Controllable Response Generation

Susana Benavidez
Andrew Kirjner
Nick Seay
Mentor: Sina Semnani

Overview

Part 1 Text Generation vs Controllable Text Generation

Part 2 Conditional Training
Weighted Decoding

Part 3 Transformer + Attribute Model:
The Mammoth and the Mouse

Challenges of Text generation:

- **Semantics (meaning)**
- **Consistency** (long text generation)
- **Logic** (reasonable and making sense)

Challenges of Text generation:

Semantics (meaning) *Not our concern*
Consistency (long text generation) *Not our concern*
Logic (reasonable and making sense) *Not our concern*

Different Goals

Information v. Enhancing interactiveness and persistence of human-machine interactions

We already have the response - how can we make it more natural?

What for? What do we want to control?

What for? What do we want to control?

- Task of generating realistic sentences whose attributes can be controlled
- What can we control? *[Prabhumoye et. al, 2020]*
  - Stylistic (politeness, sentiment, formality, etc)
  - Demographic attributes of the person writing the text (e.g. gender, age, etc)
  - Content (e.g. information, keywords, entities) to be generated (BOW)
  - Order of information, events (e.g. plot summaries)

What for? What do we want to control?

- What for? (Dialogue response generation task) *[Prabhumoye et. al, 2020]*
  - Controlling persona
  - Controlling aspects of response (politeness, formality, authority, grounding response in external source of information, controlling topic sentence, story generation (control ending, persona, plot, and topic sentence)
  - Modulate formality/politeness of emails
  - Report generation (pulling source documents into unified doc)

Techniques:

Conditional Training

Weighted Decoding

Technique: Conditional Training: Model conditioned on additional control features

- Learn a sequence-to-sequence model *P(y | x, z)*, *z*: discrete **control variable**
    - During training: determine corresponding *z* value for each sample
    - Append *z* to the end of the input sequence, *z* as START symbol for decoder; concatenate z to decoder's input at every step

Technique: Conditional Training:

Example

Input: *Yes, I'm studying law at the moment*
Baseline Response: *That sounds like a lot of fun!*

z	NIDF	Conditional Training Response
0	16.8%	*Sounds like you are a great person!*
2	18.3%	*So you are a law student?*
4	18.4%	*That sounds like a lot of fun*
6	22.8%	*That sounds like a rewarding job!*
8	24.4%	*That sounds like a rewarding career!*

- Controlling specificity via conditional training.
- Define the specificity of an utterance *y* to be the mean NIDF of the words in *y*.
- Control variable is mean NIDF (discretized into 10 equal-sized buckets) which gives outputs with a narrower NIDF range, but produces less nonsensical outputs

Decoder Techniques: What makes a good conversation?

- Weighted Decoding (control features added to the decoding scoring function at test time only)
    - Increase/Decrease probability of words with certain features
        - Extreme Weights: block words (can have unintended consequences)
    - Limitation: controllable attribute must be defined at the word-level; any desired utterance-level attribute must be redefined via word-level features

Decoder Techniques: What makes a good conversation?

- **Low-Level Controllable Attributes:**
  - **Repetition** n-gram overlap
    - External: (self-repetition across utterances)
    - Internal: (self-repetition within utterances)
    - Partner: (repeating the conversational partner)
  - **Specificity** (Normalized Inverse Document Frequency)
    - As a measure of word rareness

Decoder Techniques: Weighted Decoding

Example

Input: *Yes, I'm studying law at the moment*
Baseline Response: *That sounds like a lot of fun!*

Wt	NIDF	Weighted Decoding Response
-5.0	0.6%	*Oh.......................................*
0.0	17.1%	*That sounds like a lot of fun!*
3.0	18.3%	*That sounds like a lot of fun. How long have you been studying?*
7.0	38.5%	*I majored in practising my spiritual full time philosophy test*
10.0	71.9%	*Oh wow! Merna jean isa paino yi hao hui bu acara sya gila [...]*

- Controlling specificity via weighted decoding (use NIDF as decoding feature)
  - At the extremes, the model produces only the most rare (gibberish) or the most common tokens (useless)

Transformer
+ Attribute Model

GPT2 + PPLM Model

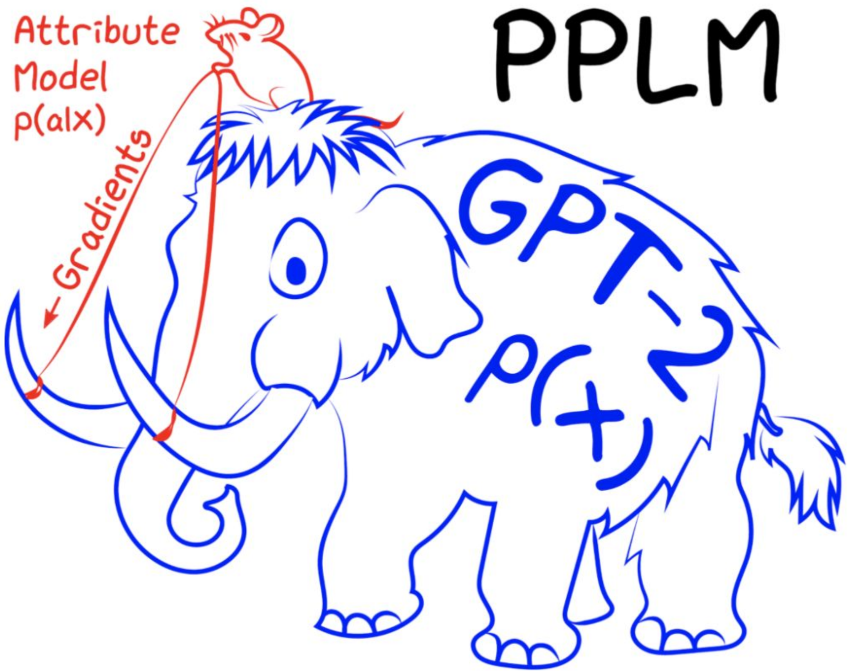


Image Courtesy of: https://eng.uber.com/pplm/

Why is GPT2 the Mammoth and PPLM the Mouse?

A General Transformer

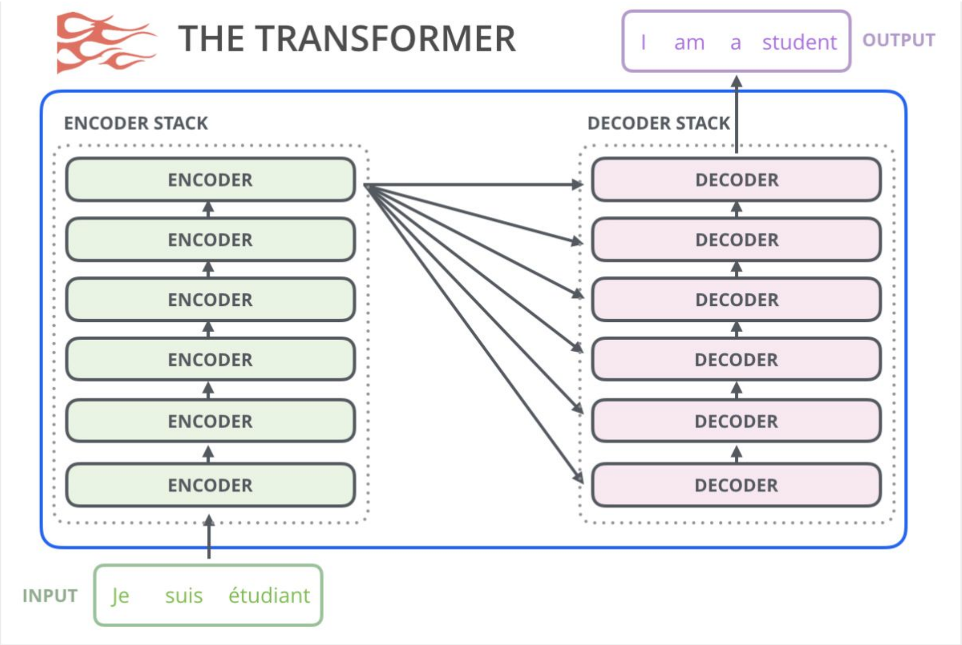


Image Courtesy of: http://jalammar.github.io/illustrated-gpt2/

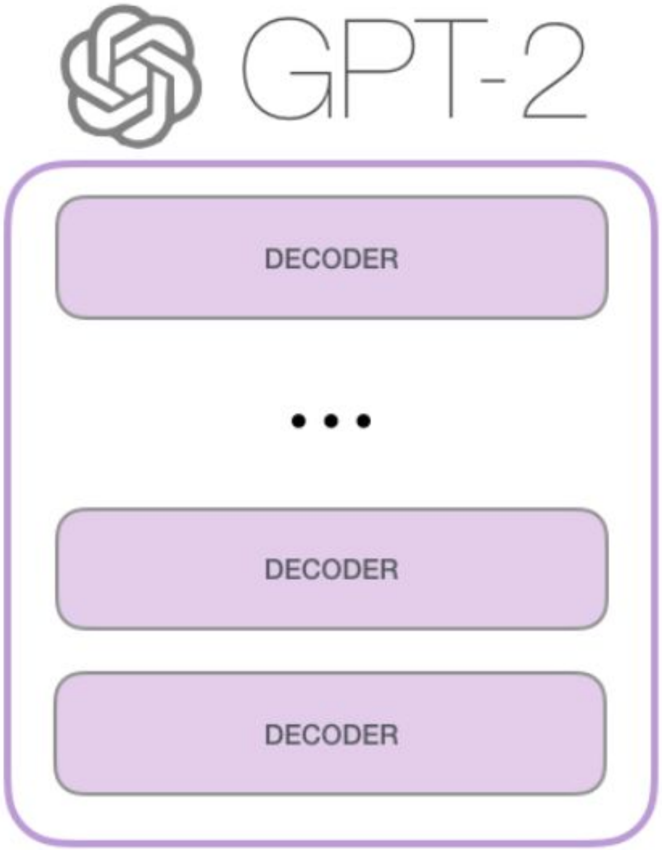


Image Courtesy of: http://jalammar.github.io/illustrated-gpt2/

Decoder Block

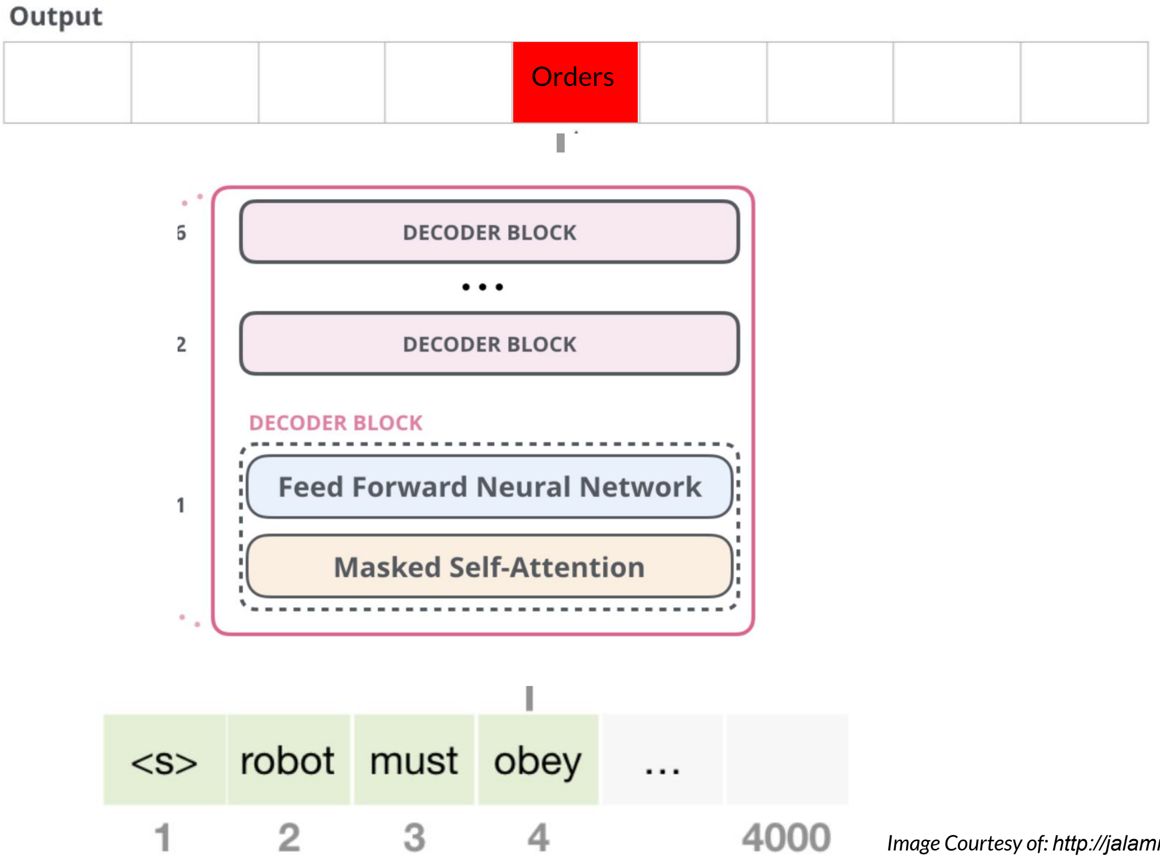


Image Courtesy of: http://jalammar.github.io/illustrated-gpt2/

Input Embeddings:

What gets passed in to the Decoder Block

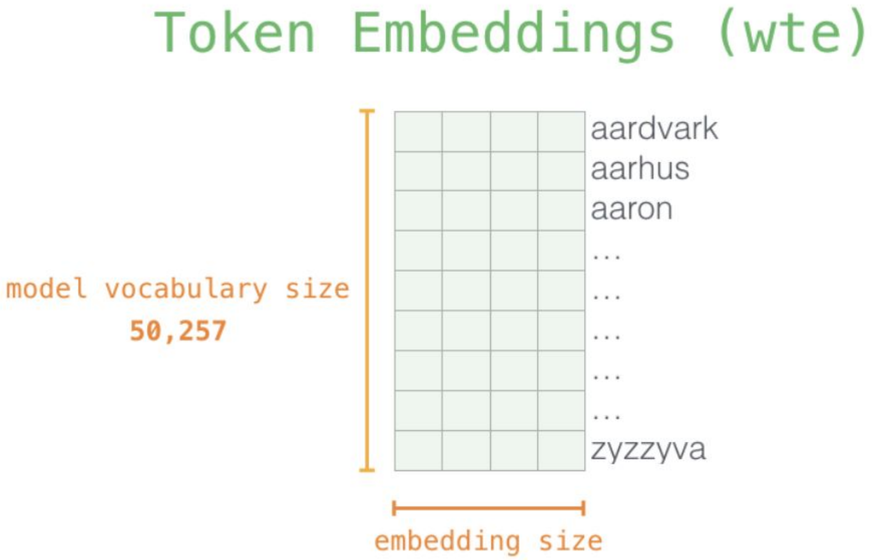


Image Courtesy of: http://jalammar.github.io/illustrated-gpt2/

Decoder Block - With Embeddings

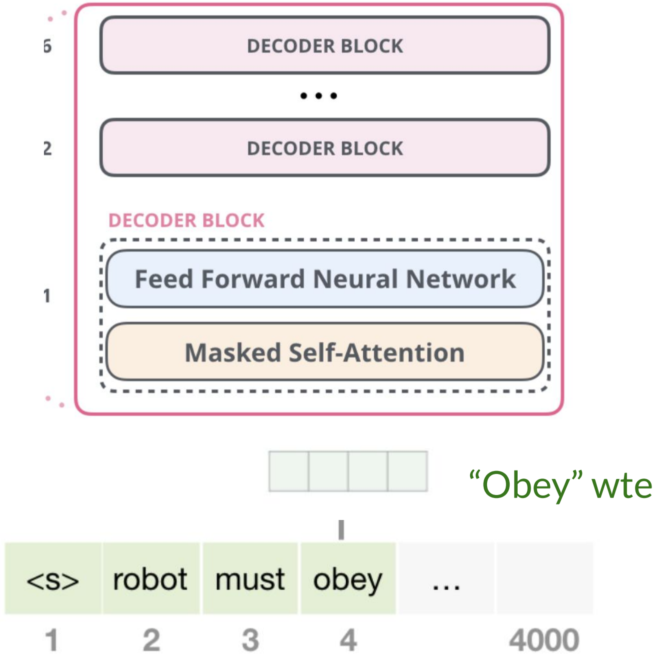


Image Courtesy of: http://jalammar.github.io/illustrated-gpt2/

GPT2 Output

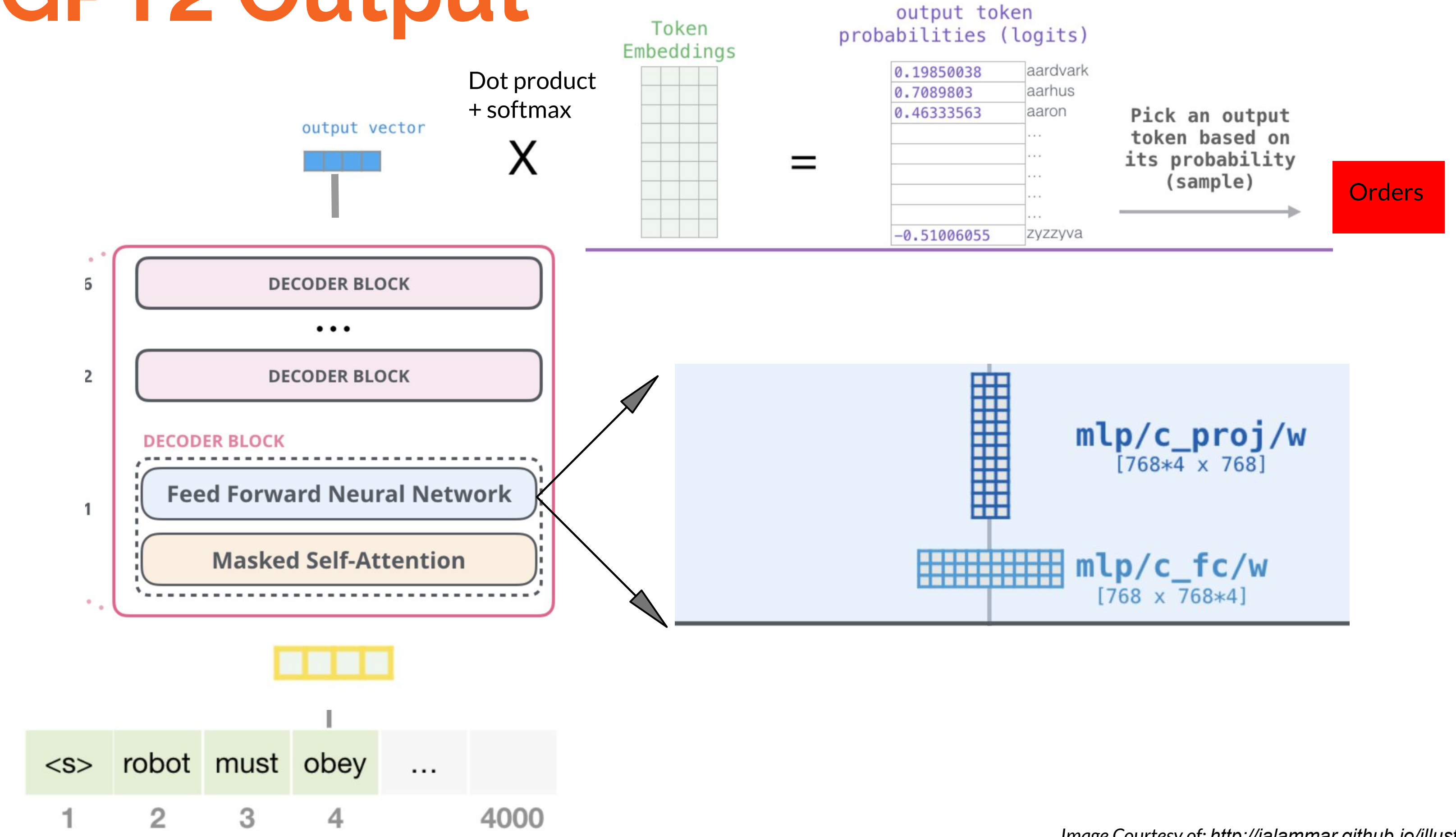


Image Courtesy of: *http://jalammar.github.io/illustrated-gpt2/*

Recall

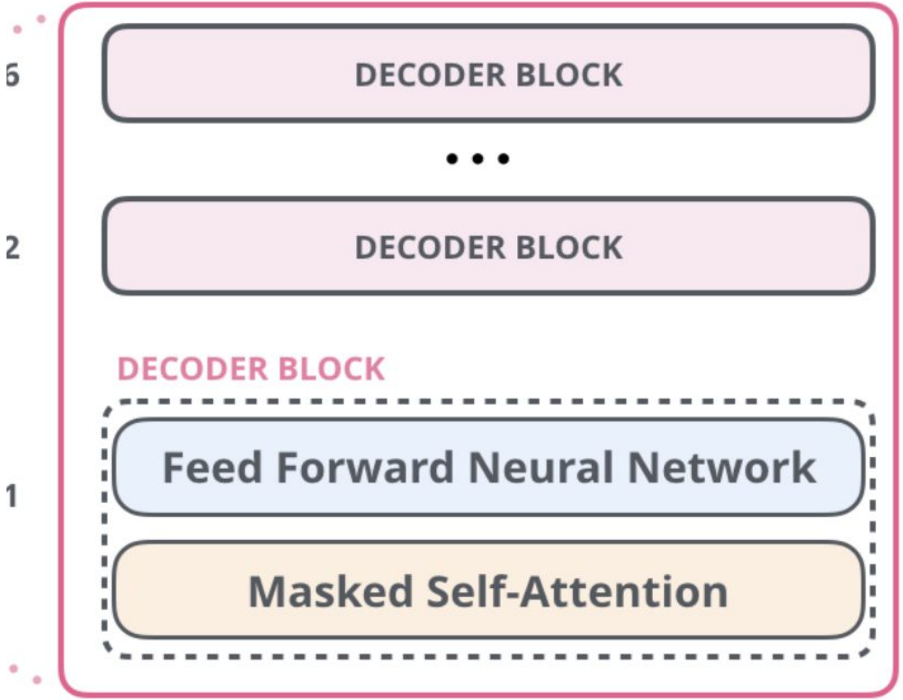


Image Courtesy of: http://jalammar.github.io/illustrated-gpt2/

Recall

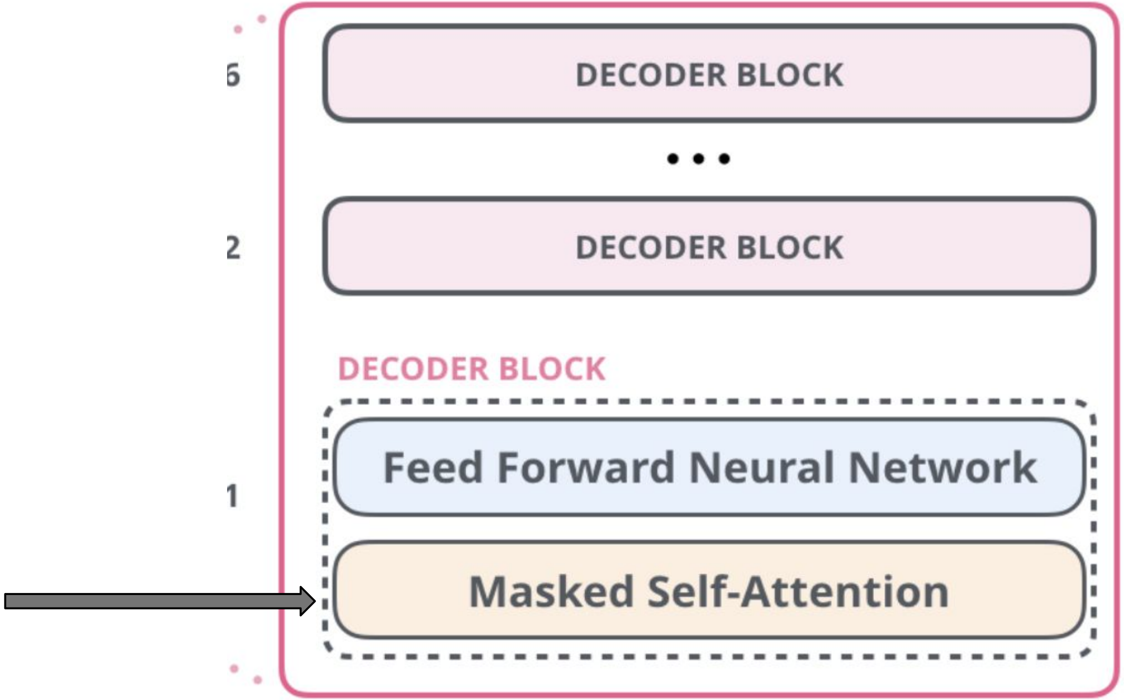
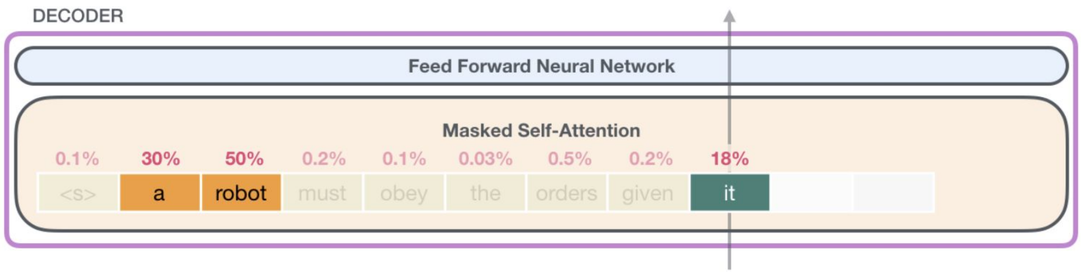


Image Courtesy of: *http://jalammar.github.io/illustrated-gpt2/*

Masked Self-Attention

Second Law of Robotics
*A robot must obey the orders given **it** by human beings except where **such orders** would conflict with the **First Law**.*



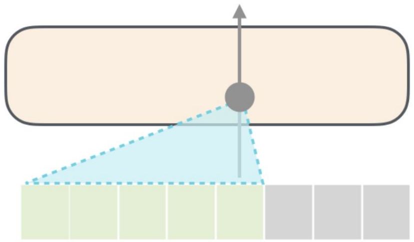


Image Courtesy of: http://jalammar.github.io/illustrated-gpt2/

Masked Self-Attention: Steps

1. Create the Query, Key, and Value (Q, K, V) vectors
2. For each input token, use its query vector to score against all the other key vectors, and then take weighted sum to get final context-dependent vector

[Alammar, 2019]

Step 1: Create Q-K-V Vectors

- Query: The query is a representation of the current word used to score against all the other words (using their keys). We only care about the query of the token we're currently processing.
- Key: Key vectors are like labels for all the words in the segment. They're what we match against in our search for relevant words.
- Value: Value vectors are actual word representations, once we've scored how relevant each word is, these are the values we add up to represent the current word.

[Alammar, 2019]

Step 1: Create Q-K-V Vectors

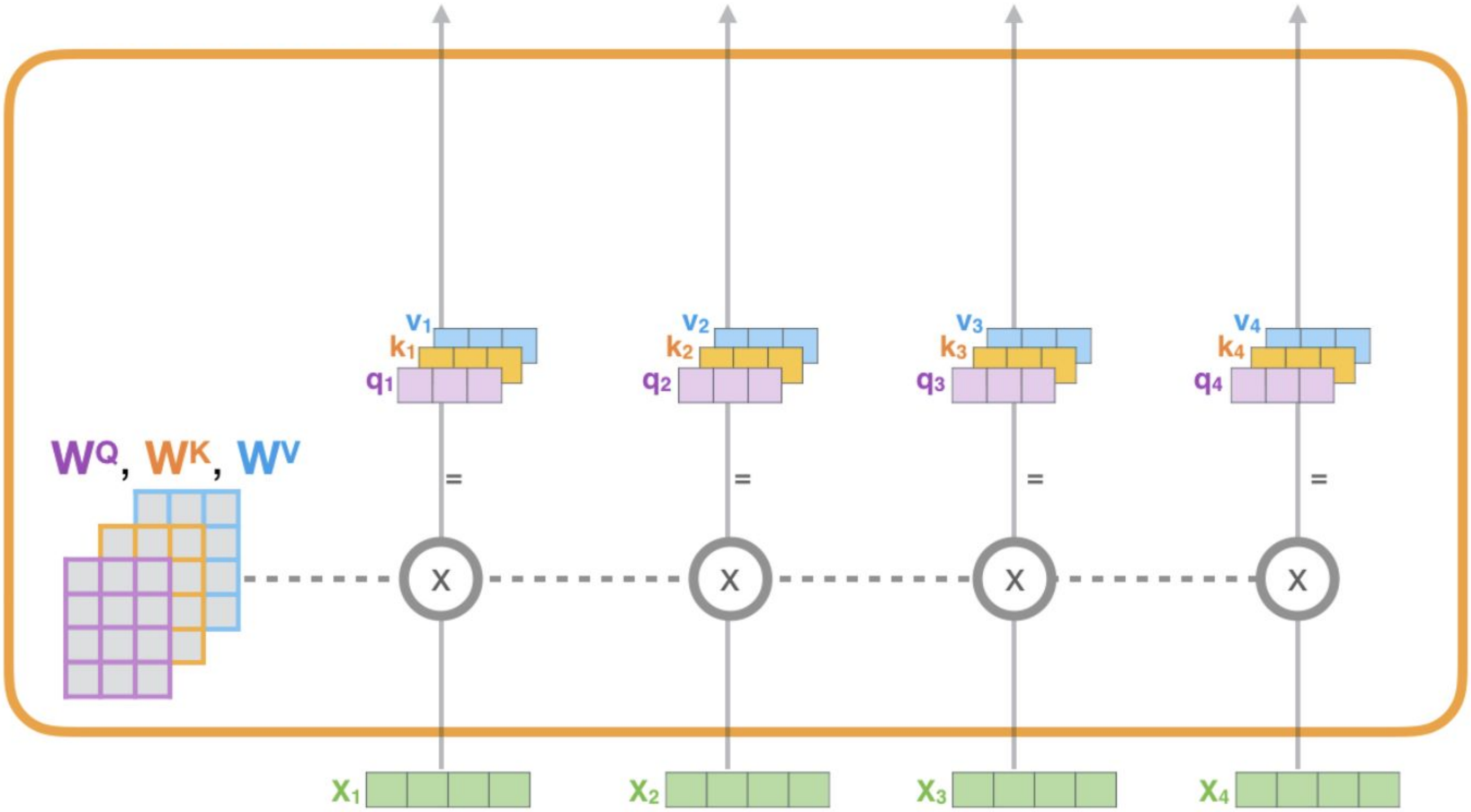


Image Courtesy of:
http://jalammar.github.io
/illustrated-gpt2/

Step 2:
Score + Sum

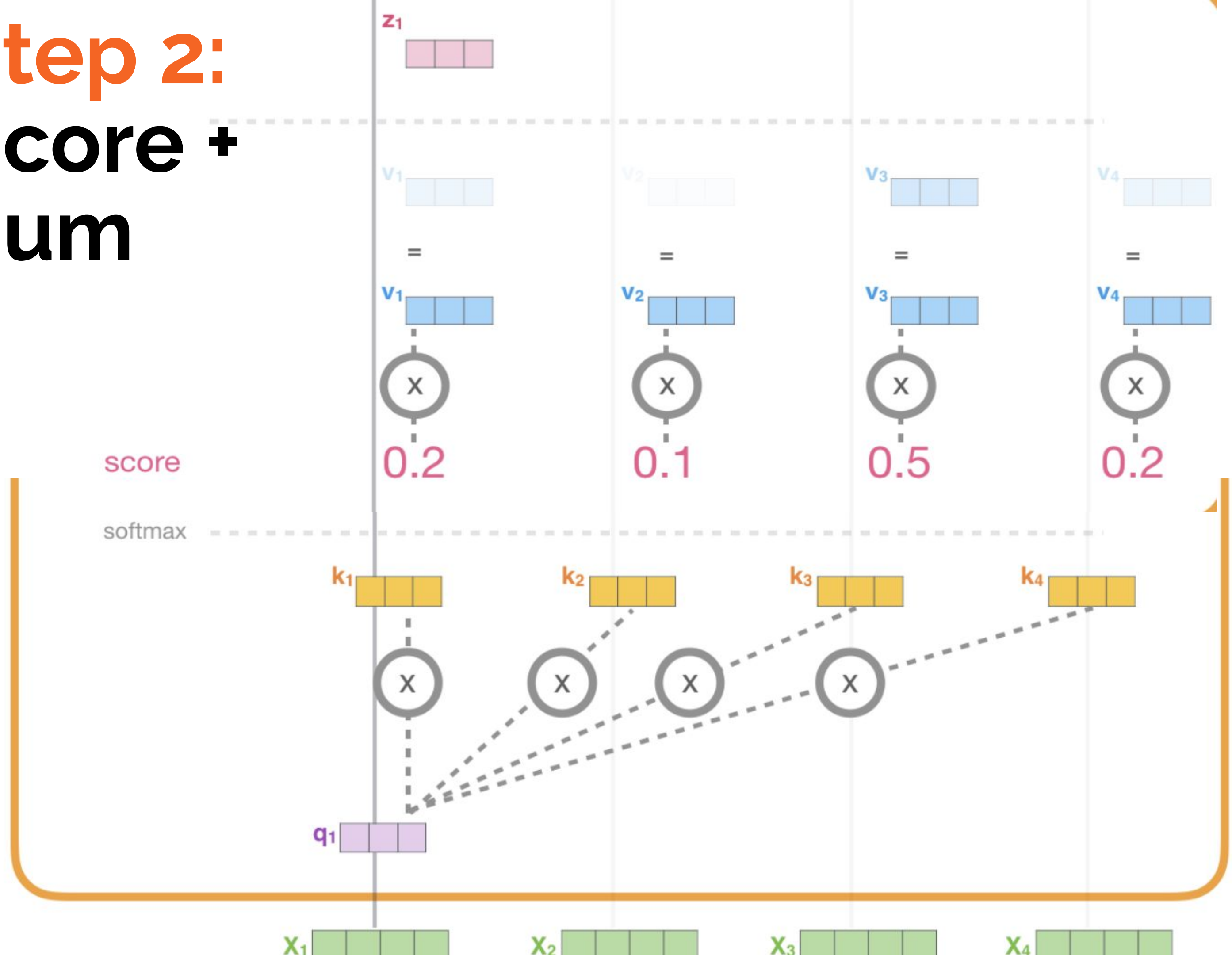


Image Courtesy of: http://jalammar.github.io /illustrated-gpt2/

Masked Self Attention: Q-K-V Vectors

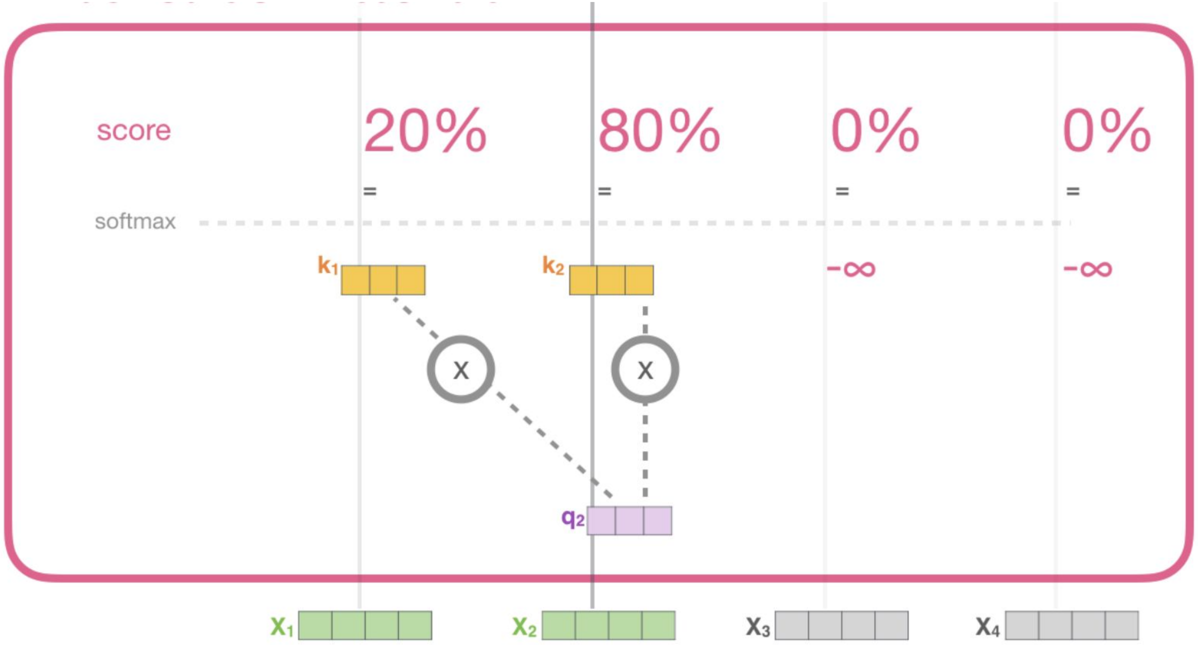


Image Courtesy of: http://jalammar.github.io/illustrated-gpt2/

GPT2 Overview

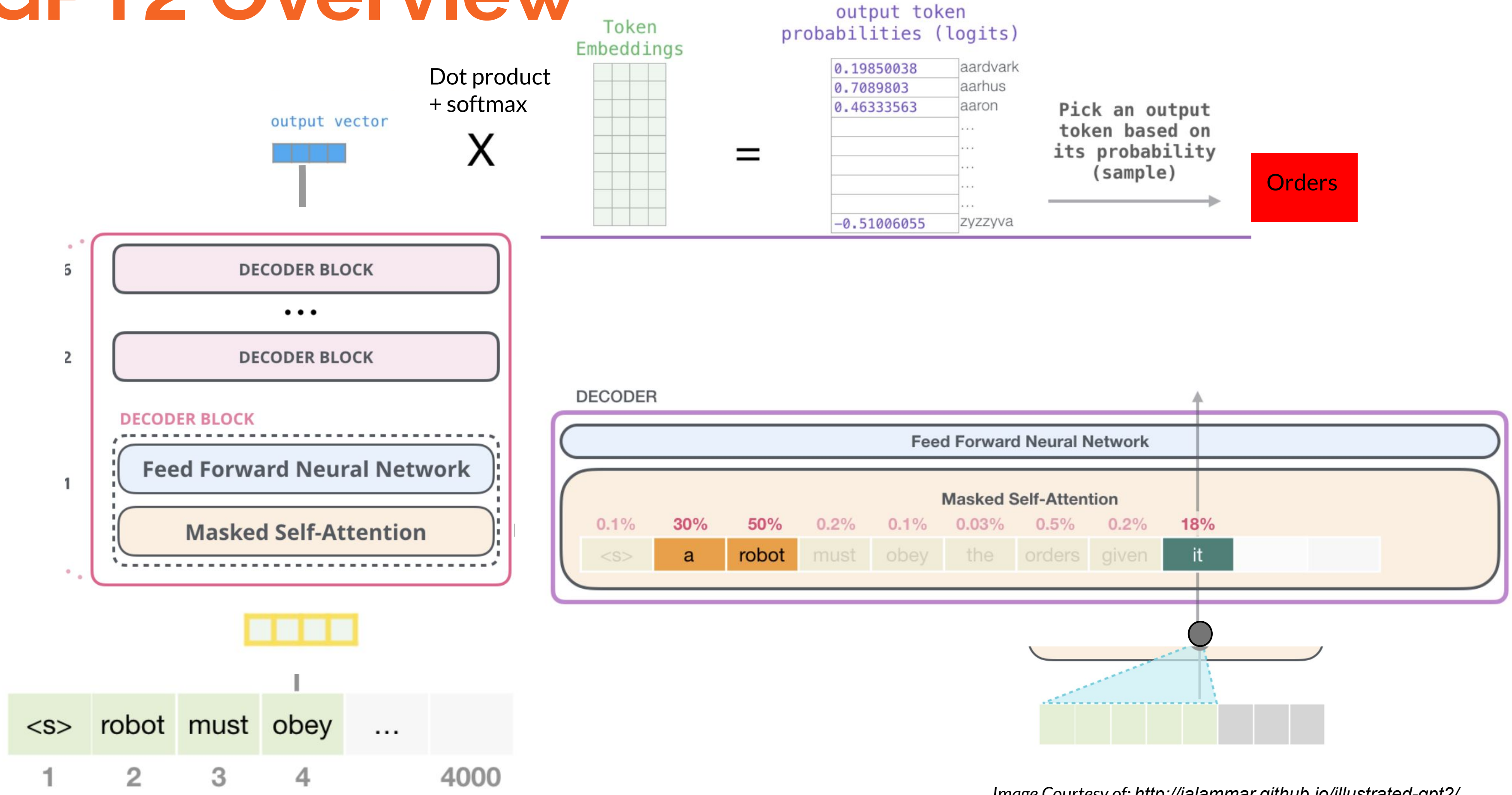
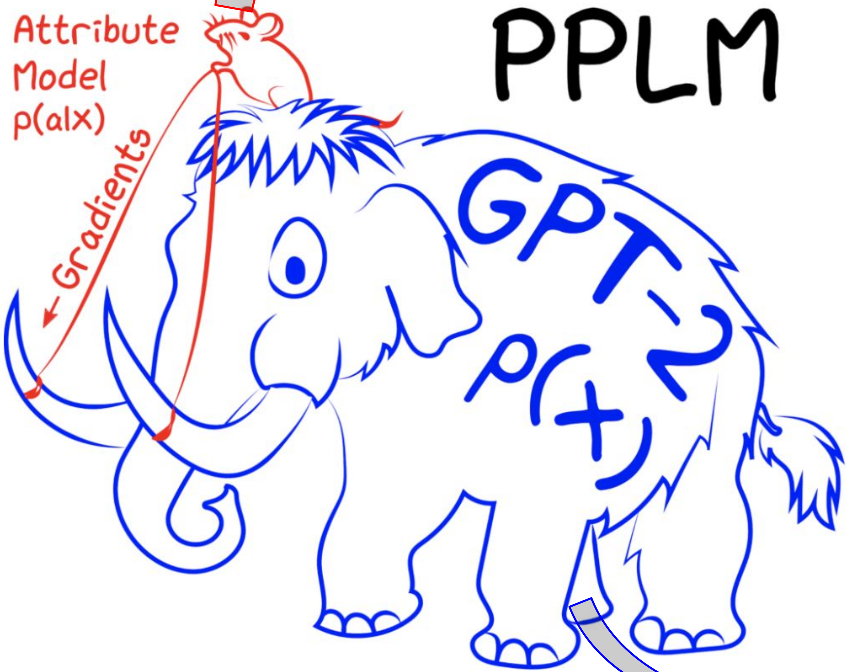


Image Courtesy of: http://jalammar.github.io/illustrated-gpt2/

Controllable Generation: GPT2 + PPLM



Bayes' Rule

$$p(x|a) \propto p(x)p(a|x)$$

Image Courtesy of: https://eng.uber.com/pplm/

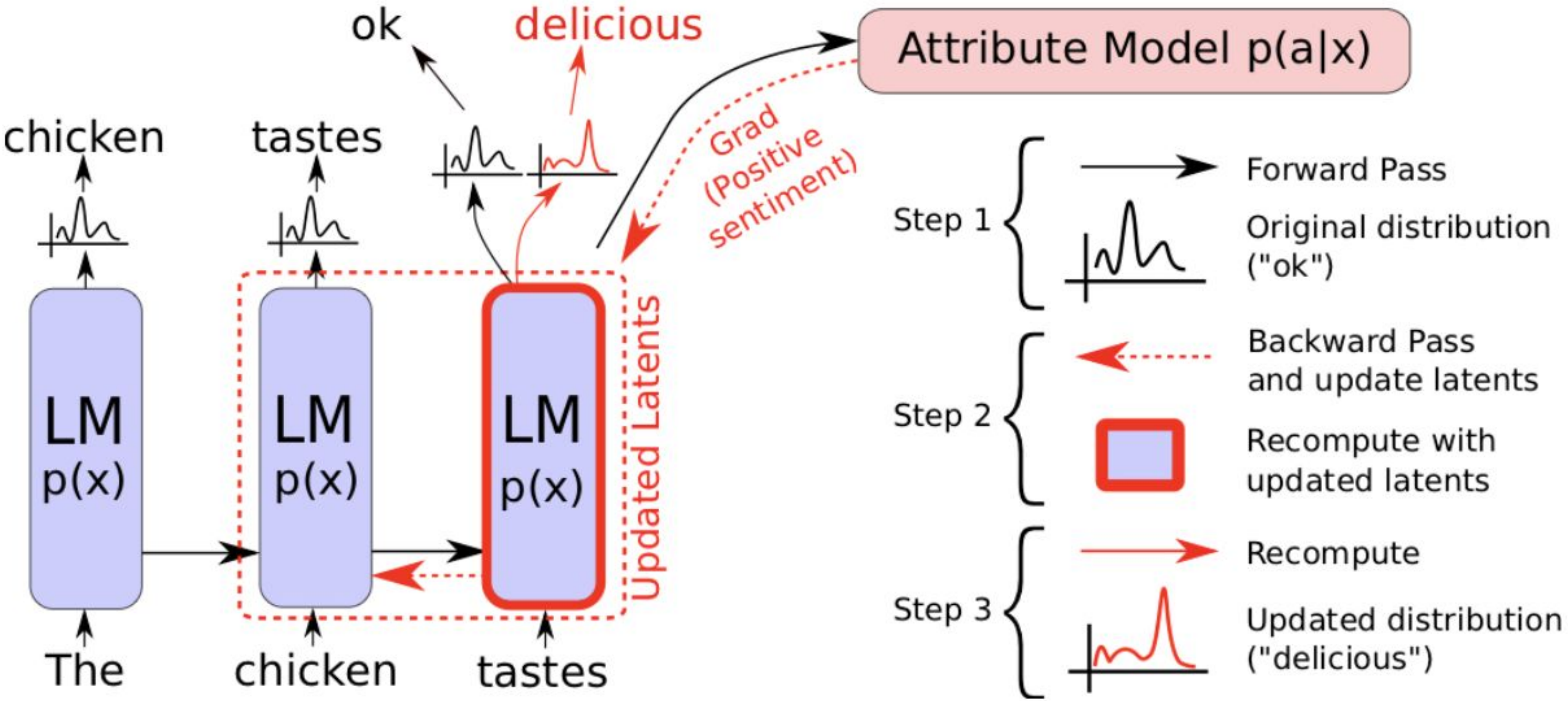
GPT2 + PPLM:

[-] <u>The chicken</u> is now out on the grill.

Image Courtesy of: https://eng.uber.com/pplm/

GPT2 + PPLM: The Three Passes

Image Courtesy of: https://eng.uber.com/pplm/



GPT2 + PPLM: Updating Gradients

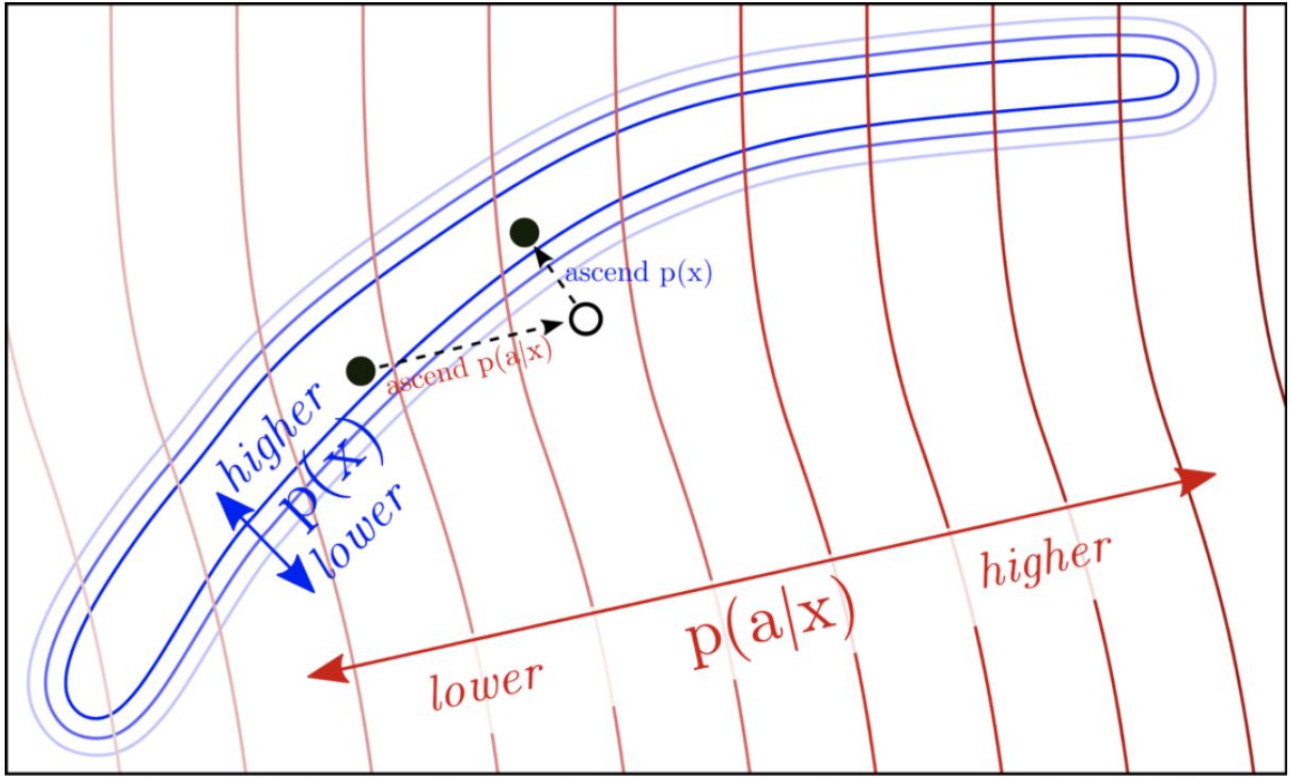
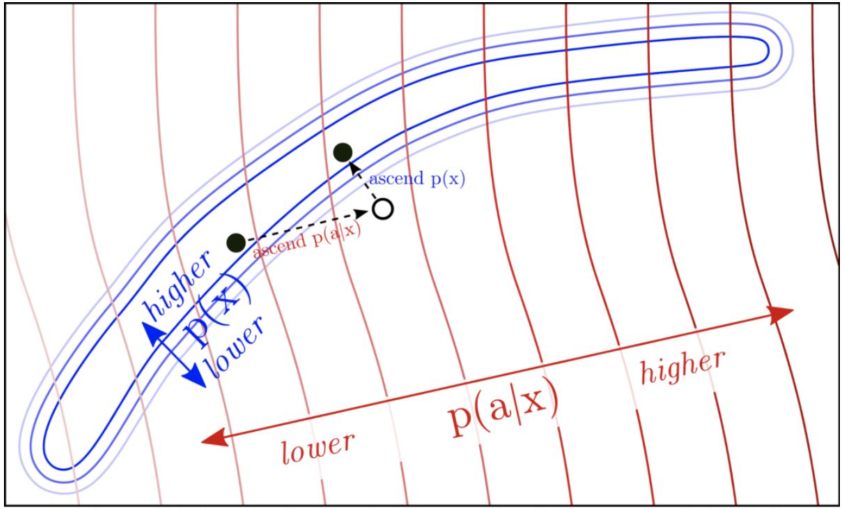


Image Courtesy of:
https://eng.uber.com/pplm/

GPT2 + PPLM: Keeping it Fluent

- Kullback–Leibler (KL) Divergence
  - Minimizes the KL divergence between the output distribution of the modified and unmodified language models
- Post-norm Geometric Mean Fusion
  - constantly ties the generated text to the unconditional p(x) LM distribution via sampling the word from the joint geometric distribution



[Dathari, 2019]

Image Courtesy of: https://eng.uber.com/pplm/

Controllable Generation: GPT2 + PPLM

[-] The chicken is now out on the grill.

[Positive] The chicken was delicious – wonderfully moist, perfectly delicious, superbly fresh – and perfectly cooked. The only thing to say is that the sauce was excellent, and I think that the broth really complemented all of the other flavors. The best part was the sauce…

Image Courtesy of: https://eng.uber.com/pplm/

Questions?

Susana Benavidez
Andrew Kirjner
Nick Seay
Mentor: Sina Semnani

Citations

Jay Alammar (2019, August 12). The Illustrated GPT-2 (Visualizing Transformer Language Models). Retrieved from http://jalammar.github.io/illustrated-gpt2/

Sumanth Dathathri, Andrea Madotto, Piero Molino, Jason Yosinski, & Rosanne Liu. (2019, December 11). Controlling Text Generation with Plug and Play Language Models. Retrieved from https://eng.uber.com/pplm/

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, & Rosanne Liu. (2019). Plug and Play Language Models: A Simple Approach to Controlled Text Generation.

Shrimai Prabhumoye, Alan W Black, & Ruslan Salakhutdinov. (2020). Exploring Controllable Text Generation Techniques.

Abigail See, Stephen Roller, Douwe Kiela, & Jason Weston. (2019). What makes a good conversation? How controllable attributes affect human judgments.