Do Generative Transformers Read Like BERT? An Empirical Analysis of Representations

Amil Merchant
CS236: Deep Generative Models
Stanford University
Stanford, CA 94305
amil@stanford.edu

Abstract

While many recent papers have studied how linguistic information is embedded in large-scale encoder-only language models such as BERT, relatively little is understood about their generative counterparts. Despite the modelling similarities with Transformer layers, the causal attention structure and disparate objectives may yield fundamentally different representations of language. In this paper, using a suite of analysis techniques (Representational Similarity Analysis, probing classifiers, and attention analysis) we investigate how representations develop in and the linguistic capabilities of the popular GPT-2 generative model. While similarities arise in how layers update representations in comparison to encoder-only models like BERT, the differences when measuring linguistic understanding and attention weights suggest that generative models are too narrowly focused on the next word to create a complete understanding of the entire input.

1 Introduction and Motivation

The advent of large-scale pre-training of Transformer models [34] has dramatically improved results for a variety of Natural Language Processing tasks [41, 26]; for example, BERT, a bi-directional encoder, topped the GLUE leaderboard in 2019 [38, 8], and modelling improvements have pushed the language understanding of these models closer to the human benchmarks [39]. Due to the scale of these improvements, recent papers have set out to study what these models understand about language, in a subfield informally known as *BERTology* [27, 32, 33, 20]. In particular, unsupervised techniques [29, 28] have been used to compare representation spaces, probing classifiers shed light on the understanding of linguistic phenomena such as part-of-speech or co-reference resolution [32, 33, 12], and attention analysis provided insight into the use of context to create representations [5, 40].

However, BERT is bi-directional and assumes text is already written before processing; it is also possible to train these large Transformer models in a generative fashion, as done for GPT-2 [24]. Despite the modelling similarities, little is understood about the effect of the change in attention structure and objective; most of the current understanding is derived from model behavior, by prompting the model with short inputs and examining the generated sentences [24, 43]. In this work, we hope to provide further insight into how representations develop in generative Transformer models such as GPT-2 and compare to encoder-only models. Specifically, we ask:

1. How do the representation spaces differ between layers of a generative model and in comparison to a encoder-only model?
2. Do generative models contain a comparable understanding of linguistic phenomena?
3. How do generative models utilize context when creating representations?

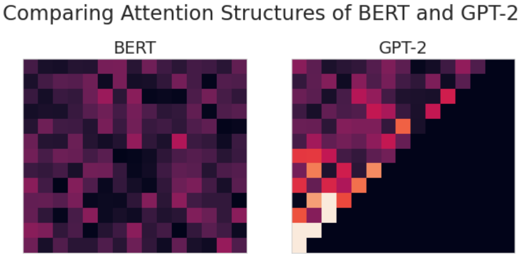

Figure 1: Schematic of attention structures of BERT and GPT-2 shows a major difference between encoder-only language models and encoder-decoder models. As text is assumed to be pre-written in BERT, contextualization can occur in both directions; however, for GPT-2, words can only depend on earlier tokens in the sequence; hence the triangular structure.

To answer these questions, we utilize a variety of analytic techniques. Representational Similarity Analysis (RSA: [18, 16]) provides a task-agnostic measure of the differences in representation spaces; probes (MLPs trained on top of a frozen model) are used to query linguistic understanding [33, 32]. Finally, attention analysis helps understand the role of contextualization. Overall, the results suggest the generative GPT-2 model adapts contextualized embeddings in a manner similar to encoder-only models, especially in the first few layers; however, the outputted representations may be too narrowly focused on the generation task to provide a complete understanding of the inputted text.[1]

2 Related Work

Understanding the influence of context, how representations develop in these neural networks, and what models understand about language have become popular subjects of study since the release of pre-trained sentence encoders such as Elmo and BERT [22, 8]. These works predominantly have taken the form of behavioral analyses which compare the perplexity of handcrafted inputs [19, 10, 9]. Other analyses generally fall into three main categories: unsupervised, probing, and attention.

The unsupervised methods generally attempt to measure similarity of representation spaces and have yielded insight into the layer-wise evolution of representations [36], progression of training [28], and effects of fine-tuning [20]. Abnar et al. [2] studied the effects of context, and other works have even correlated the representations of text to fMRI data to compare models to human understanding [1].

While these unsupervised techniques provide generic information about how representation spaces differ, to gain more granular insight into how the embeddings contrast between models, prior work has turned to supervised probes [12, 3, 7]. This technique attempts to measure the amount of linguistic information in a frozen encoder by extracting associated representations and training a small model to predict linguistic properties, including part-of-speech and dependencies [33, 32]. This line of work has shown that large language models often reconstruct the traditional NLP pipeline [32], progressively moving from syntactic understanding to higher-level semantic understanding.

Lastly, another branch of interpretability research has attempted to understand the role of context and inter-token interactions. For Transformers, this corresponds to examining attention weights, and various analyses have found interpretable patterns in the structure [6, 35, 37], including correlations to syntax [5]. While some studies argue that individual attention weights should not be used for explanation [14, 30, 4]; aggregate patterns provide an understanding of how contextualization occurs.

Despite this wide-spread interest in understanding large-scale language models, most work has focused on encoder-only models such as BERT, and there is a limited understanding of the representations of generative models such as GPT-2. Initial studies have used prompt-based generation to study the understanding of syntax. Most closely related to our work is that of Tenny et al. [33] which performs probing analysis on GPT-2 and has similar conclusions around the availability of linguistic information. We hope to provide greater context to these results via Representational Similarity Analysis and tie the decreased linguistic performance to the altered objective from training.

[1] In this paper, we assume general familiarity with GPT-2 and BERT, though dive into the architecture where relevant (i.e. attention in Section 5). To review these models, please see Radford et al. [24] and Devlin et al. [8]

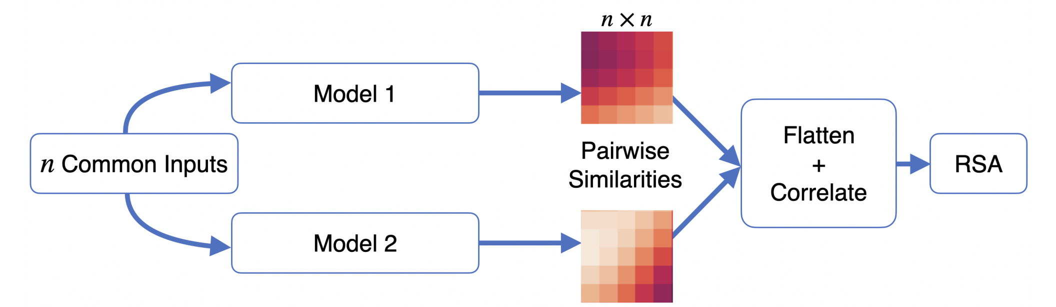

Figure 2: Schematic of how Representational Similarity Analysis is conducted. Rather than directly comparing representations between two model classes, n shared input examples are used to create $n \times n$ pairwise similarity matrices in the two representation spaces. In this paper, we exclusively use a *cosine* kernel; the matrices are then flattened and correlated to produce the similarity measure.

3 How do representations develop in a generative models like GPT-2?

We first seek to understand how representations develop in GPT-2 and compare to those developed in an encoder-only BERT model.[2] This study should help generically compare the representation spaces and elucidate how and wherein these models differ. However, note that measuring similarity in such high-dimensional vector spaces ($\mathbb{R}^{768}$ for both models) directly is often intractable as there is not a clear choice of what kernel to use. Instead, to quantitatively measure the differences in representation spaces, we turn to Representational Similarity Analysis (RSA: [16, 18]), a technique developed in the neuroscience literature to compare neuron activations. In the context of machine learning, this method has previously been used to analyze the effects of fine-tuning [20] and compare computer vision models [15, 25].

3.1 Representational Similarity Analysis

To measure similarity between two representation spaces, RSA starts with a common set of n input examples. By computing the two representations for each example (i.e. processing by different models), we obtain two comparable sets. Using kernels, we obtain two pairwise similarity matrices in $\mathbb{R}^{n \times n}$. The outputted similarity score is obtained by computing the Pearson correlation of the flattened upper triangular of these matrices, as diagrammed in Figure 2. Up to the kernel choice and normalization strategy, this method is comparable to Canonical Correlation Analysis (CCA: [13]) and Centered Kernel Alignment (CKA: [15]) which have also been used to compare neural network representations [19, 25]. In this work, we use the *cosine* kernel. As the *cosine* kernel is rotation invariant and the Pearson correlation is scale invariant, RSA can be thought of as measuring the degree of anisotropic scaling (though non-linear transformations break this analogy).

3.2 Experimental Setup

Formally, let m be the hidden size of the Transformer. Then, define $h_i^l \in \mathcal{R}^m$ to be the hidden representation of the i-th word at the l-th attention layer. Note, that some models utilize sub-word tokenization, so the representation of h_i^l is calculated via the constituent embeddings.[3] In our application of RSA, we first seek to understand the effect of each layer in the GPT-2 generative model. For $n = 500$ input sentences from the SQUAD validation set, which originate from English Wikipedia and should be similar to the data seen during pre-training [26]. We choose a random word denoted by index i. We then extract h_i^l for all layers l; used to create representations sets H^l. We then compute the similarity score between each pair of layers. The same procedure is performed for layers of BERT: B^l. Finally, we wish to compare the representations between these models; as H^l and B^l were created with the same n input examples, we can again apply RSA (see Figure 2).

[2] In this paper, we compare GPT-2 to *bert-base-cased* as both have 12 layers, 12 attention heads, and 768 hidden size. This is critical as prior work has noted that network size can often obsucre similarity measures [20].

[3] To match prior work, sub-word embeddings are aggregated via a self-attentive span extractor, with an implementation made available from Jiant NLP `https://git.io/JDfXY` [23].

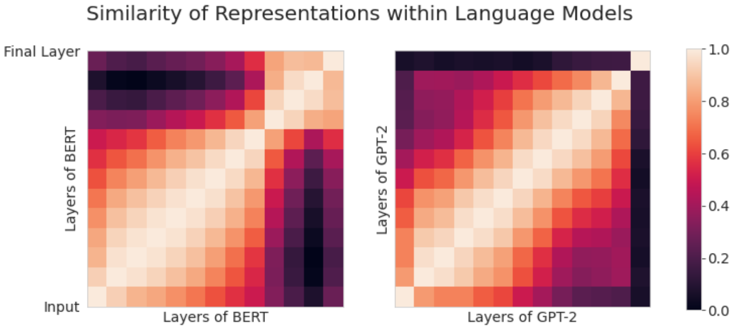

Figure 3: A comparison of layer-wise evolution of representation within BERT (left) and GPT-2 (right). Both models display a block pattern, suggesting some layers may not be *operative*.

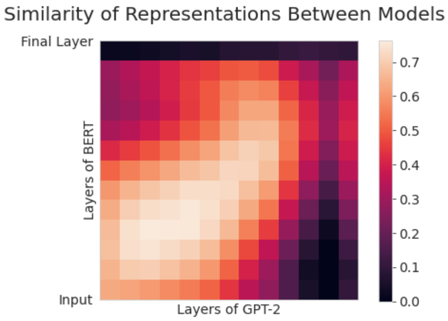

Figure 4: A comparison of the representations between GPT-2 and BERT. Early layers of both models display high similarity; however, this degrades quickly, especially for later layers in GPT-2.

3.3 Results: Early Layers are Similar but Outputs Diverge

We start by comparing the representations within both GPT-2 and BERT to explore the layer-wise evolution of representations in these language models. From the results presented in Figure 3, both BERT (left) and GPT-2 (right) display explicit block-like patterns, though the effect is slightly stronger in BERT. This result indicates that in both models there are many layers that perform minor updates (mostly rotations and isotropic scaling within blocks) and a few that are *operative* that significantly update the model's understanding of the inputted text (layers between blocks). For BERT, there appears to be one operative layer at layer 9, whereas for GPT-2 this occurs much earlier at layer 6. While these figures suggest that there may be an opportunity to shrink these Transformer models and retain the few layers that appear operative, the experiment shows that the layer-wise changes in the models are otherwise similar.

Additionally, the Representational Similarity Analysis method allows to compare the representation spaces found by GPT-2 and BERT, to see how the differences in objective and attention structure effect model representations. With the results of this experiment seen in Figure 4, we see that the two models appear quite similar in early layers.[4] Interestingly, the two models appear quite similar in early layers; suggesting that the word embeddings and first steps of contextualization are similar. As prior work notes that these layers of BERT often focus on understanding the structure of the inputted text [32, 33], we surmise that GPT-2 is performing similar updates. However, the models appear to diverge halfway through, with similarity dropping drastically from layers 7 and beyond.

What differs in the final few layers of these models? Could it relate to GPT-2 having an *operative* update earlier in the layer-wise evolution? In the rest of this paper, we attempt for a fine-grained understanding of the differences between these models, turning first to probing of linguistic phenomena.

[4] Note the similarity scale is relative; similarity scores of 0.6 have been obtained in prior work simply from pre-training two BERT models with different random seeds, so the 0.7 RSA similarity of the early layers is relatively quite high [20].

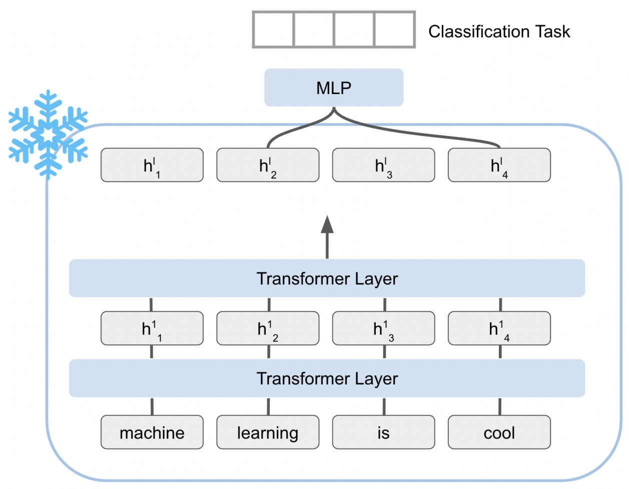

Figure 5: Schematic diagram of how edge probes are trained as an MLP on top of a frozen encoder model. For dual input tasks such as dependencies and relations understanding, two word representations are extracted from the language models and concatenated to be used as input to the MLP probe.

4 Do generative models (GPT-2) comparably understand linguistic features?

In contrast to the task-agnostic measures of similarity used in the previous section, supervised probes provide a targeted way to understand the linguistic capabilities of large-scale language models and discern differences in the representation spaces; we utilize these probing techniques to investigate why representations in laters layers of BERT and GPT-2 appear to diverge.

4.1 Experimental Setup

We approach the problem of quantitatively measuring the degree of language understanding in generative models by using supervised probes. Specifically, we explore results from the edge probes designed by Tenney et al. [33, 32], which freeze the language model and search for linguistic understanding via small MLPs trained on extracted representations. These probes are then trained via binary cross entropy losses per label, as some edge probes are multi-label and standard softmax losses would enforce an undesired exclusivity constraint.

Formally, the pre-trained model is frozen and relevant hidden states h_i^l are fed into an auxiliary MLP (a shallow, 2 layer network with a hidden size of 768 * the number of inputs). For single-input tasks such as part-of-speech, the input is h_i^l. Other tasks, such as determining dependency relations, require dual inputs h_i^l and h_j^l which are concatenated before the application of the MLP. To limit the amount of memorization [11], we only train the edge probes for 3 epochs using AdamW and use a learning rate of 1×10^{-4}.

In this paper, we focus on two[5] edge probes, Dependencies and Relations, with additional detail provided in Table 1. The Dependencies probe is a classification task based on the Universal Dependencies parse of the English Web Treebank (ETW) [31]. This task is often thought of as syntactic, as edges are classified into categories such as compound:prt, nmod, ... Good edge probe performance would indicate that the associated embeddings understand the structure of a sentence but not necessarily the meaning. To test semantic understanding, we utilize the Relations edge probe which is derived from SemEval-2010 Task 8, with labels like cause-and-effect or entity-destination. For individual examples from these edge probes, we direct the reader to Appendix 6.1.

For all probes, we report the micro-averaged F1 score on a held-out test set, to be consistent with prior work and across tasks. Note, micro-averaged F1 is used over raw accuracy as the data can be multi-label (the two are equivalent when every input has 1 label).

[5] Due to data limitations in accessing OntoNotes 5.0 (as the authors are not affiliated with a NLP lab at Stanford), the full set of edge probes is unavailable to us.

Table 1: Details of the edge probes used to measure language understanding. Note, the exact number of training examples does not match prior work as smaller train/test sets were created to feasibly run on available compute. Data splits will be made available with code.

Edge Probe	Description	Training Examples	Test Examples	Number of Lables	Example Labels
Dependencies	Syntactic Task	25110	4757	49	compound:prt nmod

Relations	Semantic Task	8000	2717	19	Cause-Effect(e1,e2)
					Entity-Origin(e2,e1)

Table 2: Results from the Dependencies and Relations edge probes show that BERT and GPT-2 have comparable language understanding near layer 6, a drastic improvement from the lexical baselines trained only on word embeddings. However, by the end of the models, GPT-2 lags behind in semantic understanding measured by the Relations task.

		Micro-Averaged F1 (%)		
Edge Probe	Model	Lexical	Layer 6	Layer 11
Dependencies	BERT	66.1	91.3	**92.3**
	GPT-2	64.7	90.7	90.7
Relations	BERT	49.5	81.3	**84.0**
	GPT-2	49.6	79.2	81.1

4.2 Results: Generative Models Lag in Semantic Understanding

Table 2 provides the results from two discussed edge probes on a selected set of layers. As baselines, we probe the de-contextualized word embeddings that are provided by both BERT and GPT-2. As Hewitt and Liang [11] explored in 2019, the MLP probes are often strong enough to find patterns even from random noise. The de-contextualized, lexical metrics therefore serve as strong baselines, and improvement over these numbers can be attributed to contextualization (the cross-token attention and further processing in the Transformers).

We start our analysis with the Dependency probe; note that both GPT-2 and BERT make significant strides in understanding the syntax structure as the model progresses; even the output performance is relatively comparable with only a $\approx 1\%$ difference in micro-averaged F1 score. These results suggest that both models well-understand syntax and do-so early (by layer 6).

Finally, we look at the relations probe to investigate semantic understanding; we note that the performance of both models is comparable up to layer 6 (as prior work notes that the noise scales from training can be large [33, 32]. However, by the output, BERT outpaces the GPT-2 model by a significant amount ($+3\%$). This result suggests that the later layers of BERT are dedicated to forming such high-level semantic understanding of text which is not obtained by GPT-2. One possible explanation for this result is that GPT-2 can read text in one direction, so there is only one chance to determine relationships between spans rather than two in the bi-directional model.[6]

5 How do generative models utilize context when creating representations?

The results so far generally suggest that bi-directional and generative models adapt and contextualize words in a similar manner for the first few layers; yet the encoder-only models eventually are able to better understand complex linguistic phenomena.

We hypothesize this result is due to the difference in objective functions. BERT is trained to predict the identity of masked words (referred to as Masked Language Modelling, MLM), whereas GPT-2 is trained on next-word prediction. The later task appears much more local in nature, to only find words that validly continue the prompt. Masked Language Modelling appears more generic, requiring far-ranging connections to deal with the large number of dropped tokens and fill in the masks.

[6] For example, consider a cause followed by an effect. GPT-2 must learn the effect $\rightarrow$ cause relationships, whereas BERT is able to learn and update in both directions, cause $\leftrightarrow$ effect.

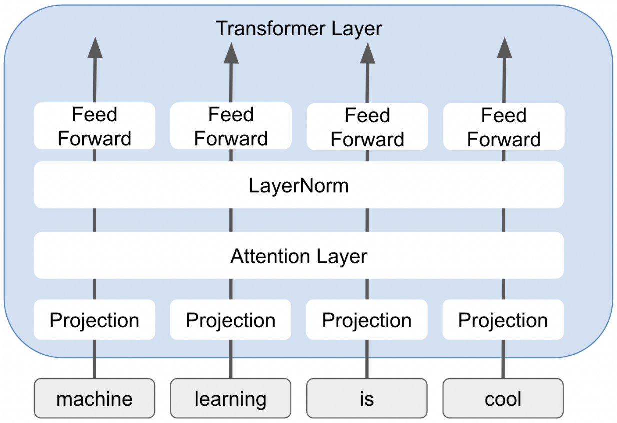

Figure 6: Schematic diagram of the first Transformer layer in a BERT or GPT-2 model. The attention layer is the main step for creating contextualized embeddings rather than just having good word embeddings.

To test this hypothesis, we turn to our final method of analysis, examining the attention structure. Attention weights provide insight into the inter-token interactions within a Transformer model and have been previously analyzed for interpretable patterns of connections [5, 6].

5.1 Attention Definition

Attention is one of core building blocks of the Transformer model diagrammed in Figure 6. The attention layer allows for inter-token interactions, core to the idea of building *contextualized* embeddings rather than good word embeddings [21]. Transformer models such as BERT and GPT-2 use the traditional form of attention referred to as Scaled Dot-Product Attention:

$$a_i = \text{Attention}(q_i, K, V) = \text{softmax}\left(\frac{q_i K^{\top}}{\sqrt{m}}\right) V$$

In the Transformer formulation, q_i is a projected token to be updated, K is the projected matrix of the tokens being attended to, m is again the dimension of the representation, and $V = K$. Note, softmax($\frac{q\dot{K}^{\top}}{\sqrt{m}}$) is referred to as the attention score. Due to the softmax, the attention scores for a single input q_i sum to 1. Why is this interesting? As attention weights have a bounded sum, looking at the relative weights is thought to provide some context to where the model is looking when creating or updating contextual representations. See Figure 7 for a depiction of attention weights for a sample sentence from the BERT and GPT-2 models.

5.2 Experimental Setup

In our case, we would like to understand the influence of context. Specifically, our hypothesis boils down to a question of whether GPT-2 focuses more on local context in comparison to BERT. To answer this question, we turn to attention weights and will compute the weighted mean absolute distance from the input token to formalize this notion of locality. That is to say, let a_{ij} be the attention score from token i to token j. The weighted mean distance for an input token i is calculated:

$$\text{Mean Absolute Distance} = \sum_j a_{ij} |i - j|$$

We would like this metric to be informative and comparable for BERT and GPT-2. Considering the examples in Figures 1 and 7, we must make three slight modfications:

1) First, we must ensure that the mean distance is not biased by sentence length. Specifically, as BERT is bi-directional, the mean distance results would not be comparable if the chosen

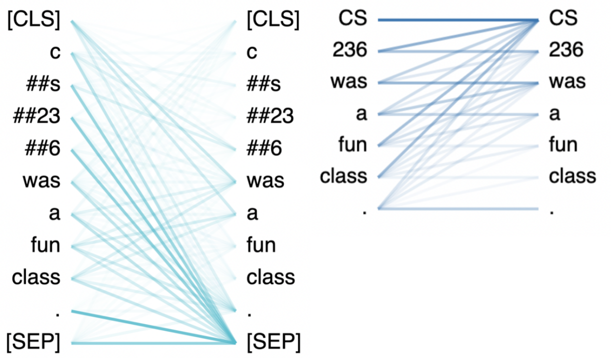

Figure 7: Examples of the attention structure for BERT (left) and GPT-2 (right). The causal structure of the attention as discussed in Figure 1 is clear. The differing number of tokens comes from the use of different tokenization strategies from GPT-2 and BERT.

token i is always in the first half of the sentence. To fix this, we choose token i to be in the latter half of a sentence.

2) We also must correct for tokenization. From Figure 7, it is clear that words are tokenized differently for BERT than GPT-2, as the earlier uses SentencePiece (sub-word) tokenization, a practice of splitting unknown words into pieces [17, 8]. To account for this difference, we compute the normalized mean absolute difference, which is equivalent to the above but divided by the length after tokenization to obtain a score from 0 to 1.

3) Finally, BERT and GPT-2 both use multi-head attention (for details see [3]). Rather than simply provide a single score mean distance score per head, we plot the sorted scores for all attention heads and look at the distribution.

For simplicity, to refer to the metric with all of these modifications as the Normalized Mean Absolute Distance in the associated results.

5.3 Results: BERT uses greater global context than GPT-2

The results for the attention analysis at a few selected layers is presented in Figure 8. Layer 1 serves as a baseline to show that at this stage of processing, the attention distributions are fairly similar (if not more global for GPT-2). However, the story flips past layer 6. At this point, BERT displays a remarkably higher propensity to focus on words/tokens that are farther away in comparison to GPT-2. This trend is also true for the last layer, where relatively BERT continues to have a more global context in comparison to GPT-2.

This trend may help understand why GPT-2 lags behind in language understanding. In attempting to optimize for the next word, GPT-2 may fail to create a complete global understanding; leading to worse semantic understanding (as seen in the edge probes) and differences in outputs (as seen in the representational similarity).

6 Conclusion

In this paper, we used a variety of analytic method to compare a generative Transformer model in GPT-2 with its bi-directional counterpart BERT. From unsupervised Representational Similarity Analysis, we find that the two models have similar word embeddings and evolve representations similarly for the first few models. However, later layers, beginning after layer 6, diverge.

We explain this divergence in two forms. First, via edge probing we find that the understanding of language is similar between both models at layer 6 but the generative model drastically lags behind in semantic understanding near the output. Attention analysis suggests that this result may arise from the greater use of context in BERT compared to GPT-2, which appears too narrowly focused on producing the next word to create a complete understanding of the input text.

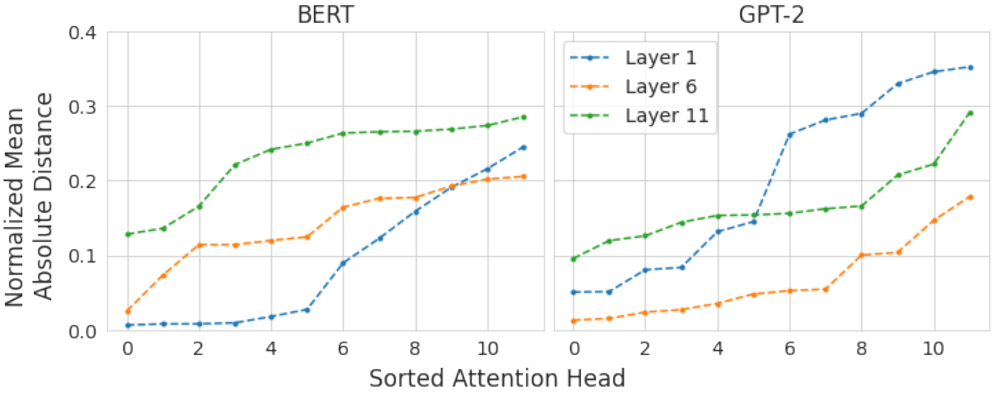

Figure 8: Normalized Mean Absolute Distance computed for the attention heads at various layers of the BERT (left) and GPT-2 (right) models. The distribution

Overall, these results suggest that generative modelling-as done in GPT-2-may not be sufficient to create a complete understanding of language. It remains to be seen if this is a fundamental issue with viewing language in 1-direction or whether it is possible to train generative models that solve these problems.

6.1 Future Work

Before ending, we would like to propose a few directions of future work that we think are promising.

- From Figure 3, it appears that the Transformer models studied in this paper only have a few *operative* layers. Is it possible to use this information for model distillation? Note, we cannot directly skip layers as later layers are not invariant to rotations / isotropic scaling of the input.
- Based on the results of section 5, it may be possible to regularize the attention and force the generative models such as GPT-2 to pay greater attention to the global context.
- Scalar mixing (i.e. a learned weighted average of the representations in various layers) has been shown to be effective for creating BERT-like encoders [32]. Could similar strategies be useful for generative models?
- The same analysis presented in this paper could be extended to the wide-variety of encoder-only models that are now on the GLUE and SuperGLUE leaderboards. Are there any interesting insights that can be gained from a large-scale analysis?

Appendix

Code

This project was implemented using the pre-trained models made available by the Hugging Face team, in particular using the Transformers and Datasets libraries [42]. The edge probing code additionally made use of Jiant [23].

We provide a GitHub repository containing sample notebooks used to perform the RSA analysis, edge probing training, and attention experiments at `https://github.com/amil5/cs236-final-project`. Note that all reproductions should be done via GPU (though an attempt was made to make the code CPU-friendly). The notebooks in this example also made use of a modified fork of the Jiant repositry available at `https://github.com/nyu-mll/jiant`, a tool made by researchers at NYU to explore large scale NLP models [23]. The 2.0 update allowed for easy integration of the HuggingFace Transformers library; however, slight modification (hence the fork) was required to register GPT-2 and ensure appropriate tokenization. The modified fork is available at: `https://github.com/amil5/jiant`.

The provided notebooks were all run using Google Colab, with a Pro subscription to prevent timeouts. No guarantees are made that the code will complete with the Free access tier or on CPUs (which seems to be okay per course guidelines).

Examples from the Edge Probe Data

To provide the reader greater insight into the edge probe data, here we provide examples of data from the Dependencies and Relations test sets. Note probes are designed so that the label corresponds to the relation from $e1$ to $e2$ in cases where ordering matters.

Table 3: Examples from the edge probe test datasets.

Edge Probe	Text	Label
Dependencies	Does anybody $\underbrace{\text{use}}_{\text{e2}}$ $\underbrace{\text{it}}_{\text{e1}}$ for anything else?	obj
	Is $\underbrace{\text{that}}_{\text{e1}}$ a money $\underbrace{\text{maker}}_{\text{e2}}$?	nsubj
	But will diplomacy $\underbrace{\text{work}}_{\text{e2}}$ $\underbrace{\text{?}}_{\text{e1}}$	punct
Relations	The $\underbrace{\text{castle}}_{\text{e1}}$ was inside a $\underbrace{\text{museum}}_{\text{e2}}$.	Component-Whole(e1, e2)
	Seniors get much $\underbrace{\text{joy}}_{\text{e1}}$ from $\underbrace{\text{animals}}_{\text{e2}}$.	Cause-Effect(e2, e1)
	The gardeners blew the $\underbrace{\text{leaves}}_{\text{e1}}$ into the $\underbrace{\text{street}}_{\text{e2}}$.	Entity-Destination(e1,e2)

Acknowledgments and Disclosure of Funding

The authors would like to thank the CS236 TAs and instructors for their help in designing and discussions around the course project. AM is a first-year ICME PhD student supported by an ICME fellowship.

References

[1] Mostafa Abdou et al. “Higher-order comparisons of sentence encoder representations”. In: *arXiv preprint arXiv:1909.00303* (2019).

[2] Samira Abnar et al. “Blackbox meets blackbox: Representational similarity and stability analysis of neural language models and brains”. In: *arXiv preprint arXiv:1906.01539* (2019).

[3] Terra Blevins, Omer Levy, and Luke Zettlemoyer. “Deep RNNs encode soft hierarchical syntax”. In: *arXiv preprint arXiv:1805.04218* (2018).

[4] Gino Brunner et al. “On identifiability in transformers”. In: *arXiv preprint arXiv:1908.04211* (2019).

[5] Kevin Clark et al. “What does bert look at? an analysis of bert’s attention”. In: *arXiv preprint arXiv:1906.04341* (2019).

[6] Andy Coenen et al. “Visualizing and measuring the geometry of BERT”. In: *arXiv preprint arXiv:1906.02715* (2019).

[7] Alexis Conneau et al. “What you can cram into a single vector: Probing sentence embeddings for linguistic properties”. In: *arXiv preprint arXiv:1805.01070* (2018).

[8] Jacob Devlin et al. “Bert: Pre-training of deep bidirectional transformers for language understanding”. In: *arXiv preprint arXiv:1810.04805* (2018).

[9] Allyson Ettinger. “What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models”. In: *Transactions of the Association for Computational Linguistics* 8 (2020), pp. 34–48.

[10] Yoav Goldberg. “Assessing BERT’s syntactic abilities”. In: *arXiv preprint arXiv:1901.05287* (2019).

[11] John Hewitt and Percy Liang. “Designing and interpreting probes with control tasks”. In: *arXiv preprint arXiv:1909.03368* (2019).

[12] John Hewitt and Christopher D Manning. “A structural probe for finding syntax in word representations”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 2019, pp. 4129–4138.

[13] Harold Hotelling. “Relations between two sets of variates”. In: *Breakthroughs in statistics*. Springer, 1992, pp. 162–190.

[14] Sarthak Jain and Byron C Wallace. “Attention is not explanation”. In: *arXiv preprint arXiv:1902.10186* (2019).

[15] Simon Kornblith et al. “Similarity of neural network representations revisited”. In: *International Conference on Machine Learning*. PMLR. 2019, pp. 3519–3529.

[16] Nikolaus Kriegeskorte, Marieke Mur, and Peter A Bandettini. “Representational similarity analysis-connecting the branches of systems neuroscience”. In: *Frontiers in systems neuroscience* 2 (2008), p. 4.

[17] Taku Kudo and John Richardson. “Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing”. In: *arXiv preprint arXiv:1808.06226* (2018).

[18] Aarre Laakso and Garrison Cottrell. “Content and cluster analysis: assessing representational similarity in neural systems”. In: *Philosophical psychology* 13.1 (2000), pp. 47–76.

[19] Rebecca Marvin and Tal Linzen. “Targeted syntactic evaluation of language models”. In: *arXiv preprint arXiv:1808.09031* (2018).

[20] Amil Merchant et al. “What Happens To BERT Embeddings During Fine-tuning?” In: *arXiv preprint arXiv:2004.14448* (2020).

[21] Jeffrey Pennington, Richard Socher, and Christopher D Manning. “Glove: Global vectors for word representation”. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, pp. 1532–1543.

[22] Matthew E Peters et al. “Deep contextualized word representations”. In: *arXiv preprint arXiv:1802.05365* (2018).

[23] Jason Phang et al. *jiant 2.0: A software toolkit for research on general-purpose text understanding models*. http://jiant.info/. 2020.

[24] Alec Radford et al. "Language models are unsupervised multitask learners". In: *OpenAI blog* 1.8 (2019), p. 9.

[25] Maithra Raghu et al. "Do Vision Transformers See Like Convolutional Neural Networks?" In: *Thirty-Fifth Conference on Neural Information Processing Systems*. 2021.

[26] Pranav Rajpurkar et al. "Squad: 100,000+ questions for machine comprehension of text". In: *arXiv preprint arXiv:1606.05250* (2016).

[27] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. "A primer in bertology: What we know about how bert works". In: *Transactions of the Association for Computational Linguistics* 8 (2020), pp. 842–866.

[28] Naomi Saphra. "Training dynamics of neural language models". In: (2021).

[29] Naomi Saphra and Adam Lopez. "Understanding learning dynamics of language models with SVCCA". In: *arXiv preprint arXiv:1811.00225* (2018).

[30] Sofia Serrano and Noah A Smith. "Is attention interpretable?" In: *arXiv preprint arXiv:1906.03731* (2019).

[31] Natalia Silveira et al. "A Gold Standard Dependency Corpus for English". In: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*. 2014.

[32] Ian Tenney, Dipanjan Das, and Ellie Pavlick. "BERT rediscovers the classical NLP pipeline". In: *arXiv preprint arXiv:1905.05950* (2019).

[33] Ian Tenney et al. "What do you learn from context? probing for sentence structure in contextualized word representations". In: *arXiv preprint arXiv:1905.06316* (2019).

[34] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems*. 2017, pp. 5998–6008.

[35] Jesse Vig and Yonatan Belinkov. "Analyzing the structure of attention in a transformer language model". In: *arXiv preprint arXiv:1906.04284* (2019).

[36] Elena Voita, Rico Sennrich, and Ivan Titov. "The bottom-up evolution of representations in the transformer: A study with machine translation and language modeling objectives". In: *arXiv preprint arXiv:1909.01380* (2019).

[37] Elena Voita et al. "Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned". In: *arXiv preprint arXiv:1905.09418* (2019).

[38] Alex Wang et al. "GLUE: A multi-task benchmark and analysis platform for natural language understanding". In: *arXiv preprint arXiv:1804.07461* (2018).

[39] Alex Wang et al. "Superglue: A stickier benchmark for general-purpose language understanding systems". In: *arXiv preprint arXiv:1905.00537* (2019).

[40] Sarah Wiegreffe and Yuval Pinter. "Attention is not not explanation". In: *arXiv preprint arXiv:1908.04626* (2019).

[41] Adina Williams, Nikita Nangia, and Samuel R Bowman. "A broad-coverage challenge corpus for sentence understanding through inference". In: *arXiv preprint arXiv:1704.05426* (2017).

[42] Thomas Wolf et al. "Huggingface's transformers: State-of-the-art natural language processing". In: *arXiv preprint arXiv:1910.03771* (2019).

[43] *Write with Transformer*. https://transformer.huggingface.co/. Accessed: 2010-09-30.