How do transformers work?

Daniel Hsu (Columbia) and Ankur Moitra (MIT)

Simons Bootcamp, September 4^{th}

PLAN

- Some history and background
- Introduce the mechanics of transformers
- Musings on how to think about them conceptually

LANGUAGE MODELS

First attempt to model natural language by Shannon ca. 1950

Prediction and Entropy of Printed English

By C. E. SHANNON

(*ManuscriptReceived Sept. 15, 1950*)

A new method of estimating the entropy and redundancy of a language is described. This method exploits the knowledge of the language statistics possessed by those who speak the language, and depends on experimental results in prediction of the next letter when the preceding text is known. Results of experiments in prediction are given, and some properties of an ideal predictor are developed.

What is the entropy of natural language? Can it be compressed?

LANGUAGE MODELS

First attempt to model natural language by Shannon ca. 1950

How well do humans predict the next word?	**vs**	How well do n-gram models predict the next word?

LANGUAGE MODELS

First attempt to model natural language by Shannon ca. 1950

How well do humans predict the next word?

VS

How well do n-gram models predict the next word?

Definition: An n-gram model is

`The quick brown fox jumped over the lazy dog`

Conditional probability of next word, given previous two words

LANGUAGE MODELS

Are n-grams a good generative model for natural text?

Absolutely not!

LANGUAGE MODELS

Are n-grams a good generative model for natural text?

Absolutely not!

- Generates gibberish or uninteresting sentences

LANGUAGE MODELS

Are n-grams a good generative model for natural text?

Absolutely not!

- Generates gibberish or uninteresting sentences
- For (say) $n = 4$, can't estimate the V^5 parameters

LANGUAGE MODELS

Are n-grams a good generative model for natural text?

Absolutely not!

− Generates gibberish or uninteresting sentences

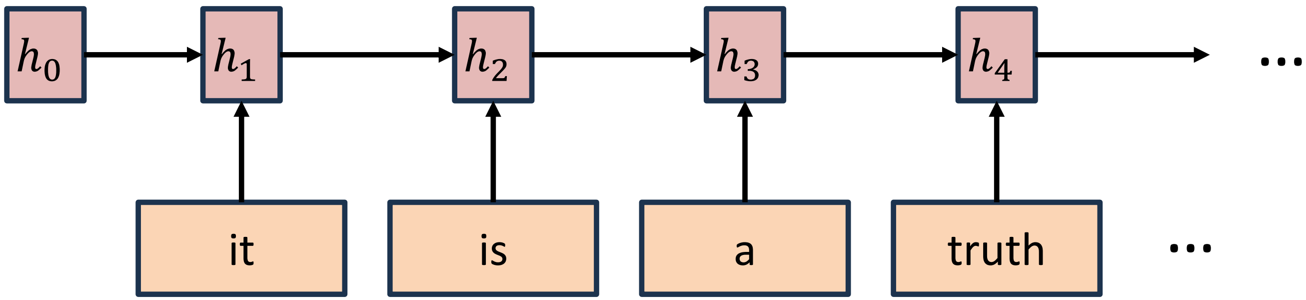
− For (say) n = 4, can't estimate the V^5 parameters

Many workarounds, e.g.

+ Clustering words, e.g. `{big, large, …}`

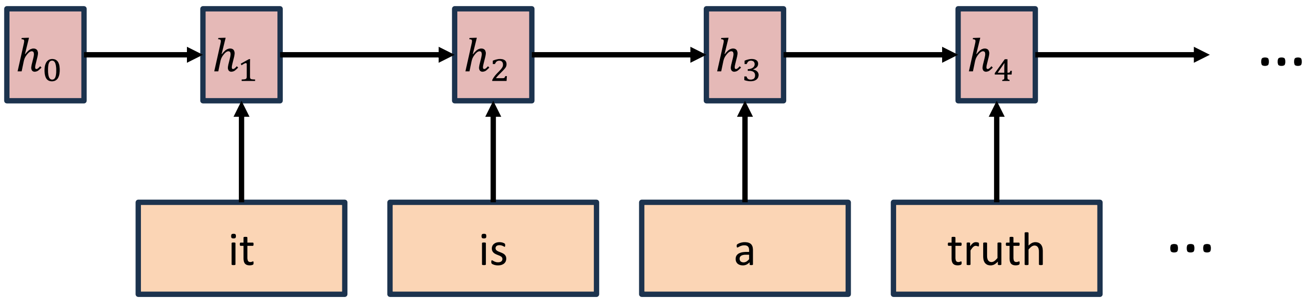
ENTER DEEP LEARNING

Depth and overparameterization makes everything better

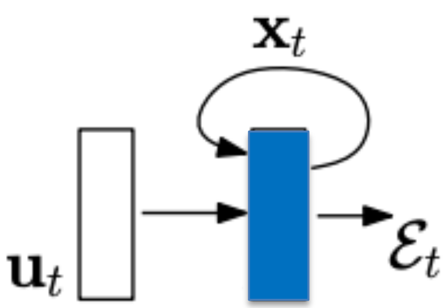


ENTER DEEP LEARNING

Depth and overparameterization makes everything better



Process tokens sequentially, accumulate information as you go



feed forward
neural network

ENTER DEEP LEARNING

But recurrent neural networks are difficult to train

ENTER DEEP LEARNING

But recurrent neural networks are difficult to train

Conceptually, we know language has long-range dependencies

e.g.
```
I grew up in France, where I spent
most of my summers. I speak fluent
                ______
```

ENTER DEEP LEARNING

But recurrent neural networks are difficult to train

Conceptually, we know language has long-range dependencies

e.g. `I grew up in France, where I spent most of my summers. I speak fluent ______`

In many models, past information gets overshadowed by what's more recent

TRANSFORMERS

In 2017, a major breakthrough enabling LLMs

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

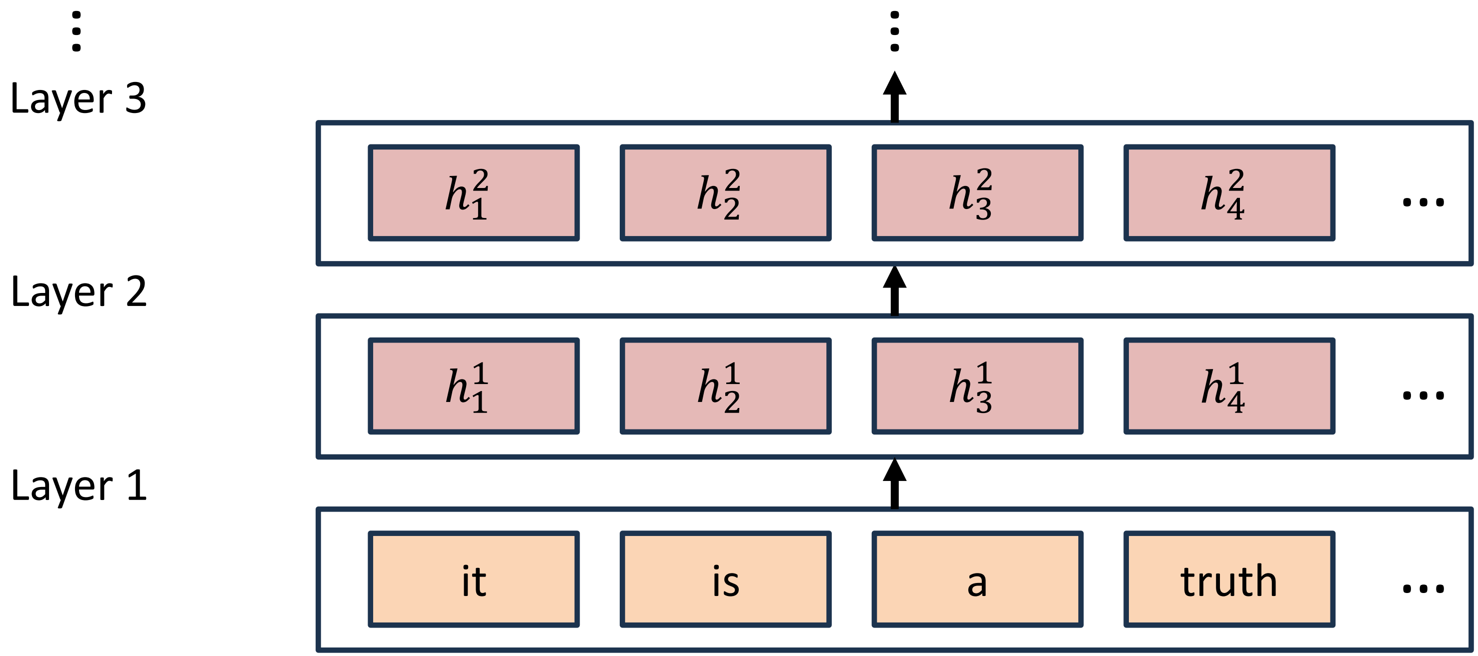
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

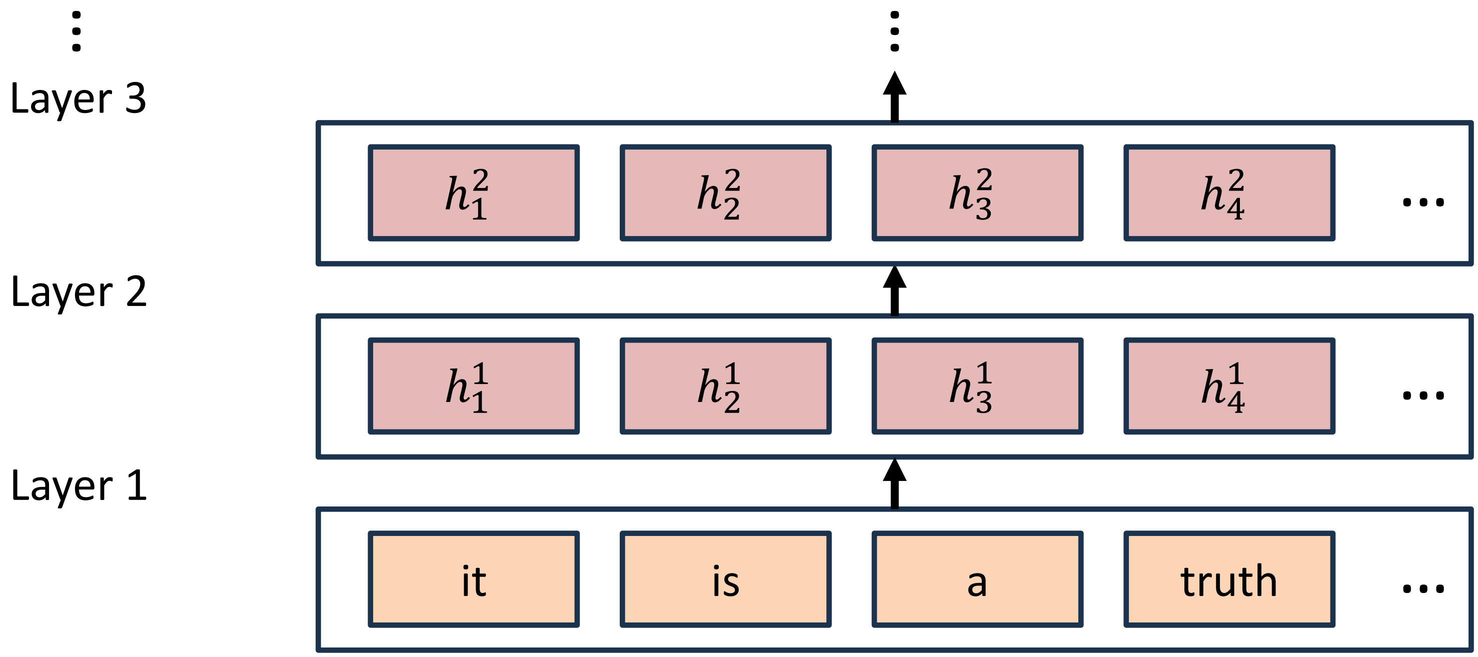
TRANSFORMERS

Transforms sequence of N tokens to sequence of N vectors by composing several sequence-to-sequence maps



TRANSFORMERS

Transforms sequence of N tokens to sequence of N vectors by composing several sequence-to-sequence maps



What computation is done in each layer?

TRANSFORMERS

Our exposition will follow

Formal Algorithms for Transformers

Mary Phuong[1] and Marcus Hutter[1]
[1]DeepMind

This document aims to be a self-contained, mathematically precise overview of transformer architectures and algorithms (*not* results). It covers what transformers are, how they are trained, what they are used for, their key architectural components, and a preview of the most prominent models. The reader is assumed to be familiar with basic ML terminology and simpler neural network architectures such as MLPs.

How do transformers work, through pseudocode?

TRANSFORMERS

Our exposition will follow

Formal Algorithms for Transformers

Mary Phuong[1] and Marcus Hutter[1]
[1]DeepMind

This document aims to be a self-contained, mathematically precise overview of transformer architectures and algorithms (*not* results). It covers what transformers are, how they are trained, what they are used for, their key architectural components, and a preview of the most prominent models. The reader is assumed to be familiar with basic ML terminology and simpler neural network architectures such as MLPs.

How do transformers work, through pseudocode?

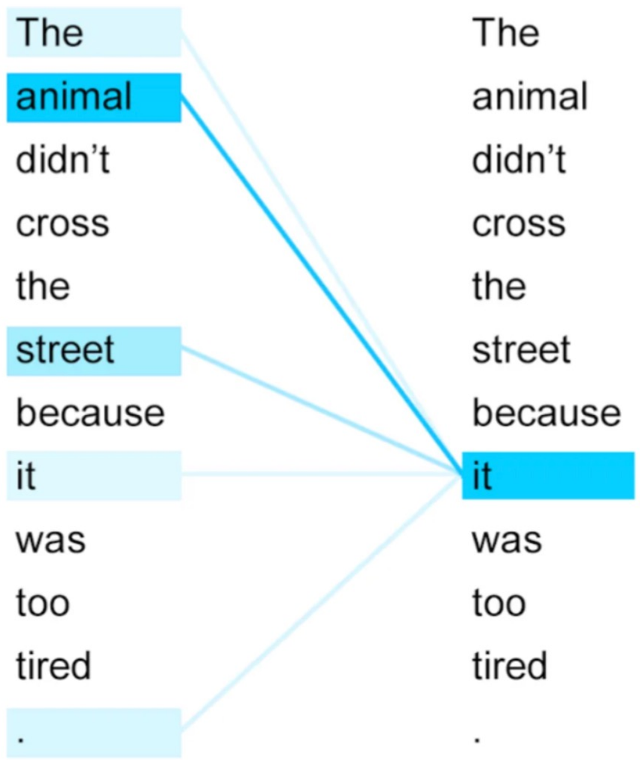
Also Jay Alammar's excellent *Illustrated Transformer*

ATTENTION

Where should you look in a sentence for purposes of translation?

ATTENTION

Where should you look in a sentence for purposes of translation?



e.g. need to decide if “it” is masculine or feminine

ASIDE: WORD EMBEDDINGS

Many ways to find a mapping that captures semantic meaning

word $\longrightarrow$ **vector**

ASIDE: WORD EMBEDDINGS

Many ways to find a mapping that captures semantic meaning

word $\longrightarrow$ **vector**

E.g. can use it to solve analogies like

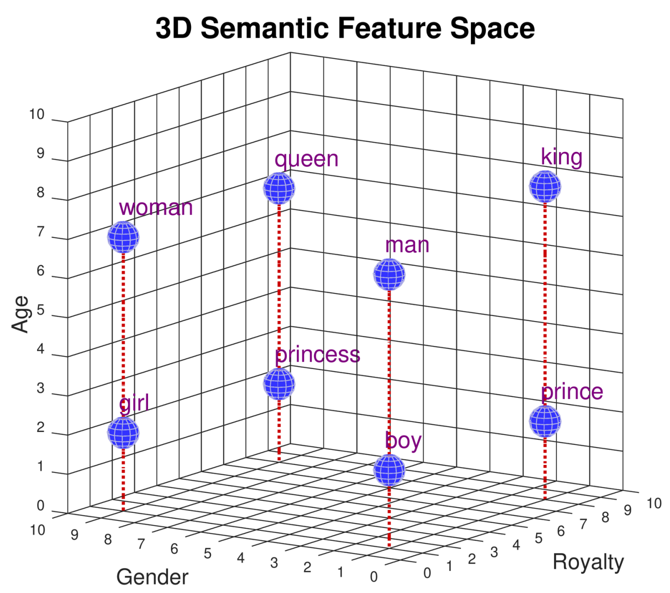
`man:woman::king:____`

ASIDE: WORD EMBEDDINGS

Many ways to find a mapping that captures semantic meaning

word ⟶ **vector**

E.g. can use it to solve analogies like



Credit: Dave Touretzky

ASIDE: WORD EMBEDDINGS

Many ways to find a mapping that captures semantic meaning

word $\longrightarrow$ **vector**

E.g. can use it to solve analogies like

`USA:burger::Canada:____`

ATTENTION

How is attention implemented?

$$\begin{bmatrix}\text{word embedding}\end{bmatrix}\begin{bmatrix}\text{query}\\ \text{matrix}\end{bmatrix} = \begin{bmatrix}\text{query vector}\end{bmatrix}$$

ATTENTION

How is attention implemented?

$$\begin{bmatrix}\text{word embedding}\end{bmatrix}\begin{bmatrix}\text{query}\\ \text{matrix}\end{bmatrix} = \begin{bmatrix}\text{query vector}\end{bmatrix}$$

$$\begin{bmatrix}\text{word embedding}\end{bmatrix}\begin{bmatrix}\text{key}\\ \text{matrix}\end{bmatrix} = \begin{bmatrix}\text{key vector}\end{bmatrix}$$

ATTENTION

How is attention implemented?

$$\begin{bmatrix}\text{word embedding}\end{bmatrix}\begin{bmatrix}\text{query}\\\text{matrix}\end{bmatrix} = \begin{bmatrix}\text{query vector}\end{bmatrix}$$

$$\begin{bmatrix}\text{word embedding}\end{bmatrix}\begin{bmatrix}\text{key}\\\text{matrix}\end{bmatrix} = \begin{bmatrix}\text{key vector}\end{bmatrix}$$

$$\begin{bmatrix}\text{word embedding}\end{bmatrix}\begin{bmatrix}\text{value}\\\text{matrix}\end{bmatrix} = \begin{bmatrix}\text{value vector}\end{bmatrix}$$

SINGLE QUERY ATTENTION

Given a query q, and keys and values for previous words compute

$$v = \sum_t \alpha_t v_t$$

weighted average of other values

SINGLE QUERY ATTENTION

Given a query q, and keys and values for previous words compute

$$v = \sum_t \alpha_t v_t \quad \text{where} \quad \alpha_t = \frac{\exp(q^T k_t/\sqrt{d})}{\sum_u \exp(q^T k_u/\sqrt{d})}$$

weighted average of other values

weights are given by softmax

SINGLE QUERY ATTENTION

Given a query q, and keys and values for previous words compute

$$v = \sum_t \alpha_t v_t \quad \text{where} \quad \alpha_t = \frac{\exp(q^T k_t / \sqrt{d})}{\sum_u \exp(q^T k_u / \sqrt{d})}$$

weighted average of other values

weights are given by softmax

\+ Can look far back for the relevant information

SINGLE QUERY ATTENTION

Given a query q, and keys and values for previous words compute

$$v = \sum_t \alpha_t v_t \quad \text{where} \quad \alpha_t = \frac{\exp(q^T k_t / \sqrt{d})}{\sum_u \exp(q^T k_u / \sqrt{d})}$$

weighted average of other values

weights are given by softmax

+ Can look far back for the relevant information

+ Similarity is computed in a natural concept space

WHAT IS ATTENTION TRYING TO DO?

Consider the sentence

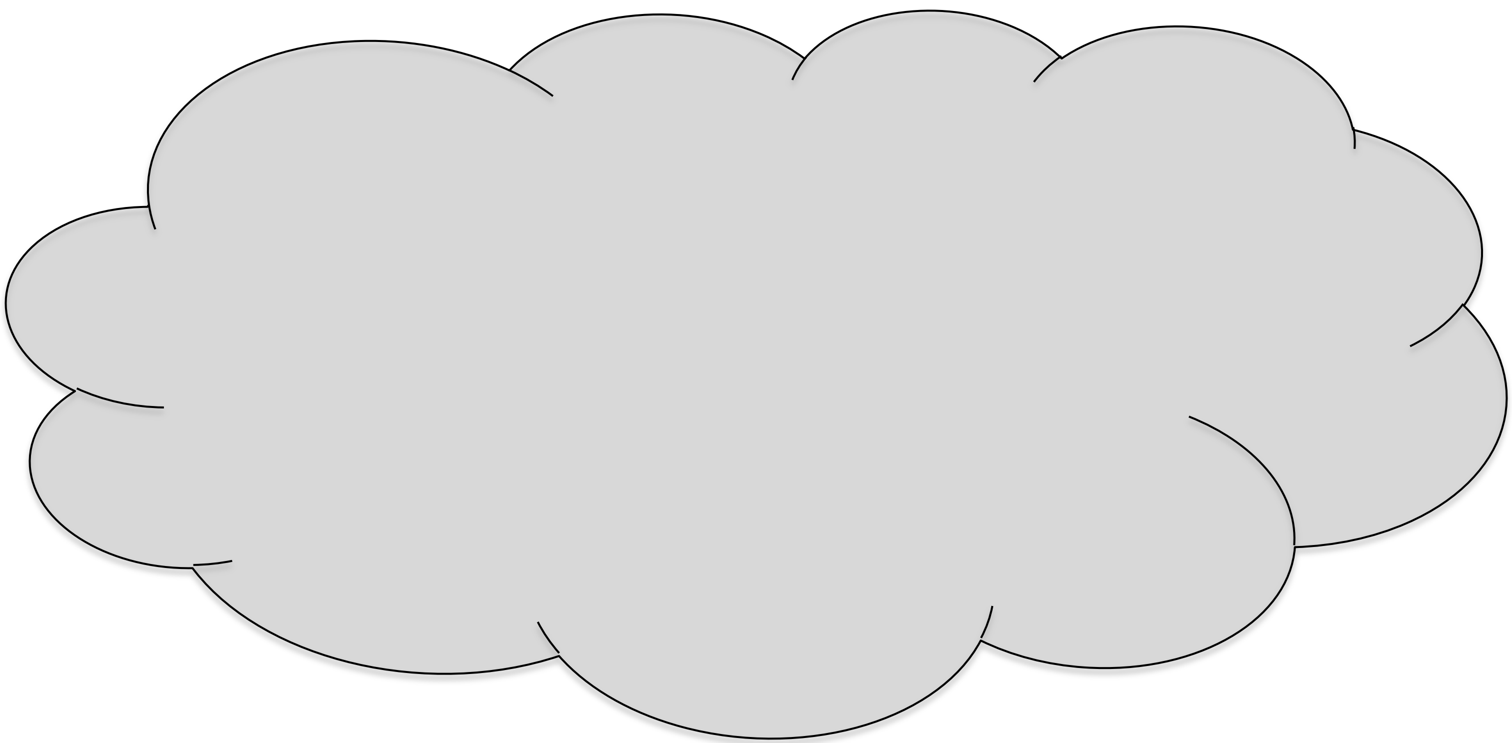
```
The ... car was ...
```

WHAT IS ATTENTION TRYING TO DO?

Consider the sentence

```
The ... car was ...
```

What does the car look like?



WHAT IS ATTENTION TRYING TO DO?

Consider the sentence

```
The ... car was ...
```

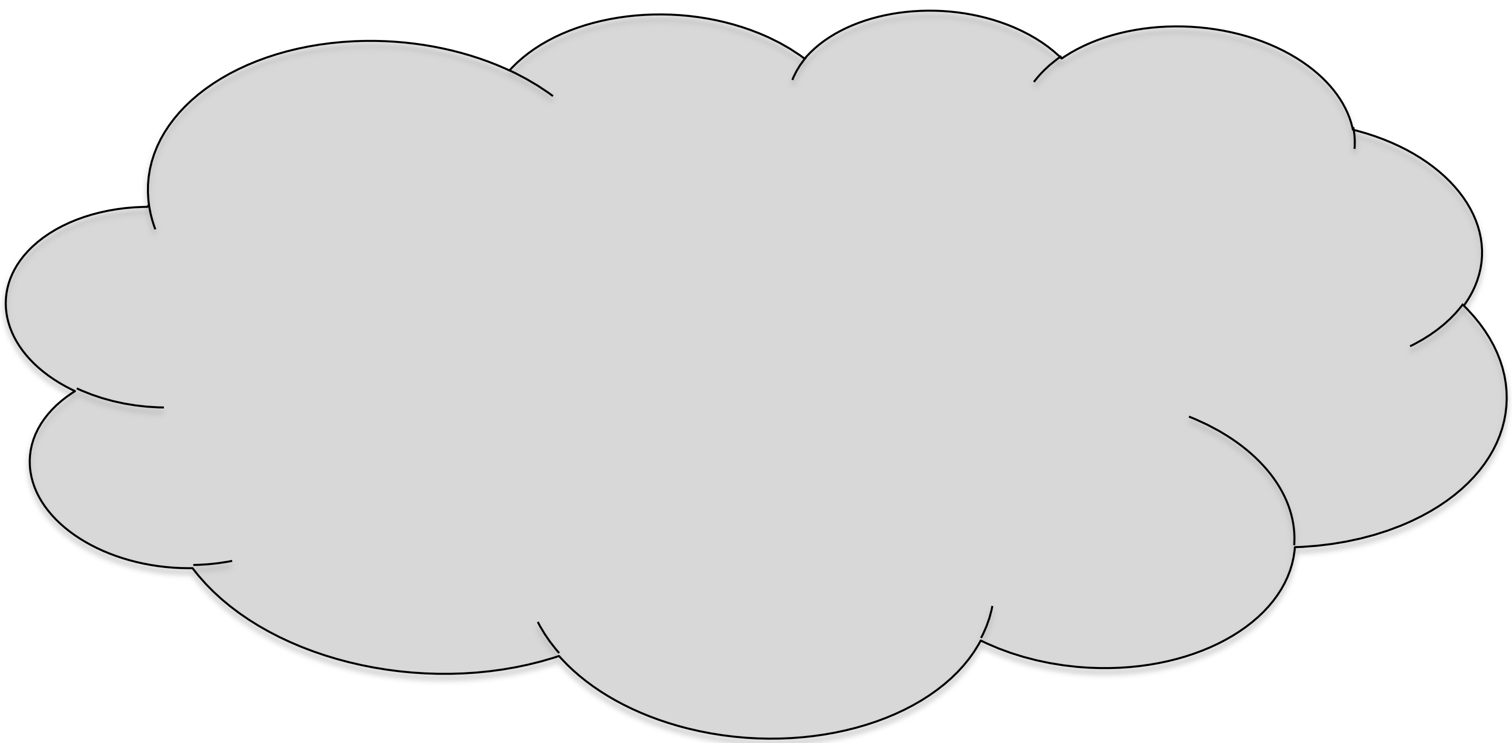
What does the car look like?

WHAT IS ATTENTION TRYING TO DO?

Consider the sentence

`The ... blue car was ...`

What does the car look like?



WHAT IS ATTENTION TRYING TO DO?

Consider the sentence

`The ... blue car was ...`

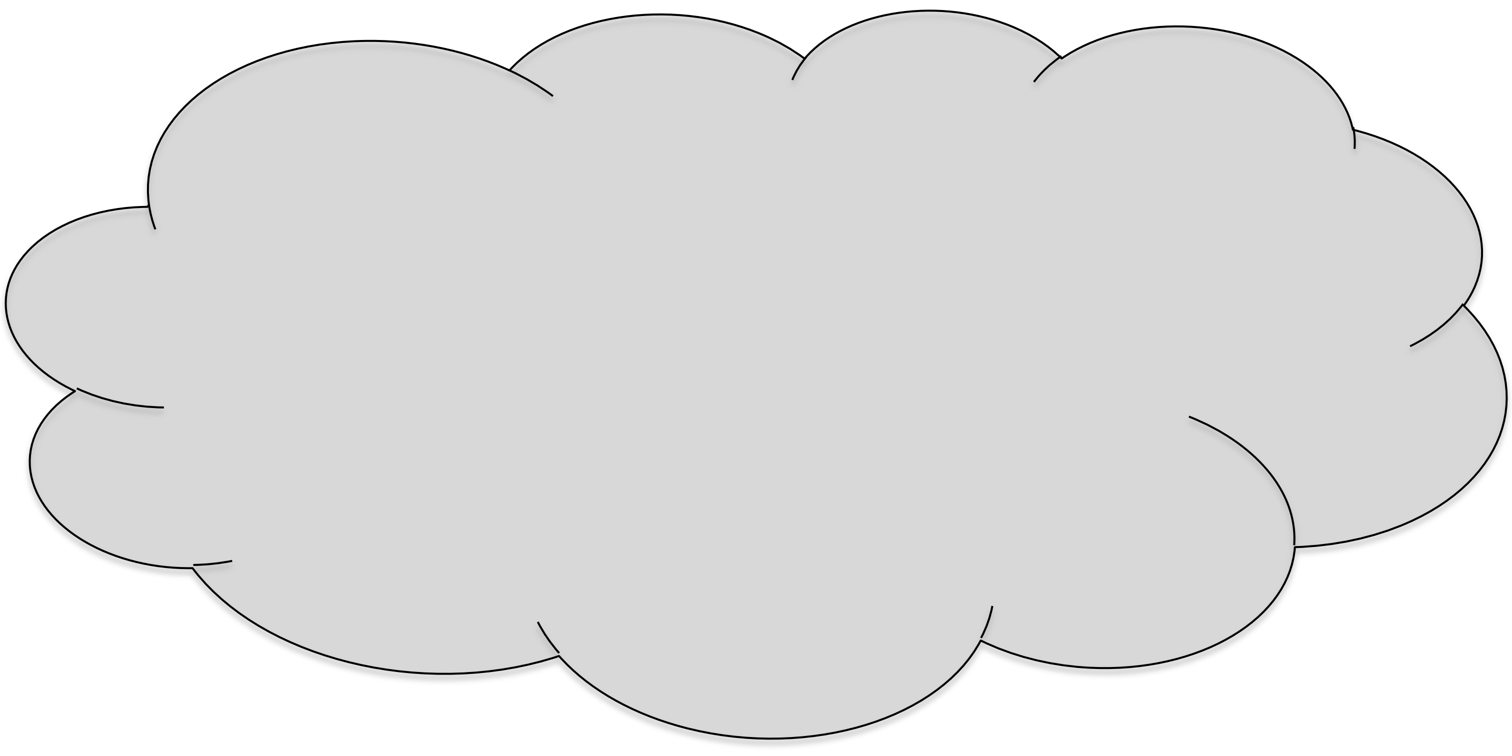
What does the car look like?

WHAT IS ATTENTION TRYING TO DO?

Consider the sentence

`The expensive blue car was ...`

What does the car look like?



WHAT IS ATTENTION TRYING TO DO?

Consider the sentence

`The expensive blue car was ...`

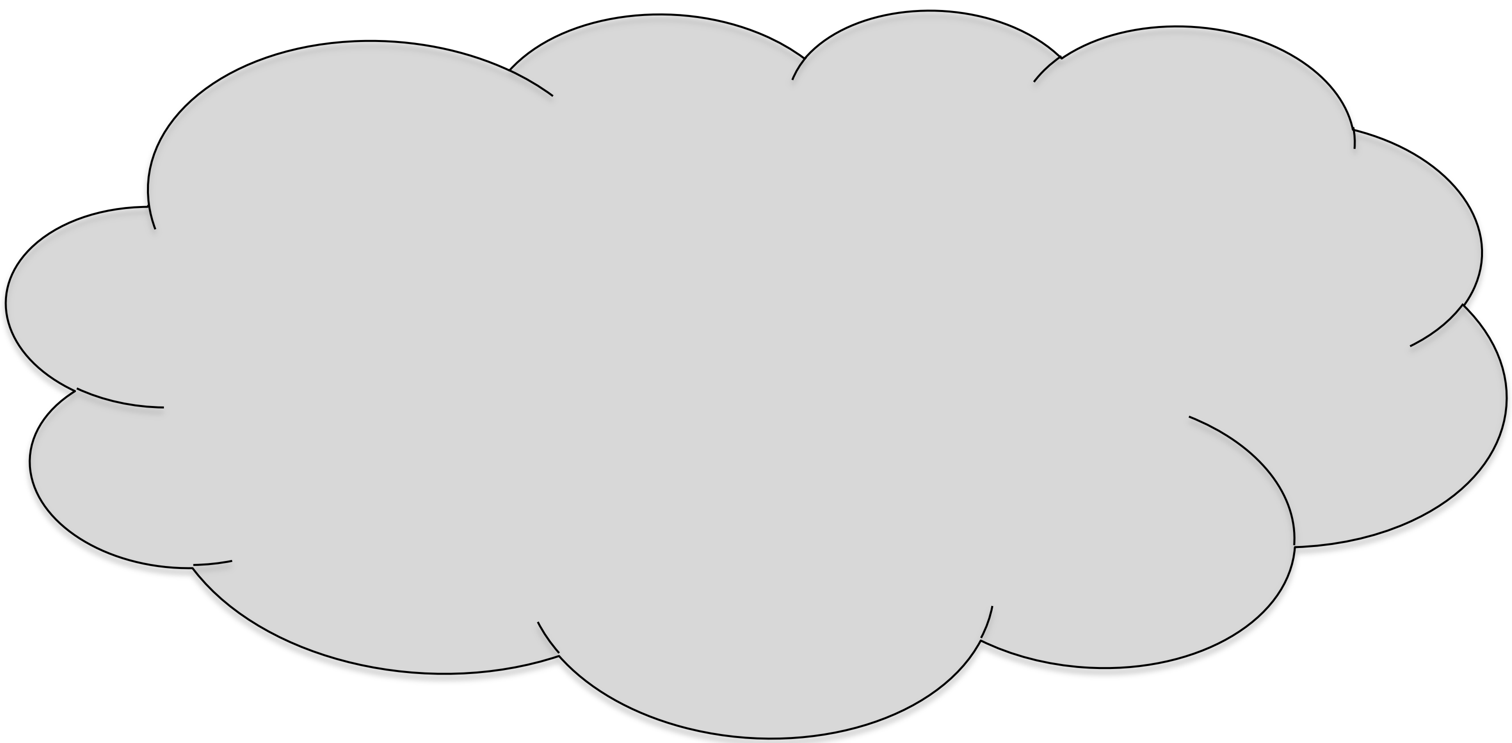
What does the car look like?

WHAT IS ATTENTION TRYING TO DO?

Consider the sentence

`The expensive blue car was totaled`

What does the car look like?

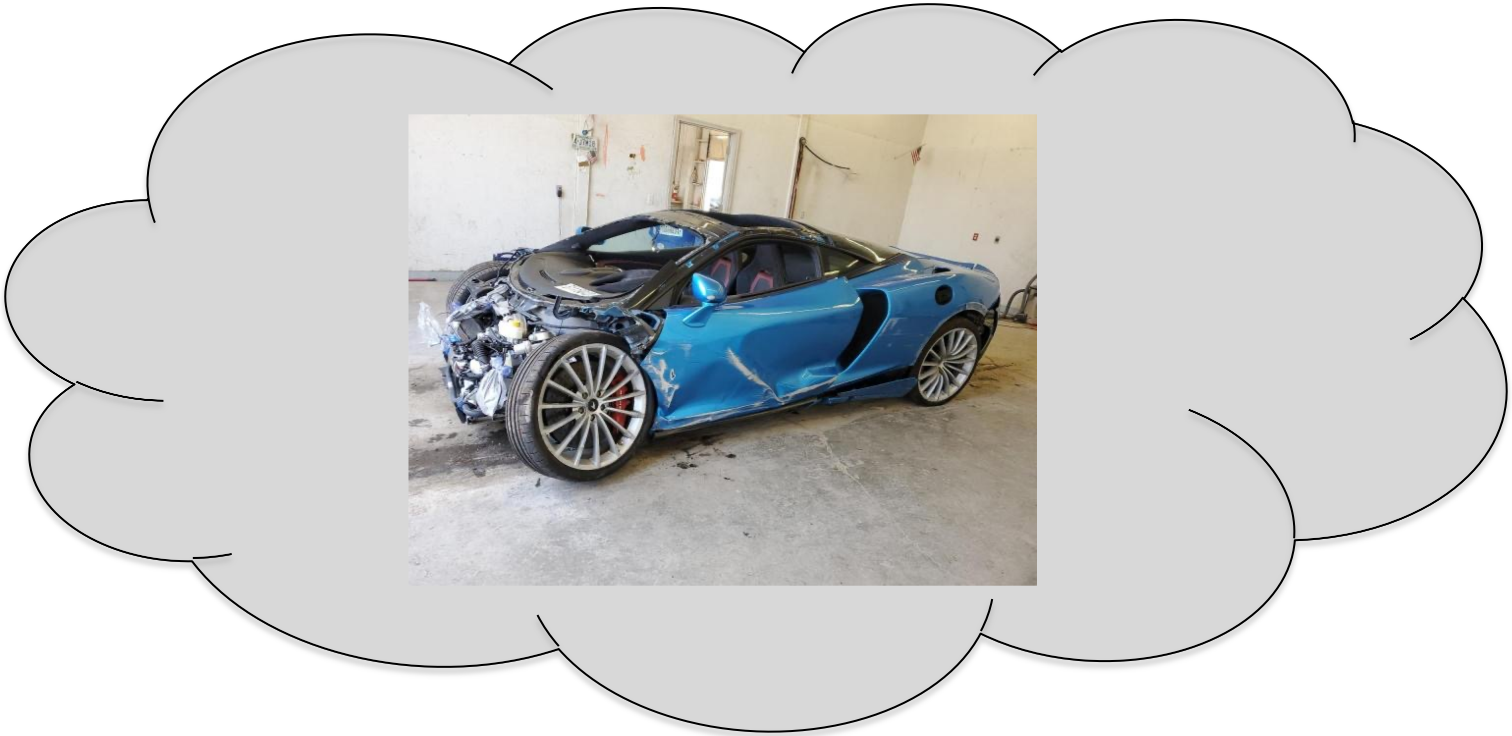


WHAT IS ATTENTION TRYING TO DO?

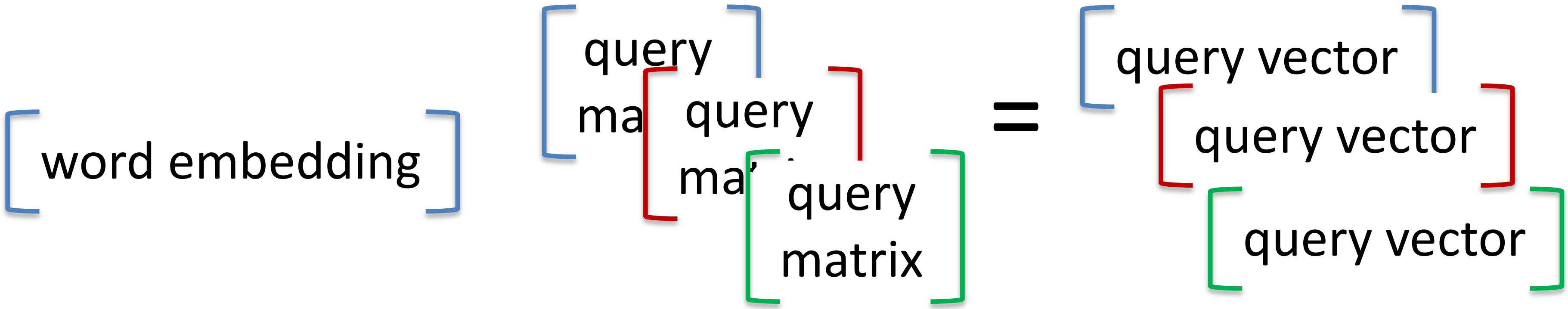
Consider the sentence

`The expensive blue car was totaled`

What does the car look like?

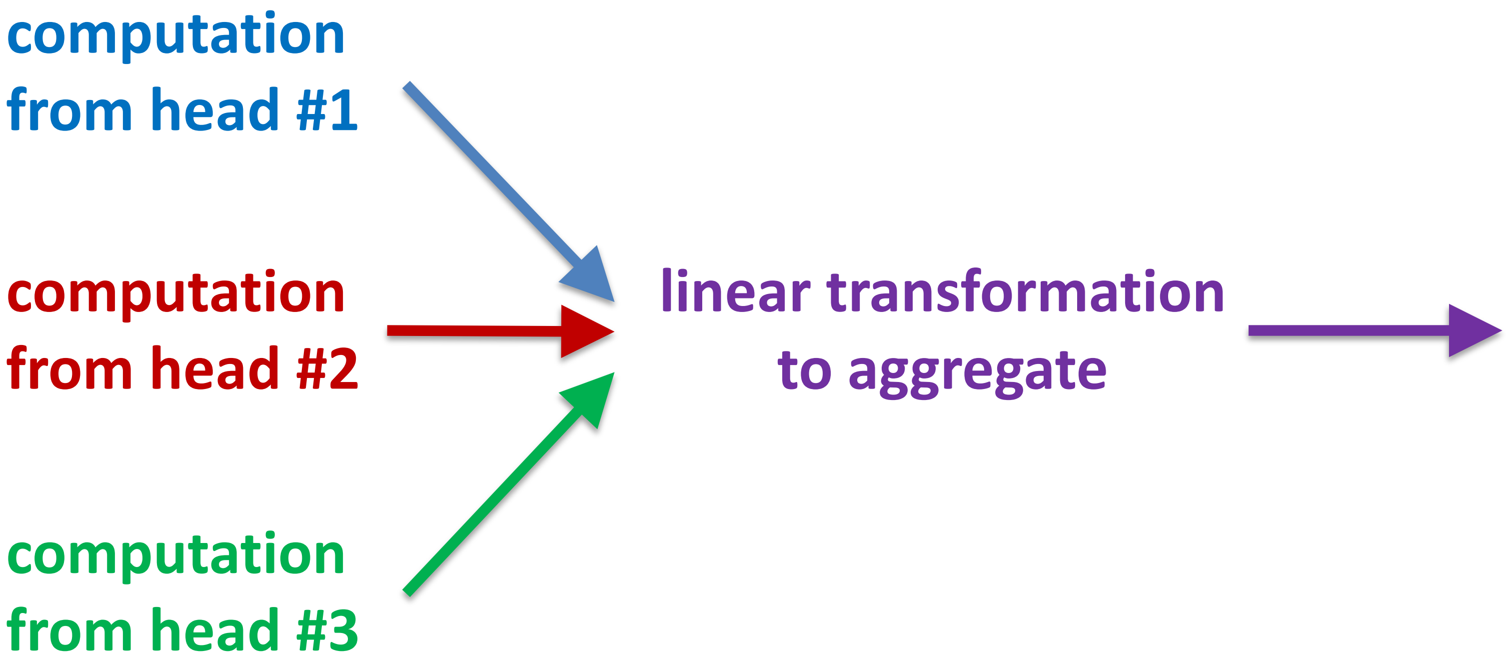


MULTI-HEADED ATTENTION

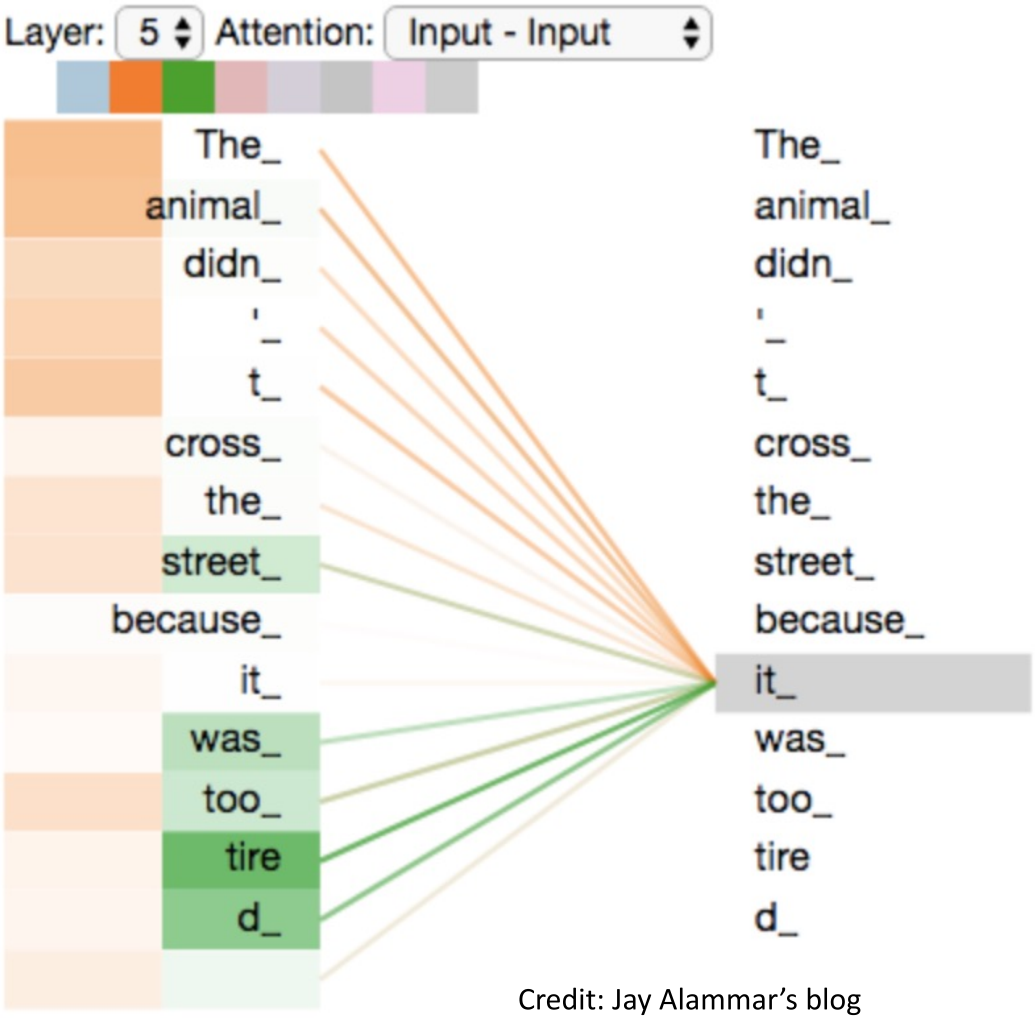


similarly for keys and values

COMBINING THE INFORMATION



MULTI-HEADED ATTENTION



Credit: Jay Alammar's blog

WHY MULTIPLE HEADS?

As a motivating example, consider

```
John and Doug planned to split the
bill but Doug didn’t have enough
money in his wallet. So he went to
the bank before they met up.
```

Who does he refer to?

WHY MULTIPLE HEADS?

Need different sorts of information, e.g.

Who are the people?

Which one doesn’t have money?

WHY MULTIPLE HEADS?

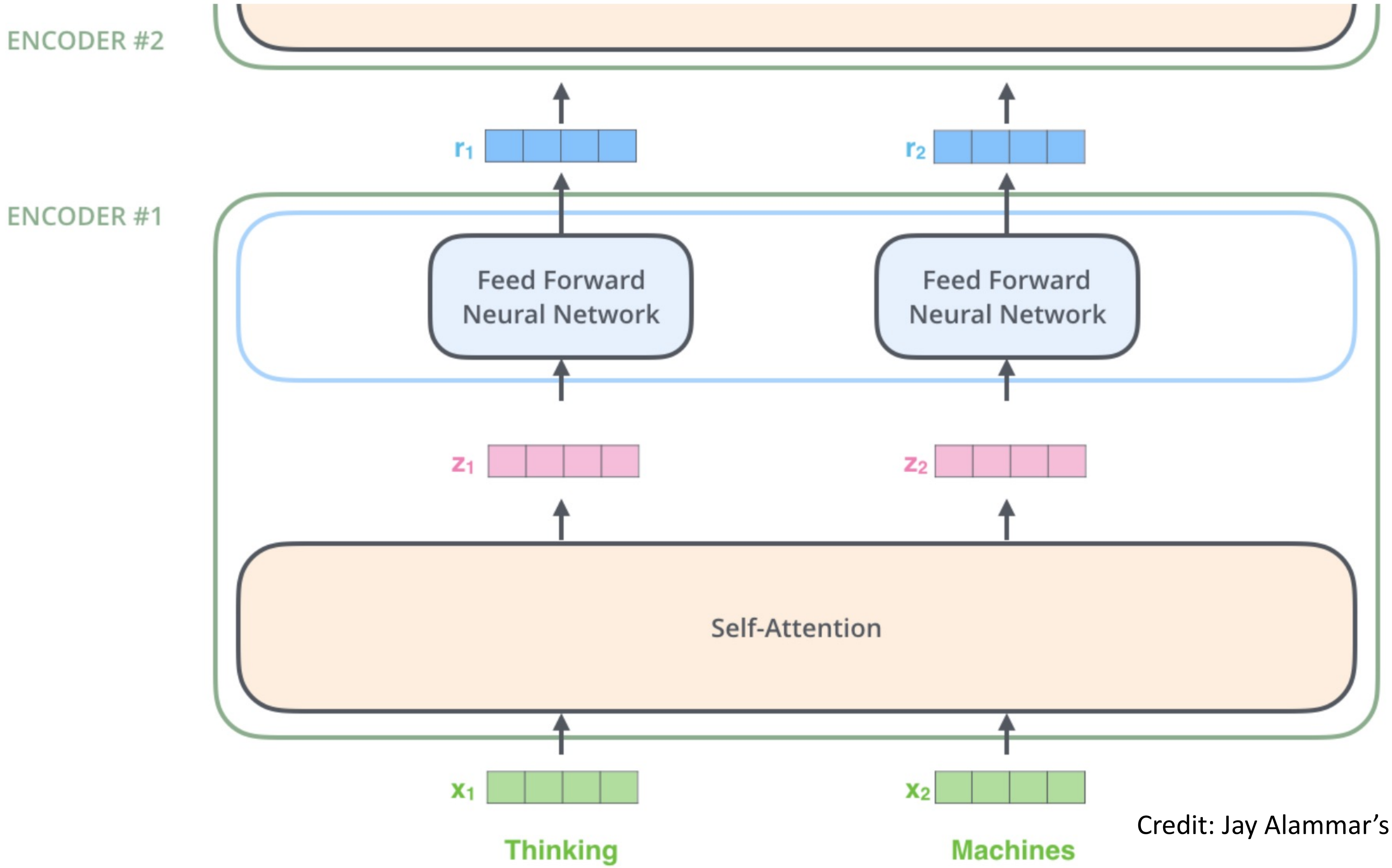
Need different sorts of information, e.g.

Who are the people?

Which one doesn’t have money?

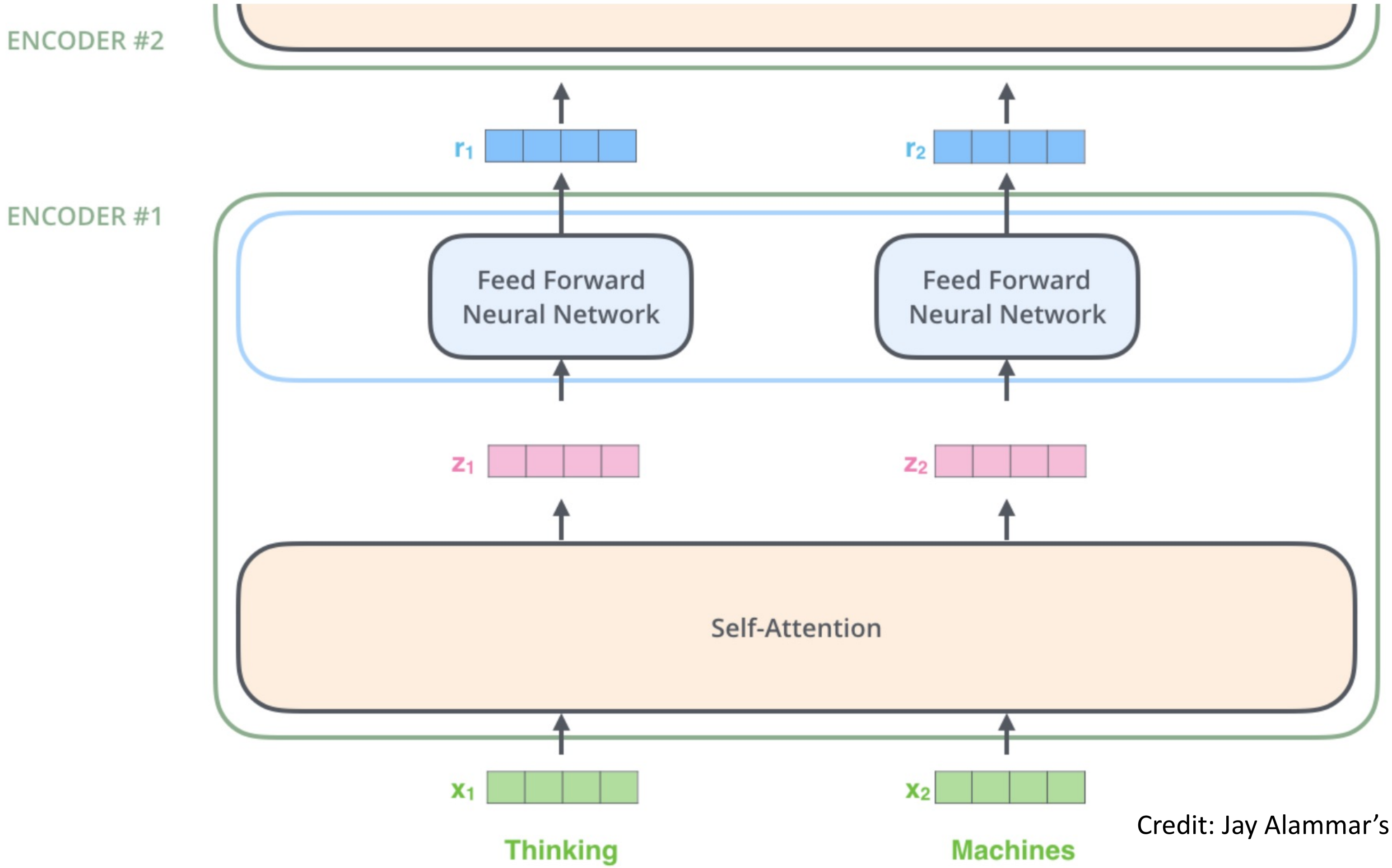
Natural approach is to have multiple channels for the flow of Information, e.g. to keep track of different attributes

ITERATING



Credit: Jay Alammar's blog

ITERATING



Credit: Jay Alammar's blog

Also intersperse normalization

OTHER CONSIDERATIONS

Architecture is designed so that computation is highly parallelizable

OTHER CONSIDERATIONS

Architecture is designed so that computation is highly parallelizable

But need to keep track of positional information

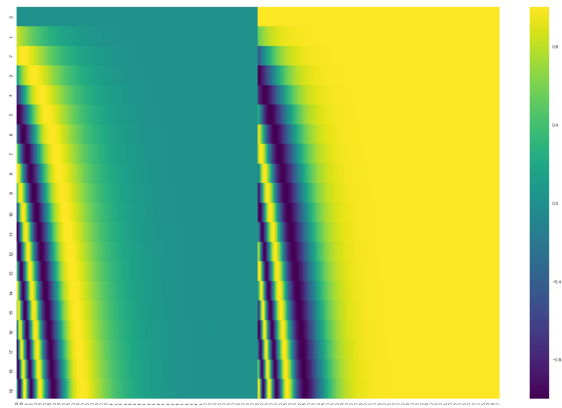
OTHER CONSIDERATIONS

Architecture is designed so that computation is highly parallelizable

But need to keep track of positional information

input vectors = word embeddings + positional embeddings

lookup table of fixed vectors to map each position to



Credit: Jay Alammar's blog

OTHER CONSIDERATIONS

Transformers are made up of an **encoder** and **decoder**

OTHER CONSIDERATIONS

Transformers are made up of an **encoder** and **decoder**

In order to use decoder for generation, need **masking**

attention computation only depends on earlier contexts

OTHER CONSIDERATIONS

Transformers are made up of an **encoder** and **decoder**

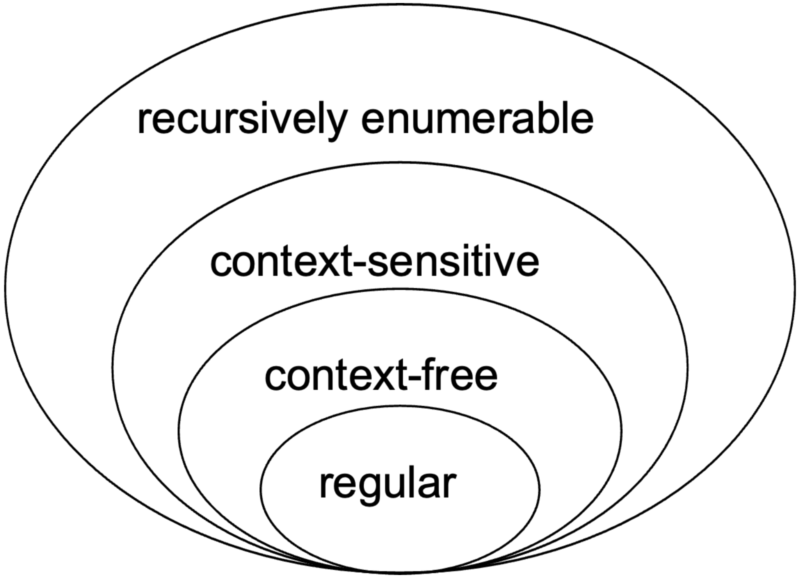
In order to use decoder for generation, need **masking**

attention computation only depends on earlier contexts

In particular, set all inner products with future contexts to $-\infty$ before computing softmax

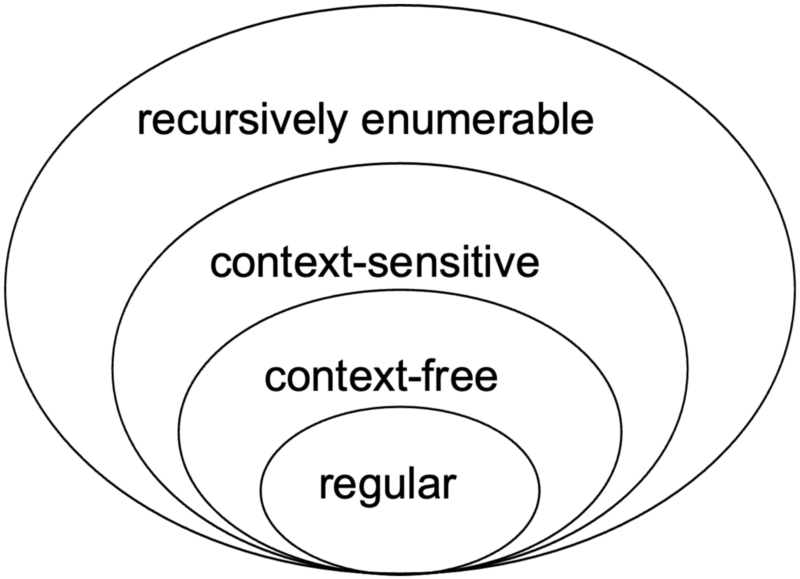
THE VIEW FROM LINGUISTICS

The Chomsky Hierarchy was an attempt to formalize the syntax of natural language



THE VIEW FROM LINGUISTICS

The Chomsky Hierarchy was an attempt to formalize the syntax of natural language



But natural language, especially corner-cases, can be quite complex

THE VIEW FROM LINGUISTICS

Recognizing all of natural language can be quite complex

STUART M. SHIEBER*

EVIDENCE AGAINST THE CONTEXT-FREENESS OF NATURAL LANGUAGE**

THE VIEW FROM LINGUISTICS

Recognizing all of natural language can be quite complex

STUART M. SHIEBER*

EVIDENCE AGAINST THE CONTEXT-FREENESS
OF NATURAL LANGUAGE**

Theorem: Swiss German is not context free

THE VIEW FROM LINGUISTICS

Recognizing all of natural language can be quite complex

STUART M. SHIEBER*

EVIDENCE AGAINST THE CONTEXT-FREENESS OF NATURAL LANGUAGE**

Theorem: Swiss German is not context free

But these are often contrived examples, don’t arise often, don’t need to solve them to get good completions

REVERSE ENGINEERING CIRCUITS FROM GPT

[Wang, Variengien, Conmy, Shlegeris, Steinhardt] considered

```
Mary and John went to a bar. John
handed a drink to _____
```

REVERSE ENGINEERING CIRCUITS FROM GPT

[Wang, Variengien, Conmy, Shlegeris, Steinhardt] considered

```
Mary and John went to a bar. John
     handed a drink to _____
```

What is the next word?

REVERSE ENGINEERING CIRCUITS FROM GPT

[Wang, Variengien, Conmy, Shlegeris, Steinhardt] considered

```
Mary and John went to a bar. John
     handed a drink to _____
```

What is the next word?

Used tools from causality to identify a circuit computing this in GPT2

REVERSE ENGINEERING CIRCUITS FROM GPT

[Wang, Variengien, Conmy, Shlegeris, Steinhardt] considered

```
Mary and John went to a bar. John
    handed a drink to _____
```

What is the next word?

Used tools from causality to identify a circuit computing this in GPT2

Copy this word, only if this no other word gets copied there

WINOGRAD SCHEMAS

For natural sentences, what kinds of logical circuits are needed?

```
I tried to put the trophy in the
   suitcase but it was too big
```

WINOGRAD SCHEMAS

For natural sentences, what kinds of logical circuits are needed?

```
I tried to put the trophy in the
   suitcase but it was too big
```

What does it refer to?

WINOGRAD SCHEMAS

For natural sentences, what kinds of logical circuits are needed?

```
I tried to put the trophy in the
suitcase but it was too big
```

What does it refer to?

```
I tried to put the trophy in the
suitcase but it was too small
```

WINOGRAD SCHEMAS

For natural sentences, what kinds of logical circuits are needed?

```
I tried to put the trophy in the
suitcase but it was too big
```

What does it refer to?

```
I tried to put the trophy in the
suitcase but it was too small
```

How about now?

PROSPECTS

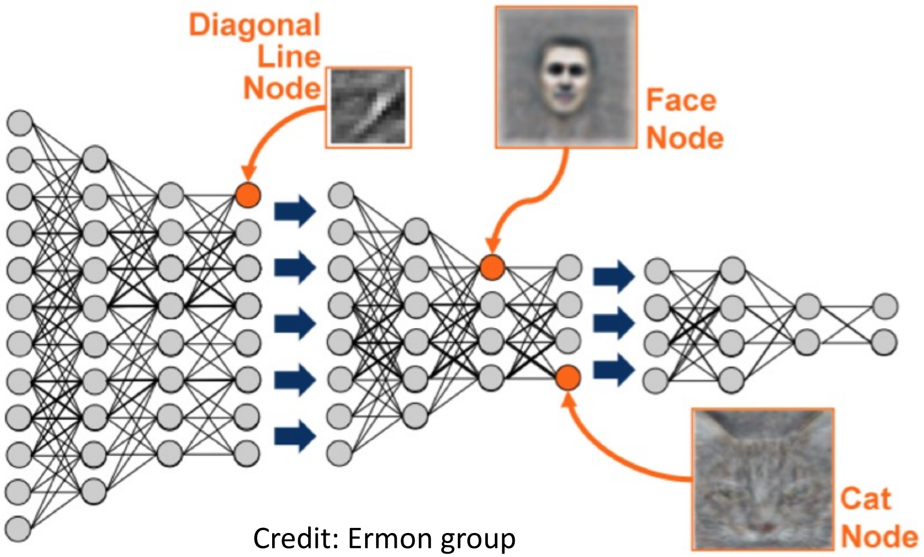
For deep learning, we have ansatzes about how we think it processes information, e.g.

> **"lower layers recognize simple features like edges, higher layers recognize composite features that are the building blocks of objects"**

PROSPECTS

For deep learning, we have ansatzes about how we think it processes information, e.g.

> **"lower layers recognize simple features like edges, higher layers recognize composite features that are the building blocks of objects"**



Credit: Ermon group

PROSPECTS

For deep learning, we have ansatzes about how we think it processes information, e.g.

What about for transformers?

PROSPECTS

For deep learning, we have ansatzes about how we think it processes information

What about for transformers?

Too general an answer can lead us down a rabbit hole, e.g.

> **"deep neural networks can approximate any continuous function arbitrarily well"**

PROSPECTS

For deep learning, we have ansatzes about how we think it processes information

What about for transformers?

Too general an answer can lead us down a rabbit hole, e.g.

> "deep neural networks can approximate any continuous function arbitrarily well"

Need theories rooted in the structure of data (language), corroborated experimentally (interpretability)

Thanks! Any Questions?