# Large language models partially converge toward human-like concept organization


## Abstract

Large language models show human-like performance in knowledge extraction, reasoning and dialogue, but it remains controversial whether this performance is best explained by memorization and pattern matching, or whether it reflects human-like inferential semantics and world knowledge. Knowledge bases such as WikiData provide large-scale, high-quality representations of inferential semantics and world knowledge. We show that large language models learn to organize concepts in ways that are strikingly similar to how concepts are organized in such knowledge bases. Knowledge bases model collective, institutional knowledge, and large language models seem to induce such knowledge from raw text. We show that bigger and better models exhibit more human-like concept organization, across four families of language models and three knowledge graph embeddings.


## 1 Introduction

The artificial intelligence community is split on the question of whether large language models (LLM) understand natural language in some non-trivial sense. Half of the community (51%) – according to a recent survey [17] – are willing to attribute non-trivial understanding to LLMs. The other half of the community (49%) argue that the illusion of understanding is the result of an Eliza effect.[1] The research question, as formulated by Mitchell and Krakauer [17], is:

> "do these systems (or will their near-term successors) actually, even in the absence of physical experience, create something like the rich concept-based mental models that are central to human understanding, and, if so, does scaling these models create even better concepts?"

We present a series of experiments designed to answer this question directly. Our findings suggest (very strongly) that the models (representations) induced by larger and better LLMs become more and more human-like.

### 1.1 Contributions

We present a series of experiments with four families of LLMs (21 models), as well as three knowledge graph embedding algorithms. Using three different methods, we compare the vector spaces of the LLMs to the vector spaces induced by the graph embedding algorithms. (This amounts to a total of 220 experiments.) We find that the vector spaces of LLMs within each family become increasingly structurally similar to those of knowledge graph embeddings. This shows that LLMs partially converge on human-like concept organization, facilitating inferential semantics [20].[2] The

[1] One famous example of this view is an article by Emily Bender and colleagues [1], who argue that these models are simply *stochastic parrots* that 'haphazardly stitch together sequences of linguistic forms' without any true understanding of the world or context.

[2] We use 'converge' in the sense of Caucheteux and King [7].

Preprint. Under review.

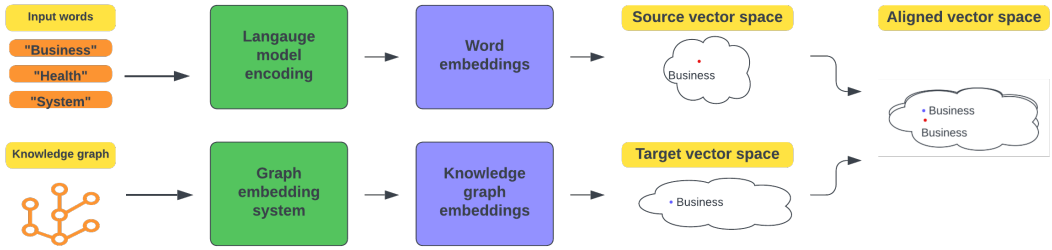


Figure 1: A simplified sketch of our experimental protocol. A vocabulary of 20K words is encoded using a language model and the corresponding entities are fetched from a pre-trained graph embedding system. The resulting vector spaces are then aligned. After alignment we evaluate retrieval performance in the target vector space. If retrieval performance is perfect, the spaces are (nearest neighbor graph) isomorphic.

sample efficiency of this convergence seem to depend on a number of factors, including polysemy and semantic category. Our findings have important implications. They vindicate the conjecture in [20] that LLMs exhibit inferential semantics, strongly contributing to the research question presented in [17], cited above. This means that LLMs partially converge toward human-like concept representations and, thus, come to partially 'understand language', in a non-trivial sense. We speculate that the human-like conceptual organization is also what facilitates out-of-distribution inferences in LLMs.

## 2 Experiments

### 2.1 Language models

We evaluate the vector spaces induced by four well-known families of language models, conducting experiments with a total of 20 different transformer-based models. The four families are OPT [34], GPT-2 [21], and Pythia [2] (non-deduplicated version at model checkpoint step 143000) and BERT [9].[3] We also evaluated the vector space of GPT-3 [5], i.e., *text-embedding-ada-002*.[4] Transformer-based LLMs use multiple layers of self-attention [29] and can model complex interactions across large context windows. Both left and right context can be considered. GPT, OPT and Pythia are decoder-only autoregressive LLMs, however, and thus only consider left context, i.e., the words preceding the next token. BERT is an encoder-only non-autoregressive LLM and considers both left and right context of the masked token to be predicted. Transformers are in general considered state-of-the-art for most NLP tasks [32]. For each of the language model families, we consider variants with increasing size in terms of the number of model parameters. See Table 1 for a model overview.

| LM | Params | LM | Params | LM | Params | LM | Params |
|---|---|---|---|---|---|---|---|
| OPT-125M | 125M | GPT-2 small | 117M | Pythia-70M | 70M | BERT-TINY | 4.4M |
| OPT-350M | 350M | GPT-2 medium | 345M | Pythia-160M | 160M | BERT-MINI | 11.3M |
| OPT-1.3B | 1.3B | GPT-2 large | 774M | Pythia-410M | 410M | BERT-SMALL | 29.1M |
| OPT-2.7B | 2.7B | GPT-2 xl | 1.5B | Pythia-1B | 1B | BERT-MEDIUM | 41.7M |
| OPT-6.7B | 6.7B | GPT-3 Ada-002 | 175B | Pythia-2.8B | 2.8B | BERT-BASE | 110.1M |
| | | | | Pythia-6.9B | 6.9B | | |

Table 1: 21 transformer-based language models used in this experiment.

[3] We also evaluated the T5 [22] LM series. T5 is trained with a multi-task objective, leading to mixed results.
[4] https://platform.openai.com/docs/guides/embeddings/what-are-embeddings

### 2.2 Knowledge graph embeddings

We experiment with three graph embedding algorithms and the vector spaces induced by running these on large-scale knowledge bases.

#### 2.2.1 BigGraph

The first vector space is that of the so-called *BigGraph* embeddings [15]. BigGraph is trained on an input knowledge graph, i.e. a list of edges, identified by its' source and target entities and a relation type. The network output is a feature vector or embedding for every entity in the graph. An inherent quality of this method is that adjacent entities are placed close to each other in the vector space. The particular embeddings used in this work are obtained by pre-training on WikiData, a well-known knowledge base[5]. Knowledge bases like Wikidata provide a structured representation of the real-world [25] and encode implicit world knowledge. The BigGraph embeddings contain all entities from the *"truthy"* Wikidata dump (2019-03-06) and thus includes URLs, dates etc; which are not (directly) included in language model vocabularies. To ensure compatability between the vector spaces, we limit ourselves to *single word* BigGraph entities. From these single word entities we pick $20,000$ common English words.[6]

#### 2.2.2 GraphVite

Our second and third knowledge base-derived vector spaces were both obtained by the GraphVite graph embedding algorithm [35]. We use the following pre-trained models; *TransE* [4] and *ComplEx* [27], both of which are pre-trained on WikiData5m [31]. We use the same entities presented in §2.2.1.

### 2.3 Graph isomophism

An isomorphism from $G_1 = (V_1, E_1)$ to $G_2 = (V_2, E_2)$ is a bijection $f : V_1 \rightarrow V_2$ such that any pair of nodes $a$ and $b$ are joined by an edge iff $f(a)$ and $f(b)$ are joined by an edge. Near-isomorphism of graphs refers to the situation where two graphs are not exactly isomorphic but exhibit strong structural similarity. Note that if the nearest neighbor graphs of two embedding spaces are isomorphic, there exists a vector space mapping with precision@1 of 1.0; see §2.7 for details on how to compute precision@1. We will evaluate to what extent (the $k$-nearest neighbor graphs of the) LLM vector spaces are isomorphic to the knowledge graph embeddings, by computing representational similarity analysis scores, as well as by evaluating the precision@$k$ of linear projections.

### 2.4 Linear projections

We present two distinct methods of mapping (or projecting) the vector space of an LLM to the vector space of BigGraph (BG) and GraphVite (GV). Let $\mathbf{M}_{\texttt{LM}} \in \mathbf{R}^{V \times d_{\texttt{e}}}$ be a matrix of word embeddings and $\mathbf{M}_{\texttt{REF}} \in \mathbf{R}^{V \times d_{\texttt{ref}}}$ a matrix of knowledge graph node embeddings, where $V$ denotes the size of the vocabulary, $d_{\text{e}}$ the dimensionality of the word embeddings for a given language model and $d_{\text{ref}}$ refers to dimensionality of the knowledge graph embeddings, with $d_{\text{ref}} = 200$ for BG and $d_{\text{ref}} = 512$ for GV.

Our first approach is to use generalized Procrustes analysis [24] to align $\mathbf{M}_{\texttt{LM}}$ with $\mathbf{M}_{\texttt{REF}}$. Since this method enforces $d_e = d_{\text{ref}}$, we use `PCA` to reduce the dimensionality of $\mathbf{M}_{\texttt{LM}}$ to the desired size. The aim of Procrustes analysis is thus to find a transformation matrix $\mathbf{\Omega}$ that minimizes the sum of squared distance between each pair of word embeddings in $\mathbf{M}_{\texttt{LM}}$ with $\mathbf{M}_{\texttt{REF}}$. This is achieved by solving the following problem:

$$\min_{\mathbf{\Omega} = s\mathbf{A}} ||\mathbf{\Omega}\mathbf{M}_{\texttt{LM}} - \mathbf{M}_{\texttt{REF}}||_F^2$$
$$s \in \mathbf{R}^+,\ \mathbf{A} \in \mathbf{R}^{d_e \times d_e} \text{ s.t. } \mathbf{A}^{\mathbf{T}}\mathbf{A} = \mathbf{I}$$

Where $F$ denotes the Frobenius norm and we have that $\mathbf{\Omega}$ can be computed using singular value decomposition. In practice, we compute $\mathbf{\Omega}$ using a subset of the full vocabulary; $V_{\texttt{train}}$.

[5] https://torchbiggraph.readthedocs.io/en/latest/pretrained_embeddings.html

[6] https://github.com/first20hours/google-10000-english/blob/master/20k.txt

Secondly, we propose utilizing $d_{\text{ref}}$ ridge regression models $f_i$ for $i = 1, 2, .., d_{\text{ref}}$, with one predictor for each dimension of the reference vector space. Each predictor $f_i$ is trained on a subset of the full vocabulary $V_{\texttt{train}}$ and learns a function $f_i : \mathbf{R}^{d_e} \rightarrow \{\mathbf{R}\}_j$, for $j = 1, 2, .., d_{\text{ref}}$, where $j$ indicates the $j$'th dimension of the reference vector space. The ridge regression models are re-trained for each language model as $d_e$ varies across these. Once trained, the models can be used to project the remaining vocabulary $V_{\texttt{test}}$. The methodology of using a separate ridge regression model for each dimension of the reference/target vector space has previously been used to decode linguistic meaning from brain activation [19].

### 2.5 Representational Similarity Analysis

To further gauge the similarity of the vector spaces induced by LLMs and knowledge graph embeddings, we present experiments using Representational Similarity Analysis (RSA) [6]. For a given language model we consider the matrix $\mathbf{M}_{\texttt{LM}}$ of word embeddings alongside the corresponding matrix $\mathbf{M}_{\texttt{REF}}$. We compute the representational dissimilarity matrices (RDMs); i.e. for each word embedding in each of the matrices we compute the euclidean distance to all other word embeddings within that respective matrix, thus generating two $V \times V$ RDMs. Once the RDMs has been computed (denote them $r_1$ and $r_2$), they are then compared used cosine similarity:

$$\cos(r_1, r_2) = \frac{r_1^T r_2}{\sqrt{r_1^T r_1 r_2^2 r_2}}$$

A cosine similarity close to 1.0 will indicate $\mathbf{M}_{\texttt{LM}}$ more closely resembles $\mathbf{M}_{\texttt{REF}}$. Note that we flatten the RDMs in practice, to get a single value as a final metric. Representational similarity analysis is a well-known method within neuroscience, see for instance [14].

### 2.6 Analogies

Finally, we carry out experiments using the WiQueen analogy dataset [12]. The dataset consists of quadruples <$w_1, w_2, w_3, w_4$> of analogies, e.g. <*Hefei,Anhui,Guiyang,Guizhou*> which corresponds to the analogy *"**Hefei** is to **Anhui**, as **Guiyang** is to **Guizhou.**"*. We encode the individual words from the analogies using each of the language models, thus obtaining a new quadruple <$e_1, e_2, e_3, e_4$> of word embeddings for each analogy and language model. We then proceed to *"solve"* the analogy mathematically by computing $e_1 - e_2 + e_3 = e_{\text{new}}$ [10]. Finally, we compare $e_{\text{new}}$ to $e_4$, by checking if $e_{\text{new}}$ and $e_4$ are nearest neighbors in the WiQueen vector space. Note how for an LLM vector space to solve the analogy task (completely) is equivalent to being (nearest neighbor graph) isomorphic to the underlying knowledge base [18].

### 2.7 Evaluation

For the experiments in which we induce a linear mapping $\mathbf{\Omega}$ and $f(\mathbf{M}_{\texttt{LM}})$ between the vector spaces of LLMs and BigGraph/GraphVite, we evaluate how close $\mathbf{M}_{\texttt{LM}}\mathbf{\Omega}$ and $f(\mathbf{M}_{\texttt{LM}})$ is to $\mathbf{M}_{\texttt{REF}}$ using precision@$k$ as our performance metric.[7] That is, for each word $w$ in $V_{\text{test}}$, we perform k-nearest neighbors of the corresponding word embedding contained within the projection of $\mathbf{M}_{\texttt{LM}}$ in the reference vector space. If $w$ is found among the k-nearest neighbors in the reference vector space, we say that the precision at $k$ (p@k) is 100%. The final precision is then scored as an average over all words in $V_{\text{test}}$. Note that for all values of $k$ this will either be a *"hit"* or a *"miss"* as there is only one relevant item to retrieve. The full vocabulary $V$ contains 20,000 words and is split into $V_{\text{train}}$ and $V_{\text{test}}$ at 80%/20% respectively, which makes a random retrieval baseline $\texttt{P@1} = \frac{1}{4000}$. In practice, our linear projections are found to be significantly more precise, which in turn reflects the growing resemblance between the vector spaces of increasingly larger language models and the vector spaces of the knowledge graph embeddings induced by the BigGraph/GraphVite graph embedding algorithms. Note that for the experiment involving analogies, we do not use a reference vector space, but instead evaluate the retrieval directly using the WiQueen data set.

For the representational similarity analysis, we simply use the cosine similarity between the RDM of each language model and the RDM of BigGraph/GraphVite as the performance metric.

[7] This is the *de facto* standard performance metric in the word vector space alignment literature [26].

## 3 Results

### 3.1 Procrustes Analysis

We report alignment precision (p@k) for $k \in \{1, 10, 20, 50\}$, with our main results depicted in Figure 2. The plots show the convergence results, i.e. the relationship between language model size and alignment precision. For all four model families, we see a consistent trend, where larger language models lead to better alignments with the reference vector spaces. Overall, GPT appears to have the most pronounced convergence properties. Note that the different graph embeddings heavily influence the precision, but that the *trend* is similar across graph embedding systems. For our best performing model GPT-3 (ada-002) projected onto the vector space of GraphVite (TransE) using Procrustes analysis, we observe a P@50 beyond 60%, which in turn means that more than 6/10 words are mapped to a relatively small neighborhood of 50 words out of a total of 4000 words in $V_{\text{test}}$. This, in our view, constitutes strong evidence that LLMs learn human-like concept organizations.

*Why do we think so?* There are two ways to see this: One is to contrast the performance figures with a random baseline. If the language model representations were orthogonal to knowledge base representations, the likelihood of perfect alignment (p@1) would be $\frac{1}{4000} = 0.00025$. The random baseline p@50 is thus 0.0125. Our best model, in contrast, scores 0.67. Another way to view this is by considering what a p@50 score of 0.67 really means. It means, simply, that two in three knowledge base nodes are aligned with a neighborhood of 50 words around the correct answer. That is, for two in three concepts, we can find a 50-word realization of this concept based on our linear projection only. What is a 50-word cluster in a language model? Typically, 50 words corresponds to a small set of taxonomic siblings and their morphological inflections. Such as {car, cars, car's, automobile, automobiles, vehicle, truck, trucks, Mercedes, taxi, cab, ...}. So a p@50 of 0.67 basically means that we can take a knowledge base node and ask a language model (in an arbitrary language) for its realization, and most of the time, we will get a fine-grained semantic class back.

### 3.2 Ridge Regression

As a secondary projection method we train $d_{\text{ref}}$ ridge regression predictors, i.e. one for each dimension of the reference vector space. These predictors are then used to project the remaining vocabulary of word embeddings $V_{\text{test}}$ for a given language model to the reference vector space. After this, retrieval can be conducted. The retrieval performance for $k = \{1, 10, 20, 50\}$ and all four model families can be found in Figure 3. The results share some characteristics with those presented in §3.1, but in some cases performance drops for the families' largest models (e.g. OPT-6.7B and Ada-002), presumably because of poor signal-to-noise ratios in the extra dimensions, which, in Procrustes Analysis, are removed through principal component analysis.

### 3.3 Analogies

Figure 4 shows the results of the four series of language models on the WiQueen data set. Again, we observe a consistent upward trend for all four model families. Note that this experiment more closely resembles a real-world task for a language model, as analogies play a central role in human commonsense reasoning [28]. The trends shown in Figure 4 are in tune with those presented in §3.1-3.2 and thus expands the evidence to support our arguments to more realistic use cases of language models.

### 3.4 Representational Similarity Analysis

The results of the representational similarity analysis for all four language model families can be found in the supplementary material. The partial convergence results are similar to those obtained with linear projection.

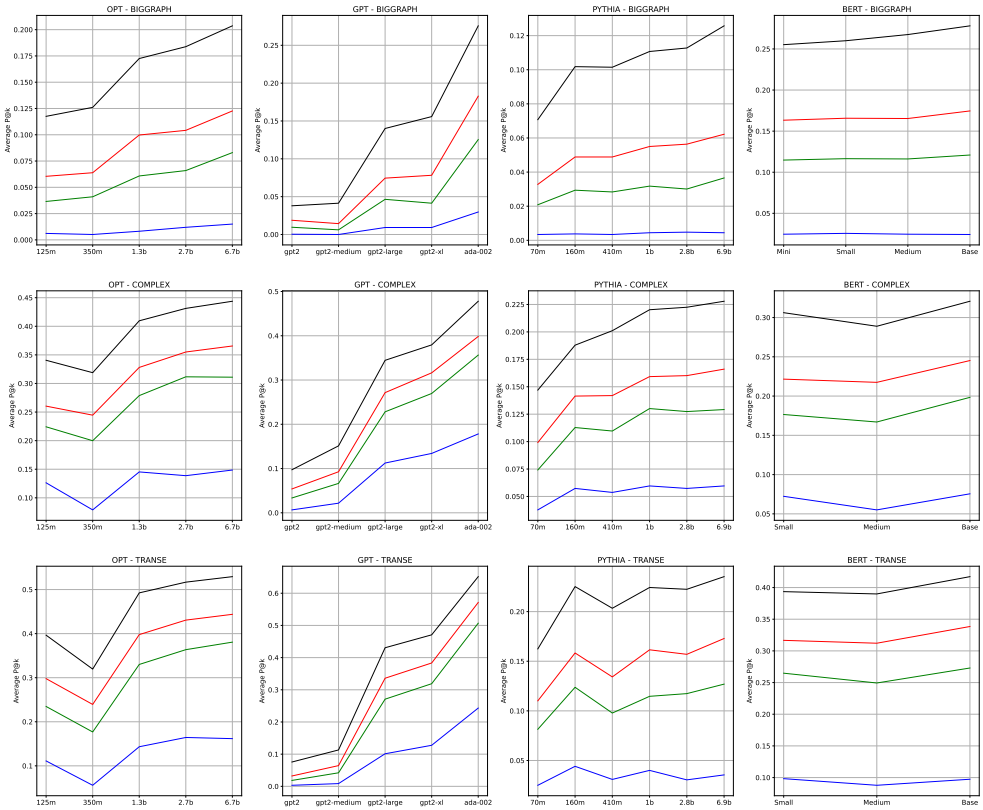


Figure 2: Plot labels: — $k = 1$, — $k = 10$, — $k = 20$, — $k = 50$. Projection onto the vector space of BigGraph (1st row), ComplEx (2nd row) and TransE (3rd row) using **generalized Procrustes analysis** and retrieval performance p@k at $k = \{1, 10, 20, 50\}$ for 4 language model families. We see results up to p@50∼0.7, and strong, positive convergence (almost) across the board. Baseline p@50 is 0.0125.

# 4 Analysis

## 4.1 LLMs and graph embeddings

We first analyzed the robustness of our partial convergence results across various settings[8], fitting a linear trend line $y = mx + b$ to each convergence line using linear least squares regression. We saw that TransE and BigGraph had more convergence results than ComplEx: Whereas 0.969 of the convergence plots had positive slopes for TransE and BigGraph, only 0.875 of the convergence plots had positive slopes for ComplEx. We also saw that OPT and Pythia had the most robust convergence properties of the LLMs (with 1.0 of the convergence plots having positive slopes), compared to about 0.80 for GPT and 0.90 for BERT. Furthermore, we observe that when utilizing ridge regression as the projection method, the retrieval performance experiences a significant decline at the largest model in some cases. The largest models also correspond to those with the highest dimensional embeddings, which may suggest that this projection method encounters difficulties when the dimensionality of the input embeddings exceeds $d = 2048$.

We notice the significantly large share of converge lines having positive slopes. In almost all of our experiments, LLMs become more and more human-like in their concept-organization.

[8] 96 total settings: 4 values of $k$, 3 reference vector spaces, 4 LM families and 2 projection methods.

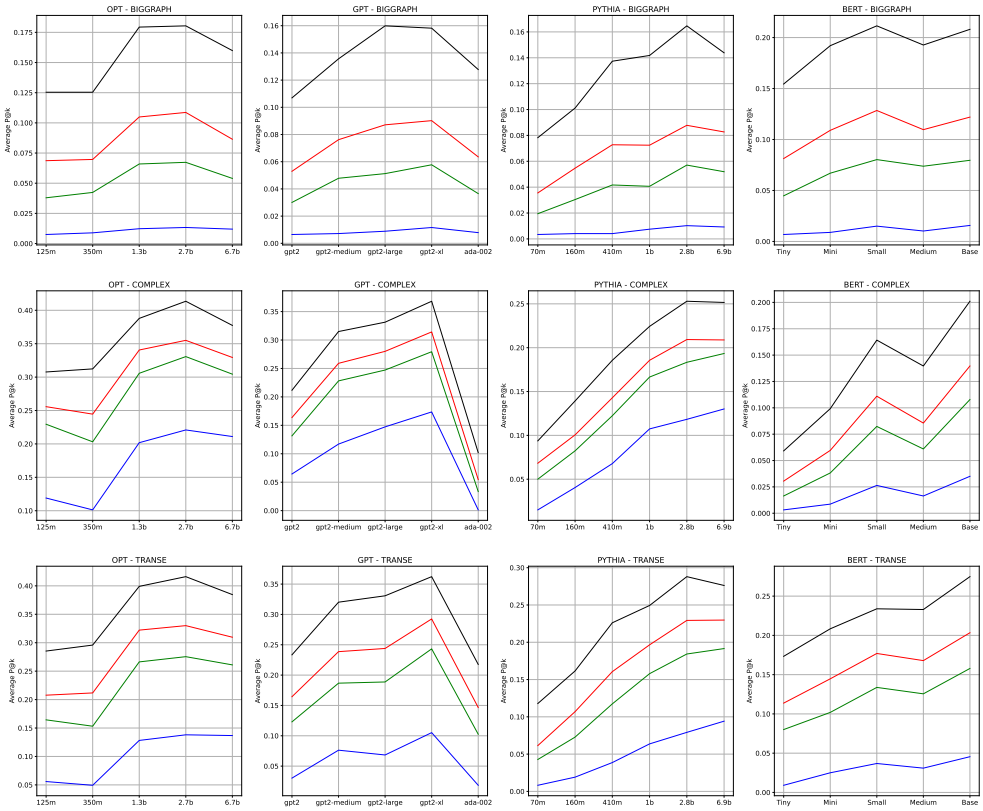


Figure 3: Plot labels: — $k = 1$, — $k = 10$, — $k = 20$, — $k = 50$. Projection onto the vector space of BigGraph (1st row), ComplEx (2nd row) and TransE (3rd row) using **ridge regression** and retrieval performance p@k at $k = \{1, 10, 20, 50\}$ for 4 language model families. We see partial convergence, except for the GPT-3 results.

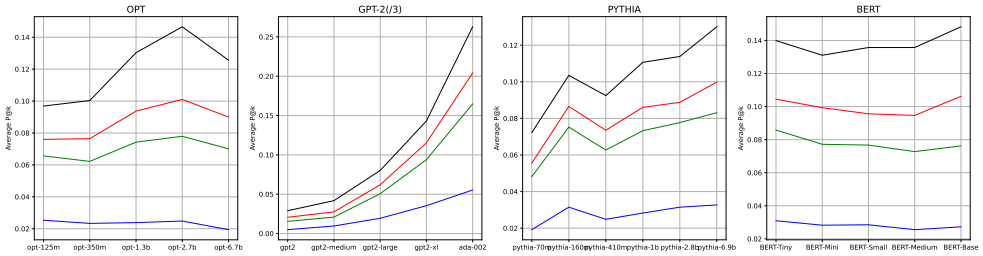


Figure 4: Plot labels: — $k = 1$, — $k = 10$, — $k = 20$, — $k = 50$. The figure presents the results of how well the language models *solve* the analogies from the WiQueen dataset and retrieve the correct the word. We report retrieval performance p@k at $k = \{1, 10, 20, 50\}$ for 4 language model families.

### 4.2 Polysemy and semantic category

We investigate the effect of polysemy and semantic category of the target words. Previous work on bilingual dictionary induction from cross-lingual vector space projections found high variance in retrieval scores across similar dimensions [13, 16]. Table 2 provides an overview of our results. Polysemy refers to the level of lexical ambiguity for target words, distinguishing between words with one, two to three and four or more distinct meanings. We obtain polysemy counts for our target words (see §2.2.1) from NLTK's WordNet interface [3]. For semantic categories, we compare our experiments with common (frequent) words (see §2.2.1) to using only *places* (geographic

locations/places etc.) or *names* (anthroponyms). The names were found in [9] and the geographic locations were found in. [10]

**Polysemy** To investigate potential sources of error, we present statistics for three bins of polysemy counts in Table 2. Our findings suggests that performance drops as the polysemy counts grows to more than four. This is intuitive because words with multiple meaning can be mapped to vastly different positions in the induced vector space, which might lead to a performance drop when retrieval is carried out in the reference vector space. These findings align well with those presented in [16], particularly that non-polysemous words tend to yield higher precision scores, but also that for some models (e.g. GPT-2), using words with two to three meanings results in better performance - these observations remain to be investigated in future work.

**Semantic category** Furthermore, we investigate the impact of semantic categories. We repeat the experiments across all settings using both the vocabulary of anthroponyms and world cities. The statistics presented in Table 2 suggests that such semantic categories has an noticeable impact on performance. Specifically that anthroponyms have a relatively low max slope coefficient, suggesting slow convergence properties while the language model size grows. In addition to this, we observe that the *places* (i.e. world cities) has a substantially higher positive rate, than other categories considered, which indicates that large language models might have better internal representations of some concepts compared to others.

| **Polysemy** | | | | **Semantic category** | | | |
|---|---|---|---|---|---|---|---|
| **Variable** | **% positive** | **Max coeff.** | **SD** | **Variable** | **% positive** | **Max coeff.** | **SD** |
| 1 | 0.812 | 0.172 | 0.038 | Common | 0.938 | 0.151 | 0.029 |
| 2-3 | 0.792 | 0.145 | 0.032 | Places | 1.0 | 0.105 | 0.020 |
| 4+ | 0.771 | 0.129 | 0.028 | Names | 0.917 | 0.061 | 0.013 |

Table 2: The effect of polysemy and word classes on the convergence trend. *Common* refer to common english words (i.e. those presented in §2.2.1).

| **Models** | **Polysemy** | BigGraph **P@50** | TransE **P@50** | ComplEx **P@50** | **Semantic category** | BigGraph **P@50** | TransE **P@50** | ComplEx **P@50** |
|---|---|---|---|---|---|---|---|---|
| OPT-6.7b | 1 | **0.595** | **0.748** | **0.590** | Common | 0.203 | **0.529** | **0.444** |
| | 2-3 | 0.490 | 0.680 | 0.483 | Places | **0.299** | 0.382 | 0.325 |
| | 4+ | 0.385 | 0.610 | 0.435 | Names | 0.210 | 0.264 | 0.222 |
| ADA-002 | 1 | **0.610** | **0.813** | **0.618** | Common | 0.276 | **0.651** | **0.478** |
| | 2-3 | 0.520 | 0.760 | 0.528 | Places | **0.373** | 0.465 | 0.367 |
| | 4+ | 0.423 | 0.675 | 0.450 | Names | 0.285 | 0.329 | 0.250 |
| Pythia-6.9b | 1 | **0.495** | **0.538** | **0.478** | Common | 0.126 | **0.235** | **0.228** |
| | 2-3 | 0.373 | 0.488 | 0.410 | Places | **0.190** | 0.204 | 0.188 |
| | 4+ | 0.323 | 0.348 | 0.310 | Names | 0.145 | 0.150 | 0.139 |
| BERT-BASE | 1 | **0.533** | **0.785** | **0.578** | Common | 0.239 | **0.508** | **0.393** |
| | 2-3 | 0.425 | 0.688 | 0.505 | Places | **0.350** | 0.359 | 0.322 |
| | 4+ | 0.320 | 0.538 | 0.448 | Names | 0.259 | 0.228 | 0.198 |

Table 3: Effect of polysemy and semantic category on the largest model from each model family. Procrustes is used as the projection method. Note that a low level of lexical ambiguity leads to better performance and that the best performing semantic category varies across the reference vector spaces.

[9] US common names
[10] World cities

### 4.3 Analogies

Ushio et al. [28] investigated how well LLMs such as GPT are able to solve analogies. They obtained the best results using GPT. This aligns well with our finding that GPT-{2,3} has solid convergence properties and obtained the overall best results; see Figure 4.

## 5 Discussion

We have seen that language models partially converge on human-like concept organizations. How surprising is this? Given the contentious debate around whether large language models 'understand' [17], including whether they induce models of knowledge, our result is important. Large language models do not only learn to use patterns in context, but as a result, they induce compressed models of knowledge. In retrospect, it is also clear, however, that some results, e.g., the near-isomorphism of word vector spaces across languages [30] or the near-isomorphism with representations from computer vision [16], already pointed in this direction. Language models for different languages likely learn similar concept geometries, because they induce models of (our knowledge of) the world. Language and computer vision models, in a similar way, share one reference, namely, the world we live in, and what we know about it.

### 5.1 Practical implications

There has already been considerable work on grounding language models in knowledge bases. This work often has focused on algorithms for joint language and graph embedding [33]. Our work suggests that similar results can be obtained with 'retro-fitting´ [11], i.e., fine-tuning the language models to improve existing similarities. See, for example, the approach taken by [12]. Our results also suggest, however, that in the limit, perhaps grounding in knowledge bases will become redundant. The systematicity of human-like conceptual organization in language models seemingly facilitates out-of-distribution capacities, e.g., enabling analogical inference.

### 5.2 Philosophical implications

Our results clearly show that language models induce inferential semantics [23, 20]. This, we believe, is an important contribution to the debate about the capacities of large language models [17]. Our results also, however, question the divide between syntax and semantics [8]. Semantics, in a way, seems to fall out of syntax. Clearly, our results have no bearing on intentionality (the aboutness of mental tokens), but they do suggest one way syntactic tokens acquire semantics.

## 6 Limitations

We have experimented with three families of autoregressive language models and one non-autoregressive family. We have compared the word vector spaces induced by such models with three vector spaces induced by graph embedding algorithms over large knowledge bases. All our experiments have been limited to English. This, of course, is a major limitation. Language characteristics may effect the quality of word vector spaces, and morpho-syntactic properties may influence how corpora and knowledge bases align. Finally, while we do error analysis over polysemy and semantic categories, we acknowledge that the set of variables that covary with performance, is probably much larger.

## 7 Conclusion

This paper weighs in on the debate around understanding in large language models and show how large language models converge toward human-like concept organization, building implicit models of the world (as we know it). Over 220 experiments, we show how language models partially converge toward human-like concept organization, with particularly strong similarities in how monosemous and common words are encoded. Our observations have important practical and philosophical implications, providing a possible explanation for the out-of-distribution capacities of large language models and making an important contribution to the debate above.

## 8 Computational requirements

A Google Colab Pro+ subscription or similar ($\geq$ 52GB RAM) is required in order to reproduce our experiments. We used an NVIDIA V100 and A100 Tensor Core GPU, provided by Google Colab.

## 9 Acknowledgement

## References


[1] Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, New York, March 2021. Association for Computer Machinery – ACM. doi: 10.1145/3442188.3445922. URL `https://dl.acm.org/doi/10.1145/3442188.3445922`.

[2] Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. Pythia: A suite for analyzing large language models across training and scaling, 2023.

[3] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly, Beijing, 2009. ISBN 978-0-596-51649-9. doi: http://my.safaribooksonline.com/9780596516499. URL `http://www.nltk.org/book`.

[4] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, page 2787–2795, Red Hook, NY, USA, 2013. Curran Associates Inc.

[5] 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`.

[6] Nili H Wingfield C, Walther A, Su L, Marslen-Wilson W, and Kriegeskorte N. A toolbox for representational similarity analysis. *PLoS Comput Biol.*, 13(4):e1005508, 2014. doi: 10.1371/journal.pcbi.1003553. URL `https://EconPapers.repec.org/RePEc:spr:psycho:v:31:y:1966:i:1:p:1-10`.

[7] Charlotte Caucheteux and Jean-Rémi King. Brains and algorithms partially converge in natural language processing. *Communications Biology*, 5:134, 02 2022. doi: 10.1038/s42003-022-03036-1.

[8] Paul M. Churchland and Patricia S. Churchland. Could a machine think? *Scientific American*, 262 1:32–7, 1990.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/v1/n19-1423. URL `https://doi.org/10.18653/v1/n19-1423`.

[10] Kawin Ethayarajh. Rotate king to get queen: Word relationships as orthogonal transformations in embedding space. 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 3503–3508, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1354. URL https://aclanthology.org/D19-1354.

[11] Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. Retrofitting word vectors to semantic lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1606–1615, Denver, Colorado, May–June 2015. Association for Computational Linguistics. doi: 10.3115/v1/N15-1184. URL https://aclanthology.org/N15-1184.

[12] Nicolas Garneau, Mareike Hartmann, Anders Sandholm, Sebastian Ruder, Ivan Vulić, and Anders Søgaard. Analogy training multilingual encoders. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(14):12884–12892, May 2021. doi: 10.1609/aaai.v35i14.17524. URL https://ojs.aaai.org/index.php/AAAI/article/view/17524.

[13] Yova Kementchedjhieva, Mareike Hartmann, and Anders Søgaard. Lost in evaluation: Misleading benchmarks for bilingual dictionary induction. 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 3336–3341, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1328. URL https://aclanthology.org/D19-1328.

[14] Nikolaus Kriegeskorte, Marieke Mur, and Peter Bandettini. Representational similarity analysis - connecting the branches of systems neuroscience. *Frontiers in Systems Neuroscience*, 2, 2008. ISSN 1662-5137. doi: 10.3389/neuro.06.004.2008. URL https://www.frontiersin.org/articles/10.3389/neuro.06.004.2008.

[15] Adam Lerer, Ledell Wu, Jiajun Shen, Timothee Lacroix, Luca Wehrstedt, Abhijit Bose, and Alex Peysakhovich. Pytorch-biggraph: A large scale graph embedding system. In A. Talwalkar, V. Smith, and M. Zaharia, editors, *Proceedings of Machine Learning and Systems*, volume 1, pages 120–131, 2019. URL https://proceedings.mlsys.org/paper_files/paper/2019/file/e2c420d928d4bf8ce0ff2ec19b371514-Paper.pdf.

[16] Jiaang Li, Yova Kementchedjhieva, and Anders Søgaard. Implications of the convergence of language and vision model geometries, 2023.

[17] Melanie Mitchell and David C. Krakauer. The debate over understanding in ai's large language models. *Proceedings of the National Academy of Sciences*, 120(13):e2215907120, 2023. doi: 10.1073/pnas.2215907120. URL https://www.pnas.org/doi/abs/10.1073/pnas.2215907120.

[18] Xutan Peng, Mark Stevenson, Chenghua Lin, and Chen Li. Understanding linearity of cross-lingual word embedding mappings. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL https://openreview.net/forum?id=8HuyXvbvqX.

[19] Francisco Pereira, Bin Lou, Brianna Pritchett, Samuel Ritter, Samuel J. Gershman, Nancy G. Kanwisher, Matthew M. Botvinick, and Evelina Fedorenko. Toward a universal decoder of linguistic meaning from brain activation. *Nature Communications*, 9, 03 2018. doi: 10.1038/s41467-018-03068-4. URL https://www.nature.com/articles/s41467-018-03068-4.

[20] Steven Piantadosi and Felix Hill. Meaning without reference in large language models. In *NeurIPS 2022 Workshop on Neuro Causal and Symbolic AI (nCSI)*, 2022. URL https://openreview.net/forum?id=nRkJEwmZnM.

[21] Alec Radford, Jeff Wu, Rewon Child, D. Luan, Dario Amodei, and Ilya Sutskever. Language Models are Unsupervised Multitask Learners. 2019. URL https://www.semanticscholar.org/paper/Language-Models-are-Unsupervised-Multitask-Learners-Radford-Wu/9405cc0d6169988371b2755e573cc28650d14dfe.

[22] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

[23] Magnus Sahlgren and Fredrik Carlsson. The singleton fallacy: Why current critiques of language models miss the point. *Frontiers in Artificial Intelligence*, 4, 2021. ISSN 2624-8212. doi: 10.3389/frai.2021.682578. URL https://www.frontiersin.org/articles/10.3389/frai.2021.682578.

[24] Peter Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10, 1966. URL https://EconPapers.repec.org/RePEc:spr:psycho:v:31:y:1966:i:1:p:1-10.

[25] Suhas Shrinivasan and Simon Razniewski. How stable is knowledge base knowledge? *CoRR*, abs/2211.00989, 2022. doi: 10.48550/arXiv.2211.00989. URL https://doi.org/10.48550/arXiv.2211.00989.

[26] Anders Søgaard, Ivan Vulić, Sebastian Ruder, and Manaal Faruqui. *Cross-Lingual Word Embeddings*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, United States, 2 edition, 2019. doi: 10.2200/S00920ED2V01Y201904HLT042.

[27] 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, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/trouillon16.html.

[28] Asahi Ushio, Luis Espinosa Anke, Steven Schockaert, and Jose Camacho-Collados. BERT is to NLP what AlexNet is to CV: Can pre-trained language models identify analogies? In *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 3609–3624, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.280. URL https://aclanthology.org/2021.acl-long.280.

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[30] Ivan Vulić, Sebastian Ruder, and Anders Søgaard. Are all good word vector spaces isomorphic? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3178–3192, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.257. URL https://aclanthology.org/2020.emnlp-main.257.

[31] Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. KEPLER: A Unified Model for Knowledge Embedding and Pre-trained Language Representation. *Transactions of the Association for Computational Linguistics*, 9:176–194, 03 2021. ISSN 2307-387X. doi: 10.1162/tacl_a_00360. URL https://doi.org/10.1162/tacl_a_00360.

[32] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6.

[33] Donghan Yu, Chenguang Zhu, Yiming Yang, and Michael Zeng. Jaket: Joint pre-training of knowledge graph and language understanding. In *AAAI 2022*, February 2022. URL https://www.microsoft.com/en-us/research/publication/jaket-joint-pre-training-of-knowledge-graph-and-language-understanding/.

[34] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 5 2022. URL https://arxiv.org/abs/2205.01068.

[35] Zhaocheng Zhu, Shizhen Xu, Jian Tang, and Meng Qu. GraphVite: A high-performance CPU-GPU hybrid system for node embedding. In *The World Wide Web Conference*. ACM, may 2019. doi: 10.1145/3308558.3313508. URL https://doi.org/10.1145.2F3308558.3313508.