In [1]:
```
%run Latex_macros.ipynb
```

Which Pre-Trained Transformer Model to use for Fine-Tuning ?

In choosing a pre-trained Transformer model, there are many choices

- Architecture type: Encoder, Decoder, Encoder/Decoder
- Size
- Training set

We offer some guidelines, assuming our goal is to Fine-Tune the Pre-Trained model to a new Target task.

Architecture

Encoder

The salient feature of the Encoder is the *unmasked* Attention.

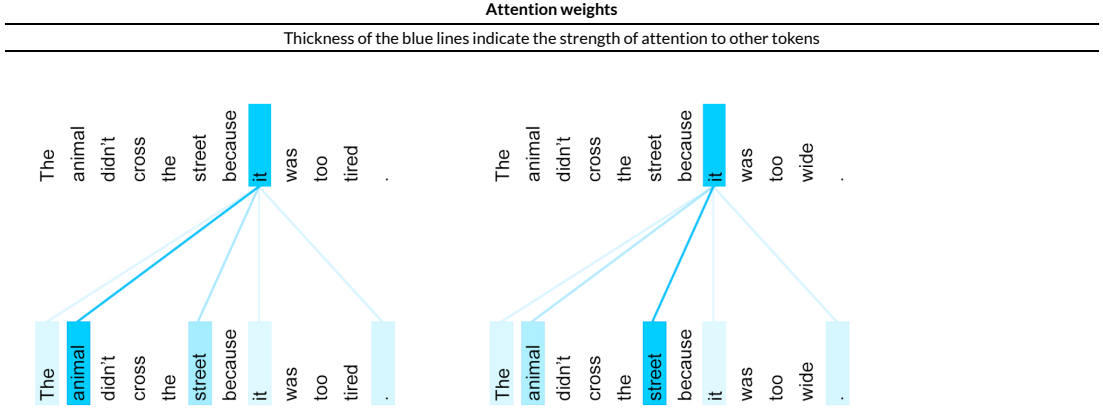
It converts the sequence $\mathbf{x}_{(1:\bar{T})}$ to the sequence $\bar{\mathbf{h}}_{(1:\bar{T})}$

- transforming the "raw" sequence (each position in isolation) $\mathbf{x}_{(1:\bar{T})}$
- into the "processed" sequence (each position informed by *all* the other positions) $\bar{\mathbf{h}}_{(1:\bar{T})}$

That is: the output positions are *context-sensitive* representations.

For example: the position corresponding to the word "it"

- in each of the two sentences
- is ambiguous in the raw sequence
- is unambiguous in the processed sequence
    - it has been informed by the tokens preceding and following its position



Picture from: https://1.bp.blogspot.com/-AVGK0ApREtk/WaiAuzddKVI/AAAAAAAAB_A/WPV5ropBU-cxrcMpqJBFHg73K9NX4vywwCLcBGAs/s1600/image2.png

Thus, an Encoder-only model is useful for

- adding contextual information to the raw input
- as a source of a sequence that can be "pooled" into a single positions
    - that is a fixed length summary of the entire sequence
    - and thus can be feed to simpler NN layers (e.g., a Fully Connected layer acting as a Classifier)

Decoder

The salient features of the Decoder are its *Autogregressive behavior* and *Causal masking*

This makes it appropriate for

- generative tasks
- where the output positions are generated one position at a time
- and each position is informed *only* by preceding (i.e., previously generated) positions

Encoder/Decoder

Most useful for Sequence to Sequence tasks where

- the input sequence may be fully accessed at all positions (**no** masking)
- the output sequence is generated autoregressively
    - causal attention to the previously generated output
    - with full access to the "processed" input (i.e., output of the Decoder)

Examples include

- language translation
- question answering

Some of these tasks might be able to be handled by a Decoder-only architecture too

- where the "raw" input $\mathbf{x}$
- is also included as part of the output
- but all of $\mathbf{x}$ precedes the first true Decoder output

$$\mathbf{x}_1, \ldots, \mathbf{x}_{\bar{T}}, \hat{\mathbf{y}}_1, \ldots$$

Since all of $\mathbf{x}$ precedes any $\hat{\mathbf{y}}$

- causal attention still allows full access to the entire length of $\mathbf{x}$ when generating any position in $\hat{\mathbf{y}}$
    - similar to Cross Attention to the Encoder output in the Encoder/Decoder

Size

Many models come in different size variations: small, medium, large.

Obviously the memory size and computing ability of the processor you are using may influenced your choice of model.

Training set

Fine-tuning a model that has been pre-trained on

- a larger training set
- that is more similar to the Target task examples

would seem to be most beneficial for Transfer Learning.