Comparative Analysis of Encoder-Only, Decoder-Only, and Encoder-Decoder Language Models

Boyu Liu
College of Liberal Arts & Sciences, University of Illinois Urbana-Champaign, Champaign–Urbana Metropolitan Area, Illinois, U.S.A.

Keywords: Large Language Models, Natural Language Processing, Transformer, Gpt, Bert.

Abstract: With the surge in Artificial Intelligence (AI) popularity sparked by ChatGPT, a plethora of Transformer-based models have emerged, and the decoder-only architecture has become the mainstream development direction of large language models (LLMs) in most big-tech companies. In the rapidly advancing field of Natural Language Processing (NLP), understanding the capabilities and limitations of different language model architectures is critical for pushing the boundaries of AI. This paper delves into the comparative analysis of encoder-only, decoder-only, and encoder-decoder models, illuminating their strengths, weaknesses, and optimal use cases within the landscape of NLP. Encoder-only models are highlighted for their efficiency and deep understanding, decoder-only models for their generative capabilities and adaptability, and encoder-decoder hybrids for their versatile application across a broad spectrum of NLP tasks. This comparative analysis provides valuable insights into the strategic deployment of these models in real-world applications and underscores the ongoing need for innovation in model architecture to optimize performance and computational efficiency.

1 INTRODUCTION

About six years ago, a renowned paper titled "Attention Is All You Need" was officially published, the first to introduce the concept of the attention mechanism. It created a brand-new model in Natural Language Processing (NLP) called the Transformer, which is groundbreaking and has become the predecessor of most of today's mainstream Language Models (LMs) (Shazeer et al. 2017). The Transformer architecture consists of two distinct components: an encoder, which processes the input data, and a decoder, which generates the output sequence. These components are crucial in shaping the subsequent evolution of models such as OpenAI's Generative Pretrained Transformer (GPT) series, which are primarily decoder-based, and Google's Bidirectional Encoder Representations from Transformers (BERT), which rely exclusively on the Transformer model's encoder mechanism. These three models have revolutionized how people approach language understanding and generation tasks, especially decoder-based models like ChatGPT, as a game changer, started a new industrial revolution of Artificial General Intelligence (AGI), and pushed the AI craze to an unprecedented height. However, before the release of GPT-3.0, the decoder-based transformer models were not as favored as encoder-decoder transformer models or even the encoder-only models like BERT. What intrinsic differences among these three approaches have influenced the rise in popularity of GPT over the others? Is decoder-only LMs always better? In which case do people use BERT, and in which case do people use GPT? Exploring the differences among LMs based on these three architectures is crucial. Different NLP tasks have distinct requirements: some need a deep understanding of context, while others prioritize creative language generation, or sometimes users need quick and accurate Q&A. By conducting a comparative analysis of encoder-only, decoder-only, and encoder-decoder hybrid LMs, researchers and practitioners can gain valuable insights into which model is more suited for specific kind of tasks. This understanding can lead to more efficient and effective deployment of these models in real-world applications, ranging from automated customer service to advanced research in linguistics and AI. This paper aims to provide a comprehensive comparative analysis of these three kinds of LMs,

Liu, B.
Comparative Analysis of Encoder-Only, Decoder-Only, and Encoder- Decoder Language Models.
DOI: 10.5220/0012829800004547

In *Proceedings of the 1st International Conference on Data Science and Engineering (ICDSE 2024)*, pages 524-530
ISBN: 978-989-758-690-3

shedding light on their strengths, weaknesses, and optimal use cases in the ever-evolving landscape of NLP.

2 BACKGROUND

2.1 Transformer Model and Encoder-Decoder Architecture

The Transformer model comprises two key segments: an encoder and a decoder, each with a stack of six identical layers (Figure 1). The encoder layers are each made up of two sub-layers: a multi-head self-attention mechanism and a position-wise feed-forward network, both augmented by residual connections and normalized on a layer basis (Shazeer et al. 2017). The decoder, mirroring the encoder's structure, includes an additional third sub-layer in each of its layers, which performs multi-head attention over the encoder's output. This feature allows the decoder to focus on relevant parts of the input sequence, thereby facilitating more accurate and contextually informed predictions. A key innovation in the Transformer is its use of masked multi-head self-attention in the decoder, which ensures predictions for a given position are dependent only on the known outputs at previous positions, making it particularly suited for sequence generation tasks.

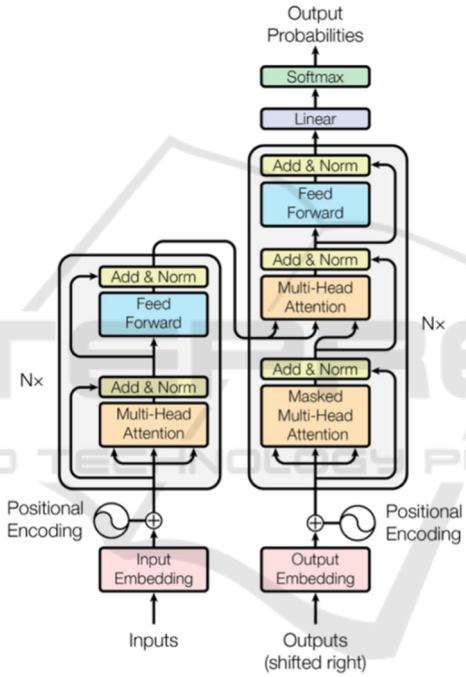

Figure 1: The Transformer – model architecture (Shazeer et al. 2017).

2.2 Encoder-Only Language Models

Encoder-only LMs only consists of a stack of encoder layers (Jacob et al. 2018). BERT is a prominent example of an encoder-only LM. These BERT-like Models primarily designed for tasks that involve understanding or processing input text, such as classification, sentimental analysis, and question answering. These models typically process the input text in a bidirectional manner, meaning that they consider the context from both the left and right sides of a token within the input sequence. This enables the model to possess a thorough grasp of the context surrounding each word.

During its pre-training phase, BERT-like models could employ two special tasks: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP) (Jacob et al. 2018, Radford et al. 2021).

2.2.1 MLM

The MLM task is formulated to address the unidirectionality constraint by randomly masking a portion of the input tokens and training the model to predict the original identity of the masked words based on their bidirectional context. This is achieved by replacing the masked tokens with a [MASK] token, random tokens, or leaving them unchanged, thereby enforcing the model to infer the missing

information from a composite understanding of the surrounding words. The MLM objective is mathematically represented as:

$$L_{MLM}(\theta) = -\sum_{i=1}^{n} \log p_\theta(x_i | x_{masked}) \quad (1)$$

where x_{masked} is the input with some tokens masked, x_i is the original token, and θ is the model parameters.

2.2.2 NSP

BERT incorporates the NSP task to learn relationships between sentences. In this binary classification task, BERT is designed to take in sentence pairs and is trained to predict if the second sentence is the logical and chronological next sentence in the document. This task enhances BERT's ability to capture the relationships between sentences, which is crucial for downstream tasks that involve understanding the structure of documents, such as question answering and natural language inference. The NSP task can be formalized as:

$$L_{NSP}(\theta) = -\frac{1}{N}\sum_{1}^{N} y_i log\, \hat{y}_i + (1 - y_i)\log(1 - \hat{y}_i) \quad (2)$$

where N is the number of sentence pairs, y_i indicates whether sentences are consecutive, and $\hat{y}_i$ is the model predicted probability.

2.3 Decoder-Only Language Models

Decoder-only LMs, like OpenAI's GPT, generally consist of multiple layers of modified Transformer decoder blocks stacked on top of each other. Each block comprises components of Masked Muti-head Self Attention, Position-wised Feed-Forward Networks, and Layer Normalization (Radford et al. 2021).

In contrast to encoder models, decoder-only models typically pertained and generate text in a unidirectional or auto-regressive manner. This means each token is generated based on the previously generated tokens without seeing future tokens in the sequence.

The autoregressive language modeling task can be mathematically represented as follows:

Given a sequence of tokens $x_1, x_2, \ldots, x_n$, the model predicts the next token x_{n+1} based on all previous tokens. The probability of the sequence can be factorized as:

$$P(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid x_1, x_2, \ldots, x_{i-1}) \quad (3)$$

For each step i, the model outputs a probability distribution over the vocabulary for the next token x_i, based on the previous tokens.

2.4 Mainstream LLMs of Different Architecture

While OpenAI's ChatGPT is undoubtedly the most popular and well-known language model in recent years, there are large numbers of decoder-only LMs, such as the latest Google AI model Gemini, Llama and Llama 2 developed by Meta, Google's Bard, LaMDA, PaLM, etc. For encoder-only LMs, there are also many other LMs besides BERT like Microsoft's DeBERTa, Google's ALBERT, and Meta's RoBERTa. Apart from encoder-only and decoder-only, seq2seq (encoder-decoder) models like Meta's BART and Google's T5 are also widely applied.

3 TRAINING EFFICIENCY

Since few studies focus specifically on training efficiency, and the actual training efficiency depends on various factors, this part will compare decoder LMs' and encoder LMs' training efficiency mainly from a theoretical perspective.

3.1 Pretraining Tasks Complexity and Time

For encoder-only LMs, tasks like MLM are inherently parallelizable since each masked token's prediction is relatively independent of others. This parallelism can lead to efficient use of computational resources, potentially reducing pretraining time. Due to their ability to process input tokens in parallel, encoder-only models can efficiently handle large batches of data, which can shorten the overall time required for pretraining.

For decoder-only LMs, the autoregressive pretraining task, where each token prediction depends on the previously predicted tokens, can limit parallel processing, potentially making pretraining more time-consuming than encoder-only models. While the sequential learning process is thorough, it might require more time to achieve similar levels of understanding and generation capabilities due to its inherent sequential processing constraints.

Encoder-decoder models are often pre-trained on a variety of complex tasks that require both understanding and generating text. While this makes them highly versatile, it also means that their pretraining can be the most time-consuming due to the complexity of the tasks and the need to learn both encoding and decoding capabilities. The use of

diverse pretraining tasks, including sequence-to-sequence transformations, can require extensive time to cover the breadth of capabilities these models are expected to learn.

3.2 Computational Resource Consumption

Mentioned in the previous part, MLM tasks parallelism in BERT like model pretraining can lead to efficient use of computational resources. However, the inability to fully parallelize the pretraining process means that decoder-only models might not utilize computational resources as efficiently as encoder-only models, potentially leading to longer pretraining times. Encoder-decoder LMs tend to require more memory and computational resources since they incorporate both encoder and decoder structures. Especially when employing complex attention mechanisms and a large number of parameters, they may have the highest resource consumption.

3.3 Parameter Efficiency

Encoder-only LMs generally achieve good performance on less data due to their deep text understanding capabilities, especially when fine-tuning task-specific objectives. We can get this conclusion from Table 1 and the analysis in 5.3.

Decoder-only LMs generally need a large volume of training data to generate high-quality text. They may not be as efficient as encoder-only models in learning from each sample, as generative tasks are inherently more complex than understanding tasks.

Encoder-decoder LMs may show certain advantages in data efficiency, particularly when tasks require both understanding and generative capabilities. They can improve sample efficiency through complex representations learned from large datasets, though this still depends on the nature of the task and the quality of the training data.

4 NATURAL LANGUAGE UNDERSTANDING ABILITY

Natural Language Understanding (NLU) refers to the ability of a LM to understand human language, which is a very important aspect to evaluate the overall ability of a LM. In this part, this paper will compare NLU ability of encoder-only LMs, decoder-only LMs, and encoder-decoder LMs based on SuperGLUE benchmark (Smith & Johnson 2020, SuperGLUE 2023).

4.1 Evaluation Metrics Used

4.1.1 Accuracy

Accuracy is a straightforward measure that quantifies the ratio of a model's correct predictions to its total predictions. It's commonly used in classification tasks, where the goal is to correctly identify the category to which a piece of data belongs. The formula for accuracy is given by:

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions} \quad (4)$$

4.1.2 F1-Score

The F1 score represents the harmonic mean of precision and recall, thereby striking a balance between the two metrics. It is used a lot in situations where there is an uneven class distribution or when false positives and false negatives carry different costs. The F1 score, which varies from 0 to 1, with 1 signifying ideal precision and recall, and 0 represents that the model did not correctly predict any positive cases.

$$F1\ Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (5)$$

4.1.3 Exact Match (EM)

It measures how often the system's answer is exactly the same as one of the correct answers. Mathematically, it can be expressed as a ratio or percentage

$$EM = \frac{Number\ of\ Exactly\ Correct}{Total\ Number\ of\ Questions} \times 100\% \quad (6)$$

4.2 Super GLUE

The SuperGLUE benchmark is a collection of natural language understanding tasks designed to evaluate and compare the performance of LMs (SuperGLUE 2023).

SuperGLUE consists of a suite of eight diverse tasks that cover a wide range of NLP abilities, including question answering, entailment, co-reference resolution, and more. These tasks are:

1) **BoolQ:** A question-answering task requiring the model to provide a boolean (yes/no) answer to a question based on a short passage. This task is evaluated with accuracy.
2) **CommitmentBank (CB)**: A textual entailment task that involves determining if a text entails,

contradicts, or is neutral to a given hypothesis. This task is evaluated using accuracy and F1.

3) **Choice of Plausible Alternatives (COPA)**: A causal reasoning task where the model must select the most plausible cause or effect from two given options for a given premise. This task is evaluated using accuracy.
4) **Multi-Sentence Reading Comprehension (MultiRC)**: A question-answering task where questions may have multiple correct answers which must be identified from a list of possible options. This task is evaluated using F1 and EM.
5) **Reading Comprehension with Commonsense Reasoning Dataset (ReCoRD)**: A task where the model fills in the blank in a sentence using a list of provided entities, requiring commonsense reasoning. The evaluation metrics used are F1 and EM.
6) **Recognizing Textual Entailment (RTE)**: Similar to CB, this task involves determining whether one sentence logically follows from another.
7) **Words in Context (WiC)**: A word sense disambiguation task that requires the model to determine whether a word is used in the same way in two different sentences. The evaluation metric used is accuracy in this task.
8) **Winograd Schema Challenge (WSC)**: A co-reference resolution task where the model must determine which noun a given pronoun refers to in a sentence, often requiring complex commonsense reasoning. The evaluation metric used is accuracy in this task.

4.3 Analysis of SuperGLUE Benchmark

Table 1 listed the top 10 LMs based on SuperGLUE scores, of which 5 are encoder-only LMs, 4 are encoder-decoder, and only one, PaLM, is decoder-only. Among the top-10 models in the SuperGLUE leaderboard, encoder-only LMs and encoder-decoder LMs outnumber decoder-only LMs by a ratio of 5 to 1 respectively. Also, in the top 30 rankings, the number of encoder-only models far exceeds the number of decoder-only models. This result indicates that encoder-only LMs and encoder-decoder LMs have generally stronger NLU capability than decoder-only LMs.

Notably, OpenAI's GPT-3 ranks 25th, just 0.3 points higher than the SuperGLUE Baseline, BERT++. However, GPT-3 has 174 billion parameters, which is 62 times more than BERT++'s 2.8 billion. Moreover, the sole decoder-only model in the top ten, PaLM, has 90 times more parameters than Vega v2, the highest-scoring model. Also, the second highest LM ST-MoE-32B has approximately 5 times the parameters as Vega v2. Thus, to achieve comparable NLU abilities, encoder-decoder LMs require significantly more parameters than encoder-only LMs, and decoder-only LMs even require a much greater number of parameters than encode-decoder LMs.

Table 1: Top 10 LMs from SuperGLUE leaderboard (Zhong et al. 2022).

Model Name	Super GLUE Score	Architecture Type	Number of Parameters
Vega v2	91.3	Encoder-only	6 billion
ST-MoE-32B	91.2	Encoder-Decoder	32 billion
METRO-LM	90.9	Encoder-only	5.4 billion
ERNIE 3.0	90.6	Encoder-only	10 billion
PaLM 540B	90.4	Decoder-only	540 billion
T5 + UDG, Single Model (Google Brain)	90.4	Encoder-Decoder	11 billion
DeBERTa / TuringNLRv4	90.3	Encoder-only	3.2 billion
SuperGLUE Human Baseline	89.8	N/A	N/A
T5	89.3	Encoder-Decoder	11 billion
Fronzen T5 1.1 + SPoT	89.2	Encoder-Decoder	11 billion
NEZHA-Plus	86.7	Encoder-only	2.8 billion

5 ZERO-SHOT/FEW-SHOT GENERALIZATION CAPABILITIES

Zero-shot generalization refers to a model's ability to apply learned knowledge to tasks without any task-specific training; Few-shot learning indicates that a machine learning model can learn a new task from a very small amount of training data. These generalization capabilities are crucial measures of a model's ability to comprehend and utilize its pre-trained knowledge in novel contexts.

5.1 Related Experiment

The research made by Wang, T. et al. studied how different pre-training objectives and architectural choices affect the zero-shot generalization abilities of language models (MMLU 5-Shot Leaderboard, 2024). Limited to BERT-like LMs' generative capability, they cannot conduct a study in its zero-shot setting, so encoder-only LMs are ignored from the study. The study finds that: A causal decoder-only model pre-trained with full language modeling (predicting the next word in a sequence) performs best in zero-shot tasks where no additional fine-tuning is done on specific tasks, and an encoder-decoder model pre-trained with masked language modeling (predicting masked words) outperforms others when fine-tuning is done on multiple tasks (MMLU 5-Shot Leaderboard, 2024). Therefore, the decoder-only Model performs the best in zero-shot generalization capability.

5.2 Massive Multitask Language Understanding (MMLU) Benchmark Analysis

The MMLU benchmark is a comprehensive test designed to evaluate the generalization abilities of language models across a wide range of subjects. It consists of multiple-choice questions derived from exams in various disciplines, from humanities to STEM fields, challenging the models to apply their knowledge to unfamiliar problems (Wang et al. 2023, Shazeer et al. 2017, Bahdanau et al. 2014). The benchmark assesses not just the depth of a model's training but also its capacity to transfer learning to new contexts without additional fine-tuning (zero-shot). In the following test, this paper will compare each model's MMLU Score with 5-shot fine-tuning (few-shot) (Table 2).

Table 2: Top 10 LMs from MMLU 5-shots leaderboard (Wang et al. 2023).

Model	MMLU Score	Architectural Type
GPT-4	86.4	Decoder-only
Gemini Ultra	83.7	Decoder-only
PaLM 2	78.3	Decoder-only
PaLM	75.2	Decoder-only
Gemini Pro	71.8	Decoder-only
Mistral 8x7b	71.3	Decoder-only
GPT-3.5	70	Decoder-only
Zephyr 7b	66.08	Decoder-only
Llama 2 65b	63.4	Decoder-only
Mistral 7b	60.1	Decoder-only

This paper obtains Top 10 LMs in MMLU score, where all the top models on the MMLU leaderboard are decoder-only. Based on the data from the MMLU leaderboard, it appears that decoder-only language models exhibit dominantly strong few-shot generalization capabilities.

In contrast, from the benchmark the encoder-decoder LMs is not showing comparable few-shot generalization capability. And due to the weakness of generative ability, encoder-only LMs are meaningless to compare.

6 CONCLUSION

In conclusion, comparative analysis of encoder-only, decoder-only, and encoder-decoder language models in this paper reveals distinct trade-offs in terms of training efficiency, NLU, and zero-shot/few-shot generalization capabilities. Encoder-only models stand out for their training efficiency, requiring less time and fewer computational resources to train while demonstrating strong NLU capabilities with the smallest parameter count. This makes them particularly suitable for tasks that prioritize language understanding over generation. On the other hand, decoder-only models, despite their need for significantly more computational resources and a more significant number of parameters to match the NLU capabilities of their counterparts, excel in generative tasks. Their superior zero-shot generalization ability further underscores their utility in scenarios where generative capacity and adaptability to new tasks without explicit training are crucial. Encoder-decoder models, requiring the most

computational resources, offer a balanced approach. They need fewer parameters than decoder-only models to achieve comparable NLU performance and more parameters than encoder-only models to reach the same level of understanding. This hybrid approach, however, enables them to effectively handle a broader range of tasks, leveraging both strong understanding and generation capabilities.

Looking forward, the evolving landscape of NLP presents numerous opportunities for further research and development. The quest for models that combine the efficiency and understanding capabilities of encoder-only models with the generative prowess and adaptability of decoder-only models continues. Future research could explore more efficient training algorithms, novel architectural innovations, or even entirely new paradigms of model design to reduce computational demands while boosting the models' efficacy over a wider array of tasks.

REFERENCES

A. Radford, K. Narasimhan, T. Salimans, Improving language understanding by generative pre-training, arXiv preprint arXiv:18e3242.3775, (2021).

D. Bahdanau, K. Cho, K., Y. Bengio, *Neural Machine Translation by Jointly Learning to Align and Translate.* Inter. Conf. Learn. Repre., 1-11,(2014)

D., Jacob, et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." arXiv preprint arXiv:1810.04805, (2018).

J. Smith,, & Johnson, A. Comparative Analysis of Encoder-Only, Decoder-Only, and Encoder-Decoder Language Models. J. Natur. Lang. Proc., **15(3)**, 123-136,(2020).

MMLU 5-Shot Leaderboard. Retrieved from https://klu.ai/glossary/mmlu-eval. [Accessed: 15 Mar. 2024].

Q. Zhong, L. Ding, Y. Zhan, Toward Efficient Language Model Pretraining and Downstream Adaptation via Self-Evolution: A Case Study on SuperGLUE. arXiv preprint arXiv:2212.01853, (2022).

SuperGLUE. Retrieved from https://super.gluebenchmark.com/ 2023. [Accessed: 15 Mar. 2024].

T. Wang, A. Roberts, D. Hesslow, What Language Model Architecture and Pretraining Objective Work Best for Zero-Shot Generalization? Adv. Neural Inform.Proc. Sys. (2023)

V., A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, *Attention is all you need.* Adv. Neural Inform.Proc. Sys.5998-6008, (2017).

V., A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, et al. *Attention Is All You Need*.Adv. Neural Inform.Proc. Sys.,5998-6008, 2017.