**Introduction to Language Modelling**

[[Shravan Kumar](https://medium.com/@shravankoninti?source=post_page-----8742ebc1edd7--------------------------------)](https://medium.com/@shravankoninti?source=post_page-----8742ebc1edd7--------------------------------)

[Shravan Kumar](https://medium.com/@shravankoninti?source=post_page-----8742ebc1edd7--------------------------------)

·

[Follow](https://medium.com/m/signin?actionUrl=https%3A%2F%2Fmedium.com%2F_%2Fsubscribe%2Fuser%2Fccc45b3ef025&operation=register&redirect=https%3A%2F%2Fmedium.com%2F%40shravankoninti%2Fintroduction-to-language-modelling-8742ebc1edd7&user=Shravan+Kumar&userId=ccc45b3ef025&source=post_page-ccc45b3ef025----8742ebc1edd7---------------------post_header-----------)

7 min read

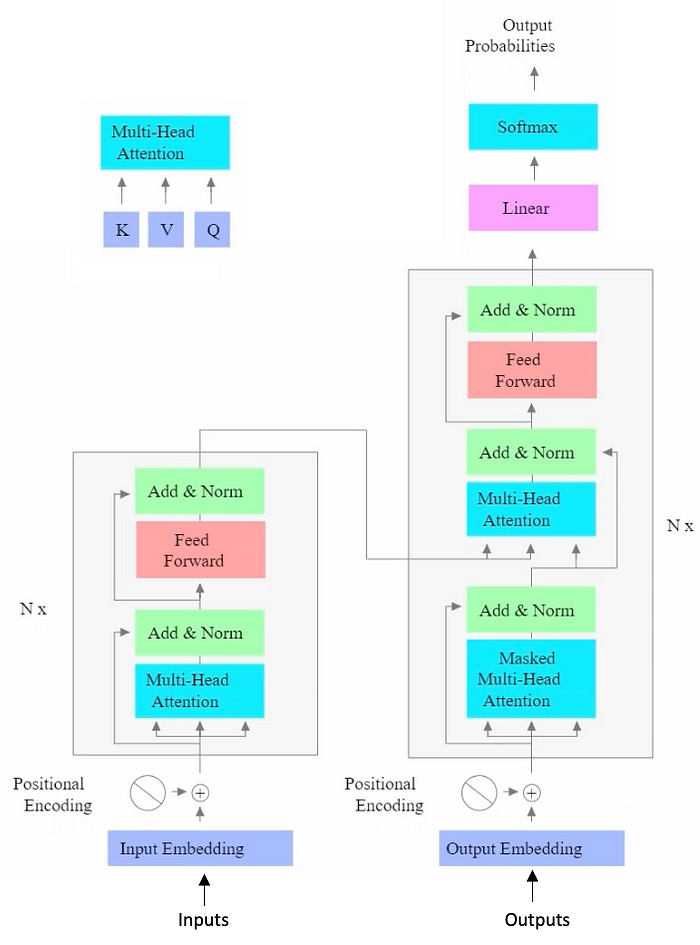
·

Nov 22

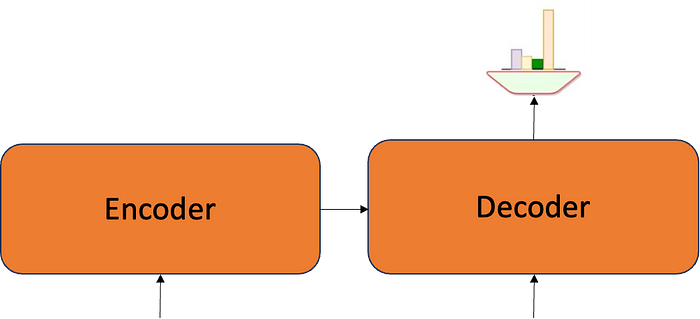
2

In my previous [blog](https://medium.com/@shravankoninti/transformers-attention-is-all-you-need-layer-normalization-1435248866d6) we learned about the components of the transformer architecture in the context of machine translation.

Here is how the transformer block at all granular steps is shown below.



If we narrow down the above flowchart into a zoomed out level and have a bird’s eye view at a high level — we can see the entire block as encoder and decoder block where the encoder takes an input and produces an output and this output is a distribution over the vocabulary.



At an even higher level of abstraction, we can think of it as a black box that receives an iput in the form of text and produces an output which is text in the target language as show below in the case of machine translation.



What if we want to use the trasformer architecture for other NLP tasks?

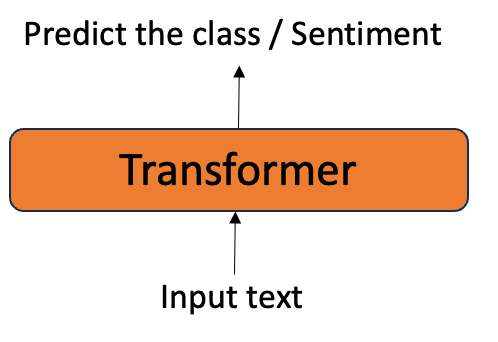
For e.g., Question & Answering



For e.g., Summarization



For e.g., Predict the class or sentiment



All of these can be abstracted into input/ouptut format with the transformer acting as the function which maps the input to output. This function has the parameters ‘theta’ which are the parameters ‘WK’, ‘WV’, ‘WQ’, FFN parameters and then repeated across all the layers. If we want to use transformer architecteure from multiuple NLP tasks then we need datasets for each of the task where we collect the data and labels associated to it.

*We need to train a separate model for each task using dataset specific to the task.*

These labels will differ for all the different tasks — for summarization we should prepare the input text and have summary as label; for question & answering we should prepare the input text , question as inputs and answer as label; for sentiment prediction we should prepare the input text and sentiment as label.

If we train the architecture from scratch ( that is by randomly initializing the parameters) for each task, it takes a long time for convergence. Often, we may not have enough labelled samples for many NLP tasks. Morever, preparing the labelled data is laborious and costly. On the other hand, we have a large amount of unlabelled text easily available on the internet.



*Can we use of such unlabelled data to train a model?*

There are two major questions we need to answer if we are going through training on unlabelled data.

* What will be the training objective in that case?
* Would that be helpful in adapting the model to downstream tasks with minimal fine-tuning (with zero or a few samples)?

**Motivation for understanding Language Modelling**

Assume that we ask question to a lay person based on a statement or some excerpt.

* *“Wow, India has now reached the moon”*

Is this sentence expresssing a positive or a negative sentiment?

* An excerpt from business today *“What sets this mission apart is the pivotal role of artificial intelligence (AI) in guiding the spacecraft during is critical descent to then moon’s surface.”*

Did the lander use AI for soft landing on the moon?

* *He likes to stay*
* *He likes to stray*
* *He likes to sway*

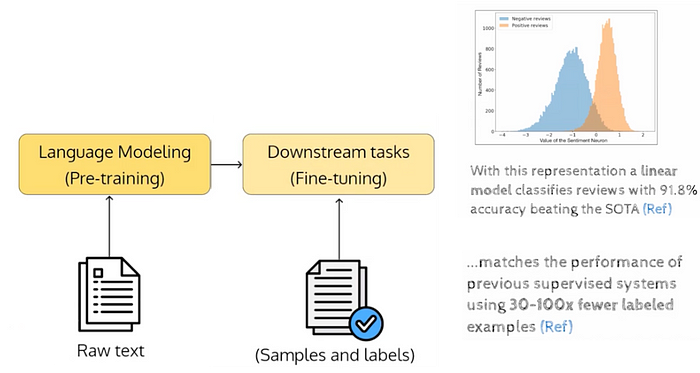
Are these meaningful sentences?

Many of the above tasks are not being trained for, so someone for a lay person can answer these questions based on some intuition, eventhough he/she many not be explicitly trained on any of these tasks. How?

We develop a strong understanding of language through various language based interactions (listening/reading) over our life time without any explicit sueprvision.

*Can a model develop basic understanding of language by getting exposure to a large amount of raw text?****[pre-training]***

*More importantly, after getting exposed to such raw data can it learn to perform well on downstream tasks with minimum supervision?****[Supervised Fine-tuning]***



By doing the 2 stage approach where we do pre-training and fine-tuning we are able to get better metrics for a certain specific task for e.g., here it is sentiment analysis. This is the motivation for understanding the language modelling and pre-training.

Now let us see the connection between pre-training and language modeling? First let us focus on what is language modeling?

let v be a vocabulary of language (i.e., collection of all unique words in the language) we can think of a sentence as a sequence X1, X2, X3,…..Xn where Xi belongs to v

for example if v = {an, apple, ate, I}, some possible sentences (not necessarily grammatically correct) are

a. An apple ate I

b. I ate an apple

c. I ate apple

d. an apple

e. …………….

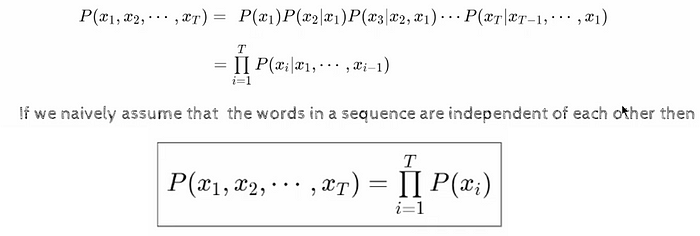
f. ………………

Intuitively some of these sentences are more probable than others.

What do we mean by that? We mean that given a very very large corpus, we expect some of these sentences to appear more frequently than others (hence more probable) we are now looking for a function which takes a sequence as input and assigns a probability to each sequence

f : (X1, X2, ……. Xn) -> [0,1]

such function is called as language model.



if we are considering the independent factor — we will have P(x2/x1) = ~P(x2) similarly for P(x3/x2,x1) = ~P(x3).

**How do we enable a model to understand language?**

Teach it the task of predicting the next token in a sequence. Why is this an important idea? Because we have tons of sequences avaialble on the web which you can use as training data. The way we do it is — we take all k-1 input tokens as our input and k is the output corresponding to the k input words. Let’s take an example of web data from wiki



Assuming we are predicting the masked words from above — if we are given 4 words then we will predict the 5th word and produce the probability for this word with a distribution over vocabulary as P(x5 | x1, x2, x3, x4). So here by using the internet data we are kind of converting a problem into supervised problem by creating the input data and output data and this transition of creating is free where we are getting the label for free.

Let’s take a sentence from the above — “He is also supporting IIT Madras’s AI4Bharat.” we can create the data and ask the model to predict for

* He is — — — (x -> x1,x2)
* He is also — — — (x -> x1,x2,x3)
* He is also supporting — — — — (x->x1,x2,x3,x4)

This way we get large amounts of training data for training this large transformer architectures and smaller amount of training required for each of the tasks — so we got large amount of training data we got which we call as pre-training data.

***Roughly speaking this tasks of predicting the next token in a sequence is called as language modelling.***

However, we know that the words in a sentence are not independent but depend on the previous words.

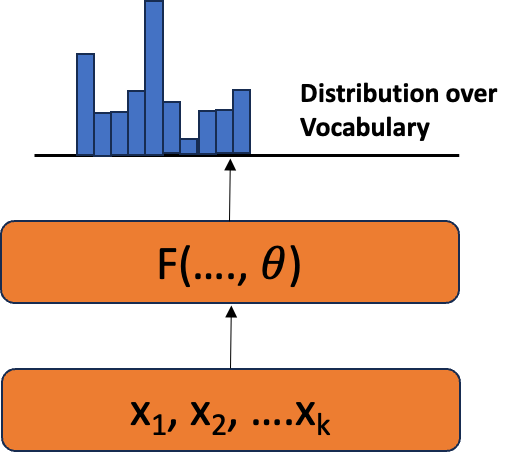
a. I enjoyed reading a book

b. I enjoyed reading a thermometer

The presence of “enjoyed” makes the word “book” more likely than “thermometer” . Hence the naive assumption does not make sense



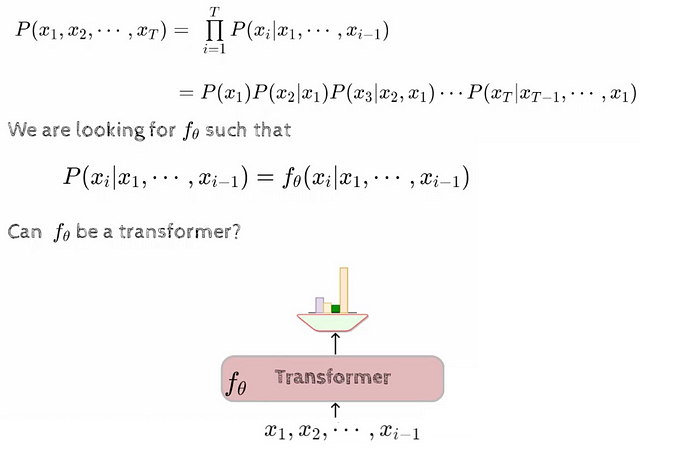
All the probabilities we are finding above for each of the ‘i’ is finding about distribution over the vocabulary. All of these distributions are conditional based distributions where the condition is changing.

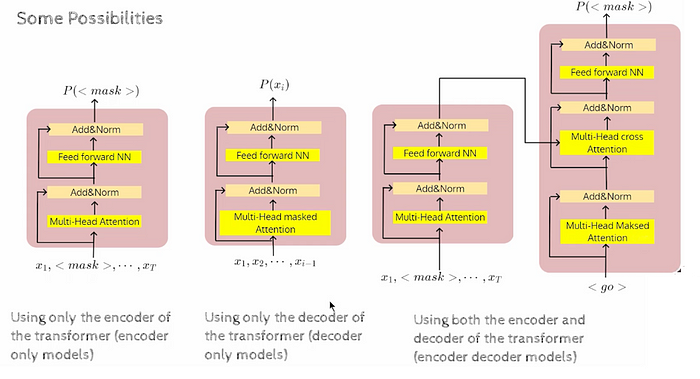


**How do we estimate these conditional probabilities?**

***One solution:***Use autoregressive models where the conditional probabilities are given by parameterized functions with a fixed number of parameters (like transformers).

**Causal Language Modelling (CLM)**





***Examples:***

* Encoder only model is — BERT
* Decoder only model is — GPT
* Encoder and Decoder model is — T5

In the upcoming blogs we will focus on each of the above models and see how they will be applicable to LLM application for different tasks.

**Please do clap**👏**or comment if you find it helpful ❤️🙏**