Contents

Chapter 3

Uncovering Interpretable Structures in Pretrained Language Models

Parts of this work were previously presented in a preprint. Please refer to [Chen et al., 2024] for the full citation.

In the previous chapter, we observed that language modelling objectives effectively complete knowledge graphs, indicating that these objectives can embed structural patterns in their model weights. At its core, a language modelling objective uses a token's local context to predict itself. Remarkably, this local approach enables models to infer broader, global structures within structured data, such as knowledge graphs, particularly when there is high contextual variety.[1] This prompts a natural question: *Can language modelling objectives capture global structures in any dataset, or are they limited to explicitly organized data like knowledge graphs?*

To answer this question, we study transformer based large language models (LLMs)[2] trained on unstructured texts. Typically, LLMs are trained using autoregressive language modelling objectives, where each token is predicted based on the model's analysis of all its preceding tokens in the context. We hypothesize that this local modelling in LLMs allows them to capture global structures, as factorization models do, *even* when trained on unstructured, potentially noisy datasets like web text. Accordingly, this chap-

[1] For example, when there is many diverse predicates in the knowledge graph.

[2] Also known as foundation models for their general intelligence capabilities and applications across diverse tasks.

ter seeks to uncover these latent global structures within LLMs. Our method decomposes the transformer's monolithic computations into an ensemble of atomic computational paths, where each path resembles a factorization model, enabling structure recovery as in knowledge graph completion (see Chapter 2). In factorization models and knowledge graph completion, structures are typically limited to trigrams, whereas here they can potentially span n-grams with sufficient compute budget.[3] Using this method, we uncover and reconstruct structures embedded within LLMs that reflect patterns from their unstructured training data – such as common English phrases and domain-specific keywords from programming. Thus, despite training on unorganized texts, i.e. data without any structures, large language models ultimately learn and encode meaningful structures underlying the data through language modelling objectives. Since these structures are intrinsic to the trained model, they provide a basis for interpreting LLM behaviour without requiring external benchmarks, enabling data-free interpretability and transparency. We explore several applications of these intrinsic structures for language models.

- **Symbolic Interface.** Constructing symbolic interfaces for neural language models by sketching their (or their components') computation with the n-gram structures embedded in the model weights.

- **Behaviour Search.** Searching key n-grams in the model internal to locate and measure specific behaviours of interest, providing a deeper, structural profiling of model behaviour beyond surface-level probing.

- **Model Diff.** Enabling data-free comparison of models by analyzing differences in their n-gram structures, e.g., before and after fine-tuning.

Our case studies establish initial evidence for these applications with a few new interpretations of LLM behaviours.

- Some feedforward networks (FFNs) appear to handle simple grammatical tasks, such as adding the suffix "-ly" to preceding tokens, complementing recent findings that FFNs store factual knowledge [Geva et al., 2021, 2022].

- LLMs acquire different bigram structures at varying speeds during pretraining. In

[3] We leave as future work scaling the method and finding n-gram structures for $n > 3$.

OLMo, unique 1-to-1 bigrams like `(&, amp)` are acquired quickly while many-to-many bigrams like `(at, least)`[4] are initially promoted and later down-weighted.

- Vertical (downstream) finetuning, such as finetuning for coding tasks, raises the ranking of coding-related n-gram structures within the LLMs.
- Alignment finetuning through RLHF [Bai et al., 2022] conceals toxic n-gram structures from the surface-level outputs. Yet significant portions of toxic n-gram structures still reside within the model, making it susceptible to "jail breaking".

These findings contribute insights toward the responsible and transparent use of LLMs.

3.1 Interpreting LLMs by Uncovering Hidden Structures

Large language models (LLMs) are becoming increasingly prevalent as the universal knowledge engine, supporting a wide range of tasks, especially generative applications [Wei et al., 2021, Radford et al., 2019, Brown et al., 2020, Touvron et al., 2023a,b]. Despite their impressive capabilities, their opaque nature raises questions about their inner workings and the need for attribution to understand model behaviour. Mechanistic interpretability (MI) has emerged as an alternative to traditional attribution methods [Lundberg, 2017], focusing on tracing model behavior to internal structures rather than to the input [Bereska and Gavves, 2024, Ferrando et al., 2024].

Most MI research seeks to reveal the learned "algorithms" embedded within model computations, often using a hypothesis-and-dataset-driven approach. This approach typically involves forming a hypothesis, selecting a probing dataset, applying techniques like path patching [Wang et al., 2022] or causal tracing [Meng et al., 2022], iteratively refining the hypothesis in response to findings. Although valuable, this hypothesis-driven MI approach may restrict open-ended exploration, which is crucial for uncovering global behavior as did in human behavior studies [Skinner, 1965, Simon et al., 1990, Zipf, 2016], mapping model knowledge, and indexing behaviors to computation. Ultimately, MI aims to uncover and label structures within the monolithic computations described by the large neural models, with which users can index, associate and attribute various model behaviours to distinct aspects of the model operations.

[4] Many-to-many refers to the fact that there are rich continuations after the token `at` and precedings before the token `least`.

As we see in Chapter 2, factorization-based models (FMs) with language modelling objectives demonstrate that, after training, recovering structures can be as straightforward as computing (parameterized) inner products between embedding matrices [Trouillon et al., 2016, Lacroix et al., 2018, Balazevic et al., 2019] – revealing that these embedding matrices, derived from language modelling optimization, often store patterns aligning with underlying structures in the data, if we query them through proper operations e.g. relational weighted inner products. Given that large language models (LLMs) are similarly composed as an embedding-encapsulated system – an embedding layer, a central transformer "body", and an unembedding layer – trained using language modelling objectives, we hypothesize that similar structures latent in the model may also emerge in these large language models. We are interested in finding the structures and investigate whether such structures could facilitate mechanistic interpretability in LLMs.

To achieve this goal, this chapter introduces a method for uncovering latent structures by decomposing a transformer's computation into a set of distinct input-to-output computational paths, each of which begins with an embedding layer and ends with an unembedding layer – mirroring factorization-based models for knowledge base completion. By isolating these paths and systematically evaluating them in the input space, our method reveals n-gram structures embedded in the model's computations, analogous to how FMs reveal relational patterns in knowledge graphs.

We further discuss the relationship between such decomposition and approximating the original computation using Taylor Expansion. Despite not fully approximating the original transformer computation, the identified n-gram structures are useful for interpreting large language models as we will elaborate in our case studies. Figure 3.1 illustrates the workflow. We present a set of case studies on several autoregressive large language models (LLMs) from *Llama* and *OLMo* families with varying sizes. Our case studies illustrate that these isolated computational paths and the n-grams they retrieve offer valuable tools for interpreting LLM in multiple scenarios:

- i) revealing inner workings of LLMs where we identify specific functions of FFNs and attention heads, such as adding "-ing" suffixes (Section 3.5.1);
- ii) analysing pretraining dynamics where we observe distinct learning patterns for various bigrams e.g., "at least" is initially promoted and later suppressed in *OLMo* (Section 3.5.2);

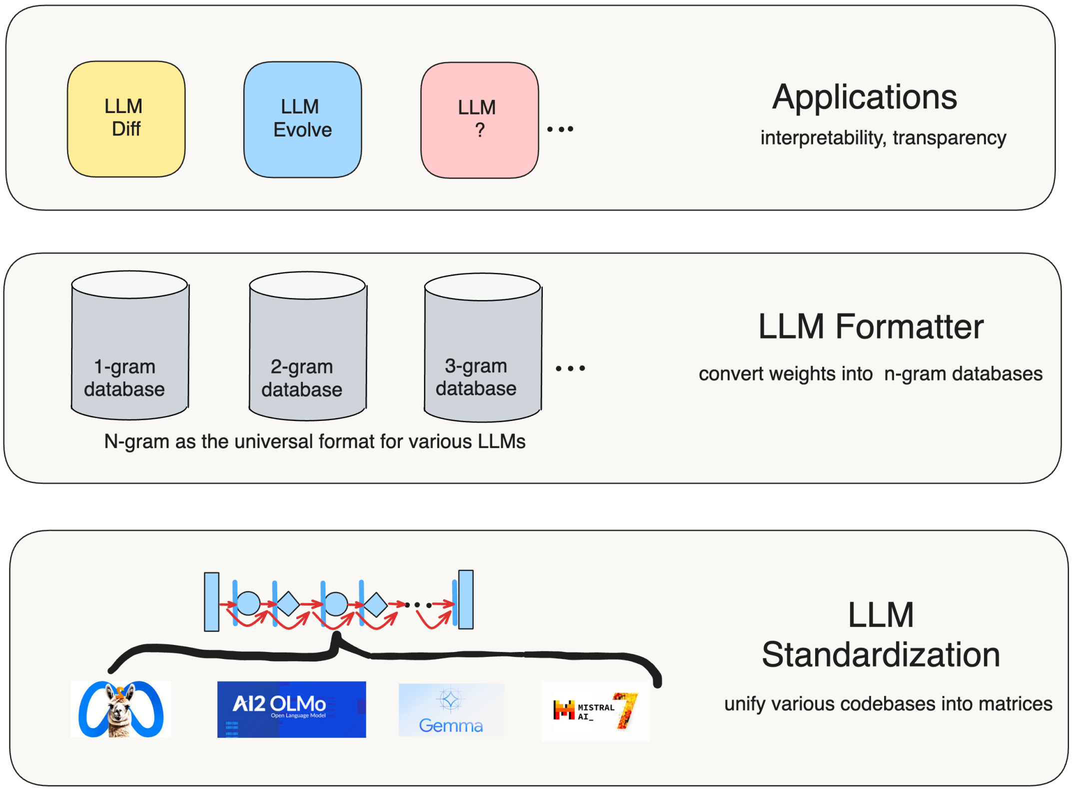

Figure 3.1: The uncovered n-gram structures can be seen as a reformatting of the corresponding large language models. These n-gram structures are derived from decomposing the transformer computations into smaller units, from where we can recompose matrix factorizations. And the identified semantic structures can support applications in interpretability and transparency.

- iii) assessing finetuning effects where we reveal model knowledge via domain-specific n-grams with applications in quantifying toxicity levels, finding, perhaps unexpectedly, that reinforcement learning from human feedback (RLHF) alignment [Bai et al., 2022] does not completely eliminate toxicity (Section 3.5.3). These findings support the development of more interpretable, transparent and responsible applications of LLMs.

3.2 Literature Review: Transformers and N-grams

Interpreting transformers. There has been much effort in interpreting the inner computations of transformer models. In particular, *mechanistic interpretability* [Ferrando et al., 2024] focuses on reverse-engineering such computations by identifying, clustering and labelling model behavior [Shah et al., 2024, Meng et al., 2022, Bricken et al., 2023] in human understandable terms and attributing them with certain model components, e.g., MLPs [Geva et al., 2021, 2022], or typical "circuits" [Conmy et al., 2023, Ferrando and Voita, 2024]. Recent work discussed limitations of currents approaches to MI. For example, Templeton et al. [2024] found it generally hard to conclude neuron-level interpretabilities, compared with feature representations; while Bolukbasi et al. [2021], Goldowsky-Dill et al. [2023] points out that conclusions drawn are generally limited to the chosen data distribution. As our approach focuses on manipulating functions, it does not require extra datasets that are used for probe fitting in methods such as Belrose et al. [2023] nor sampling, as needed by [Conmy et al., 2023, Ferrando and Voita, 2024, Voita et al., 2024]. On a high level, allowing singling out any portion of compute from the original monolithic transformer, our expansions abstract and generalize previous characterizations of the computational paths [Veit et al., 2016, Elhage et al., 2021], where non-linear components with significant roles, e.g. layernorm and MLPs, are either ignored or over-simplified for the ease of analysis. Additionally, zero ablations (or knock out) [Olsson et al., 2022] and direct logits attributions [Wang et al., 2022] are linked to particular instantiations of zeroth-order jet expansions [Chen et al., 2024].

The resurgence of n-gram models. The early applications of n-gram models for languages dates back to [Shannon, 1948], where n-grams were used to model the statistics of English. In essence, these n-grams captured structure underlying the English data they modeled: which words usually go together and which do not. The n-gram based approaches have since then been vital in natural language processing, particularly for general language modelling [Goodman, 2001] with applications like machine translation [Brants et al., 2007]. Recently, there have been regained interests in combining n-gram with neural network based approaches [e.g. Liu et al., 2024b]. Several recent works have also explored the relationships between LLMs and n-gram language models, such as analysing the representational capacity of transformers to emulate n-gram

LMs [Svete and Cotterell, 2024], and measuring the agreement between LLM predictions and curated n-gram rule sets [Nguyen, 2024].

3.3 Decomposing Transformers for Structural Recovery

Large language models are often based on the transformer architecture [Vaswani et al., 2017]. The transformer, in its original formalization, was optimized for leveraging the SIMD (single instruction multiple data) paradigm offered by the GPU for fast parallel processing sequences. Despite its efficiency, this formalization is not designed for underpinning any human-understandable structures embedded in the model. To enable structural recovery similar to how a factorization model does on a knowledge graph (Chapter 2), we need to decompose the transformer computation into smaller and easier-to-analyse units. A straightforward way is to cluster activation patterns on external datasets and treat components reacting similarly to a group of data points as a unit [Voita et al., 2024, Ferrando and Voita, 2024, Ferrando et al., 2024]. However, the recovered structures will heavily depend on the choice of data in this case, undesirable for understanding the model's global behaviour.

Luckily, transformers, despite consisting of complicated modules like self-attention, follow a simple recursive residual paradigm, where multiple identical architected residual blocks [He et al., 2016] are stacked together. We can exploit this fact to decompose computations into a set of atomic paths, each of which behave like a factorization model and enable latent structure recovery. Notation-wise, we operate at the granularity of residual blocks (e.g., self-attention or MLP blocks). This notational choice simplifies our presentation, while aligning with previous literature [Veit et al., 2016], and maintains practical relevance given the prevalence of residual computation for real-world applications [Dosovitskiy et al., 2020, Touvron et al., 2023a,b].

3.3.1 Neural Networks with Recursive Residual Links

We start by reviewing the archetypal computational structure of recursive residual nets, which feature transformers prominently. Specially, we focus on neural network architectures where the main body comprises multiple recursive residual blocks, with input and output managed respectively by an encoding and a decoding module. Such models fall

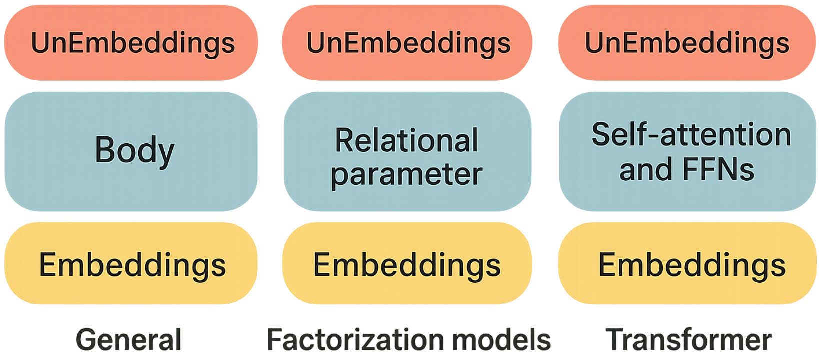

Figure 3.2: Embedding "sandwiches" are typical architectures for dealing with discrete and finite inputs to the neural networks. For example, the factorization based models for knowledge graph completion and the transformer for textual sequence completion.

into the same category of embedding-encapsulated models as the factorization models do, where the body is "sandwiched" between two embedding layers (see Figure 3.2).

Formally, let $\mathcal{Z}$ be an input space. For example, this can be sequences of tokens. Denote $c \in \mathbb{N}^+$ as the number of classes, such as the vocabulary size in a language model. Define $\mathcal{Y} = \mathbb{R}^c$ as the space of output logits, which correspond to the unnormalised over the c classes. Let $d \in \mathbb{N}^+$ represent the dimensionality of the hidden representations. We are concerned with functions $q : \mathcal{Z} \to \mathcal{Y}$ described as follows:

$$q = \upsilon \circ h_L \circ \eta, \quad \text{where } h_L : \mathbb{R}^d \to \mathbb{R}^d, \quad h_L = \bigcirc_{l=1}^{L} \beta_l, \tag{3.1}$$

where $L \in \mathbb{N}^+$ is the number of residual blocks (e.g. recursive depth), $\eta : \mathcal{Z} \to \mathbb{R}^d$ is an input encoding module (e.g. token embedding layer), $\bigcirc$ denotes repeated functional composition, and

$$\begin{aligned} &\beta_l : \mathbb{R}^d \to \mathbb{R}^d, && \text{for } l \in [L], \\ &\beta_l = \text{id} + \gamma_l, && \gamma_l : \mathbb{R}^d \to, \mathbb{R}^d \end{aligned} \tag{3.2}$$

$$\begin{aligned} &\upsilon : \mathbb{R}^d \to \mathcal{Y}, && \upsilon(x) = U \cdot \gamma_{L+1}(x), \\ &U \in \mathbb{R}^{c \times d}, && \gamma_{L+1} : \mathbb{R}^d \to \mathbb{R}^d \end{aligned} \tag{3.3}$$

are respectively residual blocks with non-linearities γ_l's (e.g., input-normalized causal self-attentions or MLPs), and the output decoding module (e.g., an unembedding projection U after a layer normalization γ_{L+1}); id is the identity map. We leave all parameters *implicit* and assume all functions are infinitely differentiable C^∞.

For transformer based language models, the model is optimized with a language modelling objective, where the next token is predicted based on analysing all the prior tokens in the local context. The function q therefore outputs unnormalised conditional probabilities (or logits) in that

$$\mathbb{P}_q(\text{“}z \text{ belongs to class } i\text{”}|z) = \text{Softmax}[q(z)]_i, \text{for } z \in \mathcal{Z}.$$

The recursive residual links are the critical ingredient that manages the information flow in the transformer. By carrying forward the outputs from each layer along with the embedded input, the recursive residual connections enable each subsequent layer to access not only the immediate computations of the previous layer but also the aggregated results from all prior layers. The recursive residual links thus facilitate the “storage” of computations from all preceding blocks along with the embedded input, leading to the accumulation of information across the model's depths.

3.3.2 Rewriting Residual Computation for Various Purposes

Although residual links have mainly been visualized as arrows connecting stacked modules in the mainstream expression of Eq. 3.1, we note that this is a perspective that renders their role in easing the training of deep networks. Such an expression of Eq. 3.1, suited for developing and training the deep residual nets, might not be suitable for analysing and interpreting them. Therefore, rewriting them in other ways become necessary for post training analysis and interpretability. Figure 3.3 summarizes several rewritings for different purposes.

Nested update accumulation Notably, as visualized in Figure 3.3 (b), we can rewrite the recursive computation of Eq. 3.1 by accumulating all the prior block outputs up to

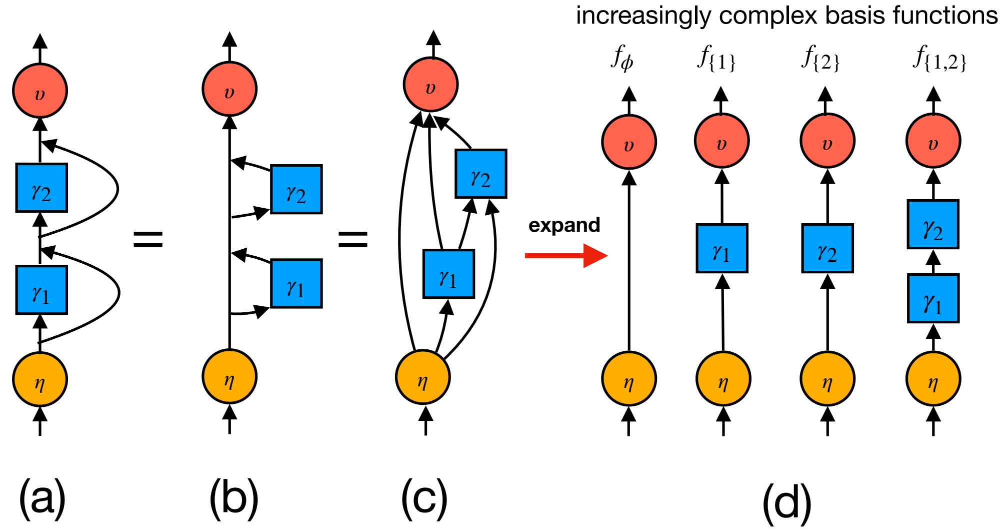

Figure 3.3: Various expressions of residual stream, each emphasizing a different aspect. (a) a visual expression adapted from [He et al., 2016, Vaswani et al., 2017], highlighting the identity shortcuts which ease the training of very deep models. (b) a visual expression adapted from [Elhage et al., 2021, nostalgebraist, 2021], highlighting the updates being written into the residual stream which serve as a communication channel. (c) a visual expression adapted from [Veit et al., 2016], highlighting the unrolling of all the residual links (d) a visualization highlighting our proposed decomposition in Section 3.3.3 into separated input-to-output computational paths which are useful for interpretability. For a linear residual net, (a)-(d) are equivalent expressions.

block $l \in [L]$, assuming $h_0 = \eta$:

$$
\begin{aligned}
h_l &= \left(\bigcirc_{j=1}^{l} \beta_j\right) \circ \eta = \eta + \sum_{j=1}^{l} \gamma_j \circ h_{j-1} \\
q &= \upsilon \circ \eta + \sum_{l=1}^{L} \upsilon \circ \gamma_l \circ h_{l-1}.
\end{aligned}
\tag{3.4}
$$

Elhage et al. [2021] introduces the term *residual stream* to describe h_l, while similar concepts like "residual bus" can be traced back to Hochreiter and Schmidhuber [1997] and Srivastava et al. [2015]. Such rewritings of recursive residual links have been widely applied in the mechanistic community [Elhage et al., 2021, nostalgebraist, 2021], highlighting the updates produced by each block (e.g. the self-attention block or the FFN block in the standard transformer) being written into the residual stream which serve as a communication channel.

Gradient paths Similarly, Veit et al. [2016] describe and study the unrolled structure of the final residual stream expressed as $h_L = \eta + \sum_{j=1}^{L} \gamma_j \circ h_{j-1}$, which reveals a number of paths from the input to the decoder (rather than the output), growing *linearly* with the network depth L. This expansion is illustrated by the three pathways (black arrows) leading to the node υ (red circle) in Figure 3.3 (c) for a case of two-layer residual architecture. Because the differentiation is a linear operator, this kind of rewriting is useful for analysing the gradient flow during backpropagation, where one can track common issues in training deep neural networks, such as gradient vanishing and gradient ensembling from different paths. However, this rewriting alone does not lend itself directly to analysing the model's intrinsic input-output functional relationships. To "mechanistically" understand the model's behaviour, a further decomposition is needed to reflect the internal structure underpinning the model's knowledge possession.

3.3.3 Rewriting Recursive Residual Networks into Factorizations

So far, we have described several rewritings of a recursive residual computation graph, each for a different purpose. For instance, Eq. 3.4 decomposes the original computational graph into a series of additive terms. Each term builds incrementally on the previous ones, forming a hierarchical structure. Despite resembling a series expansion (e.g., a Fourier Expansion), the terms in this rewriting are not sufficiently "atomic" – the interdependency among terms and their intertwined roles complicate direct interpretation.

Decomposing recursive residual networks into 2^L input-output paths To systematically decompose the nested terms in Eq. 3.4, we observe that each γ_l takes as input a sum of upstream terms. Let us consider a sum $x_1 + x_2$ as the input signal. If γ_l preserves addition, i.e. it is an additive map [Reed and Simon, 1980], then $\gamma_l(x_1+x_2) = \gamma_l(x_1)+\gamma_l(x_2)$, naturally expanding the nested terms into distinct chains of dependencies that trace back to the input when applied at all residual links. The original computational graph can then be expanded as a sum of 2^L unique paths. Each path applies L transformations, where each transformation is either γ_l or id. Formally, we can rewrite q by

$$
\begin{aligned}
q &= \upsilon \circ \left\{ \bigcirc_{l=1}^{L} (\text{id} + \gamma_l) \right\} \circ \eta \\
&= \upsilon \circ \Big(\sum_{s \in \{0,1\}^L} \bigcirc_{l=1}^{L} \gamma_l^{s_l} \Big) \circ \eta \\
&= \sum_{s \in \{0,1\}^L} \upsilon \circ \big(\bigcirc_{l=1}^{L} \gamma_l^{s_l} \big) \circ \eta \\
&= \sum_{s \in \{0,1\}^L} f_s.
\end{aligned}
\tag{3.5}
$$

Here $s = (s_1, s_2, ...s_L)$ is an L-bit binary vector in the set of $\{0,1\}^L$, indicating a unique path configuration. $s_l = 1$ represents the path using the γ_l transformation. $s_l = 0$ represents the path using the identity transformation id. $\bigcirc_{l=1}^{L} \gamma_l^{s_l}$ is the sequential composition used by the path according to s. This rewriting reveals that the original recursive residual computation behaves as an ensemble of 2^L increasingly complex input-to-output computational paths $f_s : \mathcal{Z} \to \mathcal{Y}$ sharing L core components. The complexity of a path is determined by the number of non-identity transformations it involves. Thus the hierarchy of the paths implies interesting properties of the recursive residual computation. For example, simpler paths with fewer γ_l terms might capture broad and abstract data patterns while more complex paths might capture finer details and potentially nuanced noise. Moreover, these paths include "non-continuous", where one path can skip one or several blocks and directly go to the later portion of the computation graph.

Linear recursive residual networks as an ensemble of factorization models In the real domain, linear γ's are additive maps. So if we assume all γ's are linear, such that $\gamma_l(x) = A_l x$, for $l \in [L]$, and assume the encoder $\eta(x) = Ex$ and the decoder $\upsilon(x) = Ux$ then the result of the above decomposition turns out to be an ensemble of factorization models:

$$
q = \sum_{S \in 2^{[L]}} U \left(\prod_{l \in S} A_l \right) E \tag{3.6}
$$

where $2^{[L]}$ is the power set of $[L]$ which contain 2^L elements, meaning S could for example be $\{1\}$ or $\{1,2\}$ etc. Let us denote $W_S = \prod_{l \in S} A_l$, which is a $d \times d$ projection matrix, and $f_S(x) = W_S x$ denotes the mapping of the selected path. So we have

$$q = \sum_{S \in 2^{[L]}} U W_S E^\top$$

which is exactly a generalized factorization models where $U \in \mathbb{R}^{c \times d}$, $E \in \mathbb{R}^{c \times d}$ are the two embedding matrices wrapping the W_S matrix. From this we can see that a linear transformer boils down to an ensembling of 2^L weighted matrix factorization $U W_S E^\top$, where $W_S \in \mathbb{R}^{d \times d}$ is the weighting matrix between U and E. Akin to how predicates (relations) weight the subject embeddings and the object embeddings, here W_S plays a similar role as a special kind of global predicates (and self-attention might act as local predicates as our ongoing work shows). And most importantly, the outcomes from these individual factorization models $D_S = U W_S E^\top \in \mathbb{R}^{c \times c}$ becomes a database storing the $c \times c$ interactions between the c tokens, resembling how a factorization model based scoring function stores the links on a knowledge graph. These direct readouts from the individual input-output paths thus recover the latent input-output structure underlying the model computation. When applied to language models, we are equivalently converting a large language model into a set of factorization models and thus into their associated token interaction databases – a symbolic reformatting into a set of bigram databases, where high-scoring entries reflect meaningful information structures about the training dataset. Figure 3.4 illustrates this process.

Non-linearity in γ_l's In practical residual architectures, however, γ_l are typically non-linear and do not preserve addition – meaning $\gamma_l(x_1 + x_2)$ can not be expanded into separate terms associated with each individual upstream input x_i. As a result, nested terms in Eq. 3.4 are retained and the decomposition into 2^L paths is not immediately possible. However, we show that we can still single out any target computational path from the super exponential set of block combinations as we do for the above linear γ_l case and empirically obtain meaningful structural recovery as we show in Section 3.5 Despite the practical transformer's non-linearity, we argue that this simple method resembling the factorization based models enable meaningful structure recovery, of which the effectiveness is validated with our case studies. In addition, the rewriting error can be reduced via higher-order expansions with jets as we present the method in a follow-up work of this chapter [Chen et al., 2024], where we propose to use jets expansions to

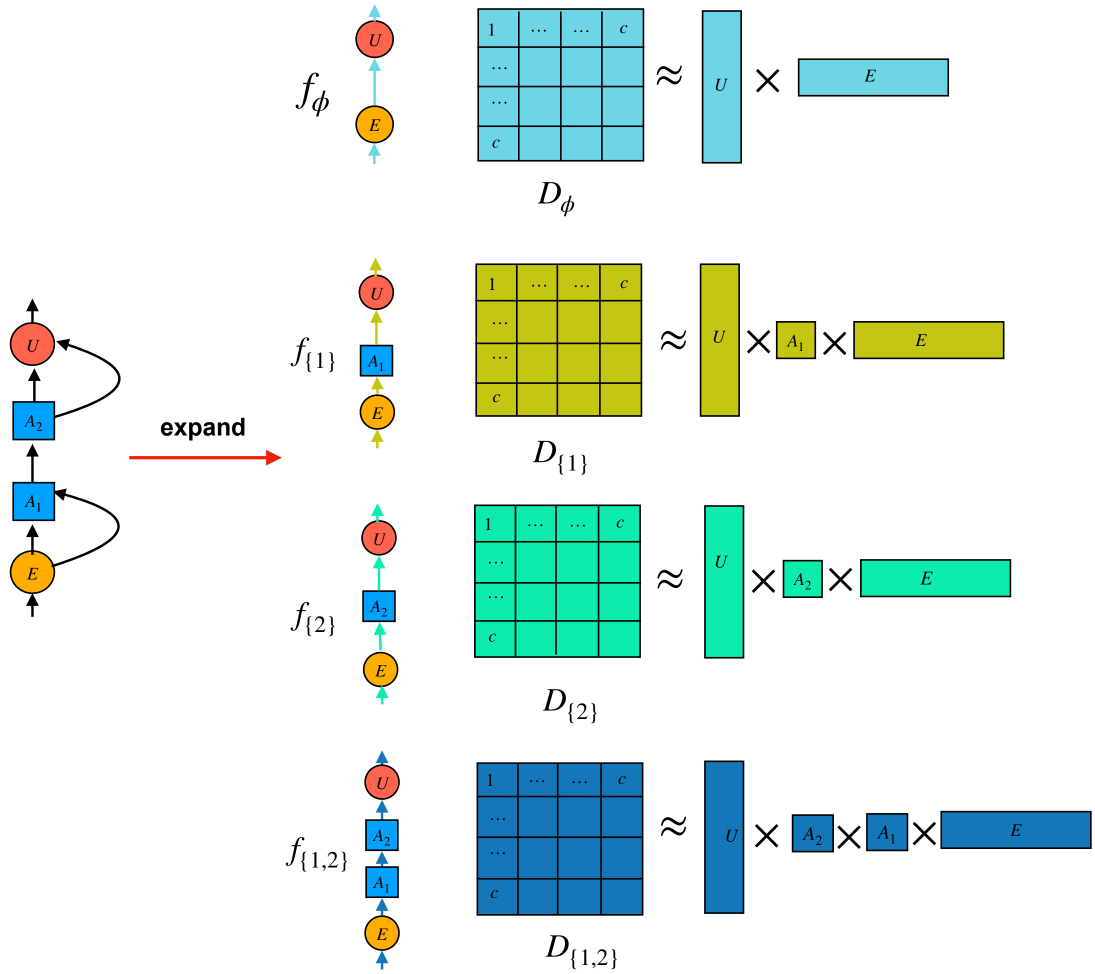

Figure 3.4: Cartoon of the process of deriving bigram databases D_S from the embedded factorization model in each expanded input-output path f_S for a two-layer recursive residual net. For example, $D_{\{1\}}$ is derived from the path $f_{\{1\}}$. These bigram databases can be used to depict their corresponding paths to a certain extent.

handle non-linearities.

3.4 Extracting N-gram Structures from Pretrained Language Models

Now that we have established that factorization models can be pinpointed within (linear) transformers, we can extract symbolic knowledge bases systematically from pretrained language models. These knowledge bases, represented in n-gram formats, can be used to analyse structural information captured by large language models, thus bridging the gap between arithmetic computations (e.g. matrix multiplications) and interpretable structures (e.g. domain keywords or other semantically meaningful units). As stated above, the practical transformer contains non-linear components such as normalization function before input to each module. Implementation-wise, we chose to incorporate these normalization functions into the input-output paths, and empirically we find these non-linearities improve the quality of the extracted bigrams compared to using purely linear paths [Elhage et al., 2021].

This section details our algorithms for extracting n-gram knowledge bases from the factorization models embedded in transformer-based LLMs, specifically on unigrams, bigrams, and trigrams. Due to computational constraints, higher-order ngrams with $n > 3$ are left for future work. Positional embeddings and the discussion on their choices (absolute learnable positional embeddings v.s. relative positional embeddings) are also excluded to avoid additional complexities beyond this study's scope.

3.4.1 Bigrams

We focus on bigrams, as they are the first studied in the literature [Elhage et al., 2021]. Algorithm 1 outlines our approach to computing pairwise token interaction scores for bigrams using token embeddings (E), an unembedding matrix (U), and paths through selected network components. The algorithm can be extended to accommodate any computational path among the 2^L possible paths through the transformer blocks. In this study, we consider the following path options and use OLMo [Groeneveld et al., 2024] as a demonstrative model in the algorithm:

1. **Direct Path**: This path processes embeddings directly without intermediate transformations, as described in Elhage et al. [2021]. Additionally, our algorithm incorporates the non-linearities presented in the OLMo architecture. The token embeddings (E) are normalized using RMS normalization (RMSNorm), and the normalized embeddings are projected onto the unembedding space to compute the interaction scores, represented as $D_{T,T+1}$. This bigram database corresponds to the path represented as f_ϕ.

2. **Single FFN Path**: This path includes a single feed-forward network (FFN) block into the direct path. The token embeddings are first normalized using RMSNorm, passed through the FFN, and normalized again. The resulting embeddings are projected onto the unembedding space to compute the interaction scores. This bigram database corresponds to the path represented as $f_{\{FFN_i\}}$.

3. **Merged Path with Multiple FFNs**: This option allows merging a list of selected FFNs along with the direct path. This bigram database corresponds to the path represented as $f_{\{FFN_{i_1},\ldots,FFN_{i_m}\}}$. For this path:

   (a) An accumulation tensor (e) is initialised with the normalized embeddings ($e \leftarrow \text{RMSNorm}(E, \epsilon)$).

   (b) For each selected FFN in the set, embeddings are normalized, processed through the FFN, and normalized again. The FFN outputs are accumulated into e.

   (c) After processing all selected FFNs, the final interaction score is computed as $D_{T,T+1}$, normalized by the number of FFNs plus one direct path ($|\mathcal{C}| + 1$).

In all paths, a SoftMax operation is applied to the unnormalised scores $D_{T,T+1}$ along the first dimension, ensuring interpretability as probabilities. In essence, the algorithm evaluates these paths over the vocabulary space by *wrapping* the selected components *with* the token embeddings (E) and the unembedding matrix (U). The final output is a 2D tensor $D_{T,T+1}$ that captures the pairwise interactions between tokens T and $T + 1$. This tensor serves as a quantitative approximation of a bigram statistic $\mathbb{P}_q(z_{T+1}|z_T, \ldots)$, revealing the token interaction dynamics embedded in the selected path(s). This bigram algorithm can be extended to encompass the full residual computation rather than focusing on partial computations. We refer to the results derived from this specific path choice as naive bigrams. However, naive bigrams have limitations: they cannot describe arbitrary paths of interest, nor do they facilitate the analysis of path contributions to model behaviour. Therefore, we skip them in the empirical study.

Algorithm 1: Bi-gram Score. Compute 2-gram token interaction graph embedded in embeddings, unembeddings and FFNs. Applicable to the OLMo architecture with vanilla attention and non-parametric RMSNorm

```
Input: Token embeddings E, unembedding matrix U, path option p, a set of
       components C along the specified path
Output: D_{T,T+1}, a 2D tensor of pairwise token interactions
Function bigram(E, U, p, C):
    if p is direct path then
        x ← RMSNorm(E, ε) ;                    // Apply RMS normalization
        D_{T,T+1} ← xU^⊤ ;                     // Project onto unembeddings
    else if p is single FFN path then
        x ← RMSNorm(E, ε);
        x ← FFN(x);
        x ← RMSNorm(x, ε);
        D_{T,T+1} ← xU^⊤;
    else if p includes Feed-Forward Networks (FFNs) then
        e ← RMSNorm(E, ε) ;                    // Initialize accumulation
        foreach FFN ∈ C do
            // Normalize embeddings for FFN computation
            x ← RMSNorm(E, ε);
            // Perform FFN computation
            x ← FFN(x);
            // Normalize FFN output and accumulate
            x ← RMSNorm(x, ε);
            e ← e + x;
        // Compute final interaction score across layers
        D_{T,T+1} ← eU^⊤;
        D_{T,T+1} ← D_{T,T+1} / (|C| + 1);
    Apply softmax on D_{T,T+1} along dimension 1;
    return D_{T,T+1}
```

3.4.2 Extension to Unigrams

Unigrams can be obtained via finding the stable state of the Markov transition equation defined via the bigrams conditional probability (Algorithm 2). The algorithm calculates unigram scores by first deriving the Markov transition matrix from bigram probabilities using the direct path, then performing an eigendecomposition to identify the steady-state eigenvector ($\lambda = 1$), which represents the unigram probabilities, and finally returning this as the unigram score.

Algorithm 2: Unigram Score. Applicable to the OLMo architecture with vanilla attention and non-parametric RMSNorm.

Input: Embeddings E, Unembeddings U, RMSNorm constant ϵ

Output: D_{T+1}, a 1D tensor storing individual token score, representing their prominence within the model.

Function `unigram`(E, U, ϵ)**:**

Obtain transitions $D_{T,T+1} \leftarrow$ `bigram`$(E, U,$ *direct path*$, \emptyset)$;

Initialize the steady state D_{T+1} as a 1D zero tensor;

Compute eigenvalues and eigenvectors $\{\lambda_i\}, \{\mu_i\} \leftarrow$ `eigen_decompose`$(D_{T,T+1})$;

`// Loop over eigenvalues to identify the stable state`

foreach λ_i, μ_i *in* $\{\lambda_i\}, \{\mu_i\}$ **do**

  if $\lambda_i == 1$ **then**

    $D_{T+1} \leftarrow \mu_i$;

return D_{T+1};

3.4.3 Extension to Trigrams

Calculating trigrams or skip n-grams becomes more nuanced because it requires unpacking the mechanism of **self-attention modules**.

Self-Attention: Beyond Immediate Tokens Self-attention enables a model to attend to tokens beyond just the immediate neighbours (e.g., bigrams). By applying one self-attention layer, the model collects information from tokens farther away in the sequence.

For instance:

- **Predicting Token $T+1$:** Using the representation at position T, one self-attention allows the model to attend to any previous token k ($k < T$). The information flow can be represented as:

$$T+1 \underbrace{\leftarrow}_{\text{time step}} T \underbrace{\leftarrow}_{\text{time step}} k$$

  Here, T passes relevant context from k to $T+1$, creating a chain of dependencies over time steps.

Skip N-Grams: Information Steps The above equation uses time steps as the coordinates for a stream of tokens. However, a different coordinate axis will reveal more informative reliance among tokens. Skip n-grams view the same information flow from an **information step** perspective, rather than a time step. For instance, the skip trigram process looks like this:

$$n+1 \quad \underbrace{\leftarrow}_{\text{information step}} \quad n \quad \underbrace{\leftarrow}_{\text{information step}} \quad n-1$$

In this view:

- n carries relevant context from $n-1$ to $n+1$.
- This contrasts with bigrams, where $n-1$ passes information directly to $n+1$ without intermediary steps.

Identifying such patterns embedded in the model can be useful to understand what kind of knowledge is being stored in the model.

Example: Skip N-Grams in a Sentence Consider the sentence: "Lemma (Properties of Jets) Let s be the function to be approximated." If there is a sufficient number of similar sentences in the training dataset, for example the training dataset contains heavy portion of maths texts, then the model would capture skip-trigrams like:

- Token z_{n-1}: "Lemma"
- Token z_n: "Let"
- Token z_{n+1}: "s"

Connecting Self-Attention with Skip Trigrams We can obtain skip trigram statistics relating to $\mathbb{P}_q(z_n|z_{n-1}, \dots, z_{n-2}, \dots)$, where dots indicate any number of interceding tokens, by focusing on paths that contain one self-attention module and possibly filtering out all paths that involve more than one self-attention. In general, paths with more self-attentions will have higher n.

Algorithm 3 describes in detail how we obtain the trigrams. During the calculation of the attention score between token T and k, the current token T becomes a bucket for storing several contextual token k along with their weightings, and pass them later to the target token $T+1$ with weighting. The big 3D tensor for describing triplet interactions among $(k, T, T+1)$ is decomposed into matrices from two steps $T \to k$ and $k \to T+1$. In other words, we trace the indirect influence of each context token k's onto the $(T, T+1)$ pairings by performing a non-contracted tensor product[5] between the $T \to k$ messaging matrix and the $k \to T+1$ messaging matrix.

Such n-gram statistics extracted directly from large language models can serve as a *data-free* tool to sketch LLMs via casting them into (symbolic) n-gram databases. Thus, they allow us to perform symbolic model comparison between *any* two models that share a common vocabulary, as opposed to taking differences in the parameter space, which is harder to interpret and only possible for models with the same architecture.

[5] It is interesting to see the non-contracted tensor products become the key operators for unpacking transformer computation and derive interpretable structures. Its contracted version, matrix products, works well when training deep neural networks on GPUs, where the SIMD paradigm prefers massive parallel ALU computation and accumulating the intermediate computation results rather than caching them all in memory and sequencing the computation. However, when we move to the interpreting neural network phase, it seems that accumulating the intermediate results all the way forward, i.e. the "deep" computation, can be less relevant compared to the "wide" computation, where non-contracted tensor product can keep track of all combinations of the indices – in language models indices correspond to tokens – without reducing them via summation. With "wide" operators like non-contracted tensor product, we can capture global information flow inside the entire vocabulary space, without collapsing higher-order token interactions. The drawback is that it requires large amounts of memory to store all the interactions. We foresee that there is a hardware lottery [Hooker, 2021] for language models interpretability akin to how training deep language models favors GPUs. For example, in this chapter, we do not use any GPUs but adopt CPUs with 1 TB memory.

Algorithm 3: Trigram Score. Compute 3-gram token interaction graph embedded in a self-attention layer via sparsely joining all attention heads. Applicable to the OLMo architecture with vanilla attention and non-parametric RMSNorm

Input: embeddings E, unembeddings U, attention weights W_q, W_k, W_v, W_o, RMSNorm constant ϵ, head size D_h, target head indices `heads`,
Output: $D_{T,k,T+1}$: a sparse 3D tensor storing interactions
$e \leftarrow$ `RMSNorm(`E`,` ϵ`)`;
Initialize $D_{T,k,T+1}$ as zero tensor;
for $h \in$ ***heads*** **do**
  Obtain current head dimensions $H = [hD_h : (h+1)D_h]$;
  Obtain QK matrix $W \leftarrow W_q^T{}_{[:,H]} W_{k[H,:]}$;
  Obtain OV matrix $V \leftarrow W_v^T{}_{[:,H]} W_{o[H,:]}$;
  Compute QK message $D_{T,k} \leftarrow \frac{eWe^T}{\sqrt{D_h}}$;
  Apply softmax normalization on $D_{T,k}$ along dimension 1;
  Sparsify $D_{T,k}$ based on threshold to obtain sparse tensor $\tilde{D}_{T,k}$;
  Compute $D_{k,T+1} \leftarrow$ `RMSNorm`$(eV, \epsilon) \cdot U^T$;
  Apply softmax normalization on $D_{k,T+1}$ along dimension 1;
  Sparsify $D_{k,T+1}$ based on threshold to obtain sparse tensor $\tilde{D}_{k,T+1}$;
  Compute $D^{(h)}_{T,k,T+1} \leftarrow$ `non_contracted_tsr_prod(`$\tilde{D}_{T,k}, \tilde{D}_{k,T+1}$`)`;
  Accumulate $D_{T,k,T+1} \leftarrow D_{T,k,T+1} + D^{(h)}_{T,k,T+1}$;
`// weighting trigrams with bigrams`
Compute $D_{T,T+1} \leftarrow$ `bigram(`E, U, ϵ`)`;
Compute $D_{T,k,T+1} \leftarrow 32 D_{T,k,T+1} + D_{T,T+1}$;
return $D_{T,k,T+1}$

Algorithm 4: Non-Contracted Tensor Product $A_{i,j} B_{j,k} = C_{i,j,k}$

Input: Two tensors A and B
Output: A 3D tensor C
Function `non_contracted_tsr_prod(`A, B`)`:
  for *each index* i *and* k **do**
    `// if vectorized, an outer product` $A_{[i,:]} \otimes B_{[:,k]}$
    for *each index* j **do**
      Compute $C_{i,j,k} = A_{i,j} \times B_{j,k}$;
  return C

3.5 Case Studies: Latent Structures for Interpreting Language Models

In this section, we explore applications of the uncovered n-gram latent structures. We present several case studies where we utilize the identified structures for understanding and interpreting large language models. To showcase the generality of the structure-revealing method, we conduct experiments with popular open-source large language model families: *Llama* [Touvron et al., 2023a,b, Rozière et al., 2024] and *OLMo* [Groeneveld et al., 2024]. Our experiments run on servers with 1 TB of memory and 128 CPUs. Unlike traditional mechanistic interpretability studies, our method does not rely on GPUs or external datasets for collecting network activation patterns, making it more accessible to resource-constrained communities.

3.5.1 Use Case 1: Analysing LLM Inner Workings

Large language models are notorious for their lack of interpretability [Zhao et al., 2024a]. The lack of interpretability is due to their inherent model complexity and size, made worse by the usual opaque training process and unknown training data. Understanding their inner workings, for example the roles of different components, can help calibrate trust for users to use them appropriately. We showcase how the bigrams and trigrams extracted along user-selected computational paths can help us discover and locate learned associations akin to studies in mechanistic interpretability [Templeton et al., 2024], but without any additional training or inference on external datasets.

Paths of individual components. By examining the representative bigrams that are captured by each MLP path, we find MLPs that might perform special linguistic functions. For example, in *OLMo*-7*B*, the path which passes through the 3rd MLP promotes the addition of the "`-ing`" suffixes to the current token. Similar MLPs with certain linguistic functions are listed in Table 3.1. Note that the relationship between functions and components are not necessarily one-to-one mappings. Particularly we find that the paths through multiple MLPs might work together to complete one linguistic function e.g. MLP 6 and MLP 18 in *Llama*-2-7*B* can add "`-ing`" suffix. One MLP might also do multiple linguistic jobs e.g. MLP 1 in OLMo 7B adding "`-ly`" and "`-_else`" suffixes.

Table 3.1: MLPs in *OLMo-7B* and *Llama-2-7B* performing linguistic functions based on jet bi-grams extracted from the corresponding jet paths. Logit values are computed after intervention.

OLMo-7B

MLP	Role	Δ logit
1	`-ly, -_else`	$-4.19, -3.35$
3	`-ing`	-0.58
9	`-'t`	-9.73
17	`-_than`	-4.26
19	`-s`	-7.42

Llama-2-7B

MLP	Role	Δ logit
6	`-ing`	-14.61
7	`-es`	-3.55
18	`-ing, -ity`	$-9.69, -11.93$
19	`-ly`	-9.14

This echos work on circuit discovery [Conmy et al., 2023, Ferrando and Voita, 2024] and superposition [Elhage et al., 2022], where the role of each component can not easily be dissected and multiple components collaborate to fulfil a function. Table 3.2 reports a role identification study on attention heads in the first self-attention of *OLMo-7B* using trigrams. Specifically, we find heads associated with maths and programming, e.g. head 1 on Maths/latex; heads promoting digits and dash composition into dates, e.g. head 25; and heads constituting phrase templates, e.g. head 15 managing a "for x purposes", where x is a placeholder. To verify the roles we revealed, we further perform preliminary intervention experiments where we ablate MLPs or attention heads and compute variations in model logits. After the interventions, the logits drop consistently for all cases, suggesting our n-grams indeed can help identify roles for selected components. Varying impact on logit differences is likely due to overdetermination [Mueller, 2024] and our partial selection of paths (e.g. for trigrams we only selected encoding-attention-decoding paths, excluding any MLP).

3.5.2 Use Case 2: Analysing Pretraining Dynamics

Pretraining an LLM is usually highly resource-intensive. Therefore, it is crucial to monitor the progress of a pretraining run to prevent wasting of time and compute. In this section, we show how bigrams can serve as an effective signalling tool to trace the pretraining dynamics, providing insights about the model's maturity. Such signals are especially useful to understand what happens with the model when the pretraining loss

Table 3.2: Several attention heads in the first residual block of *OLMo-7B* and their roles identified with jet trigrams extracted from corresponding jet paths. We also include an example trigram captured by each head.

Head Index	Role	Example 3-gram	Δlogit
2	Maths/latex	(_Lemma, _let, _s)	-0.1570
16	"for...purposes"	(_for, _use, _purposes)	-0.0019
26	Date composition	(20, 23, _-)	-0.0093
30	"into account..."	(_into, _account, _possible)	-0.0001

Table 3.3: Bi-gram evolution across pretraining steps for OLMo 7B. Each column represents a distinct step, while each row corresponds to a different rank. The table entries are the bi-grams at each step for each rank. The number of tokens seen in association with the pretraining steps is also annotated. The model gradually picks up meaningful bi-grams after starting from random bi-grams (due to random initialization).

Rank	0K [**#steps**] 0B [**#tokens**]	100K 442B	200K 885B	300K 1327B	400K 1769B	555K 2455B
0	immortal	's	at least	&amp	&amp	&amp
1	ICUirling	at least	's	at least	its own	its own
2	ords architect	its own	&amp	its own	their own	their own
3	yaml Adam	okerly	your own	your own	at least	his own
4	231 next	VENT thanks	its own	their own	your own	make sure
5	clonal	iums	iums	more than	his own	your own
6	Charg@{	you're	you're	can't	2nd	2nd
7	avoir careless	Everything v	2nd	his own	more than	at least
8	HOLD worsening	erna already	you guys	2nd	make sure	more than
9	Horse dismant	'my	more than	make sure	can't	iums

shows marginal improvements and fails to reflect the changes inside the model.

Identifying the top bigrams. To assess the model's progression, we extracted bigrams from *OLMo-7B* model checkpoints across 555K pretraining steps. Table 3.3 presents a summary of the top 10 bigrams at different stages of training. Due to space constraints, we only show the top 10 bigrams every 100K steps. Initially, the network exhibits nonsensical bigrams, such as "ICUirling". As training advances, it gradually learns more meaningful combinations, like "at least". This process of acquiring sensible bigrams stabilizes around step 200K, indicating that the model is reaching a level of maturity where the top 10 bigrams capture common meaning.

Analysing bigram learning speed. To evaluate the learning speed of these bigrams, we consider the bigrams at the final training step (555K) as the ground-truth. We then chart the hit ratios of these ground-truth bigrams at each pretraining step, as illustrated in Figure 3.5. Interestingly, even though the pretraining loss (the blue curve) shows only minor improvements after the initial 50K steps, the model's acquisition of effective bigrams continues to progress in a steady, consistent manner. This observation aligns with known phenomena in neural network training, such as double-descent and grokking, which highlight the model's ability to improve generalization capabilities even when the loss appears to stagnate [Zhang et al., 2021, Power et al., 2022]. In addition, Figure 3.6 characterizes the total pseudo-joint probability mass of top 1K bigrams from empirical data [Liu et al., 2024b]. We derive a pseudo-joint bigram probability using statistical unigrams from [Liu et al., 2024b]. We observe that the model gradually accumulates probability mass that aligns with the real corpus data distribution. Interestingly, although the overall trend is upward, the mass initially rises sharply from zero, then undergoes two noticeable dips before continuing to increase. This non-monotonic behaviour likely reflects distinct stages in the model's learning dynamics. Early in training, the model quickly captures high-frequency bigrams, resulting in the initial surge. As training progresses, it explores a broader range of token combinations, including less frequent or less relevant bigrams, temporarily redistributing probability mass away from the top 1K bigrams and causing the first dip. The second dip may result from further rebalancing, overfitting to mid-frequency patterns, or transient noise in gradient updates. Contributing factors may include optimization dynamics and noise in the training data, which we leave for future investigation. Eventually, the model reallocates probability mass more accurately and converges toward the empirical distribution, resuming its upward trajectory.

Learning schemes for different bigrams. To understand if there are any differences between the learning schemes of different bigrams, we can trace the progression of the bigram scores for selected bigrams. Figure 3.8 provides a visual comparison of how different bigrams are promoted or suppressed during the pretraining process. We analyse bigrams that exhibit different mapping relationships between the first and second tokens,

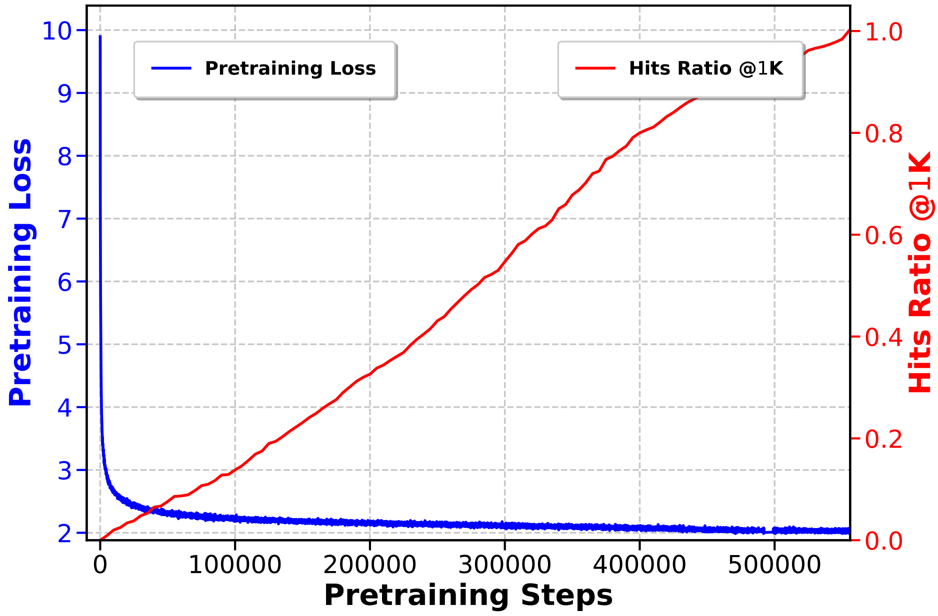

Figure 3.5: Top 1K bigram hit ratios w.r.t. the final step.

inspired by the one-to-one, one-to-many, and many-to-many relational analysis in the knowledge graph literature [Lacroix et al., 2018]. For example, "at least" is a few-to-many bigram: there are many possible tokens that can follow "at", but relatively few that commonly precede "least". The different slopes and levels of the lines indicate varying rates of learning for the respective bigrams. We observe that, the model first acquires random bigrams due to random parameter initialisation. These random bigrams, like "`ICUirling`" and "`VENT thanks`", are quickly suppressed in the early steps and never regain high scores. In contrast, few-to-many bigrams like "`at least`" are first promoted to very high scores but then get suppressed perhaps due to the model seeing more of the scope of the token "at". One-to-one bigrams like "`&amp`" (HTML code) are gradually promoted and stabilize. Many-to-many bigrams like "`make sure`" takes the most time to learn, and the scores are still increasing even at the end of pretraining. Our findings suggest that the training process effectively promotes certain "good" bigrams, but at different paces, where they might be suppressed later depending on their occurrences and linguistic nature. These insights could inform future training strategies, such as targeted training on more relevant bigrams or adjusting the training data to improve the pretraining speed.

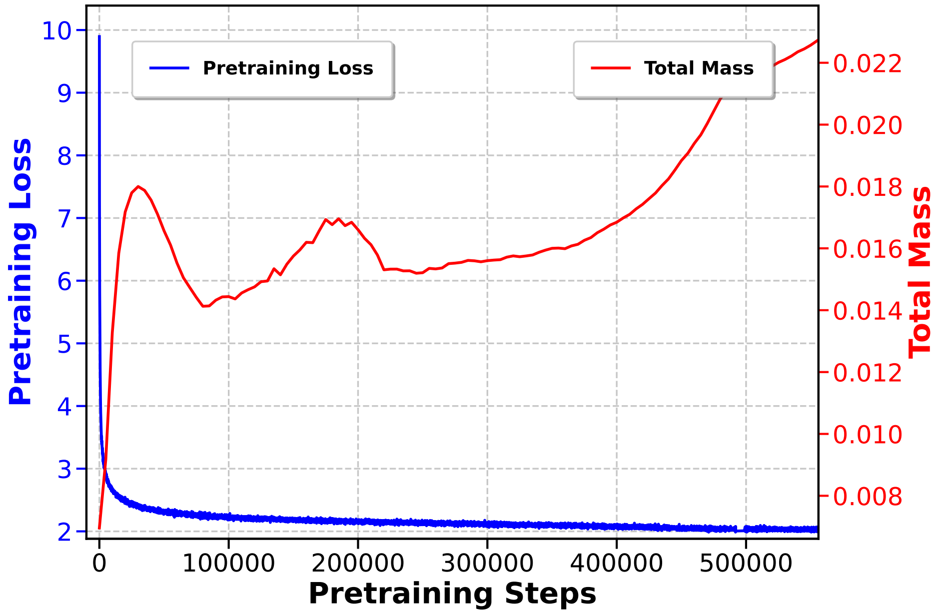

Figure 3.6: Top 1K bigram mass w.r.t. empirical data.

Figure 3.7: Analysis of *OLMo-7B*'s pretraining dynamics by measuring its bigram progression.

3.5.3 Use Case 3: Analysing Finetuning Effects

Finetuning is an important phase where the raw pretrained LLMs are guided to perform particular tasks. We would like to understand how the model inner knowledge changes during finetuning processes. While "parameter diff" can be a straightforward solution, n-grams provides an alternative approach, where the diffs are human-readable and directly reflect the change of knowledge retained by the LLMs, similar to how a `diff` command would work in Linux platforms. Such insights would allow us to better decide the mixture of data for finetuning, and the number of steps for finetuning, which are currently a mix of heuristics and trial-and-error.

Code finetuning promotes coding-relevant bigrams. We analyse the changes due to code finetuning via *diffing* bigrams extracted from *Llama-2-7B* and its finetuned versions, *Codellama-7B* and *Codellama-Python-7B*. As highlighted in Table 3.4 with orange coloring, the bigram comparison reveals coding-relevant keywords, such as "`**kwargs`", "`getters`" and "`Assertion`", suggesting bigrams can be a tool for verifying if finetun-

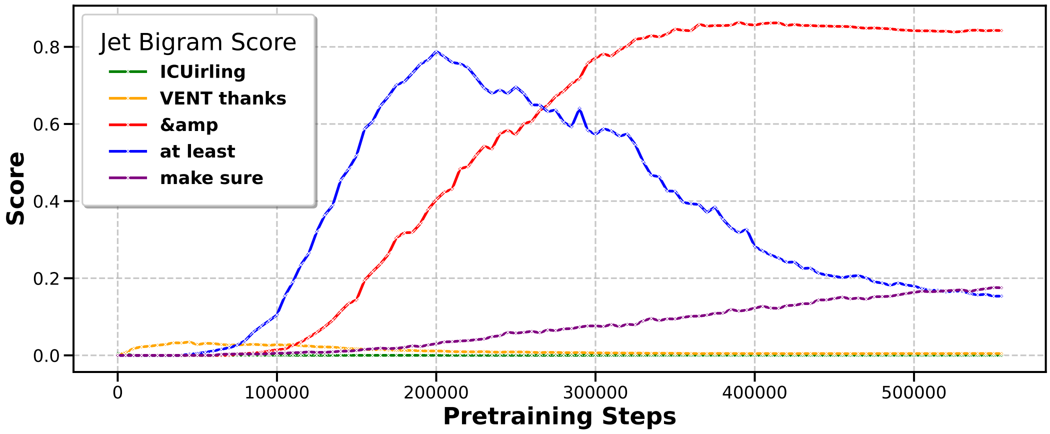

Figure 3.8: Visualization of *OLMo-7B*'s promotion and suppression dynamics of bigrams scores.

ing is effective in acquiring relevant knowledge.

Does RLHF finetuning remove toxicity? We compare the raw pretrained model, *Llama-2-7B*, with its RLHF version, *Llama-2-7B-Chat*. RLHF alignment [Bai et al., 2022] is widely believed to detoxify LLMs, as indicated by *ToxiGen* scores [Hartvigsen et al., 2022]. However, it remains easy to prompt LLMs to bypass this alignment and produce toxic content [Yi et al., 2024]. In Table 3.5, we demonstrate this with dataset-based toxicity scores on a subset of challenging prompts in the *RealToxicityPrompts* (RTP) dataset [Gehman et al., 2020]: the gap in toxicity potential between the two models *narrows* as we prepend to RTP prompts increasingly "explicit" (short) context. Specifically, for hard context, *Llama-2-7B-Chat* shows a 84% probability of producing toxic content, close to that of *Llama-2-7B*. This suggests that the RLHF model is not completely detoxified but rather hides the toxicity knowledge from the "surface", which however can be easily triggered by specific contexts. To quantify the toxicity knowledge embedded in these models, we use bigram probability scores and calculate the cumulative conditional probability mass for a set of "toxic" bigrams, which are combinations of tokens associated with toxic meanings from a predefined list of keywords. Interestingly, we observe a small change in mass from 0.03445 to 0.03377 after RLHF. Thus, although *ToxiGen* score may suggest that the model has been effectively detoxified, the bigram mass reflects retention of toxic knowledge after RLHF, aligning with the scores obtained by introducing medium or hard explicit context and computing a toxicity score (via a second scorer

Table 3.4: The bi-grams before and after code fine-tuning. For space constraints, we only show the bi-grams at every 50 ranks among the top 1,000 bi-grams. We highlight the bi-grams that are relevant to coding, such as "**kwargs" a keyword in Python programming. This demonstrates that our method has the capability to extract representative bi-grams that reflect fine-tuning quality.

Rank	LLAMA2-7B	CodeLLAMA-7B	CodeLLAMA-Python-7B
0	(_more, _than)	(_like, wise)	(_like, wise)
50	(_Now, here)	(_just, ification)	(_Like, wise)
100	(_system, atically)	(_in, _case)	(_all, udes)
150	(_all, erg)	(_get, ters)	(_no, isy)
200	(_on, ions)	(któber, s)	(output, ted)
300	(_other, world)	(_all, ud)	(Object, ive)
350	(_Just, ified)	(gebiet, s)	(_as, cii)
400	(_trust, ees)	(_Protest, s)	(_can, nab)
450	(_at, he)	(_deploy, ment)	(_transport, ation)
500	(_book, mark)	(Class, room)	(Tag, ging)
550	(_from,)	(_access, ory)	(_personal, ized)
600	(_WHEN, ever)	(_In, variant)	(_excess, ive)
650	(_where, about)	(_I, _am)	(_Add, itional)
700	(ag, ged)	(add, itionally)	(_**, kwargs)
750	(_he, he)	(_invalid, ate)	(name, plates)
800	(_all, anto)	(div, ision)	(_select, ive)
850	(_Tom, orrow)	(_process, ors)	(_Assert, ions)
900	(_for, ays)	(_Program, me)	(blog, ger)
950	(_Bach, elor)	(_set, up)	(_can, cellation)

model, [Hanu and Unitary team, 2020]) on *RealToxicityPrompts* dataset [Gehman et al., 2020]. This showcases a potential application of bigrams in constructing *data-free* indices that reveal embedded knowledge, offering complimentary views beyond traditional data-driven benchmark evaluations.

3.6 Discussion

Limitations. Isolating partial computations out of the original transformer computation graph can be seen as a truncated Taylor approximation problem, where the center is the portion we want to single out and the variate is the rest of the computation [Chen et al., 2024]. This chapter does not dive into the details of such approximation but rather

Table 3.5: Toxicity indexes for *Llama*-2-7*B* and *Llama*-2-7*B-chat* using different methods: *ToxiGen*, jet bi-grams, and *RealToxicityPrompts* challenge prompting. Higher numbers indicate higher toxicity scores on the corresponding benchmarks and higher toxic knowledge possession for jet bi-grams.

Metric	*Llama*-2-7*B*	*Llama*-2-7*B-chat*
Standard Benchmarking		
ToxiGen Score [Hartvigsen et al., 2022]	21.25	0.00
Prompt-based Benchmarking with RTP Challenging Prompting [Gehman et al., 2020]		
No Prompt	38%	23%
Very Mild	49%	35%
Medium	64%	64%
Hard	88%	84%
Data-free Benchmarking		
Jet Bi-gram Mass	0.03445	0.03377

choose to present the parallel with factorization models, where latent structures can be surfaced similarly as in knowledge base completion, echoing Chapter 2. Besides, the structures we consider are fragments of natural languages, rather than factually meaningful entities or relations. There are substantial evidences that LLMs encode real-world factual structures, for example [Petroni et al., 2019] and [Yang et al., 2024], use curated benchmarks to show pretrained language exhibit certain factual reasoning capability. We would explore similar factual structures in our approach in the future. Additionally, the n in the n-gram structures is bounded by the number of self-attention layers to unfold. For example, when no self-attention is used, we observe $n = 2$; adding a single self-attention layer increases this to $n = 3$. We speculate that there exists a systematic relationship between n and the number of self-attention layers, potentially exponential in nature. Finally, we plan to verify the relationship between the found structures and the pretraining data distribution, which requires large computing resources.

Summary. Large language models are sometimes seen as the victory symbol for the unstructured learning paradigm, where structure curation seems no longer necessary for building a powerful artificial intelligence agent – scaling model sizes on larger unstructured textual corpora is the way. This chapter, however, shows that structures are still the critical ingredients even in the large language models and exposing them is helpful for profiling the knowledge within each model checkpoint. Overall, this chapter provides initial evidence that language modelling objectives, though focused on local context and trained on unstructured data, can recognize and encode structural patterns into the transformer model weights. The key in exposing these inherent structures is to observe that transformers, the typical architecture for large language models, contain portions of computations that resemble factorization based models (FMs). Once trained with LM objectives, these portions of computations capture latent structures in the training data. To expose these structures, this chapter dissects these FMs from the monolithic computation graph of the transformer and derive their corresponding bigrams and trigrams. Akin to how structures help recover the knowledge graph in knowledge base completion, this chapter demonstrates that the uncovered n-gram structures in LLMs help reconstruct the linguistic functions acquired via the models, offering an alternative angle to interpret LLMs in a data-free way. Our case studies demonstrate the potential of using extracted n-gram patterns to debug pretraining progress, verify fine-tuning effects, and detect model toxicity. Looking ahead, LLMs could expose two complementary interfaces: a neural interface for training and prediction, and an n-gram-based symbolic interface for inspection, analysis, and control.

Implications. This chapter demonstrates that the same computation, if examined under a new perspective, can lead to new insights that are invisible in the original lens. Using transformers as an example, one view (let us call it the neuron view) is to see it as a special organization of neurons into stacked self-attention and FFNs plus embeddings on both ends; this view allows easy implementations for training on GPUs. Another view (let us call it the behavior view), which is more helpful to interpretability, is to see it as an ensemble of n-gram models describing token transition behavior. Although the neuron view is useful when building the model and training it, it might not be the best level of abstraction for understanding and interpreting model behavior due to the issue of polysemy [Elhage et al., 2022]. We believe that to understand the model better, channelling both the neuron and behaviour view is necessary. Our method provides an initial attempt to do this by reorganizing the neural computations into FMs, which brings structures in behaviours. This new lens enable new findings such that LLMs do not “digest” data points equally – some structures are acquired fast, but the others are always in learning or first learned and then suppressed. These new findings are relevant in the ongoing discourse on AI transparency and trustworthiness.

Bibliography

David Ifeoluwa Adelani, Jade Abbott, Graham Neubig, Daniel D'souza, Julia Kreutzer, Constantine Lignos, Chester Palen-Michel, Happy Buzaaba, Shruti Rijhwani, Sebastian Ruder, et al. Masakhaner: Named entity recognition for african languages. *Transactions of the Association for Computational Linguistics*, 9:1116–1131, 2021.

David Ifeoluwa Adelani, Graham Neubig, Sebastian Ruder, Shruti Rijhwani, Michael Beukman, Chester Palen-Michel, Constantine Lignos, Jesujoba O Alabi, Shamsuddeen H Muhammad, Peter Nabende, et al. Masakhaner 2.0: Africa-centric transfer learning for named entity recognition. In *2022 Conference on Empirical Methods in Natural Language Processing (EMNLP 2022), Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 4488–4508. Association for Computational Linguistics (ACL), 2022.

Khushbu Agarwal, Tome Eftimov, Raghavendra Addanki, Sutanay Choudhury, Suzanne R. Tamang, and Robert Rallo. Snomed2vec: Random walk and poincaré embeddings of a clinical knowledge base for healthcare analytics. *ArXiv*, abs/1907.08650, 2019. URL `https://api.semanticscholar.org/CorpusID:198147334`.

Divyanshu Aggarwal, Ashutosh Sathe, and Sunayana Sitaram. Exploring pretraining via active forgetting for improving cross lingual transfer for decoder language models. *arXiv preprint arXiv:2410.16168*, 2024.

Jethro Akroyd, Sebastian Mosbach, Amit Bhave, and Markus Kraft. Universal digital twin - a dynamic knowledge graph. *Data-Centric Engineering*, 2:e14, 2021. doi: 10.1017/dce.2021.10.

Ibrahim Alabdulmohsin, Hartmut Maennel, and Daniel Keysers. The impact of reinitialization on generalization in convolutional neural networks. *arXiv preprint arXiv:2109.00267*, 2021.

Jesujoba O. Alabi, David Ifeoluwa Adelani, Marius Mosbach, and Dietrich Klakow. Adapting pre-trained language models to African languages via multilingual adaptive fine-tuning. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4336–4349, Gyeongju, Republic of Korea, October 2022. International Committee on Computational Linguistics. URL `https://aclanthology.org/2022.coling-1.382`.

Prithviraj Ammanabrolu and Mark Riedl. Learning knowledge graph-based world models of textual environments. *Advances in Neural Information Processing Systems*, 34: 3720–3731, 2021.

Michael C. Anderson and Justin C. Hulbert. Active forgetting: Adaptation of memory by prefrontal control. *Annual Review of Psychology*, 72(1):1–36, 2021. doi: 10.1146/annurev-psych-072720-094140. URL `https://doi.org/10.1146/annurev-psych-072720-094140`. PMID: 32928060.

Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. Learning to learn by gradient descent by gradient descent. *Advances in neural information processing systems*, 29, 2016.

Alan Ansell, Edoardo Ponti, Anna Korhonen, and Ivan Vulić. Composable sparse fine-tuning for cross-lingual transfer. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1778–1796, 2022.

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, 2020.

Various authors. Wikipedia, the free encyclopedia, 2024. URL https://www.wikipedia.org. A collaboratively edited, free online encyclopedia.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Ivana Balazevic, Carl Allen, and Timothy M. Hospedales. Tucker: Tensor factorization for knowledge graph completion. In *EMNLP/IJCNLP*, 2019.

Pierre Baldi and Peter J Sadowski. Understanding dropout. *Advances in neural information processing systems*, 26, 2013.

Jeffrey Barrett and Kevin JS Zollman. The role of forgetting in the evolution and learning of language. *Journal of Experimental & Theoretical Artificial Intelligence*, 21(4):293–309, 2009.

Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

Shawn Beaulieu, Lapo Frati, Thomas Miconi, Joel Lehman, Kenneth O Stanley, Jeff Clune, and Nick Cheney. Learning to continually learn. *arXiv preprint arXiv:2002.09571*, 2020.

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*, 2023.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 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, 2021. Association for Computing Machinery. ISBN 9781450383097. doi: 10.1145/3442188.3445922. URL https://doi.org/10.1145/3442188.3445922.

Edward L Bennett, Marian C Diamond, David Krech, and Mark R Rosenzweig. Chemical and anatomical plasticity of brain: Changes in brain through experience, demanded by learning theories, are found in experiments with rats. *Science*, 146(3644):610–619, 1964.

Leonard Bereska and Efstratios Gavves. Mechanistic interpretability for ai safety–a review. *arXiv preprint arXiv:2404.14082*, 2024.

JM Bernardo, MJ Bayarri, JO Berger, AP Dawid, D Heckerman, AFM Smith, and M West. Generative or discriminative? getting the best of both worlds. *Bayesian statistics*, 8(3):3–24, 2007.

Jacob A Berry, Dana C Guhle, and Ronald L Davis. Active forgetting and neuropsychiatric diseases. *Molecular Psychiatry*, pages 1–11, 2024.

Tarek R Besold, Sebastian Bader, Howard Bowman, Pedro Domingos, Pascal Hitzler, Kai-Uwe Kühnberger, Luis C Lamb, Priscila Machado Vieira Lima, Leo de Penning, Gadi Pinkas, et al. Neural-symbolic learning and reasoning: A survey and interpretation 1. In *Neuro-Symbolic Artificial Intelligence: The State of the Art*, pages 1–51. IOS press, 2021.

Tolga Bolukbasi, Adam Pearce, Ann Yuan, Andy Coenen, Emily Reif, Fernanda Viégas, and Martin Wattenberg. An interpretability illusion for bert. *arXiv preprint arXiv:2104.07143*, 2021.

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, J. Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *NIPS*, 2013.

Léon Bottou. *Stochastic Gradient Descent Tricks*, pages 421–436. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN 978-3-642-35289-8. doi: 10.1007/978-3-642-35289-8_25. URL `https://doi.org/10.1007/978-3-642-35289-8_25`.

Thorsten Brants, Ashok Popat, Peng Xu, Franz Josef Och, and Jeffrey Dean. Large language models in machine translation. In *Proceedings of the 2007 Joint Conference* *on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 858–867, 2007.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Richard E Brown. Hebb and cattell: The genesis of the theory of fluid and crystallized intelligence. *Frontiers in human neuroscience*, 10:606, 2016.

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. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf`.

Jerome Bruner. *The Process of Education*. Harvard University Press, 1960.

Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. Imbalanced learning: A comprehensive evaluation of resampling methods for class imbalance. *arXiv preprint arXiv:1710.05381*, 2018. URL `https://arxiv.org/abs/1710.05381`.

Raymond B Cattell. Theory of fluid and crystallized intelligence: A critical experiment. *Journal of educational psychology*, 54(1):1, 1963.

Yihong Chen, Bei Chen, Xiangnan He, Chen Gao, Yong Li, Jian-Guang Lou, and Yue Wang. λopt: Learn to regularize recommender models in finer levels. In *KDD 2019 (Oral), Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 978–986, 2019.

Yihong Chen, Pasquale Minervini, Sebastian Riedel, and Pontus Stenetorp. Relation prediction as an auxiliary training objective for improving multi-relational graph representations. In *AKBC 2021*, 2021.

Yihong Chen, Pushkar Mishra, Luca Franceschi, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Refactor gnns: Revisiting factorisation-based models from a message-passing perspective. In *Advances in Neural Information Processing Systems*, 2022.

Yihong Chen, Kelly Marchisio, Roberta Raileanu, David Ifeoluwa Adelani, Pontus Stenetorp, Sebastian Riedel, and Mikel Artetxe. Improving language plasticity via pretraining with active forgetting. In *NeurIPS 2023*, 2023.

Yihong Chen, Xiangxiang Xu, Yao Lu, Pontus Stenetorp, and Luca Franceschi. Jet expansions of residual computation, 2024. URL `https://arxiv.org/abs/2410.06024`.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Brandon C Colelough and William Regli. Neuro-symbolic ai in 2024: A systematic review. 2024.

Together Computer. Redpajama dataset. `https://www.together.xyz/blog/redpajama`, 2023. Accessed: 2023-12-12.

Arthur Conmy, Augustine Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 16318–16352. Curran Associates, Inc.,

2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/34e1dbe95d34d7ebaf99b9bcaeb5b2be-Paper-Conference.pdf.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1269. URL https://aclanthology.org/D18-1269.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.747. URL https://aclanthology.org/2020.acl-main.747.

Ionut Constantinescu, Tiago Pimentel, Ryan Cotterell, and Alex Warstadt. Investigating critical period effects in language acquisition through neural language models. *arXiv preprint arXiv:2407.19325*, 2024.

OpenWebText Contributors. The openwebtext dataset. https://github.com/jcpeterson/openwebtext, 2019. Accessed: 2023-12-12.

Moheb Costandi. *Neuroplasticity*. MIT Press, 2016.

Common Crawl. Common crawl corpus. https://commoncrawl.org, 2023. Accessed: 2023-12-12.

Marc Peter Deisenroth, A Aldo Faisal, and Cheng Soon Ong. *Mathematics for machine learning*. Cambridge University Press, 2020.

Gilles Deleuze and Paul Patton. *Difference and Repetition*. Athlone, London, 1994.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, 2019.

Kevin P. Donnelly. Snomed-ct: The advanced terminology and coding system for ehealth. *Studies in health technology and informatics*, 121:279–90, 2006. URL https://api.semanticscholar.org/CorpusID:22491040.

Pierluca D'Oro, Max Schwarzer, Evgenii Nikishin, Pierre-Luc Bacon, Marc G Bellemare, and Aaron Courville. Sample-efficient reinforcement learning by breaking the replay ratio barrier. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=OpC-9aBBVJe.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(61): 2121–2159, 2011. URL http://jmlr.org/papers/v12/duchi11a.html.

David Steven Dummit, Richard M Foote, et al. *Abstract algebra*, volume 3. Wiley Hoboken, 2004.

Abteen Ebrahimi, Manuel Mager, Arturo Oncevay, Vishrav Chaudhary, Luis Chiruzzo, Angela Fan, John Ortega, Ricardo Ramos, Annette Rios Gonzales, Ivan Meza-Ruiz, et al. Americasnli: Evaluating zero-shot natural language understanding of pretrained multilingual models in truly low-resource languages. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6279–6299, 2022.

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. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. https://transformer-circuits.pub/2021/framework/index.html.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition. *Transformer Circuits Thread*, 2022. https://transformer-circuits.pub/2022/toy_model/index.html.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, 2024.

Javier Ferrando and Elena Voita. Information flow routes: Automatically interpreting language models at scale. *arXiv preprint arXiv:2403.00824*, 2024.

Javier Ferrando, Gabriele Sarti, Arianna Bisazza, and Marta R Costa-jussà. A primer on the inner workings of transformer-based language models. *arXiv preprint arXiv:2405.00208*, 2024.

Matthias Fey, Jan Eric Lenssen, Frank Weichert, and Jure Leskovec. Gnnautoscale: Scalable and expressive graph neural networks via historical embeddings. In *ICML*, 2021.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=rJl-b3RcF7`.

Eberhard Fuchs and Gabriele Flügge. Adult neuroplasticity: more than 40 years of research. *Neural plasticity*, 2014(1):541870, 2014.

A Garcez, M Gori, LC Lamb, L Serafini, M Spranger, and SN Tran. Neural-symbolic computing: An effective methodology for principled integration of machine learning and reasoning. *Journal of Applied Logics*, 6(4):611–632, 2019.

Jean-Baptiste Gaya, Thang Doan, Lucas Caccia, Laure Soulier, Ludovic Denoyer, and Roberta Raileanu. Building a subspace of policies for scalable continual learning. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=UKr0MwZM6fL`.

Floris Geerts and Juan L Reutter. Expressiveness and approximation properties of graph neural networks. In *International Conference on Learning Representations*, 2021.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. RealToxicityPrompts: Evaluating neural toxic degeneration in language models. In Trevor Cohn, Yulan He, and Yang Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3356–3369, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.301. URL `https://aclanthology.org/2020.findings-emnlp.301`.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020. URL `https://arxiv.org/abs/2004.07780`.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.446. URL `https://aclanthology.org/2021.emnlp-main.446`.

Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 30–45, 2022.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.

Nicholas Goldowsky-Dill, Chris MacLeod, Lucas Sato, and Aryaman Arora. Localizing model behavior with path patching. *arXiv preprint arXiv:2304.05969*, 2023.

Siavash Golkar, Micheal Kagan, and Kyunghyun Cho. Continual learning via neural pruning. In *Real Neurons* $\{\backslash\&\}$ *Hidden Units: Future directions at the intersection of neuroscience and artificial intelligence@ NeurIPS 2019*.

Joshua T Goodman. A bit of progress in language modeling. *Computer Speech & Language*, 15(4):403–434, 2001.

Marco Gori, Gabriele Monfardini, and Franco Scarselli. A new model for learning in graph domains. *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, 2:729–734 vol. 2, 2005.

C Shawn Green and Daphne Bavelier. Exercising your brain: a review of human brain plasticity and training-induced learning. *Psychology and aging*, 23(4):692, 2008.

Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, et al. Olmo: Accelerating the science of language models. *arXiv preprint arXiv:2402.00838*, 2024.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, 2020.

Axel Guskjolen and Mark S Cembrowski. Engram neurons: Encoding, consolidation, retrieval, and forgetting of memory. *Molecular psychiatry*, 28(8):3207–3219, 2023.

Kyle Hamilton, Aparna Nayak, Bojan Božić, and Luca Longo. Is neuro-symbolic ai meeting its promises in natural language processing? a structured review. *Semantic Web*, 15(4):1265–1306, 2024.

William L. Hamilton. Graph representation learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 14(3):1–159.

William L Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 1025–1035, 2017.

Laura Hanu and Unitary team. Detoxify. Github. https://github.com/unitaryai/detoxify, 2020.

Oliver Hardt, Einar Örn Einarsson, and Karim Nader. A bridge over troubled water: Reconsolidation as a link between cognitive and neuroscientific memory research traditions. *Annual review of psychology*, 61(1):141–167, 2010.

Oliver Hardt, Karim Nader, and Lynn Nadel. Decay happens: the role of active forgetting in memory. *Trends in cognitive sciences*, 17(3):111–120, 2013.

Michael Hart and Project Gutenberg Volunteers. Project gutenberg online library, 1971–2024. URL `https://www.gutenberg.org`. Free eBooks from the public domain.

Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. ToxiGen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3309–3326, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.234. URL `https://aclanthology.org/2022.acl-long.234`.

Felix Hausdorff. *Set theory*, volume 119. American Mathematical Soc., 2021.

Frederick Hayes-Roth, Donald A Waterman, and Douglas B Lenat. *Building expert systems*. Addison-Wesley Longman Publishing Co., Inc., 1983.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing, 2021a.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*, 2021b. URL https://openreview.net/forum?id=XPZIaotutsD.

Danny Hernandez, Jared Kaplan, Tom Henighan, and Sam McCandlish. Scaling laws for transfer. *arXiv preprint arXiv:2102.01293*, 2021.

Evan Hernandez, Belinda Z Li, and Jacob Andreas. Measuring and manipulating knowledge representations in language models. *arXiv preprint arXiv:2304.00740*, 2023.

F. L. Hitchcock. The expression of a tensor or a polyadic as a sum of products. *J. Math. Phys*, 6(1):164–189, 1927.

S Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation MIT-Press*, 1997.

Sara Hooker. The hardware lottery. *Communications of the ACM*, 64(12):58–65, 2021.

Roy Horan. The neuropsychological connection between creativity and meditation. *Creativity research journal*, 21(2-3):199–222, 2009.

John L Horn and Raymond B Cattell. Refinement and test of the theory of fluid and crystallized general intelligences. *Journal of educational psychology*, 57(5):253, 1966.

Ian Horrocks. Owl: A description logic based ontology language. In *International conference on principles and practice of constraint programming*, pages 5–8. Springer, 2005.

Ian Horrocks, Peter F Patel-Schneider, and Frank van Harmelen. From shiq and rdf to owl: The making of a web ontology language. *Journal of Web Semantics*, 1(1):7–26, 2003. URL https://doi.org/10.1016/j.websem.2003.07.001.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2021.

Ziniu Hu, Yuxiao Dong, Kuansan Wang, Kai-Wei Chang, and Yizhou Sun. Gpt-gnn: Generative pre-training of graph neural networks. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2020.

David Hume. An enquiry concerning human understanding. 1748. *Classics of Western Philosophy*, pages 763–828, 1999.

Jena D. Hwang, Chandra Bhagavatula, Ronan Le Bras, Jeff Da, Keisuke Sakaguchi, Antoine Bosselut, and Yejin Choi. Comet-atomic 2020: On symbolic and neural commonsense knowledge graphs. In *AAAI*, 2021.

Alex Iacob, Lorenzo Sani, Meghdad Kurmanji, William F Shen, Xinchi Qiu, Dongqi Cai, Yan Gao, and Nicholas D Lane. Dept: Decoupled embeddings for pre-training language models. *arXiv preprint arXiv:2410.05021*, 2024.

Maximilian Igl, Gregory Farquhar, Jelena Luketina, Wendelin Boehmer, and Shimon Whiteson. Transient non-stationarity and generalisation in deep reinforcement learning. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=Qun8fv4qSby`.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=6t0Kwf8-jrj`.

Prachi Jain, Sushant Rathi, Mausam, and Soumen Chakrabarti. Knowledge base completion: Baseline strikes back (again). *ArXiv*, abs/2005.00804, 2020a.

Prachi Jain, Sushant Rathi, Mausam, and Soumen Chakrabarti. Knowledge base completion: Baseline strikes back (again). *CoRR*, abs/2005.00804, 2020b. URL `https://arxiv.org/abs/2005.00804`.

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S Yu. A survey on knowledge graphs: Representation, acquisition and applications. *arXiv preprint arXiv:2002.00388*, 2020.

Rudolf Kadlec, Ondrej Bajgar, and Jan Kleindienst. Knowledge base completion: Baselines strike back. In Phil Blunsom, Antoine Bordes, Kyunghyun Cho, Shay B. Cohen, Chris Dyer, Edward Grefenstette, Karl Moritz Hermann, Laura Rimell, Jason Weston, and Scott Yih, editors, *Proceedings of the 2nd Workshop on Representation Learning for NLP, Rep4NLP@ACL 2017, Vancouver, Canada, August 3, 2017*, pages 69–74. Association for Computational Linguistics, 2017. doi: 10.18653/v1/w17-2609. URL `https://doi.org/10.18653/v1/w17-2609`.

Immanuel Kant. *Critique of Pure Reason (1st edition)*. Macmillan Company, Mineola, New York, 1781.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, 2020.

Jill L Kays, Robin A Hurley, and Katherine H Taber. The dynamic brain: neuroplasticity and mental health. *The Journal of neuropsychiatry and clinical neurosciences*, 24(2): 118–124, 2012.

Seyed Mehran Kazemi and David Poole. Simple embedding for link prediction in knowledge graphs. *Advances in neural information processing systems*, 31, 2018.

Charles Kemp, Joshua B. Tenenbaum, Thomas L. Griffiths, Takeshi Yamada, and Naonori Ueda. Learning systems of concepts with an infinite relational model. In *AAAI*, pages 381–388. AAAI Press, 2006.

Phillip Kent. Fluid intelligence: A brief history. *Applied Neuropsychology: Child*, 6(3): 193–203, 2017.

Khimya Khetarpal, Matthew Riemer, Irina Rish, and Doina Precup. Towards continual reinforcement learning: A review and perspectives. *Journal of Artificial Intelligence Research*, 75:1401–1476, 2022.

Byung-Hak Kim, Arvind Yedla, and Henry D Pfister. Imp: A message-passing algorithm for matrix completion. In *2010 6th International Symposium on Turbo Codes & Iterative Information Processing*, pages 462–466. IEEE, 2010.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Hannah Rose Kirk, Bertie Vidgen, Paul Röttger, and Scott A Hale. The benefits, risks and bounds of personalizing the alignment of large language models to individuals. *Nature Machine Intelligence*, 6(4):383–392, 2024.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Jeffrey A Kleim and Theresa A Jones. Principles of experience-dependent neural plasticity: implications for rehabilitation after brain damage. 2008.

Donald Ervin Knuth. *The art of computer programming*, volume 3. Pearson Education, 1997.

Stanley Kok and Pedro M. Domingos. Statistical predicate invention. In *ICML*, volume 227 of *ACM International Conference Proceeding Series*, pages 433–440. ACM, 2007.

Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, 2018.

Timothée Lacroix, Nicolas Usunier, and Guillaume Obozinski. Canonical tensor decomposition for knowledge base completion. In *ICML*, volume 80 of *Proceedings of Machine Learning Research*, pages 2869–2878. PMLR, 2018.

Yann LeCun. A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. *Open Review*, 62(1), 2022.

Seungpil Lee, Woochang Sim, Donghyeon Shin, Wongyu Seo, Jiwon Park, Seokki Lee, Sanha Hwang, Sejin Kim, and Sundong Kim. Reasoning abilities of large language models: In-depth analysis on the abstraction and reasoning corpus. *ACM Trans. Intell. Syst. Technol.*, January 2025. ISSN 2157-6904. doi: 10.1145/3712701. URL `https://doi.org/10.1145/3712701`. Just Accepted.

Su Young Lee, Choi Sungik, and Sae-Young Chung. Sample-efficient deep reinforcement learning via episodic backward update. *Advances in neural information processing systems*, 32, 2019.

Benedetta Leuner and Elizabeth Gould. Structural plasticity and hippocampal function. *Annual review of psychology*, 61(1):111–140, 2010.

Benjamin J Levy, Nathan D McVeigh, Alejandra Marful, and Michael C Anderson. Inhibiting your native language: The role of retrieval-induced forgetting during second-language acquisition. *Psychological Science*, 18(1):29–34, 2007.

Patrick Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. Mlqa: Evaluating cross-lingual extractive question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7315–7330, 2020a.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474, 2020b.

Belinda Z. Li, Maxwell Nye, and Jacob Andreas. Implicit representations of meaning in neural language models. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1813–1827, Online, August 2021a. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.143. URL `https://aclanthology.org/2021.acl-long.143`.

Chenchen Li, Aiping Li, Ye Wang, Hongkui Tu, and Yichen Song. A survey on approaches and applications of knowledge representation learning. In *2020 IEEE Fifth*

International Conference on Data Science in Cyberspace (DSC), pages 312–319. IEEE, 2020.

Ren Li, Yanan Cao, Qiannan Zhu, Guanqun Bi, Fang Fang, Yi Liu, and Qian Li. How does knowledge graph embedding extrapolate to unseen data: a semantic evidence view. *CoRR*, abs/2109.11800, 2021b.

Xinze Li, Zhenghao Liu, Chenyan Xiong, Shi Yu, Yu Gu, Zhiyuan Liu, and Ge Yu. Structure-aware language model pretraining improves dense retrieval on structured data. In *The 61st Annual Meeting Of The Association For Computational Linguistics*, 2023.

Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard S. Zemel. Gated graph sequence neural networks. In *ICLR (Poster)*, 2016.

Adam Liska, Tomas Kocisky, Elena Gribovskaya, Tayfun Terzi, Eren Sezener, Devang Agrawal, Cyprien De Masson D'Autume, Tim Scholtes, Manzil Zaheer, Susannah Young, et al. Streamingqa: A benchmark for adaptation to new knowledge over time in question answering models. In *International Conference on Machine Learning*, pages 13604–13622. PMLR, 2022.

Chen Liu, Jonas Pfeiffer, Anna Korhonen, Ivan Vulić, and Iryna Gurevych. Delving deeper into cross-lingual visual question answering. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 2408–2423, 2023a.

Chunan Liu, Lilian Denzler, Yihong Chen, Andrew Martin, and Brooks Paige. Asep: Benchmarking deep learning methods for antibody-specific epitope prediction. In *NeurIPS 2024, Proceedings of the Thirty-eighth Conference on Neural Information Processing Systems, Datasets and Benchmarks*, 2024a.

Hong Liu, Sang Michael Xie, Zhiyuan Li, and Tengyu Ma. Same pre-training loss, better downstream: Implicit bias matters for language models. In *Proceedings of the 40th International Conference on Machine Learning*, 2023b.

Jiacheng Liu, Sewon Min, Luke Zettlemoyer, Yejin Choi, and Hannaneh Hajishirzi. Infini-gram: Scaling unbounded n-gram language models to a trillion tokens. *arXiv preprint arXiv:2401.17377*, 2024b.

Sijia Liu, Yuanshun Yao, Jinghan Jia, Stephen Casper, Nathalie Baracaldo, Peter Hase, Yuguang Yao, Chris Yuhao Liu, Xiaojun Xu, Hang Li, et al. Rethinking machine unlearning for large language models. *Nature Machine Intelligence*, pages 1–14, 2025.

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

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019b. URL `http://arxiv.org/abs/1907.11692`.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

Jiarui Lu, Thomas Holleis, Yizhe Zhang, Bernhard Aumayer, Feng Nan, Felix Bai, Shuang Ma, Shen Ma, Mengyu Li, Guoli Yin, Zirui Wang, and Ruoming Pang. Toolsandbox: A stateful, conversational, interactive evaluation benchmark for llm tool use capabilities, 2024. URL `https://arxiv.org/abs/2408.04682`.

Scott Lundberg. A unified approach to interpreting model predictions. *arXiv preprint arXiv:1705.07874*, 2017.

Qingsong Lv, Ming Ding, Qiang Liu, Yuxiang Chen, Wenzheng Feng, Siming He, Chang Zhou, Jianguo Jiang, Yuxiao Dong, and Jie Tang. Are we really making much progress? revisiting, benchmarking and refining heterogeneous graph neural networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '21, page 1150–1160, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383325. doi: 10.1145/3447548.3467350. URL `https://doi.org/10.1145/3447548.3467350`.

Clare Lyle, Zeyu Zheng, Evgenii Nikishin, Bernardo Avila Pires, Razvan Pascanu, and Will Dabney. Understanding plasticity in neural networks. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 23190–23211. PMLR, 7 2023. URL https://proceedings.mlr.press/v202/lyle23b.html.

David JC MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

Brian MacWhinney. A unified model of language acquisition. In Judith F. Kroll and Annette M.B. De Groot, editors, *Handbook of Bilingualism: Psycholinguistic Approaches*, pages 49–67. Oxford University Press, 2005.

Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7765–7773, 2018.

Kelly Marchisio, Patrick Lewis, Yihong Chen, and Mikel Artetxe. Mini-model adaptation: Efficiently extending pretrained models to new languages via aligned shallow training. In *ACL 2023, Findings of the Association for Computational Linguistics*, 2023.

Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. volume 24 of *Psychology of Learning and Motivation*, pages 109–165. Academic Press, 1989. doi: https://doi.org/10.1016/S0079-7421(08)60536-8. URL https://www.sciencedirect.com/science/article/pii/S0079742108605368.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35: 17359–17372, 2022.

B. D. Mishra, Niket Tandon, and P. Clark. Domain-targeted, high precision knowledge extraction. *Transactions of the Association for Computational Linguistics*, 5:233–246, 2017.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. Fast model editing at scale. In *International Conference on Learning Representations*, 2021.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn. Memory-based model editing at scale. In *International Conference on Machine Learning*, pages 15817–15831. PMLR, 2022.

Sameh K Mohamed, Vít Nováček, Pierre-Yves Vandenbussche, and Emir Muñoz. Loss functions in knowledge graph embedding models. In *Proceedings of DL4KG2019-Workshop on Deep Learning for Knowledge Graphs*, page 1, 2019.

Aaron Mueller. Missed causes and ambiguous effects: Counterfactuals pose challenges for interpreting neural networks. *arXiv preprint arXiv:2407.04690*, 2024.

Deepak Nathani, Jatin Chauhan, Charu Sharma, and Manohar Kaul. Learning attention-based embeddings for relation prediction in knowledge graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4710–4723, 2019.

Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

Dai Quoc Nguyen, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Q. Phung. A novel embedding model for knowledge base completion based on convolutional neural network. In *NAACL-HLT (2)*, pages 327–333. Association for Computational Linguistics, 2018.

Timothy Nguyen. Understanding transformers via n-gram statistics. *arXiv preprint arXiv:2407.12034*, 2024.

M. Nickel, Volker Tresp, and H. Kriegel. A three-way model for collective learning on multi-relational data. In *ICML*, 2011a.

M. Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104:11–33, 2016a.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *ICML*, pages 809–816. Omnipress, 2011b.

Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A review of relational machine learning for knowledge graphs. *Proc. IEEE*, 104(1):11–33, 2016b.

Maximilian Nickel, Lorenzo Rosasco, and Tomaso A. Poggio. Holographic embeddings of knowledge graphs. In *AAAI*, pages 1955–1961. AAAI Press, 2016c.

Evgenii Nikishin, Max Schwarzer, Pierluca D'Oro, Pierre-Luc Bacon, and Aaron Courville. The primacy bias in deep reinforcement learning. In *International Conference on Machine Learning*, pages 16828–16847. PMLR, 2022.

Evgenii Nikishin, Junhyuk Oh, Georg Ostrovski, Clare Lyle, Razvan Pascanu, Will Dabney, and Andre Barreto. Deep reinforcement learning with plasticity injection. In *Workshop on Reincarnating Reinforcement Learning at ICLR 2023*, 2023. URL `https://openreview.net/forum?id=O9cJADBZT1`.

nostalgebraist. interpreting gpt: the logit lens, 2021. URL `https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens#HEf5abD7hqqAY2GSQ`.

Natasha Noy, Yuqing Gao, Anshu Jain, Anant Narayanan, Alan Patterson, and Jamie Taylor. Industry-scale knowledge graphs: Lessons and challenges: Five diverse technology companies show how it's done. *Queue*, 17(2):48–75, 2019.

Simon Nørby. Why forget? on the adaptive value of memory loss. *Perspectives on Psychological Science*, 10(5):551–578, 2015. doi: 10.1177/1745691615596787. URL `https://doi.org/10.1177/1745691615596787`. PMID: 26385996.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, 2019.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural networks*, 113: 54–71, 2019.

Denise C Park and Chih-Mao Huang. Culture wires the brain: A cognitive neuroscience perspective. *Perspectives on Psychological Science*, 5(4):391–400, 2010.

Bernhard Pastötter, Karl-Heinz Bäuml, and Simon Hanslmayr. Oscillatory brain activity before and after an internal context change—evidence for a reset of encoding processes. *NeuroImage*, 43(1):173–181, 2008.

Judea Pearl. Graphs, causality, and structural equation models. *Sociological Methods & Research*, 27(2):226–284, 1998.

Judea Pearl and Dana Mackenzie. *The book of why: the new science of cause and effect*. Basic books, 2018.

Judea Pearl and Glenn Shafer. Probabilistic reasoning in intelligent systems: Networks of plausible inference. *Synthese-Dordrecht*, 104(1):161, 1995.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 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, 2019.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.617. URL `https://aclanthology.org/2020.emnlp-main.617`.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. Unks everywhere: Adapting multilingual language models to new scripts. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10186–10203, 2021.

Jonas Pfeiffer, Naman Goyal, Xi Lin, Xian Li, James Cross, Sebastian Riedel, and Mikel Artetxe. Lifting the curse of multilinguality by pre-training modular transformers. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3479–3495, 2022.

Jean Piaget. *The Child's Conception of the World.* Harcourt, Brace & World, 1929.

Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets, 2022.

BigCode Project. Starcoder dataset. `https://huggingface.co/bigcode`, 2023. Accessed: 2023-12-12.

Alec Radford and Karthik Narasimhan. Improving language understanding by generative pre-training. 2018.

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

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

Vijaya Raghavan T Ramkumar, Elahe Arani, and Bahram Zonooz. Learn, unlearn and relearn: An online learning paradigm for deep neural networks. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL `https://openreview.net/forum?id=WN1O2MJDST`.

Roger Ratcliff. Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychological review*, 97(2):285, 1990.

Siamak Ravanbakhsh, Barnabás Póczos, and Russell Greiner. Boolean matrix factorization and noisy completion via message passing. In *International Conference on Machine Learning*, pages 945–954. PMLR, 2016.

Michael Reed and Barry Simon. *Methods of modern mathematical physics: Functional analysis*, volume 1. Gulf Professional Publishing, 1980.

Benjamin Reichman and Larry Heck. Dense passage retrieval: Is it retrieving?, 2024. URL `https://arxiv.org/abs/2402.11035`.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. URL `https://arxiv.org/abs/1908.10084`.

Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. In *International Conference on Learning Representations*.

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

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Mark R Rosenzweig. Aspects of the search for neural mechanisms of memory. *Annual review of psychology*, 47(1):1–32, 1996.

Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, and Gabriel Synnaeve. Code llama: Open foundation models for code, 2024.

Daniel Ruffinelli, Samuel Broscheit, and Rainer Gemulla. You can teach an old dog new tricks! on training knowledge graph embeddings. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=BkxSmlBFvr`.

Tomás J Ryan and Paul W Frankland. Forgetting as a form of adaptive engram cell plasticity. *Nature Reviews Neuroscience*, 23(3):173–186, 2022.

Tara Safavi and Danai Koutra. Codex: A comprehensive knowledge graph completion benchmark. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8328–8350, 2020.

Rolf Sandell. Structural change and its assessment. *International Journal of Psychology and Psychoanalysis*, 5:042, 2019. doi: 10.23937/2572-4037.1510042.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Trans. Neural Networks*, 20(1): 61–80, 2009.

Tom Schaul and Jürgen Schmidhuber. Metalearning. *Scholarpedia*, 5(6):4650, 2010.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pages 593–607. Springer, 2018.

Hans-Jörg Schmid. *Entrenchment and the psychology of language learning: How we reorganize and adapt linguistic knowledge*. American Psychological Association, 2017.

Jürgen Schmidhuber. Powerplay: Training an increasingly general problem solver by continually searching for the simplest still unsolvable problem. *Frontiers in psychology*, 4:313, 2013.

Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Towards causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021. URL `https://arxiv.org/abs/2102.11107`.

Jonathan Schwarz, Wojciech Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Progress & compress: A scalable framework for continual learning. In *International conference on machine learning*, pages 4528–4537. PMLR, 2018.

Harshay Shah, Andrew Ilyas, and Aleksander Madry. Decomposing and editing predictions by modeling model computation. *arXiv preprint arXiv:2404.11534*, 2024.

Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

Yifei Shen, Yongji Wu, Yao Zhang, Caihua Shan, Jun Zhang, Khaled B Letaief, and Dongsheng Li. How powerful is graph convolution for recommendation? *arXiv preprint arXiv:2108.07567*, 2021.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. *Advances in neural information processing systems*, 30, 2017.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal. Ai models collapse when trained on recursively generated data. *Nature*, 631(8022):755–759, 2024.

Herbert A Simon et al. Invariants of human behavior. *Annual review of psychology*, 41 (1):1–20, 1990.

Burrhus Frederic Skinner. *Science and human behavior*. Number 92904. Simon and Schuster, 1965.

Luca Soldaini and Kyle Lo. peS2o (Pretraining Efficiently on S2ORC) Dataset. Technical report, Allen Institute for AI, 2023. ODC-By, `https://github.com/allenai/pes2o`.

Balasubramaniam Srinivasan and Bruno Ribeiro. On the equivalence between positional node embeddings and structural graph representations. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=SJxzFySKwH`.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.

Zhiqing Sun, Shikhar Vashishth, Soumya Sanyal, Partha Talukdar, and Yiming Yang. A re-evaluation of knowledge graph completion methods. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5516–5522, Online, July 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.489. URL https://aclanthology.org/2020.acl-main.489.

Zhiqing Sun, Shikhar Vashishth, Soumya Sanyal, Partha P. Talukdar, and Yiming Yang. A re-evaluation of knowledge graph completion methods. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5516–5522. Association for Computational Linguistics, 2020b. URL https://www.aclweb.org/anthology/2020.acl-main.489/.

Anej Svete and Ryan Cotterell. Transformers can represent n-gram language models. *arXiv preprint arXiv:2404.14994*, 2024.

Ahmed Taha, Abhinav Shrivastava, and Larry S Davis. Knowledge evolution in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12843–12852, 2021.

Y. Tang, C. Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. Multilingual translation with extensible multilingual pretraining and finetuning. *ArXiv*, abs/2008.00401, 2020. URL https://api.semanticscholar.org/CorpusID:220936592.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*, 2024. URL https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html.

Komal K. Teru, Etienne G. Denis, and William L. Hamilton. Inductive relation prediction by subgraph reasoning. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pages 9448–9457. PMLR, 2020.

Jonathan Thomm, Giacomo Camposampiero, Aleksandar Terzic, Michael Hersche, Bernhard Schölkopf, and Abbas Rahimi. Limits of transformer language models on learning to compose algorithms. In *Proceedings of the Thirty-eighth Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024. URL `https://arxiv.org/abs/2402.05785`.

Sebastian Thrun and Lorien Pratt. *Learning to learn*. Springer Science & Business Media, 2012.

Susumu Tonegawa, Xu Liu, Steve Ramirez, and Roger Redondo. Memory engram cells have come of age. *Neuron*, 87(5):918–931, 2015.

Susumu Tonegawa, Mark D Morrissey, and Takashi Kitamura. The role of engram cells in the systems consolidation of memory. *Nature Reviews Neuroscience*, 19(8):485–498, 2018.

Kristina Toutanova and Danqi Chen. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd workshop on continuous vector space models and their compositionality*, pages 57–66, 2015.

Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. Representing text for joint embedding of text and knowledge bases. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1174. URL `https://www.aclweb.org/anthology/D15-1174`.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023a.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023b.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Eric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 2071–2080, New York, New York, USA, 6 2016. PMLR. URL https://proceedings.mlr.press/v48/trouillon16.html.

Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha Talukdar. Composition-based multi-relational graph convolutional networks. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BylA_C4tPr.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Andreas Veit, Michael J Wilber, and Serge Belongie. Residual networks behave like ensembles of relatively shallow networks. *Advances in neural information processing systems*, 29, 2016.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

Petar Veličković. Message passing all the way up, 2022. URL https://arxiv.org/abs/2202.11097.

Tom Veniat, Ludovic Denoyer, and Marc'Aurelio Ranzato. Efficient continual learning with modular networks and task-driven priors. *arXiv preprint arXiv:2012.12631*, 2020.

Elena Voita, Javier Ferrando, and Christoforos Nalmpantis. Neurons in large language models: Dead, n-gram, positional. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics ACL 2024*, pages 1288–1301, Bangkok, Thailand and virtual meeting, August 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.findings-acl.75.

Lev Vygotsky. *Thought and Language*. MIT Press, 1934.

Jiongxiao Wang, Junlin Wu, Muhao Chen, Yevgeniy Vorobeychik, and Chaowei Xiao. On the exploitability of reinforcement learning with human feedback for large language models. *arXiv preprint arXiv:2311.09641*, 2023.

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. *arXiv preprint arXiv:2211.00593*, 2022.

Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743, 2017.

Ruocheng Wang, Eric Zelikman, Gabriel Poesia, Yewen Pu, Nick Haber, and Noah D. Goodman. Hypothesis search: Inductive reasoning with language models. In *Proceedings of the Twelfth International Conference on Learning Representations (ICLR)*, 2024. URL `https://arxiv.org/abs/2309.05660`.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, Zac Kenton, Sasha Brown, Will Hawkins, Tom Stepleton, Courtney Biles, Abeba Birhane, Julia Haas, Laura Rimell, Lisa Anne Hendricks, William Isaac, Sean Legassick, Geoffrey Irving, and Iason Gabriel. Ethical and social risks of harm from language models. *CoRR*, abs/2112.04359, 2021. URL `https://arxiv.org/abs/2112.04359`.

Laura Weidinger, Jonathan Uesato, Maribeth Rauh, Conor Griffin, Po-Sen Huang, John Mellor, Amelia Glaese, Myra Cheng, Borja Balle, Atoosa Kasirzadeh, et al. Taxonomy of risks posed by language models. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 214–229, 2022.

Frank Wilcoxon. Individual comparisons by ranking methods. In *Breakthroughs in statistics*, pages 196–202. Springer, 1992.

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

World Wide Web Consortium (W3C). RDF 1.2 Primer, 2024. URL `https://w3c.github.io/rdf-primer/spec/`. Accessed: 2024.

Shirley Wu, Shiyu Zhao, Michihiro Yasunaga, Kexin Huang, Kaidi Cao, Qian Huang, Vassilis Ioannidis, Karthik Subbian, James Y Zou, and Jure Leskovec. Stark: Benchmarking llm retrieval on textual and relational knowledge bases. *Advances in Neural Information Processing Systems*, 37:127129–127153, 2024.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *ICLR*. OpenReview.net, 2019.

Xiaoran Xu, Wei Feng, Yunsheng Jiang, Xiaohui Xie, Zhiqing Sun, and Zhi-Hong Deng. Dynamically pruned message passing networks for large-scale knowledge graph reasoning. In *ICLR*. OpenReview.net, 2020a.

Xiaoran Xu, Wei Feng, Yunsheng Jiang, Xiaohui Xie, Zhiqing Sun, and Zhi-Hong Deng. Dynamically pruned message passing networks for large-scale knowledge graph reasoning. In *International Conference on Learning Representations*, 2020b. URL `https://openreview.net/forum?id=rkeuAhVKvB`.

Jiaqi Xue, Mengxin Zheng, Yebowen Hu, Fei Liu, Xun Chen, and Qian Lou. Badrag: Identifying vulnerabilities in retrieval augmented generation of large language models. *arXiv preprint arXiv:2406.00083*, 2024.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. In *ICLR (Poster)*, 2015a.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. In *ICLR (Poster)*, 2015b.

Sohee Yang, Elena Gribovskaya, Nora Kassner, Mor Geva, and Sebastian Riedel. Do large language models latently perform multi-hop reasoning? In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10210–10229, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.550. URL https://aclanthology.org/2024.acl-long.550.

Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. In *ICML*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 40–48. JMLR.org, 2016.

Yuanshun Yao, Xiaojun Xu, and Yang Liu. Large language model unlearning. *Advances in Neural Information Processing Systems*, 37:105425–105475, 2024.

Jingwei Yi, Rui Ye, Qisi Chen, Bin Zhu, Siheng Chen, Defu Lian, Guangzhong Sun, Xing Xie, and Fangzhao Wu. On the vulnerability of safety alignment in open-access llms. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 9236–9260, 2024.

Yuning You, Tianlong Chen, Zhangyang Wang, and Yang Shen. L2-gcn: Layer-wise and learned efficient training of graph convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2127–2135, 2020.

Hanqing Zeng, Hongkuan Zhou, Ajitesh Srivastava, Rajgopal Kannan, and Viktor Prasanna. GraphSAINT: Graph sampling based inductive learning method. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BJe8pkHFwS.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

Zhao Zhang, Fuzhen Zhuang, Hengshu Zhu, Zhi-Ping Shi, Hui Xiong, and Qing He. Relational graph neural network with hierarchical attention for knowledge graph completion. In *AAAI*, pages 9612–9619. AAAI Press, 2020.

Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. Explainability for large language models: A survey. *ACM Transactions on Intelligent Systems and Technology*, 15(2): 1–38, 2024a.

Wanru Zhao, Yihong Chen, Royson Lee, Xinchi Qiu, Yan Gao, Hongxiang Fan, and Nicholas Donald Lane. Breaking physical and linguistic borders: Multilingual federated prompt tuning for low-resource languages. In *The Twelfth International Conference on Learning Representations*, 2024b.

Hattie Zhou, Ankit Vani, Hugo Larochelle, and Aaron Courville. Fortuitous forgetting in connectionist networks. In *International Conference on Learning Representations*, 2022.

Zhaocheng Zhu, Zuobai Zhang, Louis-Pascal A. C. Xhonneux, and Jian Tang. Neural bellman-ford networks: A general graph neural network framework for link prediction. *CoRR*, abs/2106.06935, 2021. URL `https://arxiv.org/abs/2106.06935`.

George Kingsley Zipf. *Human behavior and the principle of least effort: An introduction to human ecology*. Ravenio books, 2016.

Difan Zou, Ziniu Hu, Yewen Wang, Song Jiang, Yizhou Sun, and Quanquan Gu. Few-shot representation learning for out-of-vocabulary words. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems, NeurIPS*, 2019.