GAMMT: Generative Ambiguity Modeling Using Multiple Transformers

Xingcheng Xu

November 2022

Outline

1. Introduction

2. The Transformer

3. Generative Ambiguity Modeling

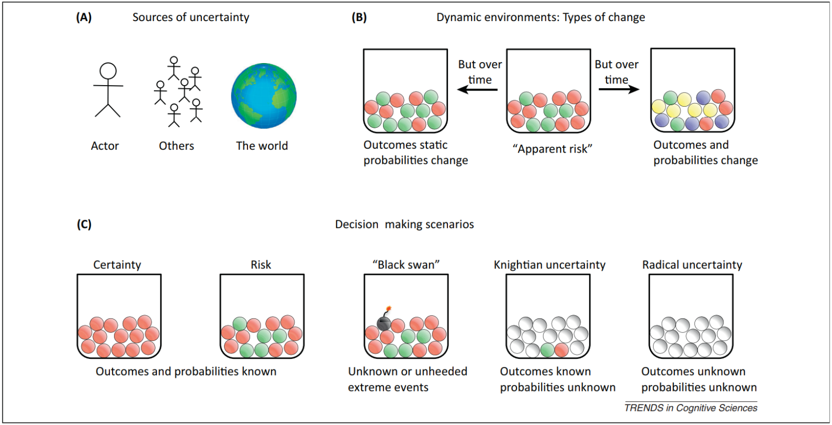
4. Deep Ambiguity Sequence Model: GAMMT

Introduction

Questions

1. What is uncertainty(risk/ambiguity/...)? How much important is it?
2. Where are the sources of ambiguity from? How to understand and model ambiguity?
3. What benefits can we get from recognizing the presence of ambiguity with respect to the pure risk situation?

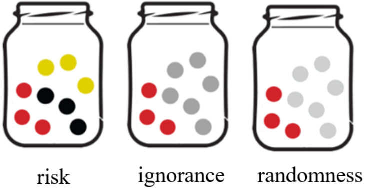
Uncertainty


(A)
Sources of uncertainty
Actor
Others
The world
(B)
Dynamic environments: Types of change
But over time
But over time
Outcomes static probabilities change
"Apparent risk"
Outcomes and probabilities change
(C)
Decision making scenarios
Certainty
Risk
"Black swan"
Knightian uncertainty
Radical uncertainty
Outcomes and probabilities known
Unknown or unheeded extreme events
Outcomes known probabilities unknown
Outcomes unknown probabilities unknown
TRENDS in Cognitive Sciences

Source: Meder et al. (2013), Trends in Cognitive Sciences.

Examples of Uncertainty

- Urn 1: 30 red, 30 black and 30 yellow balls (*risk/deterministic probability*)
- Urn 2: 30 red balls and 60 black or yellow balls (*subjective ignorance*)
- Urn 3: 90 balls, 30 in red but the number of black or yellow balls are uniformly distributed in this magic urn (*objective randomness*)

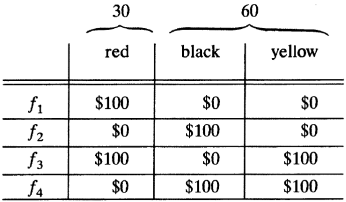


Risk and Ambiguity

- Risk refers to situations where the perceived likelihoods of events of interest can be represented by a probability measure.
- Ambiguity refers to situations where the information available to the decision-maker is too imprecise to be summarized by a probability measure (*subjective ignorance*) or results from genuine randomness in the external environment (*objective randomness*).

Ambiguity and Decision-Making

- **Ellsberg Paradox:** an urn containing 90 balls, identical except for color. You know that exactly 30 of the balls are red. Each of the remaining 60 balls is either black or yellow, but you do not know the relative numbers of black and yellow balls.[1]



	30	60	
	red	black	yellow
f_1	$100	$0	$0
f_2	$0	$100	$0
f_3	$100	$0	$100
f_4	$0	$100	$100

- **Preference:** empirically $f_1 \succ f_2$ but $f_4 \succ f_3$, which contradicts with the risk-based models.

[1] Ellsberg (1961), Machina and Schmeidler (1992)

Ambiguity and Decision-Making

- Ellsberg Paradox demonstrated that the choices under ambiguity are distinct from risk and such a distinction is *behaviorally meaningful*, and suggests that ambiguity is at least as prominent as risk in making investment decisions.
- An agent using the wrong probability measure may plausibly be aware of this possibility and thus be led to seek robust decisions. Such self-awareness and a desire for robust decisions lead naturally to consideration of sets of priors.
- Different agent groups use different probability measures to generate contents or make decisions, leading naturally to consideration of sets of probabilities.

Language, Image, ...

Generated by people (decision) or environment (semantic ambiguity, perceptual vagueness)

- incomplete knowledge, limited information (feeling, perception, experiences)
- interaction with heterogeneous agents
- genuine randomness in the external environment

Literature

- Nature Language Processing:
  - Franz. (1996). Automatic ambiguity resolution in natural language processing: an empirical approach. Berlin, Heidelberg: Springer.
  - Anjali & Babu (2014). Ambiguities in natural language processing. International Journal of Innovative Research in Computer and Communication Engineering.
  - Patel & Nenkova (2019). Modeling ambiguity in text: A corpus of legal literature.
  - Jackson (2020). Understanding understanding and ambiguity in natural language. Procedia Computer Science.
  - Yadav, Patel & Shah (2021). A comprehensive review on resolving ambiguities in natural language processing. AI Open.

Literature

- Computer Vision:
  - Casadei & Mitter (1999). Beyond the uniqueness assumption: Ambiguity representation and redundancy elimination in the computation of a covering sample of salient contour cycles. Computer Vision and Image Understanding.
  - Rust & Stocker (2010). Ambiguity and invariance: two fundamental challenges for visual processing. Current opinion in neurobiology.
  - Rupprecht et al. (2017). Learning in an uncertain world: Representing ambiguity through multiple hypotheses. ICCV.
  - Manhardt et al. (2019). Explaining the ambiguity of object detection and 6d pose from visual data. ICCV.
  - Yang et al. (2021). Probabilistic modeling of semantic ambiguity for scene graph generation. CVPR.

Literature

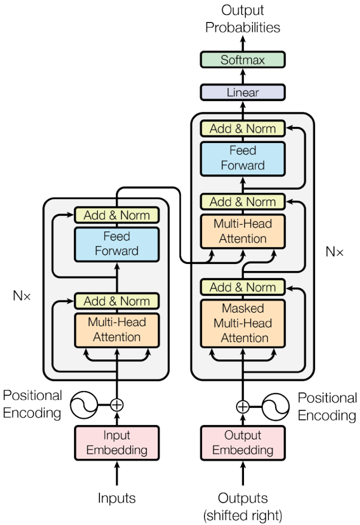
- Science:
  - Borodovsky & McIninch (1993). Recognition of genes in DNA sequence with ambiguities. Biosystems.
  - Postic et al. (2017). An ambiguity principle for assigning protein structural domains. Science advances.
  - Barbieri (2019). Evolution of the genetic code: The ambiguity-reduction theory. Biosystems.
  - Oliva et al. (2019). Accounting for ambiguity in ancestral sequence reconstruction. Bioinformatics.
- NeuroScience:
  - Levy et al. (2010). Neural representation of subjective value under risk and ambiguity. Journal of neurophysiology.
  - Bach et al. (2011). The known unknowns: neural representation of second-order uncertainty, and ambiguity. Journal of Neuroscience.
  - Chumbley et al. (2012). Learning and generalization under ambiguity: an fMRI study. PLoS Computational Biology.

The Transformer

Transformers

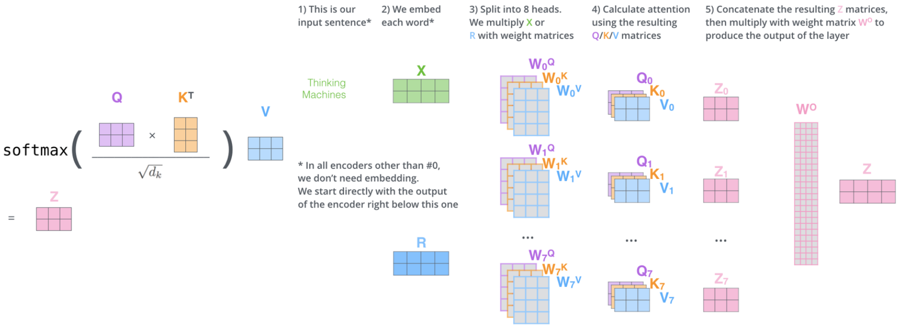
- The landmark paper:
  - The Transformer (Vaswani et al., 2017)
- Nature Language Processing:
  - BERT (Devlin et al., 2018),
  - OpenAI GPT (Radford et al., 2018), GPT-2 (Radford et al., 2019), GPT-3 (Brown et al., 2020),
  - Google T5 (Raffel et al., 2019), ...
- Computer Vision:
  - Vision Transformer (Dosovitskiy et al., 2020), ...
- Science: Structural Biology
  - AlphaFold 2 (Jumper et al., 2021), ...
- AIGC:
  - CLIP (Radford et al., 2021),
  - DALL-E 2 (Ramesh et al., 2022), Stable Diffusion (Rombach et al., 2022), eDiff-I (Balaji et al., 2022), ...

The Transformer: a Probability Machine



Source: Vaswani et al. (2017)

The Transformer: a Probability Machine



1) This is our input sentence*

2) We embed each word*

3) Split into 8 heads. We multiply X or R with weight matrices

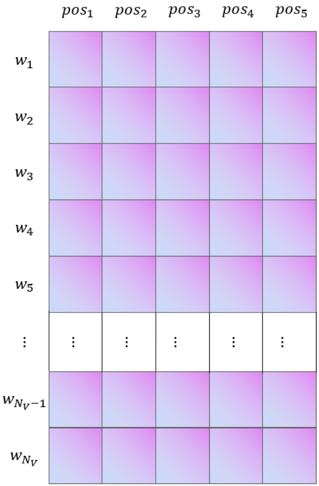
4) Calculate attention using the resulting Q/K/V matrices

5) Concatenate the resulting Z matrices, then multiply with weight matrix W^O to produce the output of the layer

* In all encoders other than #0, we don't need embedding. We start directly with the output of the encoder right below this one

Source: Alammar (2018), The Illustrated Transformer

The Transformer: a Probability Machine



$P(s_t|s_1, s_2, \cdots, s_{t-1})$

Generative Ambiguity Modeling

Conventional Wisdom in AI

Standard approaches in machine learning for sequential modeling such as language models are to factorize the joint probabilities $P(x)$ of a sequence $x = (s_1, s_2, \cdots, s_n) \in V^*$ in a space of sequences $V^* = \cup_{\ell=0}^{\infty} V^\ell$ (V is a vocabulary of tokens) as the product of conditional probabilities (e.g. GPT, GPT-2, GPT-3)

$$P(x) = \prod_{t=1}^{n} P(s_t | s_1, s_2, \cdots, s_{t-1}) \tag{1}$$

or to factorize the conditional probabilities $P(x|z)$ of a pair of sequences $x = (s_1, s_2, \cdots, s_n) \in V^*$ and $z \in V^*$ as the product of following conditionals (e.g. the original Transformer)

$$P(x|z) = \prod_{t=1}^{n} P(s_t | s_1, s_2, \cdots, s_{t-1}, z). \tag{2}$$

Sequence With Ambiguity

- $(\Omega, \{\mathcal{G}_i\}_{i=1}^{\infty}, \mathcal{G}) = (\prod_{i=1}^{\infty} \Omega_i, \{\mathcal{G}_i\}_{i=1}^{\infty}, \sigma(\cup_{i=1}^{\infty} \mathcal{G}_i))$ a filtered space modeling a sequence of experiments/games/events/objects.
- The set of possible outcomes for the i^{th} experiment is Ω_i. $\mathcal{G}_i$ is a σ-algebra on $\prod_{j=1}^{i} \Omega_j$ representing the information regarding experiments $1, 2, \cdots, i$.
- The ex ante probabilities of experiments are not known precisely and are represented by a set $\mathscr{P}$ of probability measures on $(\Omega, \mathcal{G})$, and assume that all measures in $\mathscr{P}$ are equivalent on each $\mathcal{G}_n$.

Probability Structure: Rectangularity

- $\mathscr{P}_{0,n} := \{P_{|\mathcal{G}_n} : P \in \mathscr{P}\}$, $\Omega = \Omega^{(n)} \times \Omega_{(n+1)} = \prod_{i=1}^{n} \Omega_i \times \prod_{i=n+1}^{\infty} \Omega_i$
- Probability kernel: a functional $\lambda : \Omega^{(n)} \times \mathcal{G}_{(n+1)} \to [0,1]$ satisfying $\mathcal{G}_n$-measurable and being a probability measure on $(\Omega_{(n+1)}, \mathcal{G}_{(n+1)})$.
- $\mathscr{P}$-kernel: If $\forall \omega^{(n)} \in \Omega^{(n)}$, $\exists Q \in \mathscr{P}$ satisfying

$$\lambda(\omega^{(n)}, A) = Q(\Omega^{(n)} \times A | \mathcal{G}_n)(\omega^{(n)}), \ \forall A \in \mathcal{G}_{(n+1)}.$$

- $\mathscr{P}$ is *rectangular* if $\forall n \in \mathbb{N}$, $\forall p_n \in \mathscr{P}_{0,n}$ and for every $\mathscr{P}$-kernel λ, if P is defined as

$$P(A) := \int I_A(\omega)\lambda(\omega^{(n)}, d\omega_{(n+1)})p_n(d\omega^{(n)}), \ \forall A \in \mathcal{G}, \qquad (3)$$

  then $P \in \mathscr{P}$.
- $\mathscr{P}$ is closed w.r.t. pasting of alien marginals and conditionals, endowing $\mathscr{P}$ with a recursive structure.

Example of Rectangularity: IID Model

- Experiments have a common set of possible outcomes $\overline{\Omega}$ and a common σ-algebra $\overline{\mathcal{F}}$.
- Fix a subset $\mathcal{L} \subset \mathscr{M}(\overline{\Omega}, \overline{\mathcal{F}})$ & all measures in $\mathcal{L}$ are equivalent.
- Let $P_{n,n+1}(\omega^{(n)})$ denote the restriction to $\mathcal{G}_{n+1}$ of $P(\cdot|\mathcal{G}_n)(\omega^{(n)})$.
- IID (**I**ndistinguishably and **I**ndependently **D**istributed) model:

$$\mathscr{P}^{IID} = \left\{ P \in \mathscr{M}(\Omega, \mathcal{G}) : \ P_{n,n+1}(\omega^{(n)}) \in \mathcal{L}, \ \forall n, \omega^{(n)} \in \Omega^{(n)} \right\}$$

- The set consists of all measures whose *one-step-ahead conditionals*, at every history, lie in $\mathcal{L}$ modeling partial ignorance about each experiment separately.
- If $\mathcal{L} = \{P\}$, $\mathscr{P}^{IID}$ degenerates to a probability (the conventional probability model, random walk).

Example of IID Model

- Each experiment can produce one of the three outcomes (3 words in a vocabulary): $\overline{\Omega} = \{w_1, w_2, w_3\}$ and $\overline{\mathcal{F}} = 2^{\overline{\Omega}}$.
- Probabilities are not known exactly but it is known that, for each experiment, the outcomes are given by $(0 < q < p,\ p + q \leq 1)$

$$\mathcal{L} = \{P_1 = (p, q, 1 - p - q), P_2 = (q, p, 1 - p - q)\}.$$

- The ignorance about the relation between experiments is subject to the IID model $\mathscr{P}^{IID}$.

Deep Ambiguity Sequence Model: GAMMT

Deep Ambiguity Sequence Model: GAMMT

The GAMMT: An Ambiguity Machine

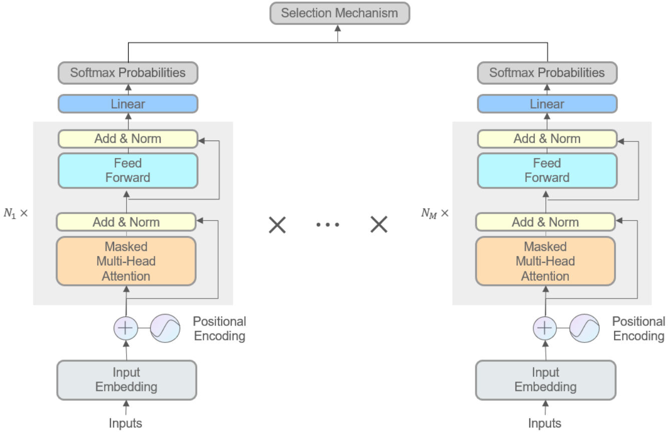
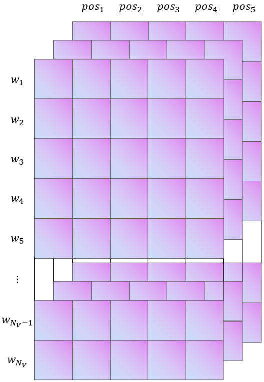


Figure: Model architecture - The Transformer decoders to approximate a set of probabilities

Deep Ambiguity Sequence Model: GAMMT

There is a set of *one-step-ahead conditional probabilities* produced by the GAMMT model at every time step/every position of a sequence. The size of the set characterizes the level of ambiguity of data.

GAMMT: $\mathcal{P}(x) = \{P(x) |\ P \in \mathscr{M}\}$, $\mathscr{M} = \{P_{\theta_1}, P_{\theta_2}, \cdots, P_{\theta_M}\}$



$P_{\theta_m}(s_t|s_1, s_2, \cdots, s_{t-1}), m \in [M]$

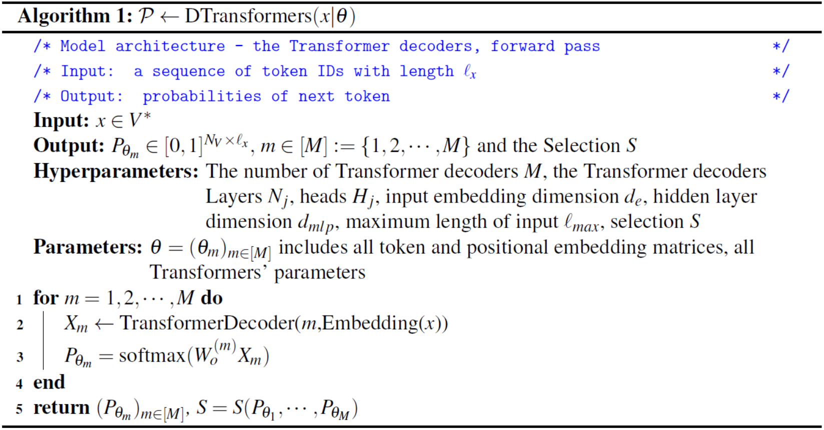
Deep Ambiguity Sequence Model: GAMMT

The model described above can be generalised to the encoder-decoder architectures with Transformers to approximate sets of conditional probabilities

$$\mathcal{P}(x|z) = \{P(x|z)|\ P \in \mathscr{M}\}$$

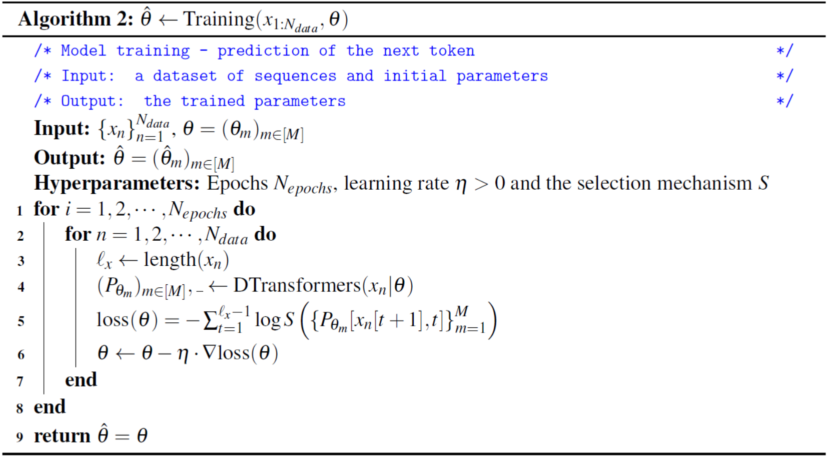
where z and x are a pair of input (source) or output (target) token sequences as the original Transformer/Google T5. (Could share the same encoder, or every parallel transformer has its only encoder.)

Algorithm: Architecture



```
Algorithm 1: $\mathcal{P} \leftarrow \mathrm{DTransformers}(x|\theta)$
  /* Model architecture - the Transformer decoders, forward pass */
  /* Input: a sequence of token IDs with length $\ell_x$ */
  /* Output: probabilities of next token */
  Input: $x \in V^*$
  Output: $P_{\theta_m} \in [0,1]^{N_V \times \ell_x}$, $m \in [M] := \{1,2,\cdots,M\}$ and the Selection $S$
  Hyperparameters: The number of Transformer decoders $M$, the Transformer decoders
                   Layers $N_j$, heads $H_j$, input embedding dimension $d_e$, hidden layer
                   dimension $d_{mlp}$, maximum length of input $\ell_{max}$, selection $S$
  Parameters: $\theta = (\theta_m)_{m\in[M]}$ includes all token and positional embedding matrices, all
              Transformers' parameters
for $m = 1,2,\cdots,M$ do
    $X_m \leftarrow \mathrm{TransformerDecoder}(m,\mathrm{Embedding}(x))$
    $P_{\theta_m} = \mathrm{softmax}(W_o^{(m)} X_m)$
end
return $(P_{\theta_m})_{m\in[M]}$, $S = S(P_{\theta_1},\cdots,P_{\theta_M})$
```

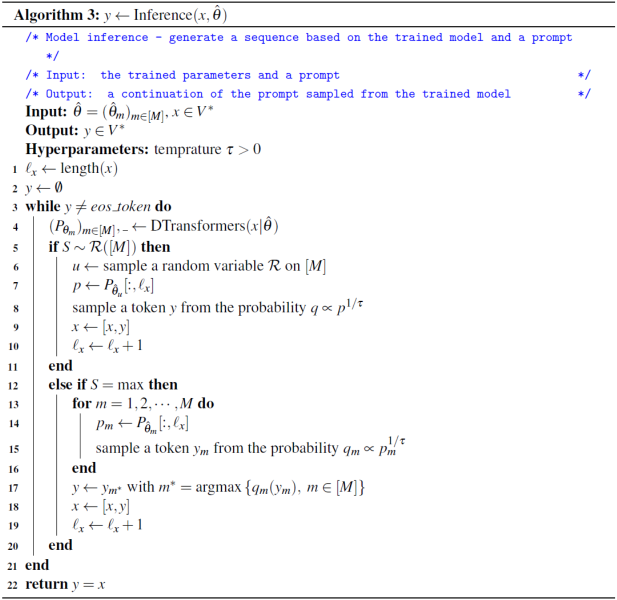
Algorithm: Training



Algorithm 2: $\hat{\theta} \leftarrow \mathrm{Training}(x_{1:N_{data}}, \theta)$

```
/* Model training - prediction of the next token                */
/* Input:  a dataset of sequences and initial parameters        */
/* Output:  the trained parameters                              */
Input: $\{x_n\}_{n=1}^{N_{data}}$, $\theta = (\theta_m)_{m \in [M]}$
Output: $\hat{\theta} = (\hat{\theta}_m)_{m \in [M]}$
Hyperparameters: Epochs $N_{epochs}$, learning rate $\eta > 0$ and the selection mechanism $S$
for $i = 1, 2, \cdots, N_{epochs}$ do
    for $n = 1, 2, \cdots, N_{data}$ do
        $\ell_x \leftarrow \mathrm{length}(x_n)$
        $(P_{\theta_m})_{m \in [M]}, \_ \leftarrow \mathrm{DTransformers}(x_n | \theta)$
        $\mathrm{loss}(\theta) = -\sum_{t=1}^{\ell_x - 1} \log S\left(\{P_{\theta_m}[x_n[t+1], t]\}_{m=1}^{M}\right)$
        $\theta \leftarrow \theta - \eta \cdot \nabla \mathrm{loss}(\theta)$
    end
end
return $\hat{\theta} = \theta$
```

Algorithm: Inference



Algorithm 3: $y \leftarrow \text{Inference}(x, \hat{\theta})$

```
/* Model inference - generate a sequence based on the trained model and a prompt
   */
/* Input:  the trained parameters and a prompt                                   */
/* Output:  a continuation of the prompt sampled from the trained model          */
Input: $\hat{\theta} = (\hat{\theta}_m)_{m \in [M]}, x \in V^*$
Output: $y \in V^*$
Hyperparameters: temprature $\tau > 0$
$\ell_x \leftarrow \text{length}(x)$
$y \leftarrow \emptyset$
while $y \neq eos\_token$ do
    $(P_{\theta_m})_{m \in [M]}, \_ \leftarrow \text{DTransformers}(x|\hat{\theta})$
    if $S \sim \mathcal{R}([M])$ then
        $u \leftarrow$ sample a random variable $\mathcal{R}$ on $[M]$
        $p \leftarrow P_{\hat{\theta}_u}[:, \ell_x]$
        sample a token $y$ from the probability $q \propto p^{1/\tau}$
        $x \leftarrow [x, y]$
        $\ell_x \leftarrow \ell_x + 1$
    end
    else if $S = \max$ then
        for $m = 1, 2, \cdots, M$ do
            $p_m \leftarrow P_{\hat{\theta}_m}[:, \ell_x]$
            sample a token $y_m$ from the probability $q_m \propto p_m^{1/\tau}$
        end
        $y \leftarrow y_{m^*}$ with $m^* = \text{argmax}\,\{q_m(y_m),\ m \in [M]\}$
        $x \leftarrow [x, y]$
        $\ell_x \leftarrow \ell_x + 1$
    end
end
return $y = x$
```

Validation on NLP Tasks

- Dataset
  - CommonCrawl
  - WebText
  - Books1, Books2
  - Wikipedia
- Tasks
  - Machine translation
  - Semantic similarity
  - Text classification
  - Reading comprehension, summarization
  - Natural language inference, textual entailment
  - Commonsense reasoning, Question answering
  - Cloze tasks, Sentence/paragraph completion tasks
  - Article generation

Ref. GPT, GPT-2, GPT-3, T5, ...

More

- Computer Vision: image as patch token sequence
- Audio Generation
- Protein Structure Prediction
- DNA/RNA Structure Prediction
- Reinforcement Learning
- ...

Concluding Remarks

- long range dependencies (attention)
- zero-shot learning (scale)
- admitting ambiguity (structure)
- high diversity of generation (structure)
- multiple representations (structure)
- parallel computing, high computational efficiency (structure)
- ...

Thanks!