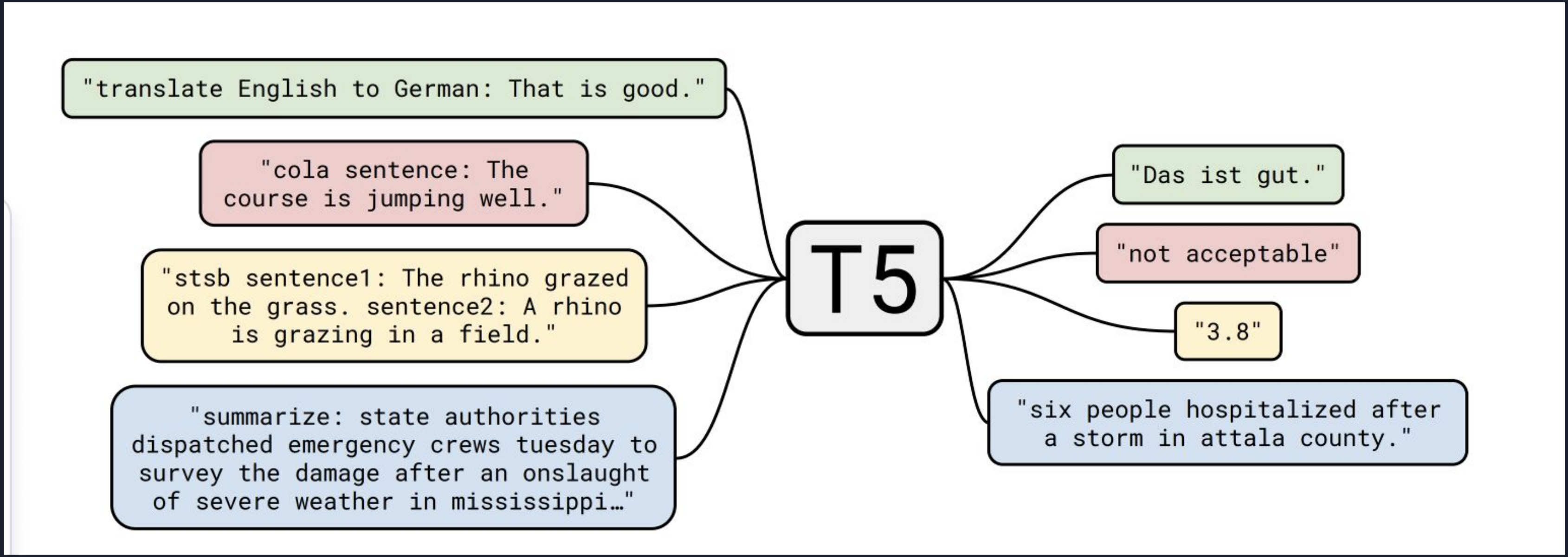
T5 **-** “**T**ext-**t**o-**T**ext **T**ransfer **T**ransformer

What is T5?

Text-to-Text Transfer Transformer (T5)

- A model that unifies **all text-based language problems into a text-to-text format**

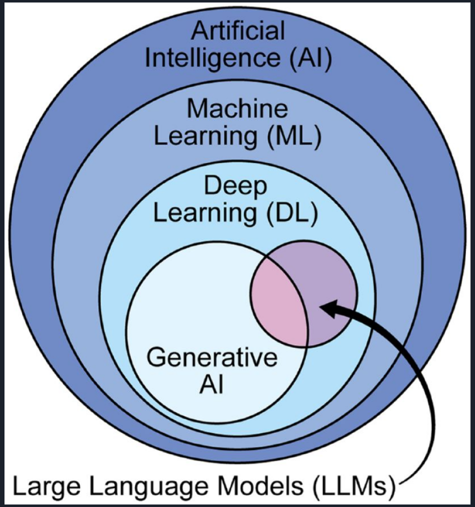


What is an LLM?

Neural networks to understand, generate and respond to human like text.

Parameters - Adjustable weights in the network that are optimised during training to predict the next word

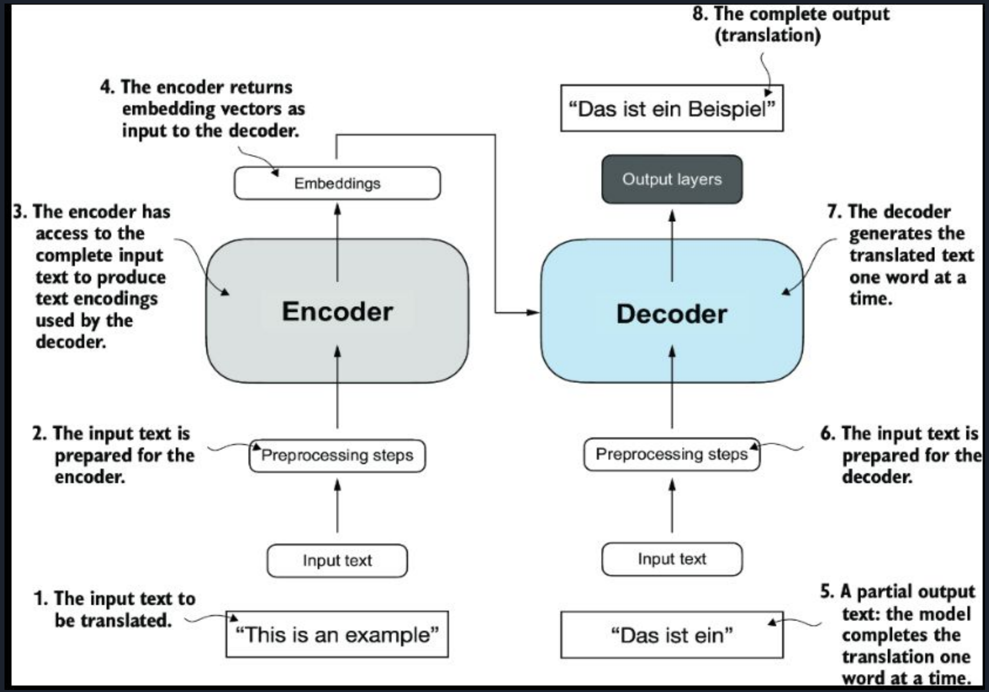
Transformers - Used as LLMs architecture

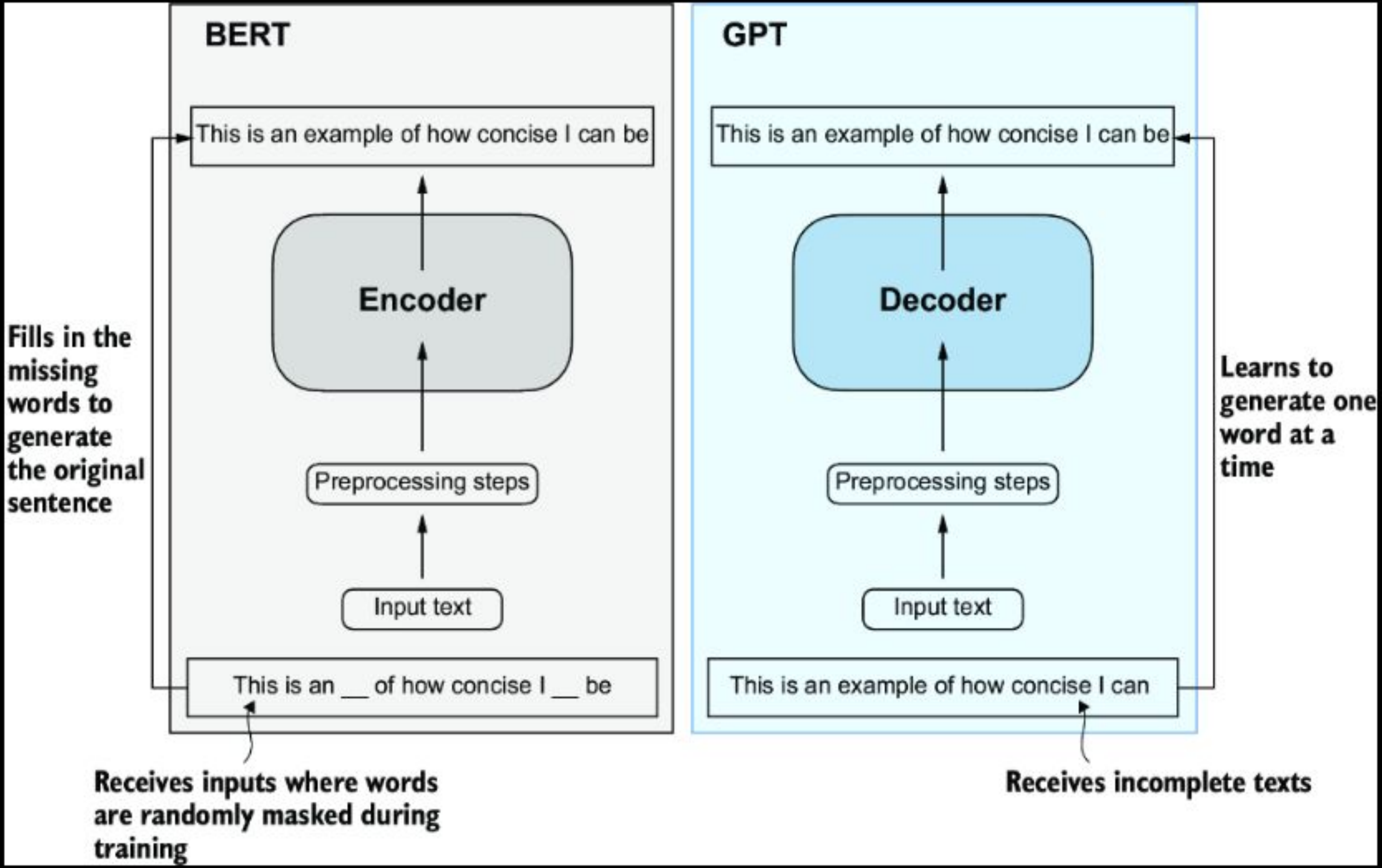


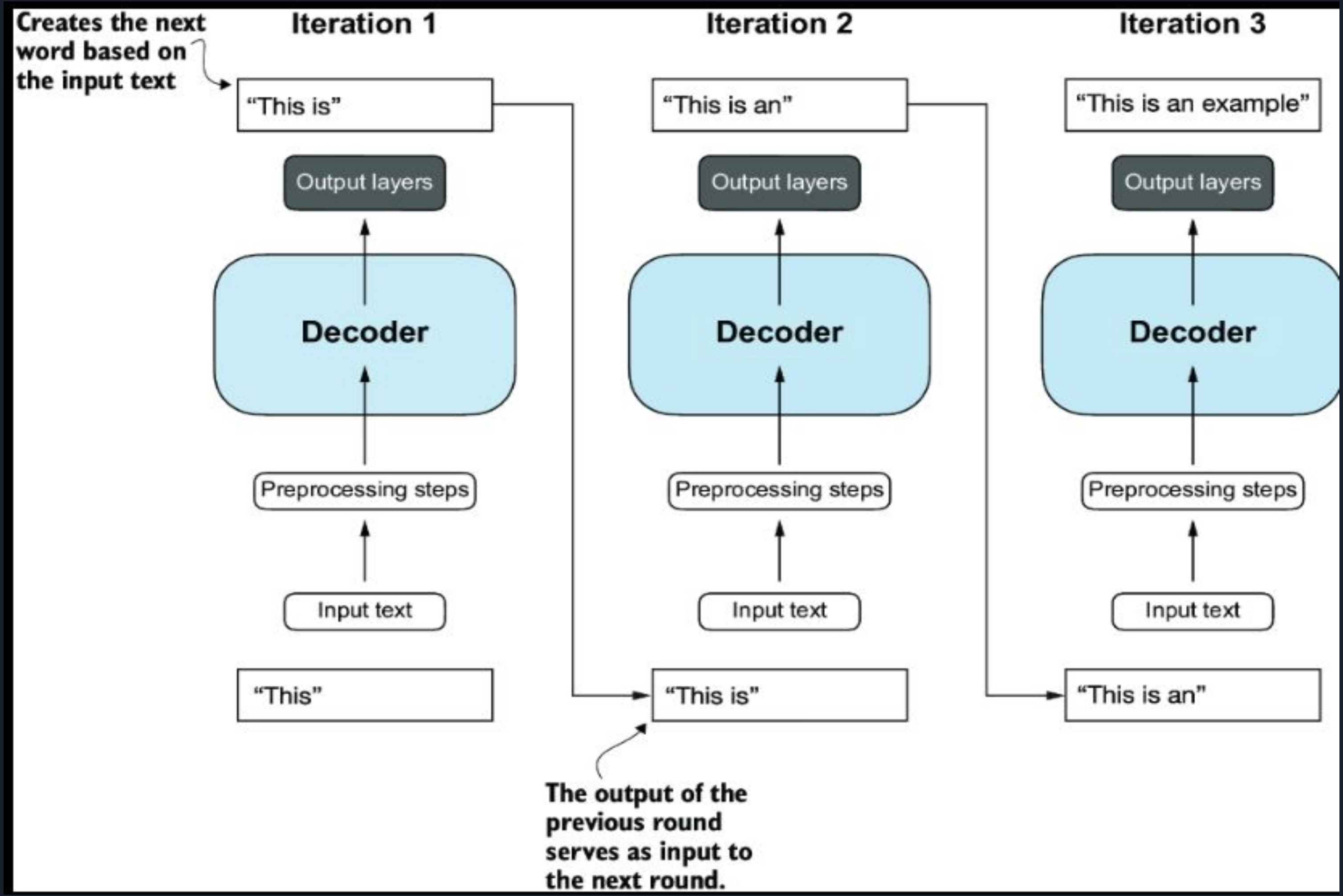
Transformer Architecture

The Transformer architecture is used due to its **proven effectiveness and suitability for the unified text-to-text framework.**

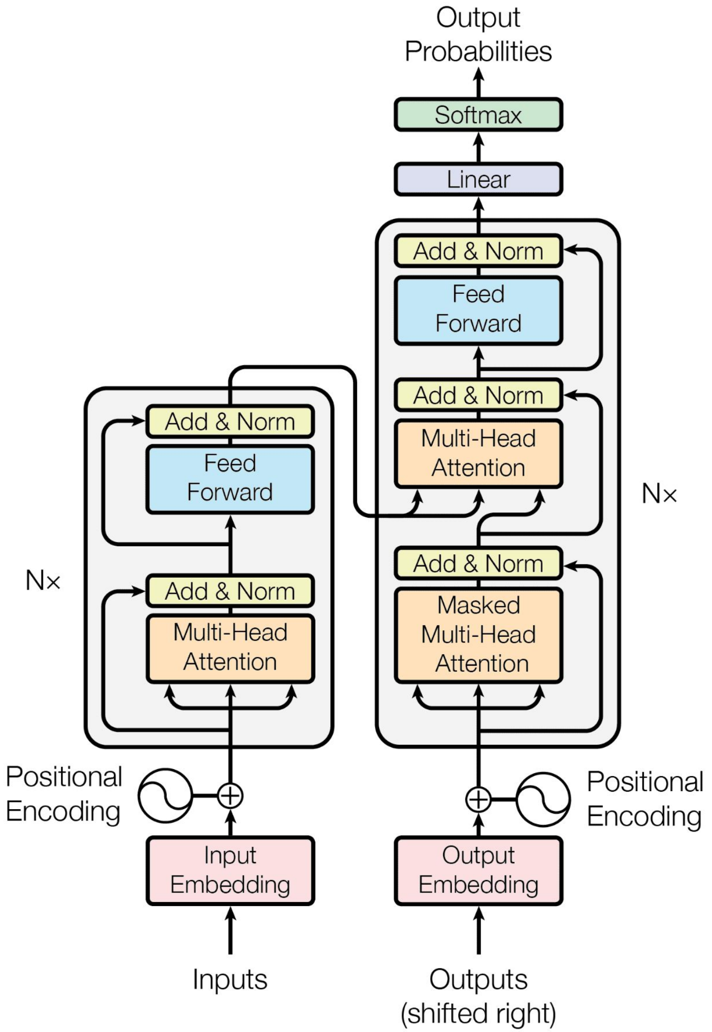
Text-to-text Framework= Input and output are both texts.




BERT
This is an example of how concise I can be
Encoder
Preprocessing steps
Input text
This is an __ of how concise I __ be
Fills in the missing words to generate the original sentence
Receives inputs where words are randomly masked during training
GPT
This is an example of how concise I can be
Decoder
Preprocessing steps
Input text
This is an example of how concise I can
Learns to generate one word at a time
Receives incomplete texts


Creates the next word based on the input text
Iteration 1
Iteration 2
Iteration 3
"This is"
"This is an"
"This is an example"
Output layers
Output layers
Output layers
Decoder
Decoder
Decoder
Preprocessing steps
Preprocessing steps
Preprocessing steps
Input text
Input text
Input text
"This"
"This is"
"This is an"
The output of the previous round serves as input to the next round.

TRANSFORMER ARCHITECTURE



Given:

- Query matrix: $Q \in \mathbb{R}^{L_q \times d_k}$
- Key matrix: $K \in \mathbb{R}^{L_k \times d_k}$
- Value matrix: $V \in \mathbb{R}^{L_k \times d_v}$

The **scaled dot-product attention** is computed as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V$$

Where:

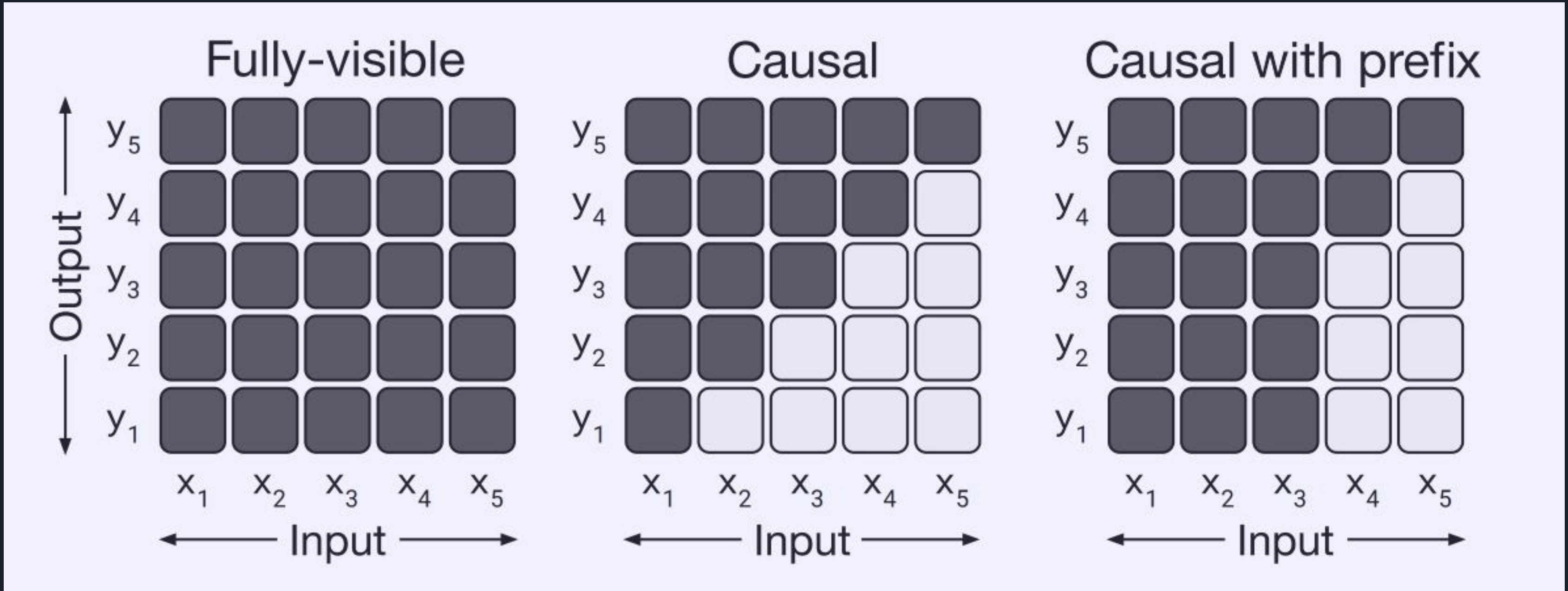
- Q, K, V are learned linear projections from the input embeddings.
- d_k is the dimensionality of keys/queries (typically $d_k = d_{model}/h$ where h … ne number of heads).

T5 uses **Pre-LayerNorm**, meaning LayerNorm is applied **before** each sub-layer (attention or FFN), unlike the original Transformer which applied it after.

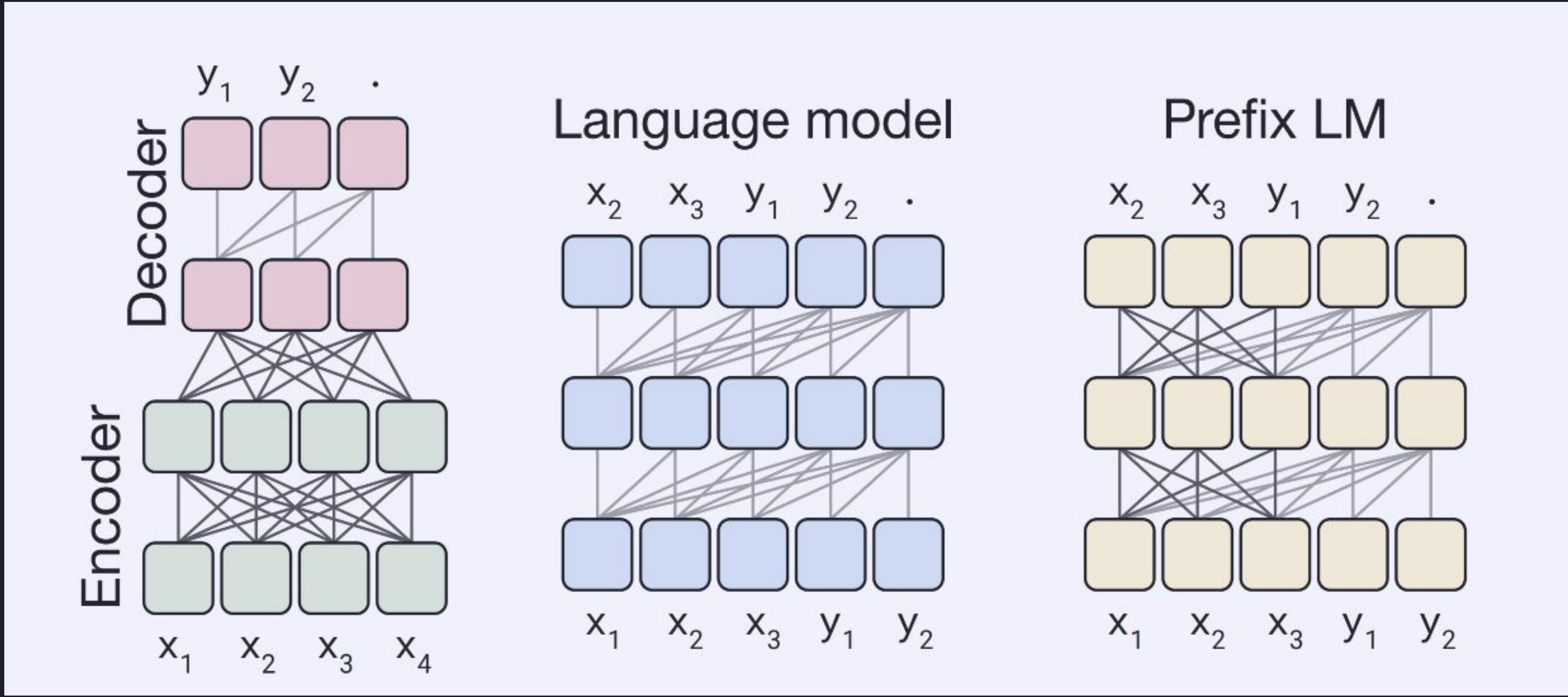
Self Attention Mechanism

Each entry of the output sequence is produced by computing a weighted average of entries of the input sequence. (show steps)

Attention Mask Patterns



Transformer Architecture Varients



Comparison of different model structures

Assumption = an L + L-layer encoder-decoder model has the same number of parameters as an 2L-layer language mode. M refers to the number of FLOPs required for an L + L-layer encoder-decoder model or L-layer decoder-only model to process a given input-target pair.

As an unsupervised objective, we will consider both a basic language modeling objective as well as our baseline denoising objective

	Architecture	Objective	Params	Cost
★	Encoder-decoder	Denoising	$2P$	M
	Enc-dec, shared	Denoising	P	M
	Enc-dec, 6 layers	Denoising	P	$M/2$
	Language model	Denoising	P	M
	Prefix LM	Denoising	P	M
	Encoder-decoder	LM	$2P$	M
	Enc-dec, shared	LM	P	M
	Enc-dec, 6 layers	LM	P	$M/2$
	Language model	LM	P	M
	Prefix LM	LM	P	M

Unsupervised Objectives

For pretraining models on unsupervised data to acquire a general purpose knowledge for NLP tasks.

Three Objectives were tested:-

1)Prefix Language Modeling

2)BERT- style Masked Language Modelling

3)Deshuffling

BERT OBJECTIVE

1) MASS Objective
2) Avoid Predicting Entire Uncorrupted Text:

    2a) Baseline Objective (Replace Corrupted Span)

        sent=The quick brown fox jumps over the lazy dog.

        Corrupted =The <x> over the <y>.

        Target O/P=<x> quick brown fox jumps <y> lazy dog <z>

    2b) Dropping Corrupted Tokens

Baseline works the best

Varying Corruption Rate - It had a limited effect on models performance (10,15,25,50%). Stick to 15%

BASELINE OBJECTIVE

Problem with I.I.D. Corruption

Denoising Objective had to make independent and identical decisions (I.I.D.) for each token whether to corrupt it or not.

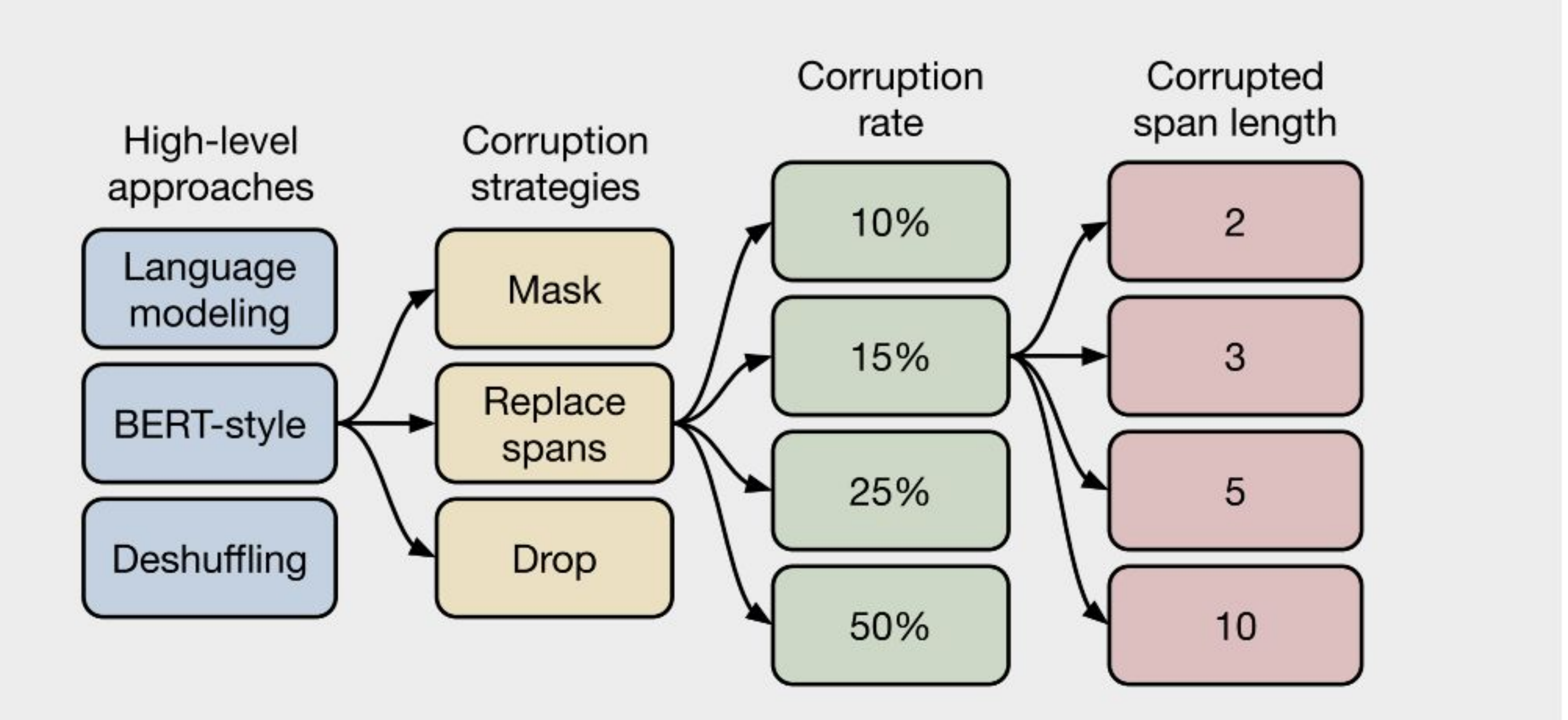
With I.I.D. not a lot of corrupted tokens appear together

SOLUTION

Specifically corrupts spans of tokens and replaces them with 1 token so sequence is overall shortened

Span Length of 3 outperformed the other lengths

Exploration of Unsupervised Objectives



List of Unlabeled Datasets

1. C4
2. Unfiltered C4
3. RealNews-like
4. WebText-like
5. Wikipedia
6. Wikipedia + Toronto Books Corpus

Vocabulary

- SentencePiece is used to encode text as wordPiece tokens.

- 32,000 wordpieces.

- trained the SentencePiece model on a mixture of 10 parts of English C4 data with 1 part each of data classified as German, French or Romanian.

SentencePiece - tool or algorithm used to encode text. It is a language-independent sub-word tokenizer and detokenizer for neural text processing.

Fine Tuning Methods

Fine tuning all of the model's parameters will lead to suboptimal results- overfitting on low resource tasks and underfitting on high resource ones

Methods:-

1) Adapter Layers
2) Gradual Unfreezing

Updating all parameters during fine tuning outperformed these methods.

Fine Tuning all the parameters

1) A standard Maximum Likelihood Objective - with teacher forcing and cross entropy loss
2) Optimizer= AdaFactor
3) Learning Rate Scheduler= Constant rate of 0.001
4) Training steps = 2^18 (262144) steps
5) Batch size= 128 sequences of length 512, so 2^16 tokens per batch
6) Checkpoint= every 5000 steps
7) Decoding at Test time= default = greedy search but for long sequences - beam search

Multitask Learning

- Trains a angle model on many tasks for performance, different checkpoints are used for different tasks
- Uses a mixed dataset

Mixing strategies:-

1. Examples Proportional Mixing - If one tasks dataset is larger than the others it'll dominate the rest. Therefore Artificial Limit - 'K' is imposed
2. Temperature Scaling Mixture - Adjust the temperature of mixing rates to mitigate the huge disparity between dataset sizes
3. Equal Mixing - each dataset have equal probability- suboptimal as it overfits on low resources and underfits on high resources

Therefore MTL underperforms in comparison to Pretraining and the finetuning

Training strategy	GLUE	CNNDM	SQuAD	SGLUE	EnDe	EnFr	EnRo
★ Unsupervised pre-training + fine-tuning	**83.28**	**19.24**	**80.88**	**71.36**	**26.98**	39.82	27.65
Multi-task training	81.42	**19.24**	79.78	67.30	25.21	36.30	27.76
Multi-task pre-training + fine-tuning	**83.11**	**19.12**	**80.26**	**71.03**	**27.08**	39.80	**28.07**
Leave-one-out multi-task training	81.98	19.05	79.97	**71.68**	**26.93**	39.79	**27.87**
Supervised multi-task pre-training	79.93	18.96	77.38	65.36	26.81	**40.13**	**28.04**

Table 12: Comparison of unsupervised pre-training, multi-task learning, and various forms of multi-task pre-training.

Scaling results

Scaling strategy	GLUE	CNNDM	SQuAD	SGLUE	EnDe	EnFr	EnRo
★ Baseline	83.28	19.24	80.88	71.36	26.98	39.82	27.65
1× size, 4× training steps	85.33	19.33	82.45	74.72	27.08	40.66	27.93
1× size, 4× batch size	84.60	19.42	82.52	74.64	27.07	40.60	27.84
2× size, 2× training steps	**86.18**	19.66	**84.18**	77.18	27.52	**41.03**	28.19
4× size, 1× training steps	**85.91**	19.73	**83.86**	**78.04**	27.47	40.71	28.10
4× ensembled	84.77	**20.10**	83.09	71.74	**28.05**	40.53	**28.57**
4× ensembled, fine-tune only	84.05	19.57	82.36	71.55	27.55	40.22	28.09

- Increasing pre training steps increases the number of pre training data
- Increasing batch size by 4 lets the model see 4 times more data and increases training speed
- Increasing the training time and increasing the model size can be complementary means of improving performance

T5 Different Models

Model	Layers (Encoder / Decoder)	dmodel	FFN Dim (dff)	Attention Heads	Parameters (approx)
T5-Small	6 / 6	512	2048	8	~60 million
T5-Base	12 / 12	768	3072	12	~220 million
T5-Large	24 / 24	1024	4096	16	~770 million
T5-3B	24 / 24	1024	16384	32	~3 billion
T5-11B	24 / 24	1024	16384	64	~11 billion