledge (Safavi and Koutra, 2021), commonsense knowledge (Da et al., 2021), linguistic knowledge (Peters et al., 2018; Goldberg, 2019; Tenney et al., 2019b), actionable knowledge

## LMs-as-KBs

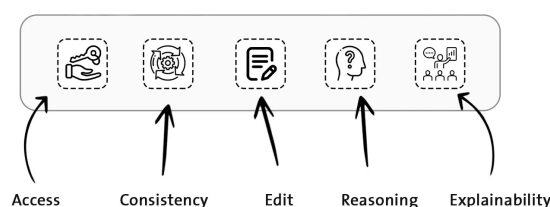


Figure 1: Aspects of LMs-as-KBs as presented in this work

(Huang et al., 2022) and more. This access to knowledge is crucial for LMs to achieve state-of-the-art results on various downstream tasks. However, as is the case with most neural systems, knowledge in LMs is encoded in a diffused manner, making it generally difficult to interpret and hard to update.

Despite these recent breakthroughs, we often do not have full control over the behavior of LMs. As a result, utilizing these models in real-world scenarios is often unsuccessful. On the other hand, Knowledge Bases (KB) are easier to control. Here, KBs refers to a data structure that stores relational information in the form of triples connecting two triplets of entities by symbolic relations (e.g.  $\langle \text{Cairo}, \text{CapitalOf}, \text{Egypt} \rangle$ ). They often follow rule-based heuristics, rendering them predictable, in addition to possessing large knowledge coverage, which primes them for use in real-world systems. These models are often used as chatbots and virtual assistants, where controlled generation of output is necessary to ensure appropriate responses (Chen et al., 2017). Therefore, KBs are a natural solution to access specific gold-standard

\* Equal Contribution

† Equal Supervision

<sup>1</sup>For an updated paper-list please check our website: <https://bkhmsi.github.io/lms-as-kbs/>.

relational information. They are repositories of knowledge, for both structured and unstructured data, and can be seamlessly queried and updated by an end user.

Since KBs can access and update relational knowledge easier than LMs can, one question naturally arises: how can we control the repository of knowledge stored implicitly in the weights of a LM as similarly as KBs can? This question was first introduced in seminal work by [Petroni et al. \(2019\)](#) and has since ignited the interest of the community with the goal of instilling LMs with desirable properties of KBs.

Several works have already approached improving LMs through the lens of KBs: [Petroni et al. \(2019\)](#); [Dhingra et al. \(2021\)](#); [Wang et al. \(2020\)](#); [Heinzerling and Inui \(2021\)](#); [Sung et al. \(2021\)](#). Many of these works include updating factoids stored within the parameters of LMs ([De Cao et al., 2021](#); [Mitchell et al., 2021](#); [Hase et al., 2021](#)) to creating new methods for extracting factual knowledge ([Petroni et al., 2019](#)). Despite significant progress towards achieving parity between LMs and KBs, LMs still lack specific aspects that KBs have. For example, given the cloze phrases “Albert Einstein was born in [MASK]” and “The hometown of Albert Einstein is [MASK]”, a user of a KB can map both queries to one triplet  $\langle \text{Albert Einstein, BornIn, X} \rangle$  that the KB readily understands and thus consistently returns the same city. On the other hand, LMs may not be as consistent, yielding potentially different answers for the same underlying factual questions.

In this survey, we propose to consolidate the work on LMs-as-KBs within one cohesive framework, with a focus on aspects related to KBs that we think are useful to integrate into LMs. Few survey papers exist that evaluate LMs in this context. For instance, [Wei et al. \(2021\)](#) evaluate knowledge-enhanced pretrained LMs by delineating the types of knowledge that can be integrated into existing LMs. [Safavi and Koutra \(2021\)](#) divide relevant work according to the level of supervision provided to the LM by a KB. Similarly, [Colon-Hernandez et al. \(2021\)](#) cover the integration of structural knowledge into LMs but forgo implicit knowledge. Our study of LMs-as-KBs from our perspective is unique compared to the focus of existing survey papers. We aim to present the current landscape of LMs-as-KBs research and highlight the existing

challenges that LMs face when applied in practice.

We observe the recent advances in LMs and explore them with respect to aspects that we find necessary for LMs to become as functional and utilizeable as KBs: *access*, *consistency*, *editability*, *reasoning*, and *explainability*. We further highlight where we are now in terms of the LMs-as-KBs framework as well as potential work for the future. Finally, we discuss the remaining challenges in the full adoption of LMs-as-KBs and propose directions for future research. This survey is structured as follows:

1. First, an overview of KBs and LMs and their intersection for LMs-as-KBs
2. Second, the enumeration of the aspects of LMs-as-KBs
3. Third, a brief summarization of each aspect with respect to recent work

We hope that by highlighting the aspects of LMs-as-KBs, we can consolidate knowledge in this ever-growing field of research. We envision that our work can provide a path for those new to the area of research to better improve LMs to be just as good, if not eventually better, than KBs.

## 2 Preliminaries

In this section, we define what LMs and KBs are, characterize the functions and attributes of KBs, and make direct comparisons between both LMs and KBs to highlight the current limitations of LMs in the context of the LMs-as-KBs framework.

### 2.1 Knowledge Bases

KBs usually adhere to a manually engineered schema that dictates the possible set of entities and relations and the interactions between them. Such a rigid schema facilitates different kinds of complex operations over the data (e.g. multi-hop reasoning) and ensures accurate, consistent and explainable answers. Examples of KBs include Wikidata ([Vrandečić and Krötzsch, 2014](#)) and the ATOMIC ([Sap et al., 2019](#)) (See Appendix A.1.1 for further details).

### 2.2 Language Models

In the context of this paper, we use the term “LM” to refer to a deep neural LM that is pre-trained on a large amount of unlabeled text in a self-supervised

setting such as masked language modeling. Examples include transformer-based models (Vaswani et al., 2017) such as BERT (Devlin et al., 2019), GPT-3 (Brown et al., 2020a), BART (Lewis et al., 2020) and T5 (Raffel et al., 2020).

### 2.3 LMs-as-KBs

KBs lack the flexibility that LMs offer in terms of extendability and expressability. KBs also require significant human effort to build and maintain. For example, populating KBs involves extracting huge amounts of relational data from unstructured text by using complex NLP pipelines such as entity linking and coreference resolution (Petroni et al., 2019). In contrast, LMs are able to implicitly capture this information without any supervision.

To better frame the current limitations of LMs with respect to the capabilities of KBs, we observe a total of five aspects of KBs that we would like LMs to excel at in order to be considered a KB. We consider **Access**, **Edit**, **Consistency**, **Reasoning**, and **Explainability and Interpretability**. The latter three are not implicitly done, but are easier to ensure within KBs than LMs.

**Access** KBs are often simple to access via manual querying of terms using specific querying languages. On the other hand, LMs cannot be queried explicitly as the knowledge is not directly encoded in specific locations in the model. Current research has focused on *how* we can query LMs similarly to how we can query KBs, focusing on specific access patterns for different types of knowledge. For example, knowledge encoded in LMs can be accessed through probing via fill-in-the-blank (cloze) prompts (Taylor, 1953) and through traditional downstream finetuning. However, there is still more work needed to improve LMs for efficient and direct access.

**Edit** Since LMs are pre-trained on a certain snapshot of data from a specific time, the knowledge it learns can be outdated (e.g. the population of some country), incorrect (Lin et al., 2021a), biased or toxic (Gehman et al., 2020; Bender et al., 2021). Perhaps more importantly, in the context of privacy, LMs may memorize sensitive information that needs to be removed (Carlini et al., 2021). Further, new information is created all the time. For example, most available LMs today would have never seen information related to the COVID-19 Omicron variant. However, updating a specific fact in a LM is not straightforward, as facts are encoded

in the weights of the model in a distributed fashion, making them inaccessible or uninterpretable (Mitchell et al., 2021). The naive way of re-training the whole model on an updated set is expensive, especially with the ever-increasing size of current LMs (Brown et al., 2020a). There are increasing evidence that show that scaling LMs to larger sizes is not the solution to generating factually correct information (Lazaridou et al., 2021; Gehman et al., 2020; Lin et al., 2021a). As a result, this would also result in catastrophic forgetting (Wallat et al., 2021). Changing a single weight may have a ripple effect that affects a large number of other implicitly memorized facts. Therefore, this task of knowledge editing is of utmost importance, especially when considering LMs in the context of KBs.

**Consistency** Language is multifaceted: the same meaning can be expressed in multiple forms. Structured KBs are built with consistency in mind; several efficient algorithms have been proposed to detect inconsistencies in KBs (Andersen and Pre-tolani, 2001) so such conflicts can be easily resolved, while other work aimed at quantifying the degree of which inconsistency arises in KBs (Picado-Muiño, 2011). Therefore, in the face of such language variability, we should expect LMs to behave consistently under semantically equivalent contexts, even across different languages. Note that consistency does not imply correctness, as a KB can be consistent but store factually incorrect information. Similarly, a LM can have an incorrect belief about a certain fact but that belief is consistent across different queries.

**Reasoning** In a KB, it can be simple to follow the path of reasoning. For instance, given the KB triplet  $\langle \text{Cairo}, \text{CapitalOf}, \text{Egypt} \rangle$ , the KB can sensibly reason that `Cairo` is in `Africa` given that it has another relation explicitly stating that `Egypt` is a part of `Africa`. On the other hand, recent work has shown that LMs can perform different forms of reasoning when finetuned on datasets that elicit reasoning capabilities within LMs (West et al., 2021; Talmor et al., 2020b,a; Hase et al., 2021). Reasoning is not readily obvious and difficult to facilitate, as LMs have been shown to perform poorly on some types of reasoning such as structured reasoning (Kassner and Schütze, 2020).

**Explainability and Interpretability** Given a KB triplet, nodes and links can easily be identified,

and the answer can easily be inferred. However, in LMs, knowledge is rarely understood by simply looking at the output. Moreover, the location of the parameters in which the knowledge comes from is unknown. Under perfect circumstances, one would want LMs to be explainable and interpretable to the end user, especially for stakeholders with no prior understanding of NLP (Lakkaraju et al., 2022). However, current research on LMs is far from achieving this goal, as many of the newer techniques focus on black box rather than white box techniques (Danilevsky et al., 2020). As a result, imbuing explainability and interpretability in LMs is core to improve LMs-as-KBs. We make the distinction, however, that KBs are inherently explainable. On the other hand, interpretability, while not latent in KBs, is still an important aspect to apply to LMs (Lipton, 2018) that we also focus on for the purposes of the survey.

### 3 Accessing Knowledge

KBs can simply be accessed by querying for specific entity nodes given the starting node and edge corresponding to the relation. In contrast, it is more difficult for LMs to access for specific pieces of knowledge. However, previous research has shown that LMs have the ability to be efficient few-shot and zero-shot learners, which shows that knowledge learned during pretraining can be potentially accessible by finetuning or prompting (Brown et al., 2020b). Specifically, finetuning LMs on downstream tasks has shown to be an effective method to tune and elicit specific knowledge for evaluation (Radford and Narasimhan, 2018; Dong et al., 2019). On the other hand, Liu et al. (2021d) has recently shown that prompting shows promise as an effective method to directly access this knowledge without any additional finetuning.

#### 3.1 Finetuning

The knowledge stored in a LM is most often inaccessible to the end-user as compared to a KB. Therefore, to retrieve specific pieces of information from a LM, one prevailing method is to finetune the model on a relevant downstream task (e.g. commonsense question answering) so it can make way for the required knowledge to surface in the output during evaluation. Previous work has shown that most knowledge encoded in a LM are acquired during pretraining, while finetuning just learns an interface to access that acquired knowledge (Da

et al., 2021; Wallat et al., 2021).

#### 3.2 Prompting

The ever-increasing size of LMs make them expensive to finetune and store in practice, despite some architectural innovations that overcomes some of these challenges (Houlsby et al., 2019). On the other hand, prompting has emerged as an alternative method to extract the wanted knowledge directly from a LM (Qin and Eisner, 2021). These prompts are often difficult to craft (Adolphs et al., 2021; Qin and Eisner, 2021; Jiang et al., 2020b), and small changes in prompts can result in large performance differences, which can in turn affect consistency (see Section 4). Research following the *pretrain, prompt, and predict* paradigm (Liu et al., 2021d) utilize prompts to induce more knowledge from LMs (Petroni et al., 2019; Davison et al., 2019; Jiang et al., 2020b). Prompting can be divided up into several categories, including **Discrete Prompts** and **Soft Prompts**.

**Discrete Prompts** Prompting often enables LMs to learn a specific subtask without extensive finetuning. This paradigm gives the model a familiar query format, which in turn leads to better responses. Many papers tackle prompting from the view of cloze-style like in Petroni et al. (2019); Davison et al. (2019); Jiang et al. (2020b); Talmor et al. (2020a); Dhingra et al. (2021); Liu et al. (2021b) in which these works use prompting to extract specific knowledge, such as commonsense (Davison et al., 2019), temporal (Dhingra et al., 2021; Liu et al., 2021b) and factual (Petroni et al., 2019), in a format known as *Discrete Prompts*. Prompting has also been applied to specific domains, such as biomedical, to extract domain-specific knowledge (Sung et al., 2021). It was first introduced in Radford and Narasimhan (2018); Radford et al. (2019) where it was shown that it could achieve decent zero-shot performance by crafting the right prompt. Knowing that prompting worked well, the same knowledge was applied to smaller LMs. Schick and Schütze (2021a,b); Gao et al. (2021) find that prompting smaller LMs improves performance, especially over supervised baselines. Others also take advantage of this improved performance and evaluate other discrete prompting approaches, such as through entailment (Wang et al., 2021a) and label token optimization (Zhang et al., 2021).

The challenge of crafting the ideal prompt for specific task is a non-trivial one. Shin et al. (2020)



Figure 2: Finegrained Aspects of LMs-as-KBs

tackles this by taking a gradient-based search to find the appropriate prompt for a specific task. AUTOPROMPT creates a template automatically to do so. On the other hand, Logan et al. (2021) takes a more manual approach to crafting prompts with comparable accuracy to manual prompts by creating *null prompts*, those of which are task-agnostic and are a simple concatenation of the input and a [MASK] token.

**Soft Prompts** Other approaches fall under the categorization of *Soft Prompts*. Those are prompts that are represented by continuous word vectors, used as input and tuned, while keeping the remainder of the model unchanged. Li and Liang (2021) achieve comparable results on generation while using very few of the model’s actual parameters by proposing prefix-tuning; task-specific vectors that can be tuned. Meanwhile, specific to extracting factual knowledge, Qin and Eisner (2021); Zhong et al. (2021) find that *Soft Prompts* carry an advantage over *Discrete Prompts* since they are more expressive and can represent multiple contexts simultaneously. Zhong et al. (2021) takes a gradient-based approach to soft prompting while Qin and Eisner (2021) improves upon AUTOPROMPT via continuous word vectors. A number of papers show that soft prompting, when optimized for specific tasks such as relation extraction and natural language understanding, can achieve better performance with minimal tuning (Lester et al., 2021; Han et al., 2021).

Other methods have been used without editing model parameters. First introduced with GPT-3 Brown et al. (2020b), **in-context learning**, which adds extra information to the model in the form of in-context demonstrations can improve performance. However, these works are still far from achieving human-level performance (Gao et al., 2021; Liu et al., 2021a).

## 4 Consistency

LMs are shown to suffer from a lack of consistency in their answers (Elazar et al., 2021). For example, they can provide different results when queried for the same fact but under a different wording (i.e. a paraphrase). In this section, we consider consistency in light of three different contexts: **Paraphrasing**, **Commonsense** and **Multilinguality**.

**Paraphrase** Bhagat and Hovy (2013) define the term *quasi-paraphrases* as ‘sentences or phrases

that convey approximately the same meaning using different words’. The word *approximately* is key here, since it does not assume the strict and logical equivalence of paraphrase. This fuzzy definition allows for a broader set of samples to be considered as paraphrases, and is of interest when considering consistency for LMs. Therefore, one method for measuring the consistency of a model is to probe it using paraphrases of the same relation for a specific subject, and test whether it always predicts the same object or not (Kassner and Schütze, 2020; Ettinger, 2020; Elazar et al., 2021). To that end, several benchmarks have been proposed to measure consistency of LMs (Ravichander et al., 2020; Elazar et al., 2021), while other datasets have been adapted for that purpose (Levy et al., 2017; Wang et al., 2021c). For instance, De Cao et al. (2021) and Mitchell et al. (2021) employ back-translation to generate paraphrases (Sennrich et al., 2016; Wieting and Gimpel, 2018) for measuring consistency after editing factual knowledge in LMs. To mitigate this lack of consistency, Elazar et al. (2021) include a new term in their loss function that aims to minimize the Kullback-Leibler (KL) divergence for instance between the output distribution of different paraphrases.

**Commonsense** Consistency, however, is not limited to paraphrasing. Previous work explores the brittleness of LMs and the addition of a negation element (e.g. not) to a probe (Kassner and Schütze, 2020; Ettinger, 2020). Specifically, a LM can maintain two contradictory beliefs in its parameters, such as “Birds can fly” and “Birds cannot fly”, showing that they are insensitive to the contextual impacts of negation. Further, Kassner and Schütze (2020) show an equal effect when mispriming the probe with a misleading distractor (e.g. Talk? Birds can [MASK]). Robust LMs will not only be consistent under different paraphrases and negations, but also under entailment. Hase et al. (2021) measure consistency under entailment, including contrapositives, after updating beliefs of a LM. Specifically, they adapt the LeapOfThought dataset (Talmor et al., 2020b) such that each datapoint has a main fact (e.g. A viper is a vertebrate) that they update and an entailed claim (e.g. A viper has a brain) that they check the truth value of with respect to the model’s updated beliefs. Similar to the efforts done for overcoming lack of consistency under different paraphrases, Hase et al. (2021) add

another loss term to their objective function to minimize the error across entailed data. On the other hand, [Kassner et al. \(2021\)](#) use a feedback mechanism that issues relevant information from a symbolic memory of beliefs as input to the system during test-time in order to improve consistency under entailment.

**Multilingual** Crosslingual LMs that are trained on several languages, such as XGLM ([Lin et al., 2021b](#)), must be consistent across the languages they support. For example, the same probe queried in different languages must provide the same fact. [Liu et al. \(2021c\)](#) design a knowledge-based multilingual LM pretraining framework using Wikidata ([Vrandečić and Krötzsch, 2014](#)) that shows improvement on crosslingual NLP tasks, but neither they nor other work measure consistency under multilinguality.

## 5 Model Editing

There have been a number of works that address the problem of model editing, with strategies ranging from simple finetuning to making use of an external memory for adding or replacing factual knowledge. To that end, [De Cao et al. \(2021\)](#) describe a set of rules that editing methods should conform to: **Generality** implies that the editing method should be capable of changing the facts of *any* LM that is not specifically trained on adaptability. For example, training using meta-learning is one way to make the models editable using a few gradient steps ([Finn et al., 2017](#)). **Reliability** means that editing a LM should only affect the targeted fact while retaining other unrelated information. **Consistency** signifies that changes should be consistent across semantically equivalent inputs. [Hase et al. \(2021\)](#) measure consistency under entailment (i.e. changing one fact must change other entailed facts in the LM), which [Mitchell et al. \(2021\)](#) call the equivalence neighborhood.

### 5.1 Task Formulation

To put it formally, given a LM  $f_\theta$  that contains a collection of implicit factual knowledge  $\mathcal{F} : \{(x, y)_i\}$  in its parameters  $\theta$  by mapping an input  $x_i$  to a potentially undesired output  $y_i$ , the goal is to obtain a new parameter set  $\theta^*$  that conforms to a dataset of revisions  $\mathcal{D} : \{(x, \hat{y})_j\}$  by predicting the desired output  $\hat{y}_j$  for  $x_j$ . In addition, the new model  $f_{\theta^*}$  should alter the base model’s output on the equivalence neighborhood  $\mathcal{P}^x$  of inputs in  $\mathcal{D}$  (i.e.

related input/output pairs such as paraphrases of  $x$ ) while leaving the model’s behavior intact on other unrelated inputs  $\mathcal{O}^x$ .

### 5.2 Finetuning

**Baseline Methods** One natural strategy to solve this problem is to finetune the model in question or retrain it from scratch on a modified training corpora that is consistent with the new facts. However, such a method would be expensive and impractical for modifying a few datapoints, especially for large LMs. Another approach is to construct a collection of supporting evidences for the modified facts and use it to finetune the model by minimizing a per-instance loss. This method can achieve high performance on the modified facts  $\mathcal{D}$ , but can significantly degrade the model’s performance on the unmodified facts ([Zhu et al., 2020](#)). To obtain a reasonable accuracy on both, one can include evidences from  $\mathcal{D}$  and the set of facts that should not be modified  $\mathcal{O}^x$  in every iteration while finetuning. However, as discussed below, further tricks need to be employed to retain the accuracy on the unmodified facts to avoid catastrophic forgetting.

**Constrained Finetuning** [Zhu et al. \(2020\)](#) tackle this problem by enforcing a norm-based constraint on the model’s weights  $\theta$  while finetuning the model on the dataset of revisions  $\mathcal{D}$  such that it minimally interferes with the facts that should not be modified. However, such a constraint on the parameter space ignores the highly non-linear nature of LMs ([De Cao et al., 2021](#)).

### 5.3 Hyper Networks

Another class of methods uses a set of small neural networks (also known as a hyper-networks or learned optimizers) that learns to predict the shift in weights for editing a targeted fact ([Ha et al., 2017](#)). Specifically, [De Cao et al. \(2021\)](#) propose KNOWLEDGEEDITOR that additionally applies a constraint on the update in the function space as opposed to the parameter space like in [Zhu et al. \(2020\)](#). The intuition behind this is to predict identical output distributions to the original one for all unrelated inputs  $\mathcal{O}^x$ . However, this method fails when editing very large models. Therefore, to scale to larger models, [Mitchell et al. \(2021\)](#) propose Model Editing Networks with Gradient Decomposition (**MEND**) that trains hyper-networks on the standard finetuning gradient of a given correction. The trick they employ is decomposing the gradi-

ent into its rank-one outer product form to learn a function  $g$  that scales nicely with a model’s dimensionality, making it much less computationally expensive than prior methods. In a parallel line of work, Hase et al. (2021) introduce a training objective for Sequential, Local, and Generalizing model updates (SLAG). They show that SLAG outperforms previous methods when updating multiple beliefs in sequence.

#### 5.4 Direct Editing

Meng et al. (2022) propose another method they call Rank-One Model Editing (ROME) for modifying factual knowledge in large neural networks. Specifically, they consider the transformer MLP modules as simple key-value memories (Geva et al., 2021), and editing a specific fact is just a matter of locating the relevant MLP weights by causal tracing then writing to it directly the new key-value pair using a rank-one modification.

## 6 Reasoning

It has been a long-standing goal of AI to reason over explicit knowledge given to it in order to reach conclusions (McCarthy, 1959; Newell and Simon, 1956; Metaxiotis et al., 2002). The advent of LMs has brought this dream closer to reality (Clark et al., 2020). Concretely, LMs are shown to leverage the knowledge they learn during pretraining to perform well on reasoning tasks expressed in natural language rather than in formal representations. Such tasks include: commonsense (Da et al., 2021), natural language inference (Bowman et al., 2015), mathematical (Saxton et al., 2019; Polu et al., 2022), rule-based (Clark et al., 2020), inductive (Misra, 2021) and abductive reasoning (Bhagavatula et al., 2020). However, we focus on types of reasoning that KBs have been shown to perform better on NLP tasks.

### 6.1 Symbolic Reasoning

**Logical Reasoning** Previous work show that LMs can perform rule-based reasoning by emulating the process of binding facts with low level first order logic rules to deduce a conclusion (Clark et al., 2020; Talmor et al., 2020b; Gontier et al., 2020). LMs can also generate proofs demonstrating their ‘chain of thought’ (Saha et al., 2020; Tafjord et al., 2021; Polu et al., 2022). Wei et al. (2022) show that inducing a prompt that mimics the reasoning process improves the performance on reasoning

tasks such as math word problems.

**Mathematical Reasoning** In another line of work, Nye et al. (2021) show that LMs can perform complex multi-step computations when asked to generate the results of intermediate steps. Polu et al. (2022) show that by using expert iteration (i.e. proof search (Polu and Sutskever, 2020) interleaved with curriculum learning) a LM can solve complex mathematical problems.

**Limitations** Despite these successes, the best models are still unable to chain more than 2 or 3 non-trivial steps of complex reasoning (Polu et al., 2022). Further, it largely remains an open question of whether the LMs are indeed “reasoning” or just emulating the thought process of humans (Wei et al., 2022).

### 6.2 Commonsense Reasoning

Commonsense reasoning is the ability to reason about the underlying nature of situations humans encounter on a day-to-day basis such as the effects of Newtonian physics and the intentions of others. LMs are shown to possess a certain amount of commonsense knowledge in their parameters (Petroni et al., 2019; Davison et al., 2019; Zhou et al., 2020; Cui et al., 2021). As a result, Huang et al. (2019); Talmor et al. (2019); Sap et al. (2019); West et al. (2021) introduce datasets to evaluate the extent to which LMs can reason over the knowledge they learned during pretraining. In addition to implicit reasoning, Talmor et al. (2020b) show a way in which LMs can systematically reason over both explicit input statements given to the model by a user and the implicit knowledge stored in the parameters of the model. However, despite succeeding in commonsense leaderboards, Merrill et al. (2021) suggest that those models fail to understand the underlying semantics leading them to commit trivial mistakes.

## 7 Explainability & Interpretability

While the field of NLP has traditionally been guarded by the use of inherently explainable and interpretable techniques, the move from predominantly statistical NLP methods to black box neural models have only motivated the necessity to return to the study of explainability and interpretability.

While literature in NLP and computer science combine explainability and interpretability as similar, if not, identical aspects (Došilović et al., 2018),



we aim to clarify these confusions by creating clear definitions for both terms. In short, we define **Interpretability** as the inspection of the inner workings of the model and understanding the reasoning behind model predictions by evaluating internal mechanisms. On the other hand, **Explainability** is a focus on the outward appearance of a model, namely whether the model’s outputs are explainable in a post-hoc setting.

Since our main goal is to apply these LMs-as-KBs in practice, LMs that are not explainable and interpretable are often undeployable. For example, prior work has identified that BERT (Devlin et al., 2019), while powerful, is inherently opaque with respect to its inner workings (Rogers et al., 2020). KBs, on the other hand, are far easier to read, as their fixed schemas are simpler to interpret. Having the same level of explainability inherent in LMs as we do in KBs would vastly improve their practical use. However, while KBs are naturally explainable, they do not inherently have interpretable qualities. So we make the distinction that, while we talk about both aspects with respect to LMs, we in tangent explain interpretability to cover all aspects of explainable AI in LMs.

## 7.1 Interpretability

Transformer models (Vaswani et al., 2017; Devlin et al., 2019; Liu et al., 2019; Raffel et al., 2020) are composed of different building blocks: the encoder, decoder, self-attention, and more. However, we know little of how these building blocks work. Even as newer LMs are devised, some model capabilities remain uninterpreted, and problematic behaviors become evident even after a model has been in use for a long period of time. As a result, a deep understanding of the mechanisms that drive these models is imperative for long-term use and applied success in the real world.

**Probing** Because little is known about how neural models function, many researchers opt to understand these models via probing. In its most basic form, one type of probing is done by way of the use of a simple linear classifier to associate internal representations with external properties (Belinkov et al., 2020). Probing allows researchers to answer questions about how the models function, their structure, or the decisions that these models make, especially regarding how these models learn linguistic structures (Tenney et al., 2019b; Hewitt and Liang, 2019; Hewitt and Manning, 2019; Be-

linkov et al., 2020).

**Attention** With the introduction of attention-based LMs (Vaswani et al., 2017), researchers have attempted to use attention (Bahdanau et al., 2015) to interpret the inner workings of the models. However, attention heads have been found to be fairly uninterpretable, as Michel et al. (2019) has found that multiple attention heads have little impact on performance. Other transformer-based attention research has evaluated whether these models actually learn linguistic and syntactic structure and the relationship between them (Tenney et al., 2019a; Jawahar et al., 2019; Clark et al., 2019; Vig and Belinkov, 2019).

However, many previous works have also contended whether attention can convey proper explanations (Jain and Wallace, 2019; Serrano and Smith, 2019; Wiegrefe and Pinter, 2019). Jain and Wallace (2019) and Serrano and Smith (2019) argue that attention cannot be used as a *faithful* explanation for the model. On the other hand, Wiegrefe and Pinter (2019) contradicts both statements by noting that there is a *plausible* chance that attention could be correlated with the model. However, this claim is dependent on one’s definition of explanation. Mohankumar et al. (2020) follows up prior work to note that the distribution of attention fails to fall on important words and strays to unimportant tokens. As a result, the definition of whether attention can adequately provide an explanation for the inner workings of LMs remains opaque.

**Transformer Mechanisms** We evaluate transformers and whether they are explainable via other mechanisms, such as the feedforward layers (Zhao et al., 2021; Geva et al., 2021; Meng et al., 2022). Zhao et al. (2021) propose a tool to measure nonlinearities in LMs by taking into account geometry space of embeddings, finding that the non-linearity of the self-attention feedforward layers and MLPs of a LM follow similar patterns, but their functions are less known. Geva et al. (2021) extrapolate from this learned fact and find that feedforward layers in LMs are just key value memories; as a result Meng et al. (2022) are able to use their method of causal tracing to locate the knowledge and use the key value pairs to access knowledge within the feedforward layers and make modifications to it.

**Mechanistic Interpretability** Akin to reverse engineering software, if we could reverse engineer transformers, we could garner more understanding

about the inner workings of these models. [Elhage et al. \(2021\)](#) introduces mathematical conception as a way to understand the internal operations within. In addition, they also discover that attention heads can explain in-context learning within smaller models. Their promising results highlight the potential for further development of mathematical tools to understand computational patterns.

**Causal tracing** Introduced in [Meng et al. \(2022\)](#), causal tracing is another form of accessing in which the model is traced for the path of information within. To do so, the model is run multiple times while corruptions are introduced to the system, then the system is restored to tease out which changes restored the original results. Their results show that it is possible to identify activations that are related to a model’s factual predictions.

## 7.2 Explainability

The more we better explain the output of existing LMs, the better we can tailor these systems to real-world use. These explanations for model behavior could in turn be used to correct model shortcomings and improve help the end user gain trust in a system.

**Influence Functions** One way to explain a black box model is through a technique known as an influence function. First introduced by [Hampel \(1974\)](#), [Koh and Liang \(2020\)](#) applies influence functions to neural models by using second-order optimization techniques; presenting a method that enable influence functions to be used to interpret model outputs.

[Han et al. \(2020\)](#) applies influence functions to LMs, finding that influence functions may be better suited for more complex tasks, such as natural language inference, despite these methods being computationally prohibitive ([Pezeshkpour et al., 2021b](#)). Gradient-based and non-Hessian information *influence attribution* methods ([Pezeshkpour et al., 2021b](#)) have been introduced to speed up computation. ([Pezeshkpour et al., 2021a](#)) also introduce the combination of influence attribution with saliency maps to find artifacts in training data more accurately than using influence functions alone.

Influence functions have also been used for interactive debugging by way of using humans as feedback after interpreting their output ([Zylberajch et al., 2021](#)). In this way, we show promise that there is a potential to integrate these explainability tools with humans in a more practical setting.

**Explanations** Researchers have introduced the potential for LMs to generate coherent explanations of their decisions. LMs such as T5 ([Raffel et al., 2020](#)) are able to generate explanations that achieve state-of-the-art performance on explainability benchmarks ([Narang et al., 2020](#)). Despite the push to use explanations as a way to improve explainability within LMs, these explanations are still inconsistent and fickle. [Camburu et al. \(2020\)](#) show that by using an adversarial framework to interject modified inputs, they are able to show that LMs generate a large number of inconsistencies in their explanations.

Other methods have shown that LMs can learn to generate the reasoning process behind their decisions using prototype networks ([Schramowski et al., 2021](#)), through highlighting fragments of the input text to justify the output ([Lei et al., 2016](#)), or using human-provided explanations in the training process ([Camburu et al., 2018](#); [Paranjape et al., 2021](#)).

## 8 Future Work & Limitations

In this paper, we review the literature with respect to five aspects that LMs need to be proficient at to qualify as KBs. However, despite these recent breakthroughs, the community still has a long way to go to enable the real-world deployment of LMs. For instance, pretrained LMs need explicit tuning on a consistency corpus ([Elazar et al., 2021](#)) in order to behave similarly under different paraphrases. They are also sensitive to word-order, negation, priming, and patterns ([Kassner and Schütze, 2020](#)) and are unreliable out-of-the-box. Furthermore, previous work demonstrates that there are theoretical limitations to transformers that prevents them from performing certain types of reasoning tasks ([Hahn, 2020](#); [Bhattamishra et al., 2020](#); [Helwe et al., 2021](#)). They also lack social intelligence ([Liang et al., 2021](#)), an understanding of time ([Dhingra et al., 2021](#); [Lazaridou et al., 2021](#)), causality ([Li et al., 2021](#)), dealing with uncommon facts ([Poerner et al., 2020](#); [Jiang et al., 2020a](#)) and counterfactual reasoning ([Feng et al., 2021](#)). We hope that by uncovering the limitations of LMs from the perspective of KBs, we can continue to motivate exploring intrinsically the positives of KBs and apply the knowledge to improving LMs.

## Acknowledgements

Special thanks to Siddharth Verma for many helpful discussions and comments on the paper. Ahmed El-Kholy for the graphic in Figure 1.

## References

- Leonard Adolphs, Shehzaad Dhuliawala, and Thomas Hofmann. 2021. How to query language models? *ArXiv*, abs/2108.01928.
- Armen Aghajanyan, Bernie Huang, Candace Ross, Vladimir Karpukhin, Hu Xu, Naman Goyal, Dmytro Okhonko, Mandar Joshi, Gargi Ghosh, Mike Lewis, and Luke Zettlemoyer. 2022. Cm3: A causal masked multimodal model of the internet. *ArXiv*, abs/2201.07520.
- Armen Aghajanyan, Dmytro Okhonko, Mike Lewis, Mandar Joshi, Hu Xu, Gargi Ghosh, and Luke Zettlemoyer. 2021. **HTLM: Hyper-Text Pre-Training and Prompting of Language Models**. *arXiv:2107.06955 [cs]*. ArXiv: 2107.06955.
- Sungjin Ahn, Heeyoul Choi, Tanel Pärnamaa, and Yoshua Bengio. 2017. **A neural knowledge language model**.
- KimAllan Andersen and Daniele Pretolani. 2001. **Easy cases of probabilistic satisfiability**. *Annals of Mathematics and Artificial Intelligence*, 33:69–91.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.
- Yonatan Belinkov, Sebastian Gehrmann, and Ellie Pavlick. 2020. **Interpretability and analysis in neural NLP**. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 1–5, Online. Association for Computational Linguistics.
- Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. **On the dangers of stochastic parrots: Can language models be too big?** In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, FAccT '21*, page 610–623, New York, NY, USA. Association for Computing Machinery.
- Rahul Bhagat and Eduard Hovy. 2013. **What Is a Paraphrase?** *Computational Linguistics*, 39(3):463–472.
- Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Scott Yih, and Yejin Choi. 2020. Abductive commonsense reasoning. *ArXiv*, abs/1908.05739.
- S. Bhattamishra, Kabir Ahuja, and Navin Goyal. 2020. On the practical ability of recurrent neural networks to recognize hierarchical languages. *ArXiv*, abs/2011.03965.
- 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*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020a. **Language models are few-shot learners**. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020b. **Language models are few-shot learners**.
- Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. In *NeurIPS*.
- Oana-Maria Camburu, Brendan Shillingford, Pasquale Minervini, Thomas Lukasiewicz, and Phil Blunsom. 2020. Make up your mind! adversarial generation of inconsistent natural language explanations. In *ACL*.
- Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, Alina Oprea, and Colin Raffel. 2021. **Extracting Training Data from Large Language Models**. *arXiv:2012.07805 [cs]*. ArXiv: 2012.07805.
- Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. 2017. **A survey on dialogue systems**. *ACM SIGKDD Explorations Newsletter*, 19(2):25–35.
- Junyoung Chung, Çağlar Gülçehre, Kyunghyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *ArXiv*, abs/1412.3555.
- Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. **What does bert look at? an analysis of bert’s attention**.

- Peter Clark, Oyvind Tafjord, and Kyle Richardson. 2020. Transformers as soft reasoners over language. *ArXiv*, abs/2002.05867.
- Pedro Colon-Hernandez, Catherine Havasi, Jason Alonso, Matthew Huggins, and Cynthia Breazeal. 2021. Combining pre-trained language models and structured knowledge.
- Leyang Cui, Sijie Cheng, Yu Wu, and Yue Zhang. 2021. On commonsense cues in BERT for solving commonsense tasks. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 683–693, Online. Association for Computational Linguistics.
- Jeff Da, Ronan Le Bras, Ximing Lu, Yejin Choi, and Antoine Bosselut. 2021. Analyzing commonsense emergence in few-shot knowledge models.
- Marina Danilevsky, Kun Qian, Ranit Aharonov, Yannis Katsis, Ban Kawas, and Prithviraj Sen. 2020. A Survey of the State of Explainable AI for Natural Language Processing. *arXiv:2010.00711 [cs]*. ArXiv: 2010.00711.
- Joe Davison, Joshua Feldman, and Alexander Rush. 2019. Commonsense knowledge mining from pre-trained 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 1173–1178, Hong Kong, China. Association for Computational Linguistics.
- Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing Factual Knowledge in Language Models. *arXiv:2104.08164 [cs]*. ArXiv: 2104.08164.
- Michiel de Jong, Yury Zemlyanskiy, Nicholas FitzGerald, Fei Sha, and William Cohen. 2021. Mention memory: incorporating textual knowledge into transformers through entity mention attention.
- Cyprien de Masson d’Autume, Sebastian Ruder, Lingpeng Kong, and Dani Yogatama. 2019. Episodic memory in lifelong language learning. In *NeurIPS*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Bhuwan Dhingra, Jeremy R. Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W. Cohen. 2021. Time-Aware Language Models as Temporal Knowledge Bases. *arXiv:2106.15110 [cs]*. ArXiv: 2106.15110.
- Bhuwan Dhingra, Manzil Zaheer, Vidhisha Balachandran, Graham Neubig, Ruslan Salakhutdinov, and William W. Cohen. 2020. Differentiable reasoning over a virtual knowledge base.
- Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Russell Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.
- Filip Karlo Došilović, Mario Brčić, and Nikica Hlupić. 2018. Explainable artificial intelligence: A survey. In *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 0210–0215.
- Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhishava Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. 2021. Measuring and Improving Consistency in Pretrained Language Models. *arXiv:2102.01017 [cs]*. ArXiv: 2102.01017.
- Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*. <https://transformer-circuits.pub/2021/framework/index.html>.
- 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.
- Fuli Feng, Jizhi Zhang, Xiangnan He, Hanwang Zhang, and Tat-Seng Chua. 2021. Empowering language understanding with counterfactual reasoning. *ArXiv*, abs/2106.03046.
- Thibault Févry, Livio Baldini Soares, Nicholas FitzGerald, Eunsol Choi, and Tom Kwiatkowski. 2020. Entities as experts: Sparse memory access with entity supervision. *CoRR*, abs/2004.07202.
- Thibault Févry, Livio Baldini Soares, Nicholas FitzGerald, Eunsol Choi, and Tom Kwiatkowski. 2020. Entities as experts: Sparse memory access with entity supervision. In *EMNLP*.
- Chelsea Finn, P. Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*.
- Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners.

- Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. 2020. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. *ArXiv*, abs/2009.11462.
- Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories.
- Yoav Goldberg. 2019. Assessing bert’s syntactic abilities. *ArXiv*, abs/1901.05287.
- Nicolas Gontier, Koustuv Sinha, Siva Reddy, and Christopher Joseph Pal. 2020. Measuring systematic generalization in neural proof generation with transformers. *ArXiv*, abs/2009.14786.
- Alex Graves, Greg Wayne, and Ivo Danihelka. 2014. Neural turing machines.
- Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. REALM: Retrieval-Augmented Language Model Pre-Training. *arXiv:2002.08909 [cs]*. ArXiv: 2002.08909.
- David Ha, Andrew Dai, and Quoc V. Le. 2017. Hypernetworks.
- Michael Hahn. 2020. Theoretical limitations of self-attention in neural sequence models. *Transactions of the Association for Computational Linguistics*, 8:156–171.
- Frank R. Hampel. 1974. The influence curve and its role in robust estimation. *Journal of the American Statistical Association*, 69(346):383–393.
- Xiaochuang Han, Byron C. Wallace, and Yulia Tsvetkov. 2020. Explaining black box predictions and unveiling data artifacts through influence functions.
- Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. 2021. Ptr: Prompt tuning with rules for text classification.
- Peter Hase, Mona Diab, Asli Celikyilmaz, Xian Li, Zornitsa Kozareva, Veselin Stoyanov, Mohit Bansal, and Srinivasan Iyer. 2021. Do Language Models Have Beliefs? Methods for Detecting, Updating, and Visualizing Model Beliefs. *arXiv:2111.13654 [cs]*. ArXiv: 2111.13654.
- Bin He, Di Zhou, Jinghui Xiao, Xin Jiang, Qun Liu, Nicholas Jing Yuan, and Tong Xu. 2020. BERT-MK: Integrating graph contextualized knowledge into pre-trained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2281–2290, Online. Association for Computational Linguistics.
- Benjamin Heinzerling and Kentaro Inui. 2021. Language Models as Knowledge Bases: On Entity Representations, Storage Capacity, and Paraphrased Queries. *arXiv:2008.09036 [cs]*. ArXiv: 2008.09036.
- Chadi Helwe, Chloé Clavel, and Fabian M. Suchanek. 2021. Reasoning with transformer-based models: Deep learning, but shallow reasoning. In *3rd Conference on Automated Knowledge Base Construction*.
- John Hewitt and Percy Liang. 2019. Designing and Interpreting Probes with Control Tasks. *arXiv:1909.03368 [cs]*. ArXiv: 1909.03368.
- John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. 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 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *ICML*.
- Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos QA: Machine reading comprehension with contextual commonsense reasoning. 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 2391–2401, Hong Kong, China. Association for Computational Linguistics.
- Wenlong Huang, P. Abbeel, Deepak Pathak, and Igor Mordatch. 2022. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. *ArXiv*, abs/2201.07207.
- Sarthak Jain and Byron C. Wallace. 2019. Attention is not explanation.
- Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.
- Haozhe Ji, Pei Ke, Shaohan Huang, Furu Wei, Xiaoyan Zhu, and Minlie Huang. 2020. Language generation with multi-hop reasoning on commonsense knowledge graph.
- Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Martinen, and Philip S. Yu. 2021. A Survey on Knowledge Graphs: Representation, Acquisition and Applications. *arXiv:2002.00388 [cs]*. ArXiv: 2002.00388.

- Zhengbao Jiang, Antonios Anastasopoulos, Jun Araki, Haibo Ding, and Graham Neubig. 2020a. [X-FACTR: Multilingual factual knowledge retrieval from pretrained language models](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5943–5959, Online. Association for Computational Linguistics.
- Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020b. [How Can We Know What Language Models Know?](#) *arXiv:1911.12543 [cs]*. ArXiv: 1911.12543.
- Nora Kassner and Hinrich Schütze. 2020. [Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly](#).
- Nora Kassner, Oyvind Tafjord, Hinrich Schütze, and Peter Clark. 2021. [BeliefBank: Adding Memory to a Pre-Trained Language Model for a Systematic Notion of Belief](#). *arXiv:2109.14723 [cs]*. ArXiv: 2109.14723.
- Yoon Kim. 2014. [Convolutional neural networks for sentence classification](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.
- Pang Wei Koh and Percy Liang. 2020. [Understanding black-box predictions via influence functions](#).
- Himabindu Lakkaraju, Dylan Slack, Yuxin Chen, Chenhao Tan, and Sameer Singh. 2022. [Rethinking explainability as a dialogue: A practitioner’s perspective](#). *ArXiv*, abs/2202.01875.
- Angeliki Lazaridou, Adhiguna Kuncoro, Elena Gribovskaya, Devang Agrawal, Adam Liska, Tayfun Terzi, Mai Gimenez, Cyprien de Masson d’Autume, Tomas Kocisky, Sebastian Ruder, Dani Yogatama, Kris Cao, Susannah Young, and Phil Blunsom. 2021. [Mind the Gap: Assessing Temporal Generalization in Neural Language Models](#). *arXiv:2102.01951 [cs]*. ArXiv: 2102.01951.
- Tao Lei, Regina Barzilay, and T. Jaakkola. 2016. [Rationalizing neural predictions](#). In *EMNLP*.
- Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. [The power of scale for parameter-efficient prompt tuning](#).
- Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. [Zero-shot relation extraction via reading comprehension](#). In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342, Vancouver, Canada. Association for Computational Linguistics.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. [BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. [Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks](#). *arXiv:2005.11401 [cs]*. ArXiv: 2005.11401.
- Xiang Lisa Li and Percy Liang. 2021. [Prefix-tuning: Optimizing continuous prompts for generation](#).
- Zhongyang Li, Xiao Ding, Kuo Liao, Ting Liu, and Bing Qin. 2021. [Causalbert: Injecting causal knowledge into pre-trained models with minimal supervision](#). *ArXiv*, abs/2107.09852.
- Paul Pu Liang, Chiyu Wu, Louis-Philippe Morency, and Ruslan Salakhutdinov. 2021. [Towards understanding and mitigating social biases in language models](#). In *ICML*.
- Stephanie C. Lin, Jacob Hilton, and Owain Evans. 2021a. [Truthfulqa: Measuring how models mimic human falsehoods](#). *ArXiv*, abs/2109.07958.
- Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O’Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona Diab, Ves Stoyanov, and Xian Li. 2021b. [Few-shot learning with multilingual language models](#). *ArXiv*, abs/2112.10668.
- Zachary Chase Lipton. 2018. [The mythos of model interpretability](#). *Queue*, 16:31 – 57.
- Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2021a. [What makes good in-context examples for gpt-3?](#)
- Leo Z. Liu, Yizhong Wang, Jungo Kasai, Hannaneh Hajishirzi, and Noah A. Smith. 2021b. [Probing across time: What does roberta know and when?](#)
- Linlin Liu, Xin Li, Ruidan He, Lidong Bing, Shafiq Joty, and Luo Si. 2021c. [Knowledge Based Multilingual Language Model](#). *arXiv:2111.10962 [cs]*. ArXiv: 2111.10962.
- Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021d. [Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing](#). *ArXiv*, abs/2107.13586.
- Qi Liu, Dani Yogatama, and Phil Blunsom. 2022. [Relational memory augmented language models](#). *ArXiv*, abs/2201.09680.

- Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2020. K-bert: Enabling language representation with knowledge graph. In *AAAI*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, M. Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.
- Robert L Logan, Ivana Balavzević, Eric Wallace, Fabio Petroni, Sameer Singh, and Sebastian Riedel. 2021. Cutting down on prompts and parameters: Simple few-shot learning with language models. *ArXiv*, abs/2106.13353.
- Robert L. Logan, Nelson F. Liu, Matthew E. Peters, Matt Gardner, and Sameer Singh. 2019. [Barack’s wife hillary: Using knowledge-graphs for fact-aware language modeling](#).
- Jean Maillard, Vladimir Karpukhin, Fabio Petroni, Wen tau Yih, Barlas Oğuz, Veselin Stoyanov, and Gargi Ghosh. 2021. Multi-task retrieval for knowledge-intensive tasks. In *ACL/IJCNLP*.
- John McCarthy. 1959. [Programs with common sense](#). In *Proceedings of the Teddington Conference on the Mechanization of Thought Processes*, pages 75–91, London. Her Majesty’s Stationary Office.
- Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual knowledge in gpt. *arXiv preprint arXiv:2202.05262*.
- William Cooper Merrill, Yoav Goldberg, Roy Schwartz, and Noah A. Smith. 2021. Provable limitations of acquiring meaning from ungrounded form: What will future language models understand? *Transactions of the Association for Computational Linguistics*, 9:1047–1060.
- Kostas S. Metaxiotis, Dimitris Askounis, and John E. Psarras. 2002. Expert systems in production planning and scheduling: A state-of-the-art survey. *Journal of Intelligent Manufacturing*, 13:253–260.
- Paul Michel, Omer Levy, and Graham Neubig. 2019. [Are sixteen heads really better than one?](#)
- Kanishka Misra. 2021. On semantic cognition, inductive generalization, and language models. *ArXiv*, abs/2111.02603.
- Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. 2021. [Fast Model Editing at Scale](#). *arXiv:2110.11309 [cs]*. *ArXiv*: 2110.11309.
- Akash Kumar Mohankumar, Preksha Nema, Sharan Narasimhan, Mitesh M. Khapra, Balaji Vasan Srinivasan, and Balaraman Ravindran. 2020. [Towards transparent and explainable attention models](#).
- Sharan Narang, Colin Raffel, Katherine Lee, Adam Roberts, Noah Fiedel, and Karishma Malkan. 2020. [Wt5?! training text-to-text models to explain their predictions](#). *ArXiv*, abs/2004.14546.
- A. Newell and H. Simon. 1956. [The logic theory machine—a complex information processing system](#). *IRE Transactions on Information Theory*, 2(3):61–79.
- Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, Charles Sutton, and Augustus Odena. 2021. [Show your work: Scratchpads for intermediate computation with language models](#). *ArXiv*, abs/2112.00114.
- Franz Josef Och, Daniel Gildea, Sanjeev Khudanpur, Anoop Sarkar, Kenji Yamada, Alex Fraser, Shankar Kumar, Libin Shen, David Smith, Katherine Eng, Viren Jain, Zhen Jin, and Dragomir Radev. 2004. [A smorgasbord of features for statistical machine translation](#). In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 161–168, Boston, Massachusetts, USA. Association for Computational Linguistics.
- Bhargavi Paranjape, Julian Michael, Marjan Ghazvininejad, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2021. [Prompting contrastive explanations for commonsense reasoning tasks](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4179–4192, Online. Association for Computational Linguistics.
- Matthew E. Peters, Mark Neumann, Robert L. Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. [Knowledge enhanced contextual word representations](#).
- Matthew E. Peters, Mark Neumann, Luke Zettlemoyer, and Wen tau Yih. 2018. [Dissecting contextual word embeddings: Architecture and representation](#). In *EMNLP*.
- 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.
- Pouya Pezeshkpour, Sarthak Jain, Sameer Singh, and Byron C. Wallace. 2021a. [Combining feature and instance attribution to detect artifacts](#).
- Pouya Pezeshkpour, Sarthak Jain, Byron C. Wallace, and Sameer Singh. 2021b. [An empirical comparison of instance attribution methods for nlp](#).

- David Picado-Muiño. 2011. [Measuring and repairing inconsistency in probabilistic knowledge bases](#). *Int. J. Approx. Reasoning*, 52:828–840.
- Nina Poerner, Ulli Waltinger, and Hinrich Schütze. 2020. [E-BERT: Efficient-yet-effective entity embeddings for BERT](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 803–818, Online. Association for Computational Linguistics.
- Stanislas Polu, Jesse Michael Han, Kunhao Zheng, Mantas Baksys, Igor Babuschkin, and Ilya Sutskever. 2022. [Formal mathematics statement curriculum learning](#). *ArXiv*, abs/2202.01344.
- Stanislas Polu and Ilya Sutskever. 2020. [Generative language modeling for automated theorem proving](#). *ArXiv*, abs/2009.03393.
- Guanghui Qin and Jason Eisner. 2021. [Learning how to ask: Querying lms with mixtures of soft prompts](#).
- Yujia Qin, Yankai Lin, Ryuichi Takanobu, Zhiyuan Liu, Peng Li, Heng Ji, Minlie Huang, Maosong Sun, and Jie Zhou. 2021. [Erica: Improving entity and relation understanding for pre-trained language models via contrastive learning](#). In *ACL/IJCNLP*.
- Alec Radford and Karthik Narasimhan. 2018. [Improving language understanding by generative pre-training](#).
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. [Language models are unsupervised multitask learners](#).
- Colin Raffel, Noam M. Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *ArXiv*, abs/1910.10683.
- 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.
- Ryokan Ri, Ikuya Yamada, and Yoshimasa Tsuruoka. 2021. [mluke: The power of entity representations in multilingual pretrained language models](#). *ArXiv*, abs/2110.08151.
- Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. [A Primer in BERTology: What we know about how BERT works](#). *arXiv:2002.12327 [cs]*. *ArXiv*: 2002.12327.
- Corby Rosset, Chenyan Xiong, Minh Hieu Phan, Xia Song, Paul Bennett, and Saurabh Tiwary. 2020. [Knowledge-aware language model pretraining](#). *ArXiv*, abs/2007.00655.
- Tara Safavi and Danai Koutra. 2021. [Relational world knowledge representation in contextual language models: A review](#).
- Swarnadeep Saha, Sayan Ghosh, Shashank Srivastava, and Mohit Bansal. 2020. [Prover: Proof generation for interpretable reasoning over rules](#). *ArXiv*, abs/2010.02830.
- Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. 2019. [Atomic: An atlas of machine commonsense for if-then reasoning](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3027–3035.
- David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. 2019. [Analysing mathematical reasoning abilities of neural models](#). *ArXiv*, abs/1904.01557.
- Timo Schick and Hinrich Schütze. 2021a. [Exploiting cloze questions for few shot text classification and natural language inference](#).
- Timo Schick and Hinrich Schütze. 2021b. [It’s not just size that matters: Small language models are also few-shot learners](#).
- Patrick Schramowski, Felix Friedrich, Christopher Tauchmann, and Kristian Kersting. 2021. [Interactively generating explanations for transformer language models](#). *CoRR*, abs/2110.02058.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. [Improving neural machine translation models with monolingual data](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.
- Sofia Serrano and Noah A. Smith. 2019. [Is attention interpretable?](#)
- Taylor Shin, Yasaman Razeghi, Robert L. Logan, Eric Wallace, and Sameer Singh. 2020. [Autoprompt: Eliciting knowledge from language models with automatically generated prompts](#).
- Haitian Sun, Pat Verga, Bhuwan Dhingra, Ruslan Salakhutdinov, and William W. Cohen. 2021. [Reasoning over virtual knowledge bases with open predicate relations](#). *CoRR*, abs/2102.07043.
- Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. [Ernie: Enhanced representation through knowledge integration](#). *ArXiv*, abs/1904.09223.
- Mujeen Sung, Jinhyuk Lee, Sean Yi, Minji Jeon, Sungdong Kim, and Jaewoo Kang. 2021. [Can language models be biomedical knowledge bases?](#)



- Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. 2021. [ProofWriter: Generating implications, proofs, and abductive statements over natural language](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3621–3634, Online. Association for Computational Linguistics.
- Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. 2020a. [oLMPics – On what Language Model Pre-training Captures](#). *arXiv:1912.13283 [cs]*. ArXiv: 1912.13283.
- Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. [CommonsenseQA: A question answering challenge targeting commonsense knowledge](#). 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 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.
- Alon Talmor, Oyvind Tafjord, Peter Clark, Yoav Goldberg, and Jonathan Berant. 2020b. [Leap-Of-Thought: Teaching Pre-Trained Models to Systematically Reason Over Implicit Knowledge](#). *arXiv:2006.06609 [cs]*. ArXiv: 2006.06609.
- Wilson L Taylor. 1953. “cloze procedure”: A new tool for measuring readability. *Journalism quarterly*, 30(4):415–433.
- Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019a. [Bert rediscovers the classical nlp pipeline](#).
- Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Sam Bowman, Dipanjan Das, and Ellie Pavlick. 2019b. [What do you learn from context? probing for sentence structure in contextualized word representations](#). In *International Conference on Learning Representations*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, undefinukas Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS’17*, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.
- Pat Verga, Haitian Sun, Livio Baldini Soares, and William W. Cohen. 2020. [Facts as experts: Adaptable and interpretable neural memory over symbolic knowledge](#). *CoRR*, abs/2007.00849.
- Jesse Vig and Yonatan Belinkov. 2019. [Analyzing the structure of attention in a transformer language model](#).
- Denny Vrandečić and Markus Krötzsch. 2014. [Wiki-data: A free collaborative knowledgebase](#). *Commun. ACM*, 57(10):78–85.
- Jonas Wallat, Jaspreet Singh, and Avishek Anand. 2021. [BERTnesia: Investigating the capture and forgetting of knowledge in BERT](#). *arXiv:2106.02902 [cs]*. ArXiv: 2106.02902.
- Chenguang Wang, Xiao Liu, and Dawn Song. 2020. [Language Models are Open Knowledge Graphs](#). *arXiv:2010.11967 [cs]*. ArXiv: 2010.11967.
- Sinong Wang, Han Fang, Madian Khabsa, Hanzi Mao, and Hao Ma. 2021a. [Entailment as few-shot learner](#).
- Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhiyuan Liu, Juan-Zi Li, and Jian Tang. 2021b. [Kepler: A unified model for knowledge embedding and pre-trained language representation](#). *Transactions of the Association for Computational Linguistics*, 9:176–194.
- Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2021c. [KEPLER: A Unified Model for Knowledge Embedding and Pre-trained Language Representation](#). *Transactions of the Association for Computational Linguistics*, 9:176–194.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. [Chain of thought prompting elicits reasoning in large language models](#). *ArXiv*, abs/2201.11903.
- Xiaokai Wei, Shen Wang, Dejiao Zhang, Parminder Bhatia, and Andrew Arnold. 2021. [Knowledge enhanced pretrained language models: A comprehensive survey](#).
- Peter West, Chandra Bhagavatula, Jack Hessel, Jena D. Hwang, Liwei Jiang, Ronan Le Bras, Ximing Lu, Sean Welleck, and Yejin Choi. 2021. [Symbolic Knowledge Distillation: from General Language Models to Commonsense Models](#). *arXiv:2110.07178 [cs]*. ArXiv: 2110.07178.
- Jason Weston, Sumit Chopra, and Antoine Bordes. 2015. [Memory networks](#).
- Sarah Wiegrefe and Yuval Pinter. 2019. [Attention is not not explanation](#).
- John Wieting and Kevin Gimpel. 2018. [ParaNMT-50M: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462, Melbourne, Australia. Association for Computational Linguistics.
- Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. [Luke: Deep contextualized entity representations with entity-aware self-attention](#). In *EMNLP*.
- Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. [Kgbert: Bert for knowledge graph completion](#). *ArXiv*, abs/1909.03193.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. [Qa-gnn: Reasoning with language models and knowledge graphs for question answering](#).

Dani Yogatama, Cyprien de Masson d’Autume, and Lingpeng Kong. 2021. [Adaptive Semiparametric Language Models](#). *arXiv:2102.02557 [cs]*. ArXiv: 2102.02557.

Ningyu Zhang, Luoqi Li, Xiang Chen, Shumin Deng, Zhen Bi, Chuanqi Tan, Fei Huang, and Huajun Chen. 2021. Differentiable prompt makes pre-trained language models better few-shot learners. *ArXiv*, abs/2108.13161.

Yue Zhang and Joakim Nivre. 2011. [Transition-based dependency parsing with rich non-local features](#). In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 188–193, Portland, Oregon, USA. Association for Computational Linguistics.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. [ERNIE: enhanced language representation with informative entities](#). *CoRR*, abs/1905.07129.

Sumu Zhao, Damian Pascual, Gino Brunner, and Roger Wattenhofer. 2021. [Of non-linearity and commutativity in bert](#).

Zexuan Zhong, Dan Friedman, and Danqi Chen. 2021. [Factual probing is \[mask\]: Learning vs. learning to recall](#).

Xuhui Zhou, Yue Zhang, Leyang Cui, and Dandan Huang. 2020. [Evaluating commonsense in pre-trained language models](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9733–9740.

Chen Zhu, Ankit Singh Rawat, Manzil Zaheer, Srinadh Bhojanapalli, Daliang Li, Felix Yu, and Sanjiv Kumar. 2020. [Modifying Memories in Transformer Models](#). *arXiv:2012.00363 [cs]*. ArXiv: 2012.00363.

Hugo Zylberajch, Piyawat Lertvittayakumjorn, and Francesca Toni. 2021. [HILDIF: Interactive debugging of NLI models using influence functions](#). In *Proceedings of the First Workshop on Interactive Learning for Natural Language Processing*, pages 1–6, Online. Association for Computational Linguistics.

## Appendix

### A Models

In the main sections, we detail more general aspects of LMs. To follow the sections, we cover the different models that fall under the LMs-as-KBs

paradigm and the solutions proposed by these models that improve LM performance.

These models often incorporate knowledge explicitly through a combination of several means: via some form of **pretraining** strategy that explicitly encodes entity-level or relation-level data, via the integration of **external memory** to an existing LM, via an **attention-based mechanism**, or via a **retrieval-based** model that gathers the appropriate nodes of a KG. Models that implicitly contain knowledge, such as existing pretrained LMs, are not covered in this section, as we focus on explicit incorporation of knowledge.

#### A.1 External Memory

Existing LMs are often unable to store localized information about facts and specific knowledge; on the other hand, KBs have been known to store information about millions of entities in an interpretable fashion. Knowing this, existing research focus on taking advantage of the storage capabilities of KBs and integrating this capability into LMs (Wei et al., 2021).

Heinzerling and Inui (2021) introduce initial work understanding how LMs-as-KBs can be used in more general settings, especially in integrating knowledge about general and rare entities, since representing millions of entities within LMs is difficult when LMs have a limited vocabulary. Recent work often forgo the potential to deploy these LMs-as-KBs in the real world. To improve existing LM and overcome memory limitations, recent work focuses on the goal of improving memory for LMs to store more information about external information, such as information about entities in a sentence.

External memory has previously been used for neural networks (Bahdanau et al., 2015; Weston et al., 2015; Graves et al., 2014; Ahn et al., 2017). Ahn et al. (2017) specifically propose the **NKLM** to take advantage of external knowledge and exploit factual knowledge. Now, a large number of works now integrate external memory to improve performance on knowledge-intensive tasks. To summarize these works, we focus on detailing the different strategies of integrating external memory within LMs and how they can be further improved on.

##### A.1.1 External Knowledge Graphs

Prior work on LMs-as-KBs involves the explicit creation of external knowledge graphs (KG) that

| Name            | Author                           | External KG | Non-Parametric Memory | Pretraining | Attention | Retrieval |
|-----------------|----------------------------------|-------------|-----------------------|-------------|-----------|-----------|
| QA-GNN          | Yasunaga et al. (2021)           | ✓           |                       |             |           |           |
| KnowBERT        | Peters et al. (2019)             |             | ✓                     | ✓           |           |           |
| KGLM            | Logan et al. (2019)              | ✓           |                       |             |           |           |
| NKLM            | Ahn et al. (2017)                |             | ✓                     |             |           |           |
| REALM           | Guu et al. (2020)                |             |                       |             |           | ✓         |
| EAE             | Févy et al. (2020)               |             | ✓                     |             |           |           |
| FAE             | Verga et al. (2020)              |             | ✓                     | ✓           |           |           |
| OPQL-LM         | Sun et al. (2021)                |             | ✓                     | ✓           |           |           |
| TOME            | de Jong et al. (2021)            |             | ✓                     | ✓           | ✓         |           |
| RAG             | Lewis et al. (2021)              |             | ✓                     |             |           | ✓         |
| LUKE            | Yamada et al. (2020)             |             |                       | ✓           | ✓         |           |
| mLUKE           | Ri et al. (2021)                 |             |                       | ✓           | ✓         |           |
| ERNIE           | Zhang et al. (2019)              |             | ✓                     | ✓           |           |           |
| ERICA           | Qin et al. (2021)                |             |                       | ✓           |           |           |
| KEPLER          | Wang et al. (2021c)              |             |                       | ✓           | ✓         |           |
| SPALM           | Yogatama et al. (2021)           |             | ✓                     |             |           |           |
| GRF             | Ji et al. (2021)                 | ✓           |                       |             |           |           |
| RelationLM      | Liu et al. (2022).               |             | ✓                     |             |           |           |
| HTLM            | Aghajanyan et al. (2021)         |             |                       | ✓           |           |           |
| CM3             | Aghajanyan et al. (2022)         |             |                       | ✓           |           |           |
| K-BERT          | Liu et al. (2020)                |             | ✓                     |             |           |           |
| ERNIE           | Sun et al. (2019)                |             |                       | ✓           |           |           |
| KG-BERT         | Yao et al. (2019)                |             |                       | ✓           |           |           |
| BERT-MK         | He et al. (2020)                 | ✓           |                       |             |           |           |
| KALM            | Rosset et al. (2020)             |             | ✓                     | ✓           |           |           |
| DrKIT           | Dhingra et al. (2021)            |             |                       | ✓           |           |           |
| MBPA++          | de Masson d'Autume et al. (2019) |             | ✓                     |             |           |           |
| Multitask Model | Maillard et al. (2021)           |             |                       |             |           | ✓         |

Table 1: Models and the associated modifications to improve knowledge within their parameters.

are incorporated into models, such as in the model **KGLM** (Logan et al., 2019). KGLM looks at explicitly incorporating KGs into LMs. KGLM can access facts which are stored in a KG, growing constantly as new facts are added. The model selectively grows the KG if a fact is not there or refers back to the KG to pick an existing fact.

While not explicitly a model, **GRF** (Ji et al., 2020) propose a method to enable multi-hop reasoning with transformers. To do so, the model first encodes multi-relational graphs to obtain representations for concepts, then reasons over multiple relational paths to generate the concept, and finally chooses the output by determining the probability of obtaining the concept from the KG versus choosing a word from its innate vocabulary.

Other work also evaluate other domains. He et al. (2020) introduce **BERT-MK**, which integrates contextualized knowledge from a medical KG, which shows improvement over existing biomedical models on entity typing and relation classification tasks.

### A.1.2 Non-parametric Memory

Non-parametric memory can be seen as memory that is used in addition to internal LM memory. Existing work integrates non-parametric memory

with LMs to improve performance on downstream tasks.

Ahn et al. (2017) adapts an early version of non-parametric memory with neural models using **NKLM**. NKLM is able to combine symbolic knowledge provided from a KG with an RNN (Hochreiter and Schmidhuber, 1997).

Other work adopt a similar strategy via embeddings to incorporate entity knowledge. Early work on **KnowBERT** (Peters et al., 2019) introduces entity vectors that are computed from mention-span representations that are obtained from BERT (Devlin et al., 2019) to form entity-span representations. Other work follow similarly: Févy et al. (2020) introduce **EAE**, a LM augmented with entity memory to keep track of facts about entities. Building off of this, **FAE** (Verga et al., 2020) adds an additional memory that encodes triples from a symbolic KB that can be accessed with key-value memory to extract facts and improve predictions. Other models such as **ERNIE**, **KEPLER**, and **KALM** use existing algorithms to embed entity and relation knowledge into embeddings (Zhang et al., 2019; Wang et al., 2021b; Rosset et al., 2020). **RAG** uses vector indices of Wikipedia to access latent information during inference on knowledge-

intensive tasks (Lewis et al., 2021)

Introduced by (Dhingra et al., 2020), **DrKIT** includes a virtual knowledge base (VKB), which is a "soft knowledge base" that is used in conjunction with neural methods to compensate for structure and find answers to questions. Sun et al. (2021) improve the idea of a VKB and introduces an additional strategy to learn entity and information through **OPQL**, utilizing key-value memory to learn relationships between entities and relations. de Jong et al. (2021) take the idea of a VKB and insert it into **TOME** as non-parametric memory to improve reasoning over various knowledge sources.

Episodic non-parametric memory can also be introduced to LMs to remember specific knowledge, including both short-term and long-term contexts (de Masson d'Autume et al., 2019; Yogatama et al., 2021; Liu et al., 2022).

Liu et al. (2020) try a different strategy by introducing a visible matrix for their model **K-BERT** to control the impact of certain knowledge injected.

### A.1.3 Attention over Memory

Attention mechanisms over specific types of memory can extract salient information during inference. Early iterations of attention over memory can be seen in KnowBERT (Peters et al., 2019) where they introduce **KAR**, a form of multi-headed attention between word representations and knowledge-enhanced entity span vectors. On the other hand, in TOME (de Jong et al., 2021), they employ attention over an entire VKB within a transformer model to retrieve the relevant pieces of knowledge.

Other work modify existing transformer layers to add an additional self-attention mechanisms over entity knowledge, specifically looking at using the attention to identify the query mechanisms required depending on the attending token and token attended to (Yamada et al., 2020; Ri et al., 2021).

## A.2 Pretraining

Models have been known to encode knowledge within their parameters through unsupervised methods during pretraining. For example, given the example `Obama is from [MASK]`, using a masked language modeling (MLM) objective, we would expect the model to predict `Hawaii`. However, there is no explicit integration of any entity-level or relation-level knowledge through this objective. As a result, a number of work have sought to incorporate entity-level or relation-level knowl-

edge through pretraining strategies and loss function modifications.

Various models focus on different pretraining strategies, whether that be through the augmentation of the input or modification of loss functions or others. FAE adapts from the the EAE model to introduce a pretraining scheme modeled as a cloze-type Question Answering (QA) task (Verga et al., 2020). Other pretraining tasks include an augmentation of the existing masked language modeling (MLM) task by masking the entities and predicting the entities during pretraining (Sun et al., 2021; Yamada et al., 2020; Ri et al., 2021; Zhang et al., 2019) or introducing multiple preexisting pretraining tasks to be used in conjunction with each other, such as MLM (Yamada et al., 2020; Ri et al., 2021). Sun et al. (2019) introduce three levels of masking for their pretraining task: basic-level, phrase-level, and entity-level masking. Other work introduce a special loss function in conjunction with the standard MLM objective that focuses on predicting entities (Rosset et al., 2020), linking entities to text (Févy et al., 2020; Verga et al., 2020; Sun et al., 2021; de Jong et al., 2021), objectives focusing on semantic understanding of relations and entities (Qin et al., 2021), or knowledge embedding objective where entities and relations are encoded in KGs as distributed representations KE (Wang et al., 2020).

Some models are trained on inputs augmented specifically to encode better representations of entities and relations. For example, de Jong et al. (2021) inserts start and end entity tokens around each entity in the input. Others mask both relations and entities during pretraining, like in Sun et al. (2021). Yao et al. (2019) also imbues entity and relation information by taking the entities and relations in a sequence as input into existing LMs.

Aghajanyan et al. (2021, 2022) look at methods to incorporate more knowledge with data not commonly used in unimodal and multimodal settings. In Aghajanyan et al. (2021, 2022), both **HTLM** and **CM3** models apply scraped HTML and find that pretraining with size hints and prompting with BART (Lewis et al., 2020) can creative effective transfer to a wide range of downstream tasks and supervision levels. These approaches show the potential for web-scraped data to be used as viable signals for model pretraining on knowledge-intensive tasks.

### A.3 Retrieval

Some models can capture more knowledge in modular and interpretable ways via retrieval methods. [Guu et al. \(2020\)](#) introduce **REALM** which contain ways to extract knowledge by applying retrieval-based methods during pretraining, finetuning, and inference. Their method allows retrieval of Wikipedia knowledge and ability for the model to decide what kinds of information to query during inference. [Lewis et al. \(2021\)](#)'s RAG follow the same retrieval-based approach but instead integrates a non-parametric seq2seq to improve on tasks outside of open-domain extractive question answering. In the case of multi-task settings, [Mailard et al. \(2021\)](#) show that it is indeed possible to develop a universal multi-task retriever using non-task-specific methods using a passage and query encoder shared across all tasks.