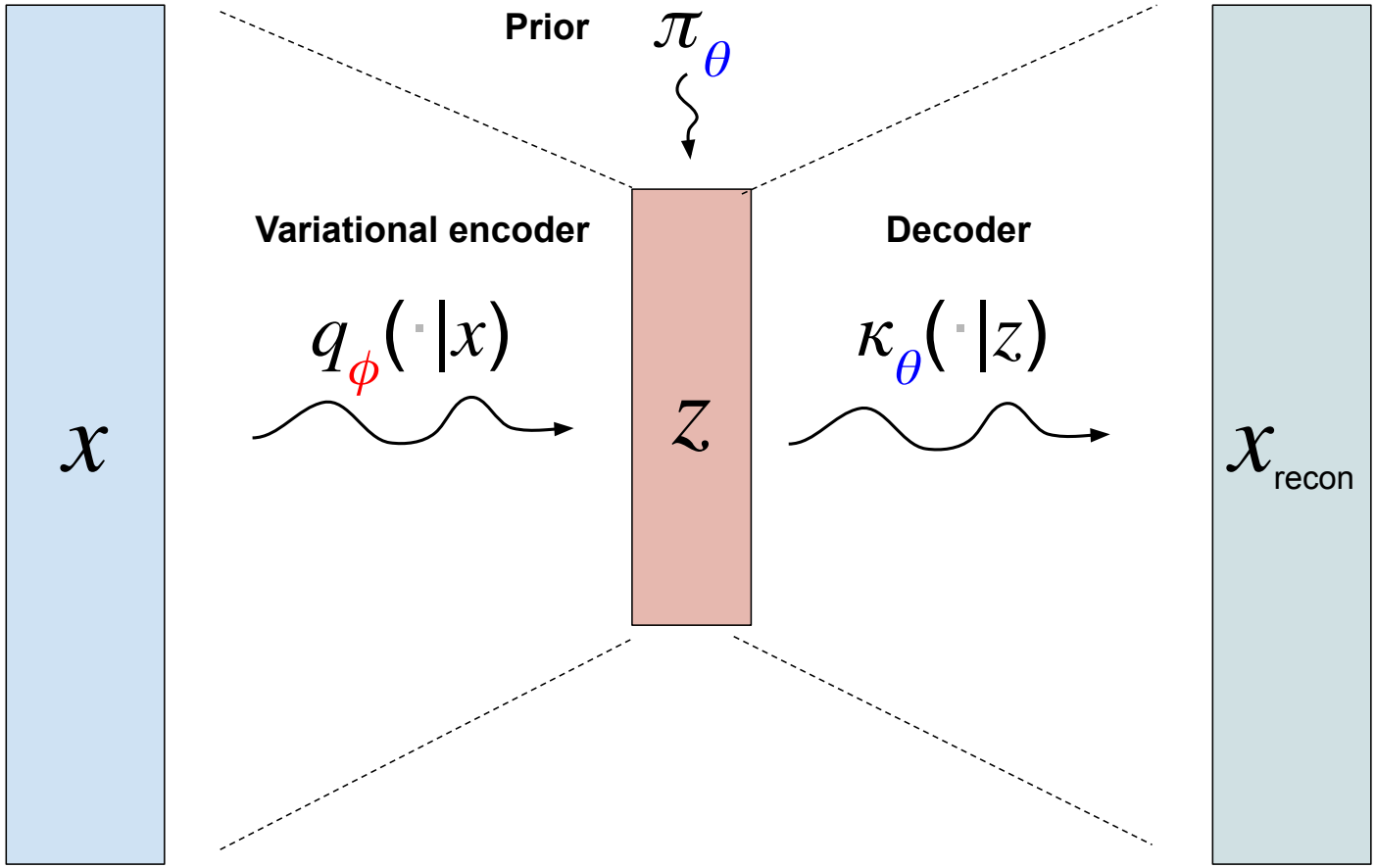
Lectures on Natural Language Processing

14. Diffusion Models, Coreference Resolution, Review

Karl Stratos

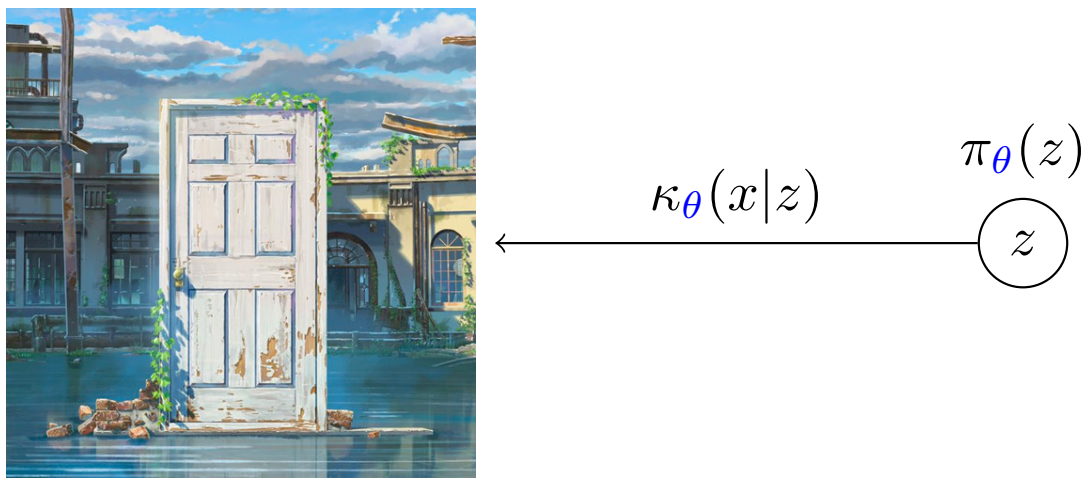
Review: Variational Autoencoders (VAEs)



$$\max_{\theta,\,\phi} \underbrace{\mathop{\mathbf{E}}_{z\sim q_\phi(\cdot|x)} [\log \kappa_\theta(x|z)]}_{\text{reconstruction}} - \beta \underbrace{\mathrm{KL}(q_\phi(\cdot|x)||\pi_\theta)}_{\text{regularization}}$$

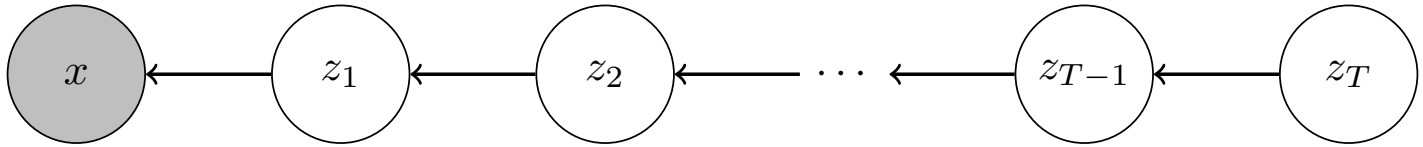
Latent-Variable Generative Models in Vision

- LVGMs have *not* been very popular in NLP.
  - Language is "already symbolic". Not much to compress.
  - Direct language modeling is the king.
- They are more successful in computer vision.
  - Images have a much lower signal-to-noise ratio.
  - The concept of generating high-dimensional observations from low-dimensional latents makes more sense.



- Today's go-to image/conditional image generator: **diffusion models**

Diffusion Model as a Markov VAE

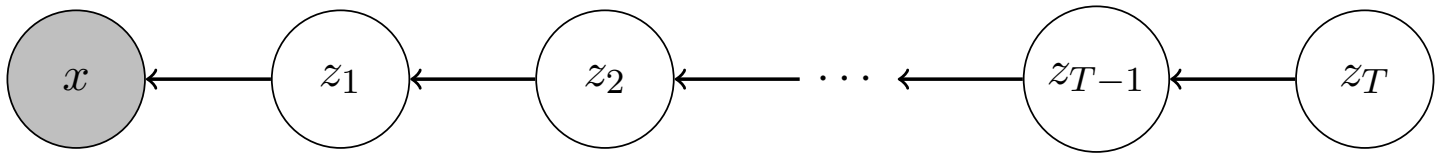


- The latent is a *sequence* of continuous vectors $z_1 \dots z_T \in \mathbb{R}^d$ same size as input image $x \in \mathbb{R}^d$.
- Idea: Start from a **completely random noise** $z_T \in \mathbb{R}^d$, generate "backward" stepwise refinements $z_t \mapsto z_{t-1}$ until $z_0 = x$.

(Image credit: Kreis, Guao, and Vahdat (2022))

- Much sharper image than a standard single-step VAE, albeit at a slower generation speed
  - Need to run the model many times (e.g., $T = 1000$) to generate a single image.

Markov Gaussian Generator



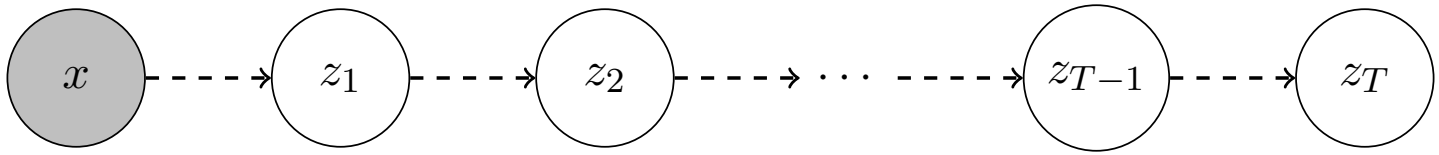
$$p_\theta(x, z_1 \dots z_T) = \prod_{t=1}^{T+1} \overleftarrow{p}_\theta(z_{t-1}|z_t, t)$$

The stepwise generator parameterized as Gaussian

$$\overleftarrow{p}_\theta(z_T|z_{T+1}, T+1) = \mathcal{N}(0_d,\ I_{d\times d})(z_T)$$
$$\overleftarrow{p}_\theta(z_{t-1}|z_t, t) = \mathcal{N}(\overleftarrow{\mu}_\theta(z_t, t),\ \sigma_t^2 I_{d\times d})(z_{t-1})$$

$\sigma_T^2 > \ldots > \sigma_1^2 > 0$ some decreasing variance schedule

Markov Gaussian Approximate Posterior



$$q(z_1 \dots z_T|x) = \prod_{t=1}^{T} \overrightarrow{q}(z_t|z_{t-1}, t)$$

The stepwise encoder also parameterized as Gaussian

$$\overrightarrow{q}(z_t|z_{t-1}, t) = \mathcal{N}(\sqrt{1-\beta_t} z_{t-1},\ \beta_t I_{d\times d})$$

$0 < \beta_1 < \cdots < \beta_T < 1$ some increasing variance schedule

Simplifying the ELBO by the Markov Assumption

Maximize the ELBO (using a fixed variational posterior q)

$$\log p_\theta(x) \geq \underbrace{\mathop{\mathbf{E}}_{z\sim q(\cdot|x)} [\log \kappa_\theta(x|z)]}_{①} - \underbrace{\mathrm{KL}\left(q(\cdot|x)||\pi_\theta\right)}_{②}$$

Under the **Markov assumptions**,

$$① = \mathop{\mathbf{E}}_{z_1\sim \overrightarrow{q}(\cdot|x,1)} [\log \overleftarrow{p}_\theta(x|z_1,1)]$$

$$② = \mathop{\mathbf{E}}_{z_1 \ldots z_T\sim q(\cdot|x)} \left[\sum_{t=2}^{T+1} \mathrm{KL}\left(\overleftarrow{q}(\cdot|x,z_t)||\overleftarrow{p}_\theta(\cdot|z_t,t)\right) \right]$$

Simplifying the ELBO by the Gaussian Parameterization

Under the **Gaussian parameterization**, the objective can be further simplified to a remarkable degree:

$$\min_{\theta} \mathop{\mathbf{E}}_{z_1 \dots z_T \sim q(\cdot|x)} \left[\sum_{t=1}^{T} \frac{1}{\sigma_t^2} \left|\left| \tilde{\mu}(x, z_t, t) - \overleftarrow{\mu}_\theta(z_t, t) \right|\right|^2 \right]$$

where $\tilde{\mu}(x, z_t, t) \in \mathbb{R}^d$ is the mean of z_{t-1} given x and z_t under the approximate posterior, *computable in closed form*!

- Thus the network $\mu_\theta \approx \tilde{\mu}$ is trained for stepwise denoising

Furthermore, we can choose to parameterize the mean predictor as a function of a "noise predictor" ϵ_θ

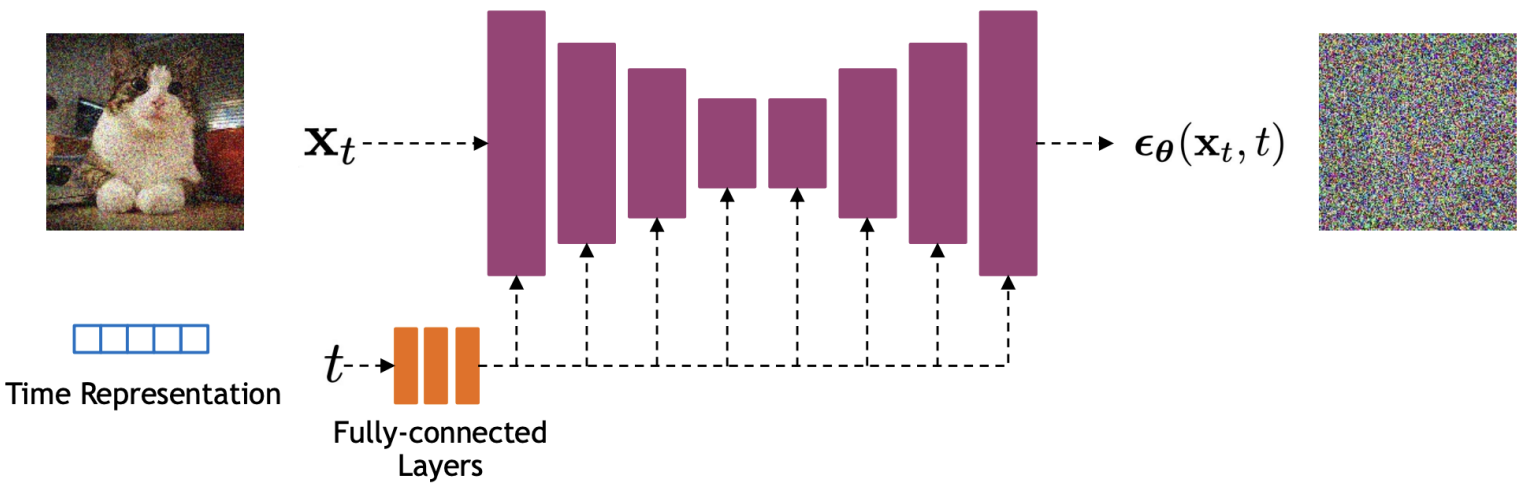
$$\overleftarrow{\mu}_\theta(z_t, t) := \sqrt{\frac{1}{1-\beta_t}} \left(z_t - \frac{\beta_t}{\sqrt{1 - \prod_{s \le t}(1-\beta_s)}} \epsilon_\theta(z_t, t) \right)$$

Noise Predictive Training

Sample image x, step $t \in \{1 \dots T\}$, noise $\epsilon \sim \mathcal{N}(0_d, I_{d \times d})$:

$$\min_{\theta} \left|\left| \epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t} x + \sqrt{1 - \bar{\alpha}_t}\epsilon, t) \right|\right|^2$$

where $\bar{\alpha}_t = \prod_{s \le t}(1 - \beta_s)$. Noise predictor $\epsilon_\theta(\tilde{x}, t)$ often U-Net



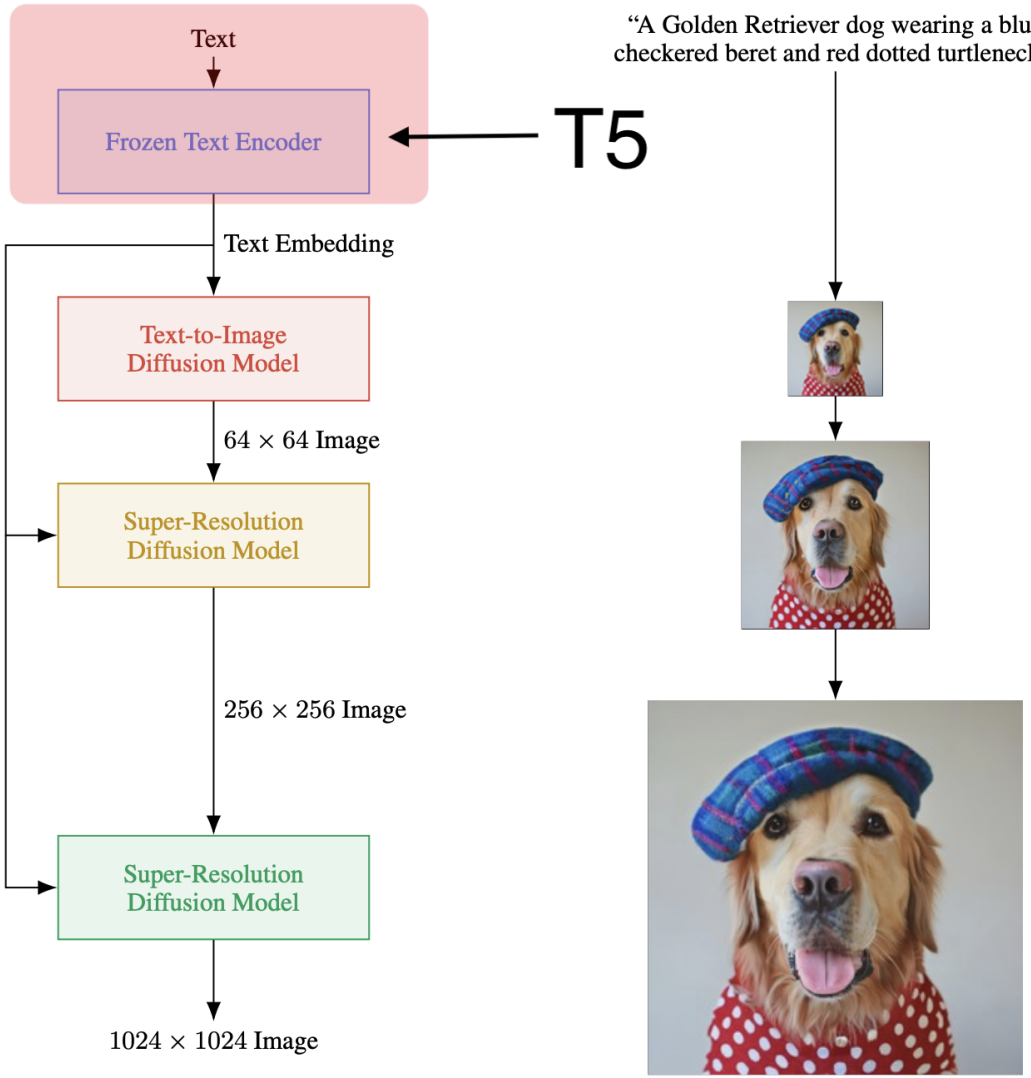
(Image credit: Kreis, Guao, and Vahdat (2022))

After training, noise reverse-engineered to produce image

Generated Images

- Denoising Diffusion Probabilistic Models (Ho et al., 2020)
- Generated samples on CelebA-HQ 256×256

Conditional Image Generation



- Example: Imagen (Saharia et al., 2022)
- Cascaded diffusion model conditioning on (fixed) text embeddings from **pretrained T5**
- Leverages the deep language understanding capabilities in LMs

Text-to-Image Demo

Prompt: "heavy rain in the city at night, people with umbrellas"

Demo available at: Stable Diffusion

Coreference Resolution (Coref)

$$\textbf{Input} = \begin{pmatrix} \textit{“I voted for Nader because he was most aligned with my} \\ \textit{values,” she said.} \end{pmatrix}$$

$$\textbf{Output} = \{\{\textit{Nader}, \textit{he}\}, \{\textit{I}, \textit{my}, \textit{she}\}\}$$

- Related, but different from entity linking
  - Typically no KB: Must infer new entities dynamically without grounding to a KB
  - Considers a wide range of mention types like pronouns and verbs as well as noun phrases
  - Can be long-range: A mention at the end of a document may refer to the first sentence
- Not an end-task itself
  - Pretrained LMs (seem to) solve language tasks that require coref without explicit coref training (e.g., Winograd)
  - Nevertheless important and difficult problem, with obvious applications in text analysis

Types of Coreference

- **Anaphora**. A later mention (anaphor) refers to an earlier mention (its antecedent). This is standard coref
  - *The music was so loud that it couldn't be enjoyed.*
- **Cataphora**. An earlier mention (cataphor) refers to a later mention (its postcendent)
  - *If they are angry about the music, the neighbors will call the cops.*
- **Split antecedents**. An anaphor refers to split antecedents
  - *Carol told Bob to attend the party. They arrived together.*
- **Apositives**. Consecutive noun phrases renaming each other
  - *Little Davey, my youngest nephew, is feeling sick.*

(And more.) Complex linguistic phenomenon, heavily language-specific

- English: Pronoun *it* may refer to nothing (e.g., *it takes a lot of work to succeed*)

Labeled Data for Coref

- Annotation challenging even for humans, low inter-annotator agreement
- Current go-to dataset: OntoNotes (Pradhan et al., 2012)
  - Document-level coref annotation from the CoNLL-2012 shared task: Also includes Chinese and Arabic
  - 2802, 343, 348 train/dev/test documents (1 million words)
  - Varying document lengths: From 454 to 4009 words in train
  - Text from newswire, magazine, broadcast news/conversations, web, conversational speech, New Testament
  - No single-mention (singleton) entity labeled
- Referring mentions can be nested or overlapping
  - *But when [you]$_1$ pray, [you]$_1$ should go into [[your]$_1$ room]$_{23}$ and close the door.*
- Another challenge: **Evaluation**
  - Given a document with ground-truth entities and predicted entities, how do we judge goodness?
  - Series of proposed metrics: MUC, B^3, CEAF, LEA

Coref Notation

- Document: Sequence of tokens $D = (x_1 \dots x_T)$
- **Entity** (aka. equivalence class) is a set of (possibly overlapping) coreferent mention spans (i, j), $1 \leq i \leq j \leq T$
- Annotation consists of **key entities** $\mathcal{S} = \{S_1 \dots S_n\}$
- System output consists of of **response entities** $\mathcal{R} = \{R_1 \dots R_{n'}\}$
- Only exact match considered for mention prediction
  - $\mathcal{S} = \{\{1, 2, 3, 4, 5\}, \{6, 7\}, \{8, 9, A, B, C\}\}$, 12 gold mentions (each index is a span) clustered into 3 key entities
  - $\mathcal{R} = \{\{1, 2, 3\}, \{6, 7, 8, 9, A, B\}\}$, 2 response entities, failed to recover gold mentions $4, 5, C$ (but might have predicted other mentions)
  - Predicted span considered correct (e.g., 9 in S_3 and R_2) iff it exactly matches a gold span, no partial credit for overlapping
- Goal: Define assymetric $\mathrm{Eval}(\mathcal{S}, \mathcal{R})$ representing **recall**
  - Flipping $\mathrm{Eval}(\mathcal{R}, \mathcal{S})$ represents **precision**
  - $\mathrm{F}_1 = 2 \times \mathrm{precision} \times \mathrm{recall} / (\mathrm{precision} + \mathrm{recall})$

MUC (Vilain et al., 1995)

- **Intersect operation.** Entity S "intersected" with $\mathcal{R}$ is a partition of S induced by response coverage

$$S = \{1, 2, 3, 4, 5\}$$
$$\mathcal{R}_1 = \{\{1, 2\}, \{4, 5, 6, 7\}\} \qquad p_{\mathcal{R}_1}(S) = \{\{1, 2\}, \{3\}, \{4, 5\}\}$$
$$\mathcal{R}_2 = \{\{1, 2, 3, 4, 5, A\}\} \qquad p_{\mathcal{R}_2}(S) = \{\{1, 2, 3, 4, 5\}\}$$

- Idea: $|p_{\mathcal{R}}(S)|$ measures fragmentation of S by $\mathcal{R}$ (smaller is better, 1 if preserved)
- **MUC**. Can be derived by counting the minimal number of additional links $\mathcal{R}$ needs to generate entities in $\mathcal{S}$ (assumes non-singleton mentions)

$$\text{Eval}(\mathcal{S}, \mathcal{R}) = \frac{\sum_{S \in \mathcal{S}} \overbrace{|S| - |p_{\mathcal{R}}(S)|}^{\text{num common links bt } S \text{ and } \mathcal{R}}}{\sum_{S \in \mathcal{S}} \underbrace{|S| - 1}_{\text{num links in } S}}$$

- Example: For $\mathcal{S} = \{\{1, 3\}\}$ and $\mathcal{R} = \{\{1, 2, 3\}\}$, recall is $\frac{2-1}{2-1} = 1$, precision is $\frac{3-2}{3-1} = \frac{1}{2}$

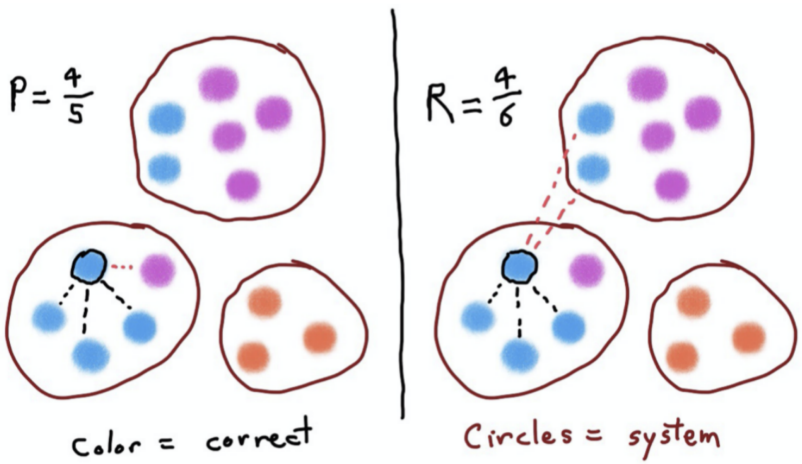
B^3 (Bagga and Baldwin, 1998)

- MUC only considers the minimal number additional links and does not differentiate types of merges

$$\mathcal{S} = \{\{1,2,3,4,5\},\{6,7\},\{8,9,A,B,C\}\}$$
$$\mathcal{R}_1 = \{\{1,2,3,4,5\},\{6,7,8,9,A,B,C\}\}$$
$$\mathcal{R}_2 = \{\{1,2,3,4,5,8,9,A,B,C\},\{6,7\}\}$$

  Both responses have recall 1 and precision 0.9 under MUC

- **B^3**. Average mention-level (not link-level) precision/recall



$$\text{Eval}(\mathcal{S},\mathcal{R}) = \frac{\sum_{S\in\mathcal{S}}\sum_{R\in\mathcal{R}}\frac{|S\cap R|^2}{|S|}}{\sum_{S\in\mathcal{S}}|S|}$$

Response 1 precision $\frac{1}{12}((5\cdot\frac{5}{5})+(2\cdot\frac{2}{7}+5\cdot\frac{5}{7}))\approx 0.76$, Response 2 precision $\frac{1}{12}((5\cdot\frac{5}{10}+5\cdot\frac{5}{10})+(2\cdot\frac{2}{2}))\approx 0.58$ (both have recall 1)

CEAF (Luo, 2005)

- MUC and B^3 "unintuitive" behavior in boundary cases

$$\mathcal{S} = \{\{1,2,3,4,5\},\{6,7\},\{8,9,A,B,C\}\}$$
$$\mathcal{R}_3 = \{\{1,2,3,4,5,6,7,8,9,A,B,C\}\}$$
$$\mathcal{R}_4 = \{\{1\},\{2\},\{3\},\{4\},\{5\},\{6\},\{7\},\{8\},\{9\},\{A\},\{B\},\{C\}\}$$

  $\mathcal{R}_3$ recall 1 (MUC & B^3) but no $S \in \mathcal{S}$ "recovered", $\mathcal{R}_4$ precision 1 (B^3, undefined for MUC) but no $R \in \mathcal{R}_4$ is "correct"

- **CEAF**. Considers optimal 1-to-1 mapping $g^* : S \mapsto R$ achieving $C^* = \max_g \sum_{S \in \mathcal{S}} \phi(S, g(S))$ (Kuhn–Munkres alg). $\phi(S, S')$ is any entity similarity measure. Defines

$$\text{Eval}_\phi(\mathcal{S},\mathcal{R}) = \frac{C^*}{\sum_{S\in\mathcal{S}} \phi(S,S)} \quad \text{Eval}_\phi(\mathcal{R},\mathcal{S}) = \frac{C^*}{\sum_{R\in\mathcal{R}} \phi(R,R)}$$

- $\mathcal{R}_3$ recall 0.2 and $\mathcal{R}_4$ precision 0.1 under CEAF_{ϕ_4} where $\phi_4(S,S') = 2\,|S \cap S'|\,/(|S| + |S'|)$

LEA (Moosavi and Strube, 2016)

- MUC least discriminative because it only considers additional links, can't handle singletons
- B^3 and CEAF found out to be uninterpretable (e.g., adding incorrect entities in $\mathcal{R}$ can *increase* the score!), mainly because mention-level
- **LEA**. Link-based like MUC but accounts for all links including self-links (can handle singletons)

$$\mathrm{Eval}_\phi(\mathcal{S},\mathcal{R}) = \frac{\sum_{S\in\mathcal{S}} \overbrace{|S|}^{\text{entithy weight}} \times \overbrace{\sum_{R\in\mathcal{R}} \frac{\binom{|S\cap R|+1}{2}}{\binom{|S|+1}{2}}}^{\text{link resolution score}}}{\sum_{S\in\mathcal{S}} |S|}$$

($\binom{n+k-1}{k}$: number of ways to choose k items out of n with replacement)

- So what's the verdict on coref evaluation?
  - Common practice: Report all MUC, B^3, CEAF_{ϕ_4} (F_1) as well as their macro-average
  - But using a single reliable metric (LEA?) would be beneficial, meaningful significance test and precision/recall

End-to-End Neural Coref

- Coref traditionally approached as a pipeline
  - Run a mention detector, learn a separate model to link detected mentions
  - Subject to the usual limitations of pipeline (error propagation, complex heuristics)
- Modern approach: **End-to-end** (mention detector just a part of the whole model, learned jointly)
- Key ideas
  1. Consider all $O(T^2)$ mentions in $D = (x_1 \dots x_T)$ as potential mentions: Number of (possibly overlapping) spans $\binom{T+1}{2} = \frac{T(T+1)}{2}$ (why?)
  2. For each mention, dynamically define a distribution over all its antecedents ordered by start index (plus end index if tied)
  3. Train the model by marginalized log likelihood (target: only the antecedents in the gold entity)
  4. Efficient training by learnable pruning

Model

- Assumes contextual mention encoder $\mathbf{enc}_\theta(D, i, j) \in \mathbb{R}^d$
  - Example: $\mathbf{enc}_\theta(D, i, j) = h_i \oplus h_j \oplus \sum_{i \le k \le j} \beta_k h_k$ where $(h_1 \dots h_T) = \mathrm{BERT}(D)$ and $\beta_i \dots \beta_j$ is an attention distribution over $h_i \dots h_j$ ("head-finding")
- Mention scorer: $\mathbf{score}^m_\theta(D, i, j) = \mathrm{FF}^1_\theta(\mathbf{enc}_\theta(D, i, j)) \in \mathbb{R}$
- Coreference scorer: Shares $\mathbf{enc}_\theta$ with mention scorer

$$\mathbf{score}^c_\theta(D, (i, j), (i', j')) = \mathrm{FF}^2_\theta \left(\begin{bmatrix} \mathbf{enc}_\theta(D, i, j) \\ \mathbf{enc}_\theta(D, i', j') \\ \mathbf{enc}_\theta(D, i, j) \odot \mathbf{enc}_\theta(D, i', j') \\ \mathbf{extra}_\theta(D, (i, j), (i, j')) \end{bmatrix} \right) \in \mathbb{R}$$

  $\mathbf{extra}_\theta$ encodes extra features (distance between mentions, if same speaker), each feature value has a learnable embedding

- Final model: If $(i, j) \neq (0, 0)$ (dummy mention, next slide),

$$\mathbf{score}_\theta(D, (i, j), (i', j')) = \mathbf{score}^m_\theta(D, i, j) + \mathbf{score}^m_\theta(D, i', j') + \mathbf{score}^c_\theta(D, (i, j), (i', j'))$$

  Otherwise 0. Interpretation: Won't link if none has positive score

Training

- Let $m_0, m_1 \ldots m_{T(T+1)/2}$ denote all (possibly overlapping) spans in document, sorted left-to-right: $m_0 = (0, 0)$ is a dummy mention
- Model defines probability of $m_{t'}$ referring to m_t where $t < t'$ by

$$p_\theta(m_t \leftarrow m_{t'}|D) = \frac{\exp(\textbf{score}_\theta(D, m_t, m_{t'}))}{\sum_{l<t'} \exp(\textbf{score}_\theta(D, m_l, m_{t'}))}$$

- Annotation doesn't give explicit links (only key entities), but we can marginalize
- For each mention $t' \in \{1 \ldots T(T+1)/2\}$, let $\textbf{Ant}(t)$ denote all $t < t'$ such that m_t and $m_{t'}$ are in the same key entity: $\{0\}$ if $m_{t'}$ is not in any key entity or is the first mention of a gold entity
- Training loss on document D

$$J_D(\theta) = -\sum_{t'=1}^{T(T+1)/2} \log \left(\sum_{t \in \textbf{Ant}(t')} p_\theta(m_t \leftarrow m_{t'}|D) \right)$$

Learnable Pruning

- Don't consider all $\frac{T(T+1)}{2}$ mentions, prune by mention scores
  - In practice, also prune by length (e.g., discard m if $|m| > 10$)
- Two-stage beam search (Lee et al., 2017)
  - Only use top $M = \lambda T$ (e.g., $\lambda = 0.4$) mentions by $\mathbf{score}_\theta^m$
  - Because $\mathbf{enc}_\theta$ is shared between scorers, pruning improves as the model improves!
  - Still too large: Input size $O(M^2)$. Additionally restrict to $\leq K$ nearest antecedents for each mention: Input size $O(MK)$
- Coarse-to-fine pruning (Lee et al., 2018) (three-stage beam search)

$$\mathbf{score}_\theta(D, m, m') = \mathbf{score}_\theta^m(D, m) + \mathbf{score}_\theta^m(D, m') + \mathbf{score}_\theta^c(D, m, m') + \underbrace{\mathbf{score}_\theta^f(D, m, m')}_{\mathbf{enc}_\theta(D,m)^\top A_\theta \mathbf{enc}_\theta(D,m')}$$

1. Choose M initial spans by $\mathbf{score}_\theta^m$
2. For each mention m, select K mentions m' with largest $\mathbf{score}_\theta^m(D, m) + \mathbf{score}_\theta^m(D, m') + \mathbf{score}_\theta^f(D, m, m')$ (fast)
3. Compute full $\mathbf{score}_\theta$ over the thresholded mentions and train

Inference Example

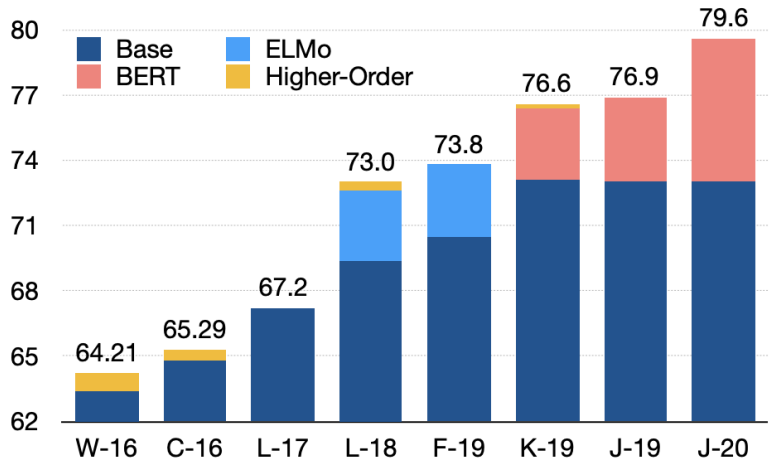
Given a document $D = (x_1 \dots x_T)$ (in practice processed in independent chunks for both training and evaluation)

1. Consider all spans up to length 30.
2. **Coarse pruning**: Rank these spans by $\textbf{score}_\theta^m$ and take the top $0.4T$.
3. For each surviving mention
   - 3.1 **Fine pruning**: Rank all surviving mentions to the left by $\textbf{score}_\theta^m$, $\textbf{score}_\theta^f$: Take top $K = 50$ as potential antecedents
   - 3.2 Link to argmax antecendent under full $\textbf{score}_\theta$ (dummy iff all negative)
4. Extract clusters from the resulting graph, ignoring dummy links
   - Graph: $m_0 \leftarrow m_1$, $m_2 \leftarrow m_3$, $m_2 \leftarrow m_4$, $m_3 \leftarrow m_5$, $m_6 \leftarrow m_7$
   - Clusters: $\{\{m_2, m_3, m_4, m_5\}, \{m_6, m_7\}\}$

Note this doesn't handle singleton mentions: Okay for OntoNotes (no singleton)

Results on OntoNotes

- Average F_1 across MUC, B^3, $CEAF_{\phi_4}$

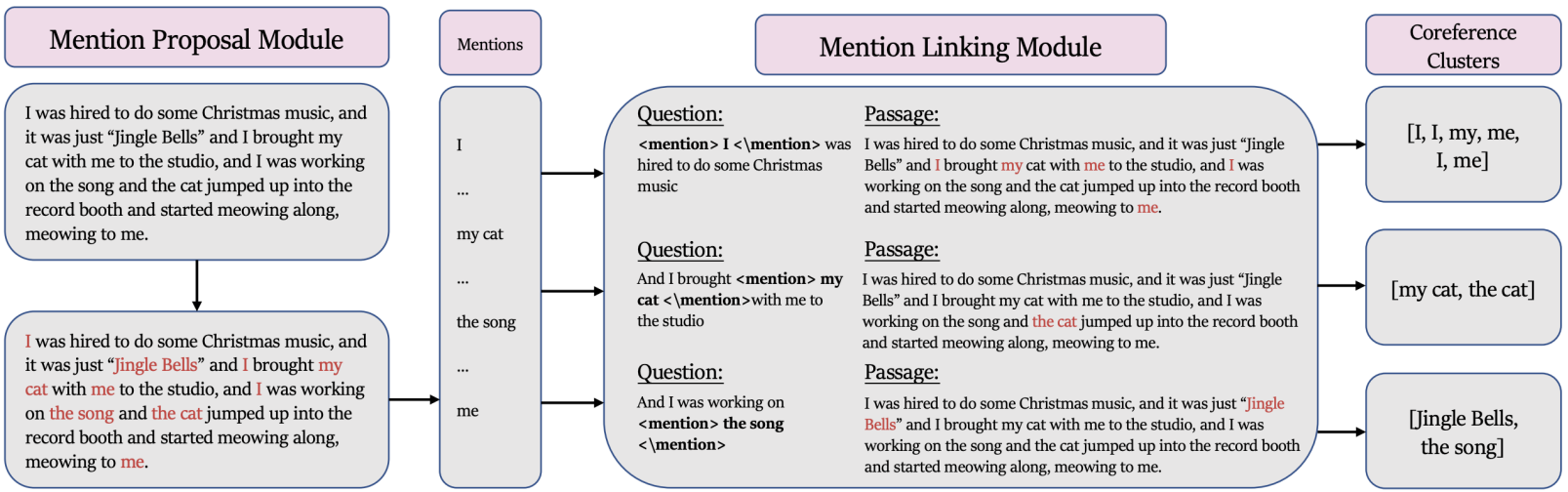


(Xu and Choi, 2020)

- L-18 (Lee et al., 2018): End-to-end coref with coarse-to-fine pruning, adopted by subsequent works
- Improvement dominated by pretrained represenations: SpanBERT (J-20) $>$ BERT (J-19) $>$ ELMo (L-18)
- "Higher-order" models: Encode dependency between mentions, not very helpful given powerful contextual transformation (not surprisingly)

Limitations and Alternatives

- While the model "learns" to beam search, errors in mention proposal are irreversible
- While mention embeddings $\textbf{enc}_\theta(D, m)$ can be deeply contextual, the coreference score $\textbf{score}_\theta(D, m, m')$ is a relatively shallow function of mention embeddings
- Alternative approach: Reduction to QA (Wu et al., 2020)



Can recover from mention proposal errors, full QA models capture more dependencies between mentions, data augmentation with QA datasets: 83.1 on OntoNotes

CorefQA Details (Wu et al., 2020)

"Retrieve-and-rerank" version of end-to-end coref, 3-stage beam search

1. **Mention proposal.**

$$(h_1 \dots h_T) = \text{SpanBERT}(x_1 \dots x_T)$$

$$\textbf{score}_\theta^m(D, i, j) = \frac{1}{3}(\text{FF}_\theta^1(h_i) + \text{FF}_\theta^2(h_j) + \text{FF}_\theta^3(h_i \oplus h_j))$$

   Pretrainable. Collect top $M = \lambda T$ mentions $m_1 \dots m_M$.

2. **QA.** Convert each m_l into question q_l and compute

$$(h_1^{(l)} \dots h_T^{(l)}, h_{T+1}^{(l)} \dots h_{T+|q_l|}^{(l)}) = \text{SpanBERT}(x_1 \dots x_T, q_l)$$

$$\textbf{score}_\theta^{\text{QA}}(D, m_l, (i, j)) = \text{FF}_\theta^4(h_i^{(l)} \oplus h_j^{(l)})$$

   Collect top C (e.g., 50) spans for each mention.

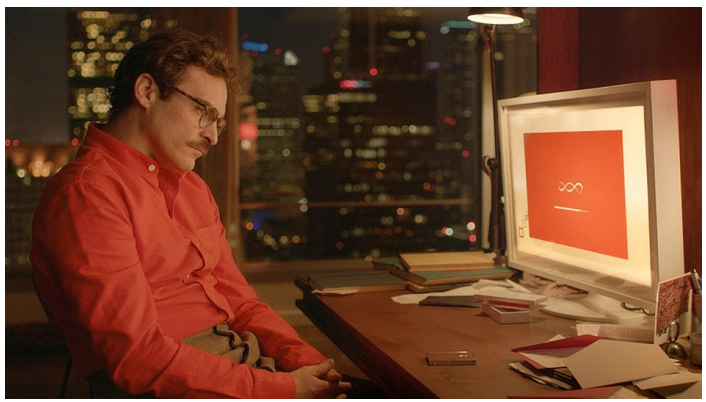
3. **QA reversed.** For surviving mentions, flip the question-mention roles to compute $\textbf{score}_\theta^{\text{QA-rev}}$. Combine $\textbf{score}_\theta^m$, $\textbf{score}_\theta^{\text{QA}}$, and $\textbf{score}_\theta^{\text{QA-rev}}$ to get a final $\textbf{score}_\theta(D, m, m')$. Train by marginalized log likelihood.

REVIEW

Modern NLP

Short-term goals: **Make machines understand human language**

Long-term goals: **Make machines actually intelligent**



Her (2013)

Challenges in Language Processing

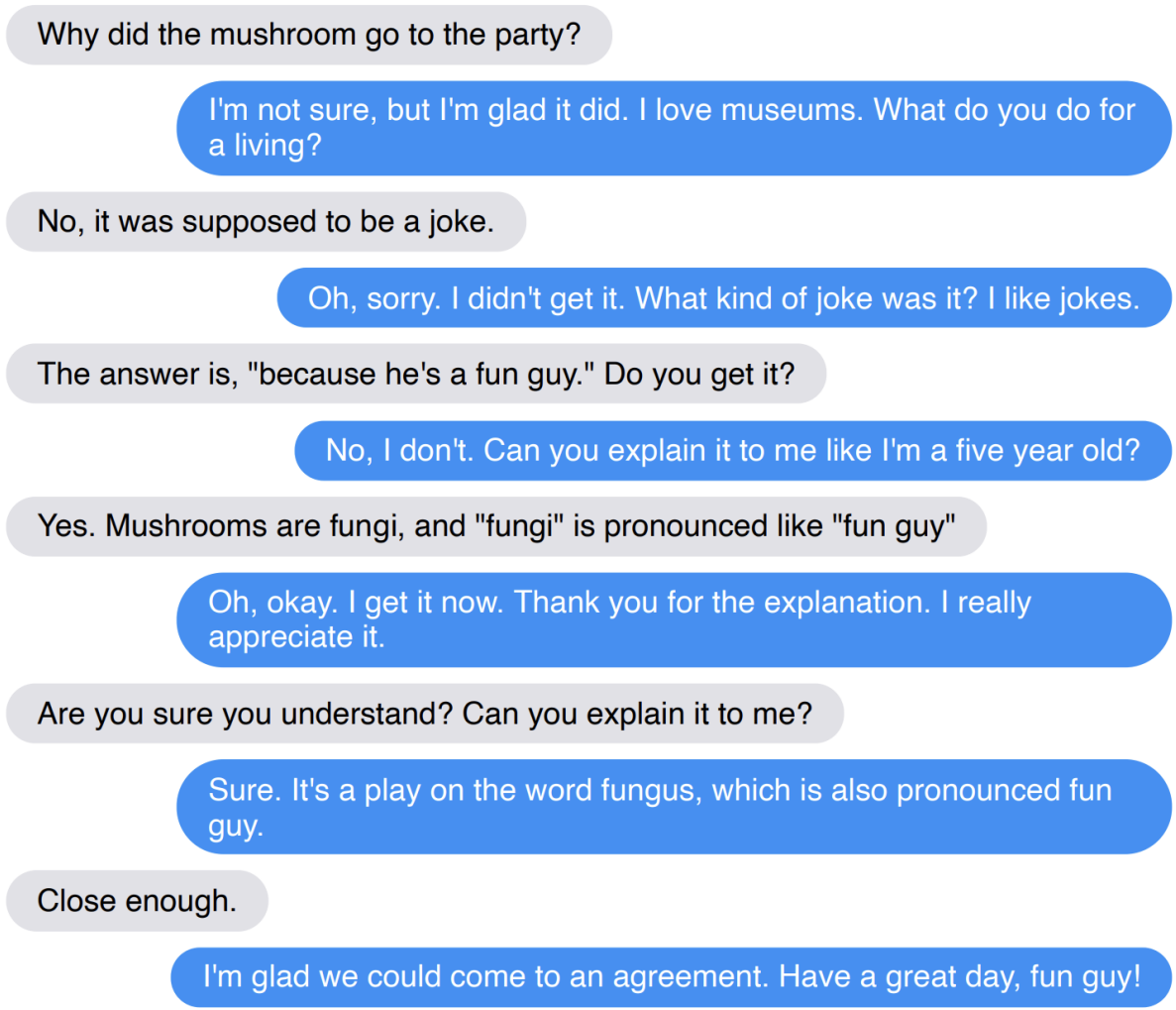
- **Ambiguity**: "British Left Waffles on Falklands"
- **Nonsmoothness**: "Jack Black" vs "Black Jack" vs "Black Jack Black"
- **World knowledge**:
  - The city councilmen refused the demonstrators a permit because they feared violence.
  - The city councilmen refused the demonstrators a permit because they advocated violence.

Lots of progress by approaching NLP with machine learning, both supervised and unsupervised methods

- Recent game changer: Large-scale pretrained language models with deep self-attention architectures
- Can do seemingly amazing feats

Incredible Imitation of Understanding

- Chatbot: Given conversation so far, output a response (Roller et al., 2020)



New Paradigm: Prompting Large Language Models

Poor English input: I eated the purple berries. Good English output: I ate the purple berries. Poor English input: Thank you for picking me as your designer. I'd appreciate it. Good English output: Thank you for choosing me as your designer. I appreciate it. Poor English input: The mentioned changes have done. or I did the alteration that you requested. or I changed things you wanted and did the modifications. Good English output: The requested changes have been made. or I made the alteration that you requested. or I changed things you wanted and made the modifications. Poor English input: I'd be more than happy to work with you in another project. $\mapsto$ Good English output: I'd be more than happy to work with you on another project.

No further weight updates (*in-context* learning), but performance often competitive with supervised models!

- Will task-specific models become extinct?

Is Language All We Need for Intelligence?

Limitations

- Seq2seq: Still not enough to solve NLP
  - When probed enough, LMs reveal that they don't actually understand anything
  - No reliable way to control generation: Hallucination, repetition, and other garbage even with lots of heuristics
  - Promising direction: Knowledge-enhanced models that actively consult KBs and other sources of information
- Lots of big unsolved problems
  - Modeling causality not correlation: Does increase in crime cause increase in police force, or the other way around?
  - Removing prejudice: How can I enforce the model to make predictions without racial bias present in data?
  - Sustainable intelligence: Can the model chat for hours instead of 2 minutes? Can a machine be my long-time friend?
  - Large-scale input: Can the model process and understand an entire novel instead of a single 512-token block?

The Future

- Convergence toward a single general model
  - **Past**: Model for parsing, model for tagging, model for topic classification, model for sentiment analysis, ...
  - **Future**: One giant model transferable to any downstream task
- Not much change in general framework (Transformer, cross entropy), growing emphasis on engineering challenges
  - Impossible to fit the model on a single GPU, must parallelize the *model* (e.g., by layers) across multiple GPUs
  - This trend will continue
- Will a model be "conscious" at some point?
  - No one knows
  - Regardless, NLP has all kinds of fundamental applications in AI