LLM  
  
**what is LLM?**  
  
GPT is a large language model or an LLM that can generate human-like texts.  
\*GPT or Generative Pre-trained Transformer.  
  
LLM is an instance of something else called a foundation model.  
  
Foundation Model: Pre-trained on large amounts of unlabeled and self-supervised data; meaning the model learns from patterns in the data in a way that produces generalized and adaptable output.  
  
LLM are instances of foundation models applied specifically to text and text-like things(like code).

Large language models are trained on large datasets of text, such as books, articles, and conversations.  
  
Large means these can be trained on gigabytes in size and trained on enormous amounts of text data, petabytes of data.   
  
Ex: Document of 1GB in size; stores 178 million words  
1 PB = 1 million GB  
  
LLMs are also among the biggest models when it comes to parameter count.  
  
A parameter is a value that the model can change independently as it learns, and the more parameters a model has, the more complex it can be.  
  
Ex: GPT-3, pre-trained on a corpus of actually 45 terabytes of data, and it uses 175 billion ML parameters.  
  
How do they work:  
  
**LLM = DATA + ARCHITECTURE + TRAINING**  
  
Data: text data as discussed above  
  
Architecture: This is a neural network and for GPT that is a transformer.  
  
And the transformer architecture enables the model to handle sequences of data like sentences or lines of code.  
  
Transformers are designed to understand the context of each word in a sentence by considering it in relation to every other word.  
  
This allows the model to build a comprehensive understanding of the sentence structure and meaning of the words within it.  
  
**TRAINING:**  
  
And then this architecture is trained on all of this large amount of data.  
  
Now, during training, the model learns to predict the next word in a sentence.  
  
Example: “the sky is..” it starts off with a random guess, “the sky is bug.”   
  
But with each iteration, the model adjusts its internal parameters to reduce the difference between its predictions and actual outcomes.  
  
The model keeps doing this gradually improving its word predictions until it can reliably generate coherent sentences like forget about “bug”, it can figure out its “blue”.  
  
Now, the model can be fine-tuned on a smaller, more specific dataset.  
  
Here the model refines its understanding to be able to perform this specific task more accurately.  
  
Fine-tuning is what allows a general language model to become an expert at a specific task.  
  
OK, so how does this all fit into number 3, business applications?

**Business Applications:**Example 1: Well, for customer service applications, businesses can use LLMs to create intelligent chatbots that can handle a variety of customer queries, freeing up human agents for more complex issues.  
  
Example 2: Another good field, is content creation. That can benefit from LLMs which can help generate articles, emails, social media posts, and even YouTube video scripts.   
  
Example 3: LLMs can even contribute to software development, and they can do that by helping to generate and review code.  
  
As large language models continue to evolve, we’re bound to discover more innovative applications.

**What are transformers?**

Specifically, I used a GPT-3, or a generative pre-trained transformer model. 3 means third generation.  
  
GPT-3 is an auto-regressive model that produces text that looks like it was written by a human.  
  
GPT-3 can write poetry, craft emails and evidently come up with its own jokes.  
  
GPT-3 is just one example of a transformer.

Something that transforms from one sequence into another.   
  
A language translation is a great example.  
  
Ex: why did the banana cross the road translate into French..  
  
Transformer consists of two parts: Encoder and decoder  
  
Encoder works on the input sequence  
  
Decoder Operates on the target output sequence.  
  
Transformers work is through sequence-to-sequence learning, where transformer takes sequence of tokens; like words in a sentence and predicts the next word in the output sequence.  
  
It does this through iterating through encoder layers, so the encoder generates encoding that define which part of the input sequence are relevant to each other and then passes these encodings to the next encoder layer.  
  
The decoder takes all of these encodings and uses their derived context to generate the output sequence.  
  
Transformers are a form of semi-supervised learning  
  
By semi-supervised, we mean that they are pre-trained in an unsupervised manner with a large, unlabeled data set, and then they’re fine-tuned through supervised training to get then to perform better.  
  
what makes transformers different than neural networks and RNN( which takes data in a sequential order) whereas transformers do not process the data in a order because it uses a mechanism called attention mechanism.  
  
This mechanism provides context around items in the input sequence, so rather than starting our translation with the word “why” because it’s at the start of the sentence(why did the banana cross the road translate into French). The transformer attempts to identify the context that bring meaning in each word in the sequence.  
  
This attention mechanism that gives transformers a huge leg up over other algorithms like RNN which must run in sequence.  
  
Transformers run multiple sequences in parallel and this vastly speeds up training times.  
  
Input  
  
Encoder  
  
Decoder  
  
Output  
  
  
  
Transformers examples:  
  
Document summaries-summarizes, writing blog posts  
Translation  
playing chess  
performing image processing that even rivals the capabilities of convolutional neural networks  
  
Transformer is a powerful deep learning model because of the attention mechanism which can be parallelized and it's getting over time.



References:  
  
IBM Technology YouTube Channel