

Integrating Domain-Knowledge into Deep Learning

Russ Salakhutdinov

Machine Learning Department
Carnegie Mellon University
Canadian Institute for Advanced Research

Carnegie
Mellon
University



Impact of Deep Learning

- ▶ Speech Recognition
- ▶ Computer Vision
- ▶ Recommender Systems
- ▶ Language Understanding
- ▶ Drug Discovery and Medical Image Analysis

Domain knowledge

- ▶ Two key ingredients of a Statistical Machine Learning system
 - ▶ Model architecture/class
 - ▶ Learning algorithms to learn from data
- ▶ How do we incorporate domain knowledge into either or both these ingredients?
- ▶ We can consider three classes of domain knowledge:
 - ▶ Relational
 - ▶ Logical
 - ▶ Scientific

Relational Knowledge

- ▶ Simple relations among entities
 - ▶ (father, Bob, Alice)
- ▶ Available via relational databases, or knowledge graphs
- ▶ Statistical Relational Models
 - ▶ Probabilistic Graphical Models (PGMs) to model relationships amongst entities
 - ▶ Probabilistic Relational Models (via Bayes Nets), Relational Dependency Networks
- ▶ Embeddings
 - ▶ Instead of distributional semantics, represent entities via vectors in some vector space
 - ▶ Learn these vector representations via predicting an entity given its “context”
- ▶ We show how to incorporate relational information in Deep Learning via knowledge graph propagation

Logical Knowledge

- ▶ Propositional and First Order Logic (FOL) based knowledge
 - ▶ In contrast to simpler tuple based relational knowledge
 - ▶ E.g. if object has a wing, and a beak, it is a bird
- ▶ Encode logical knowledge into Probabilistic Graphical Models
- ▶ Bayesian Networks from Horn clauses, Probabilistic Context Free Grammars, Markov Logic Networks
- ▶ We incorporate logical information (and more general constraints) into Deep Learning via distillation (student-teacher) framework

Scientific Knowledge

- ▶ Partial and Stochastic Differential Equations
 - ▶ Newton Laws of Motion
 - ▶ Navier-Stokes fluid dynamics equations
 - ▶ ...
- ▶ Conservation laws and principles, Invariances
- ▶ Learning PDEs from data
- ▶ Regularizing dynamical system (e.g. state space models) via PDEs

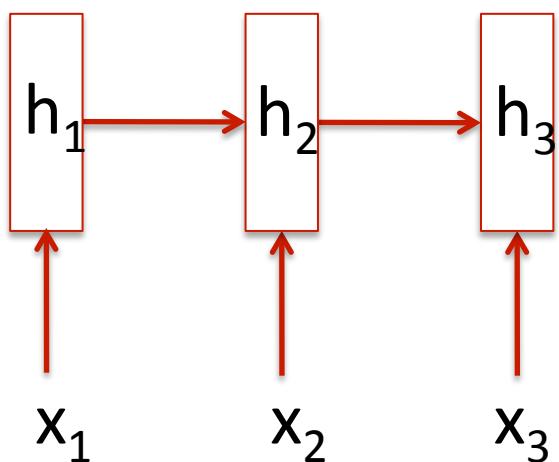
Reading Comprehension

- ▶ **Context:** "...arrested Illinois governor **Rod Blagojevich** and his chief of staff John Harris on corruption charges ... included **Blagojevich** allegedly conspiring to sell or trade the senate seat left vacant by President-elect Barack Obama..."
- ▶ **Query:** President-elect Barack Obama said Tuesday he was not aware of alleged corruption by **X** who was arrested on charges of trying to sell Obama's senate seat.
- ▶ **Answer:** Rod Blagojevich

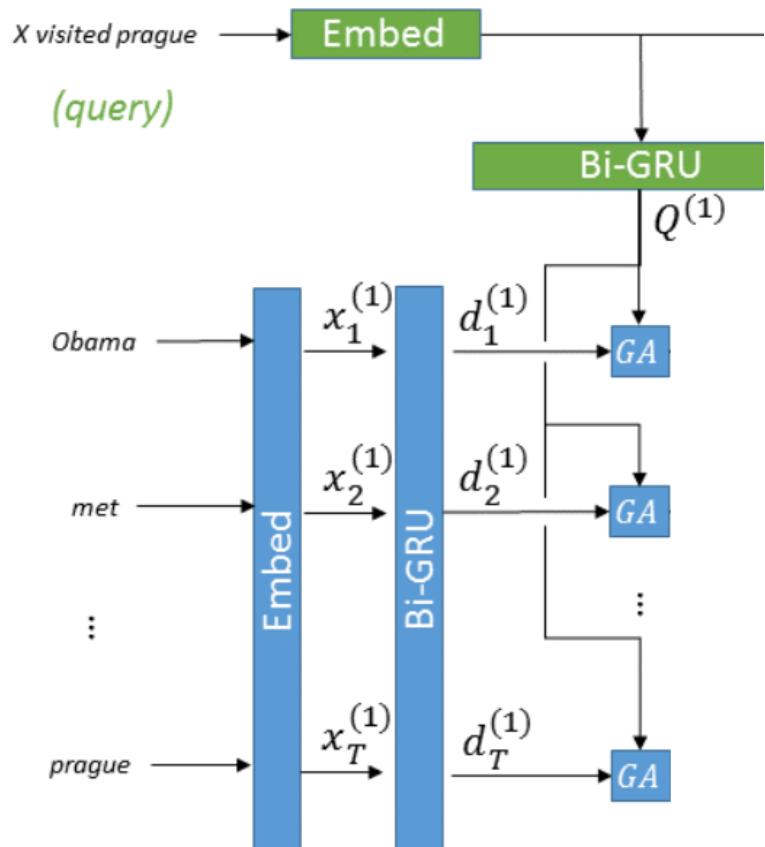
Recurrent Neural Networks (RNNs)

$$\mathbf{h}_t = \phi(\mathbf{U}\mathbf{h}_{t-1} + \mathbf{W}\mathbf{x}_t + \mathbf{b})$$

Nonlinearity Hidden State at previous time step Input at time step t



Gated Attention Mechanism

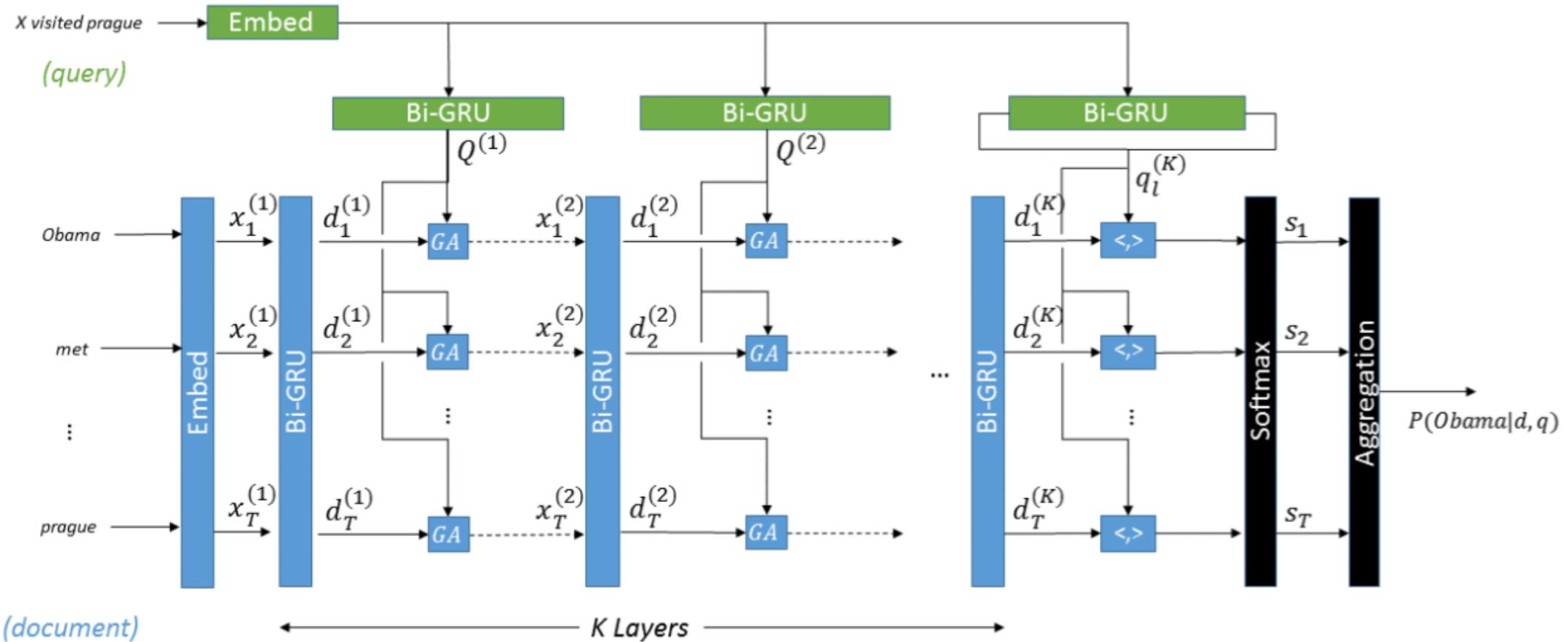


- Use Recurrent Neural Networks or Transformers to encode a document and a query.
- Use element-wise multiplication to model the interactions between document and query:

$$x_i = d_i \odot q_i$$

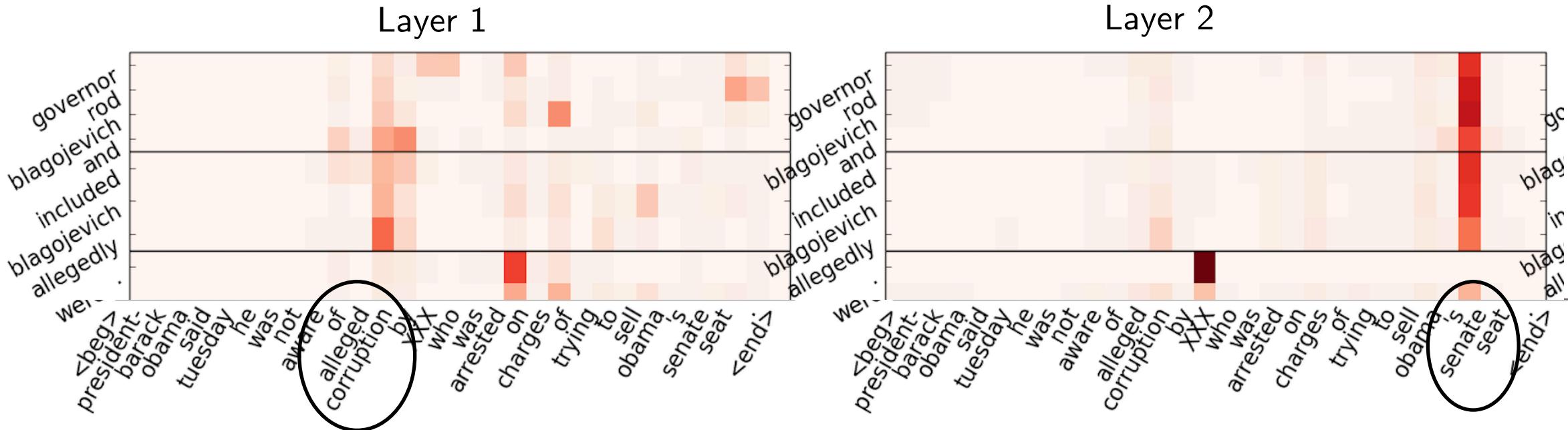
Multi-Hop Architecture

- Reasoning over multiple sentences requires several passes over the context

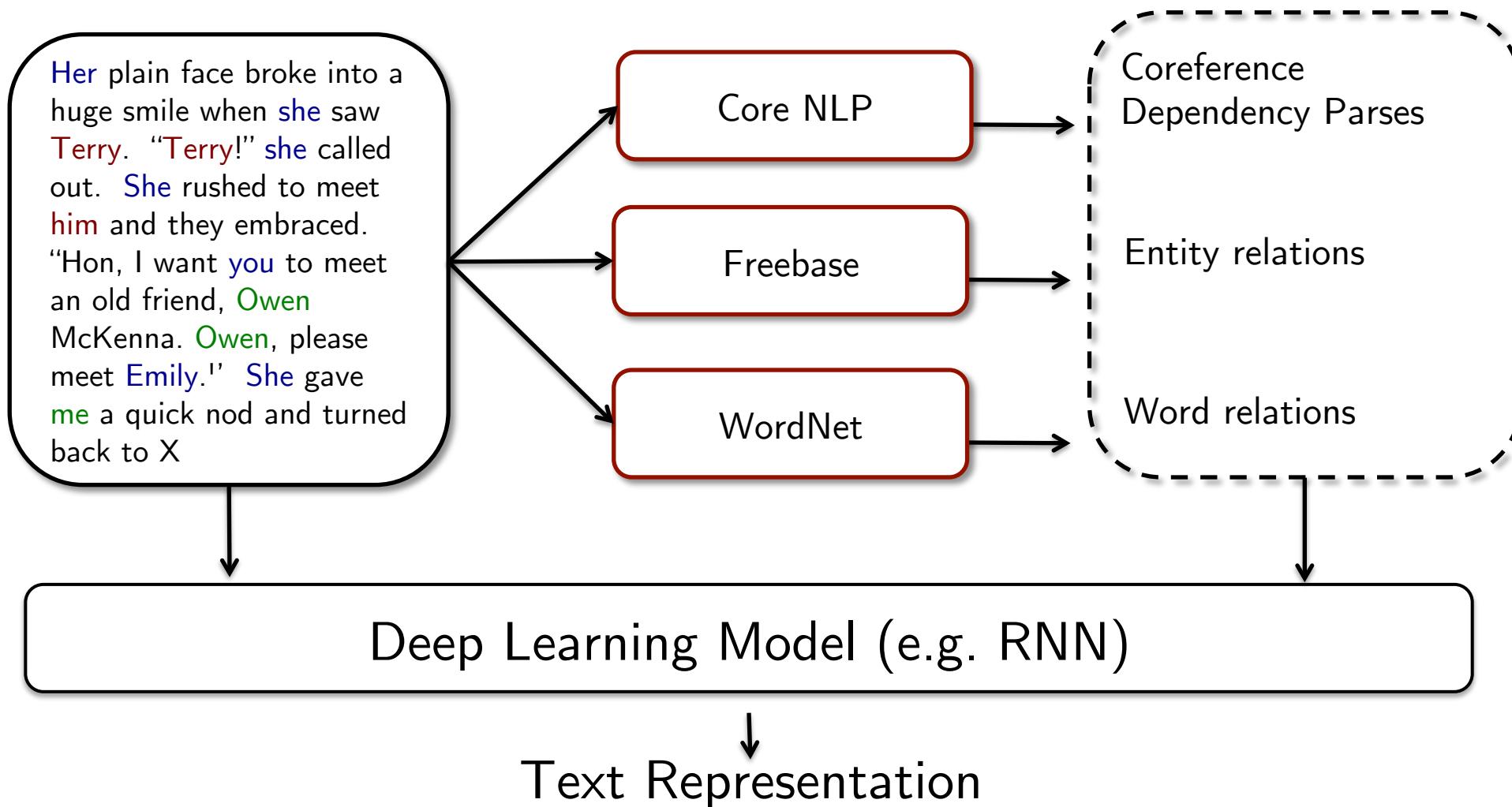


Reasoning and Attention

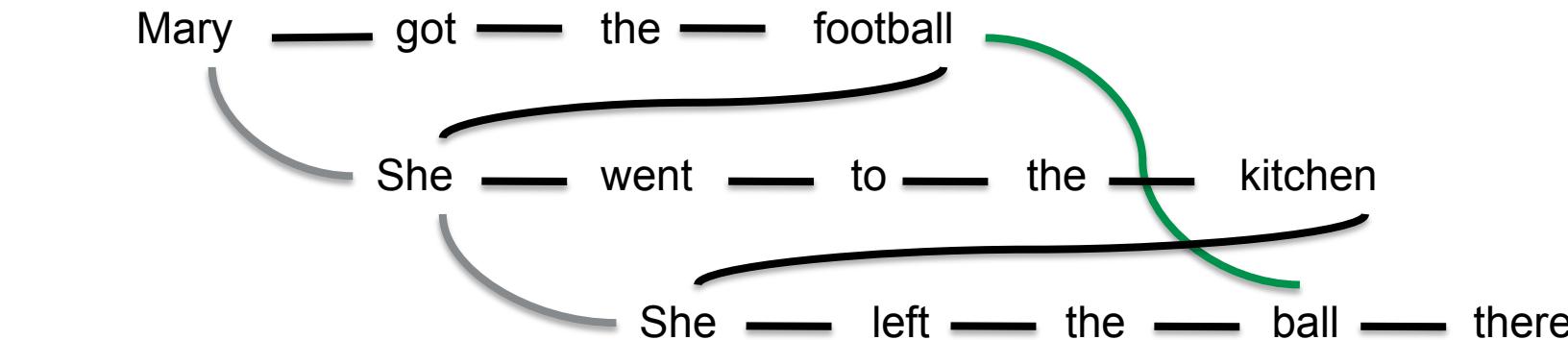
- ▶ **Context:** "...arrested Illinois **governor Rod Blagojevich** and his chief of staff John Harris on corruption charges ... **included Blagojevich** allegedly conspiring to sell or trade the **senate seat** left vacant by President-elect Barack Obama..."
- ▶ **Query:** "President-elect Barack Obama said Tuesday he was not aware of **alleged corruption** by X who was arrested on charges of trying to sell Obama's **senate seat**."
- ▶ **Answer: Rod Blagojevich**



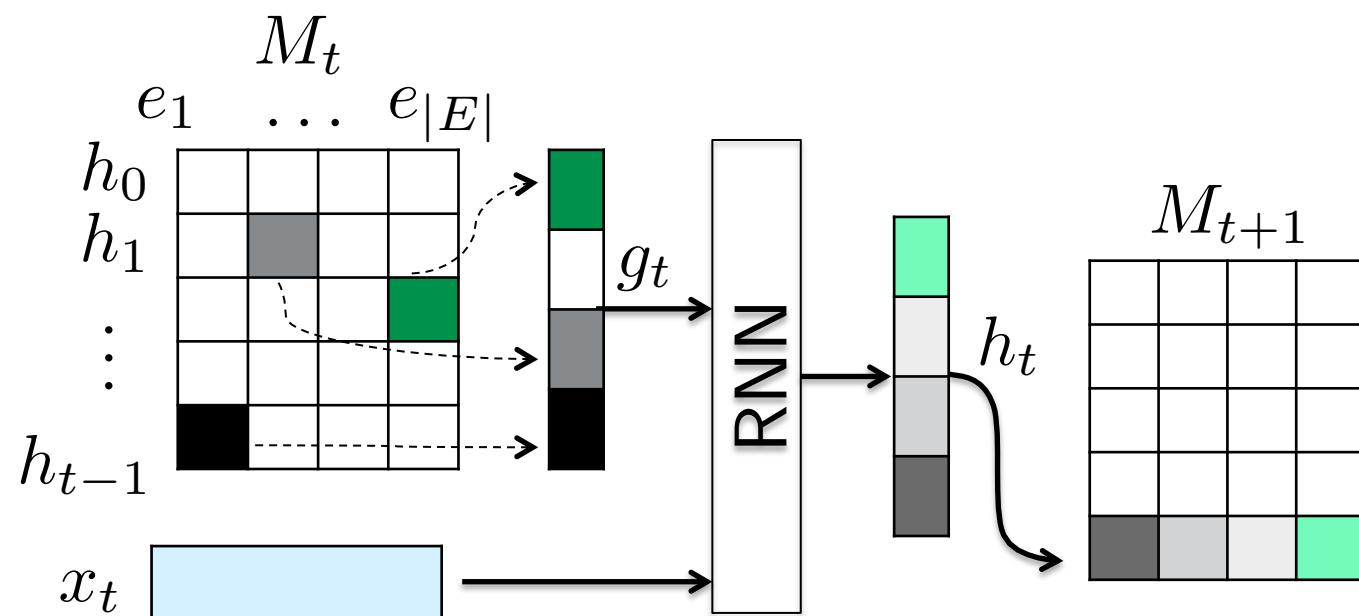
Incorporating Prior Knowledge



Explicit Memory (Dhingra et al., 2018)



Memory as Acyclic Graph Encoding (MAGE) - RNN



Open Domain Question Answering

- ▶ Finding answers to factual questions posed in Natural Language:

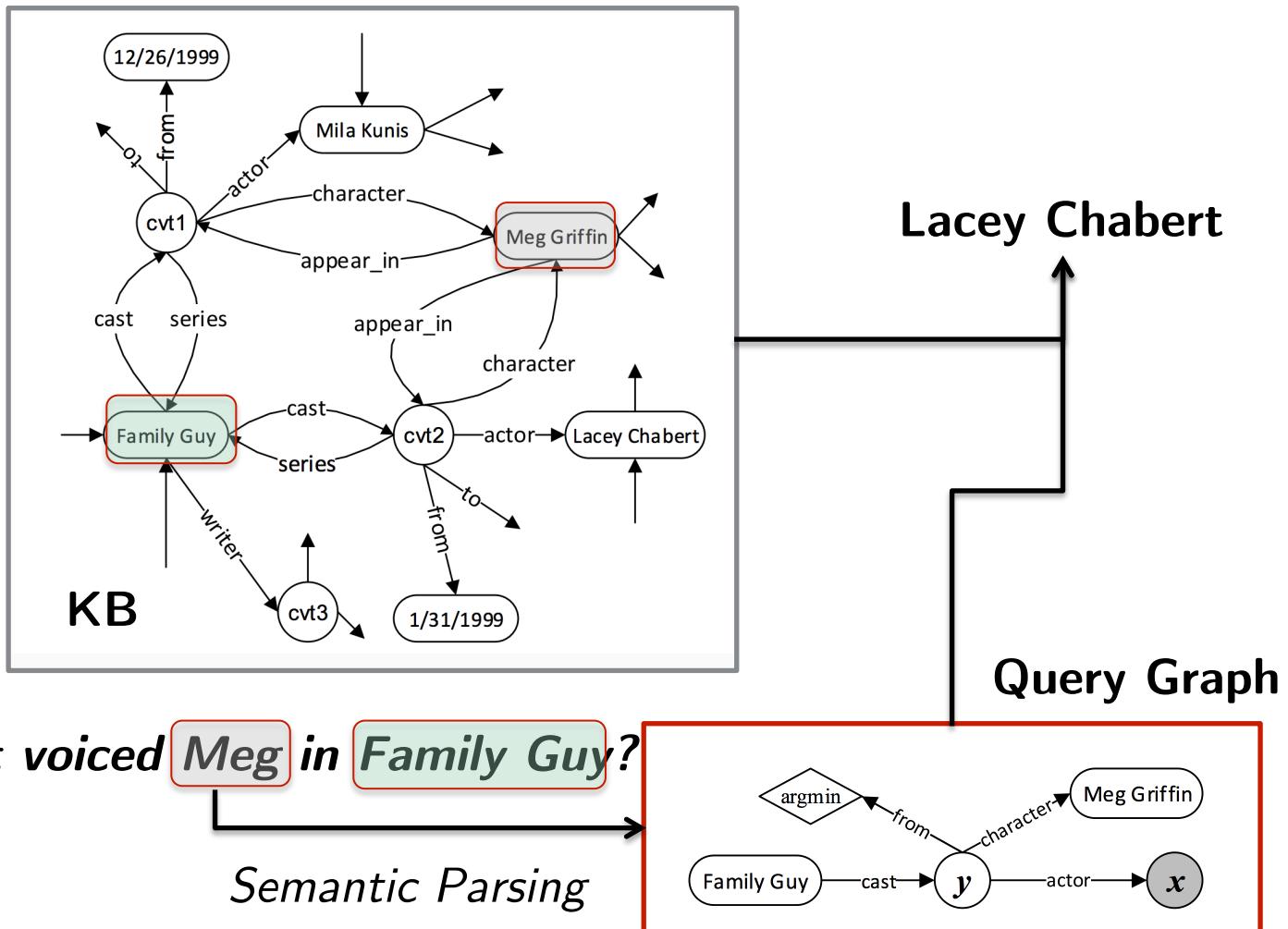
Who voiced Meg in Family Guy?

A. Lacey Chabert, Mila Kunis

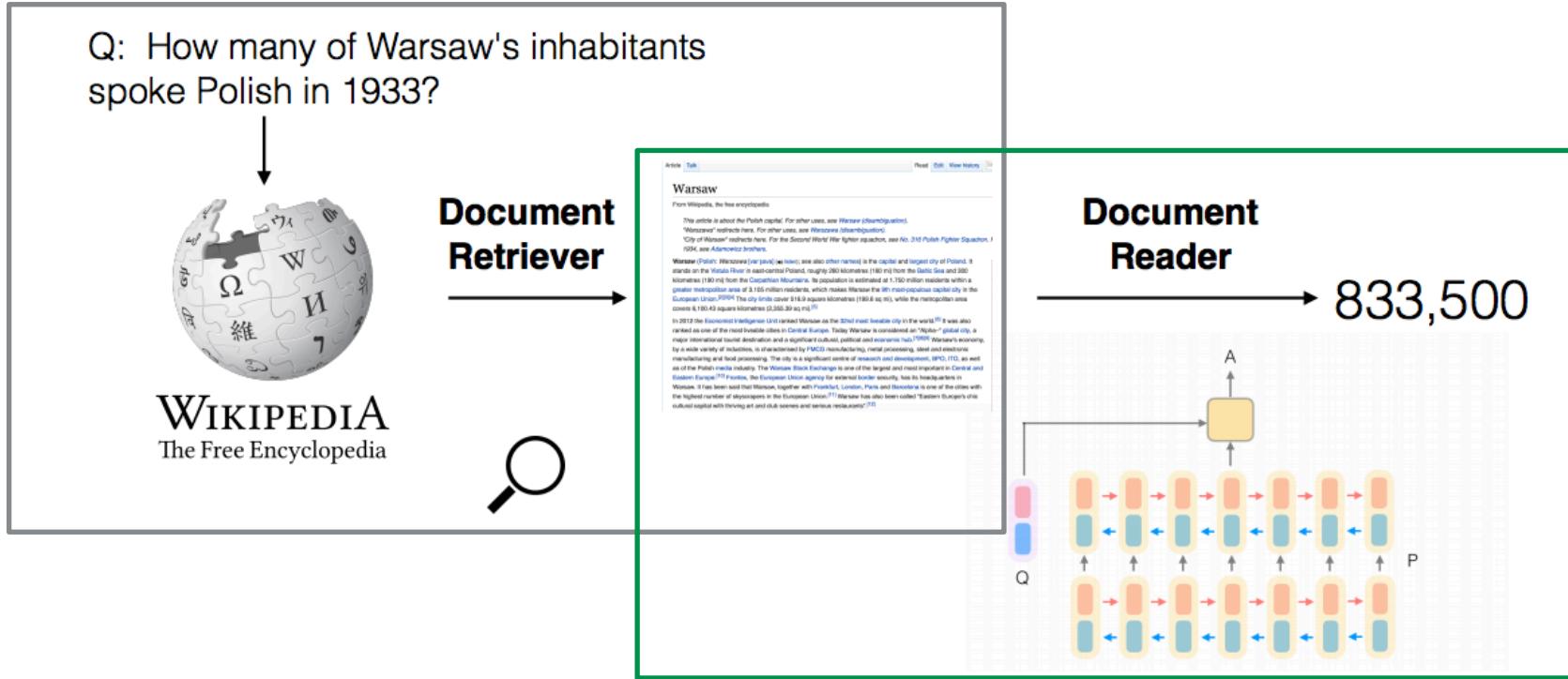
Who **first** voiced Meg in Family Guy?

A. Lacey Chabert

Knowledge Base as a Knowledge Source



Unstructured Text as a Knowledge Source



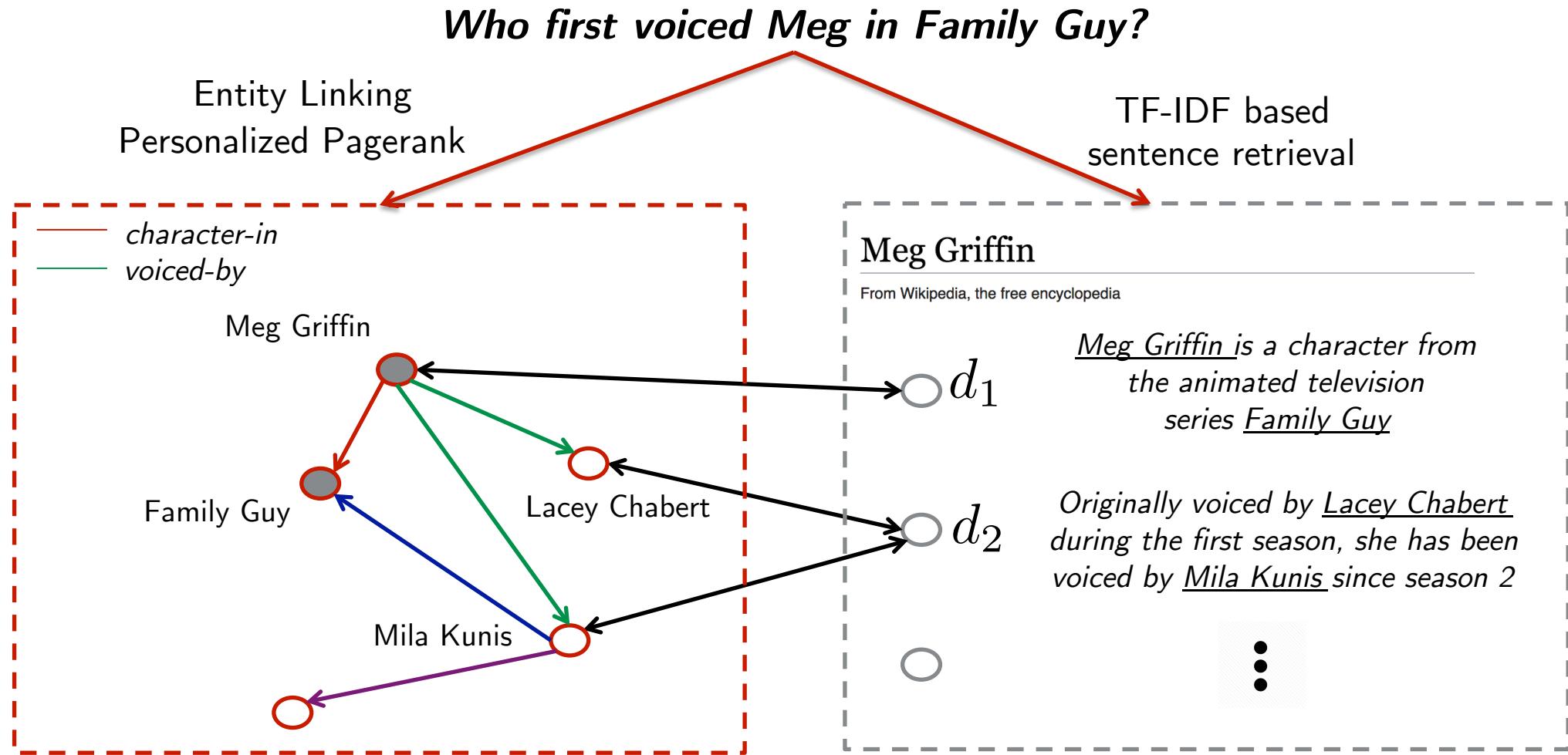
Step 1 (Information Retrieval):

Retrieve passages relevant to the Question using shallow methods

Step 2 (Reading Comprehension):

Perform deep reading of passages to extract answers

Text Augmented Knowledge Graph (Dhingra, Sun, et al., 2018)



Reading Graphs

Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and a natural language question $q = (w_1, \dots, w_T)$ learn a function $y_v = f(v) \forall v \in \mathcal{V}$, s.t. $y_v \in \{0, 1\}$ and $y_v = 1$ if and only if v is an answer for q .

$$P(y_v = 1 | \mathcal{G}, q) = \frac{\exp h_q^T h_v}{\sum_{v'} \exp h_q^T h_{v'}}$$

h_q -- Question Representation from an LSTM

h_v -- Node Representation from a Graph Convolution Network

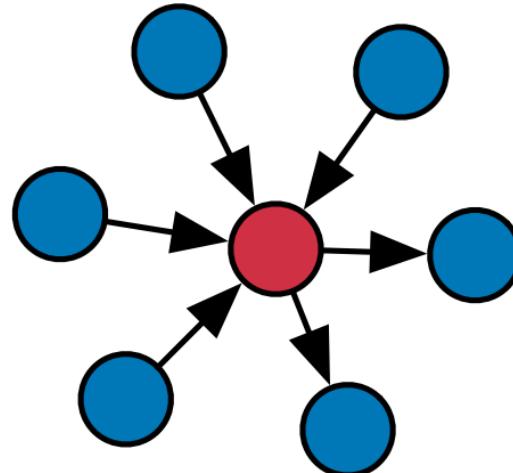
Graph Convolution Network

For each v :

Initialize $h_v^{(0)}$

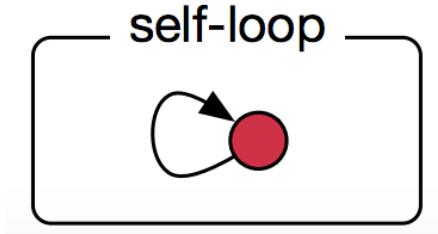
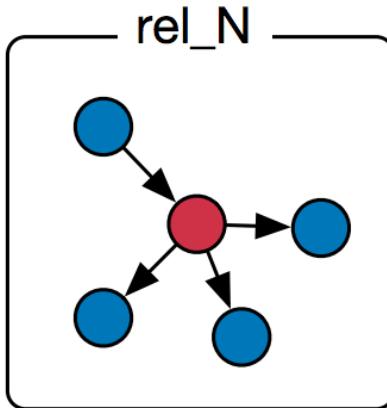
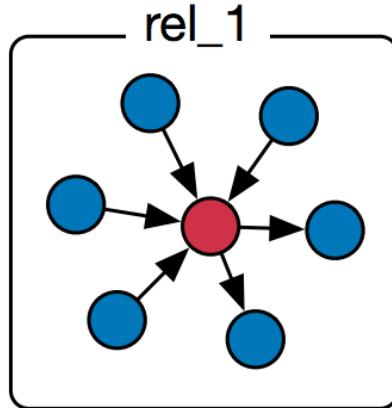
$$h_v^{(t)} = f(W_1 h_v^{(t-1)} + W_2 \sum_{v' \in N(v)} \alpha_{v'} h_{v'}^{(t-1)})$$

Repeat for $t = 1, \dots, T$



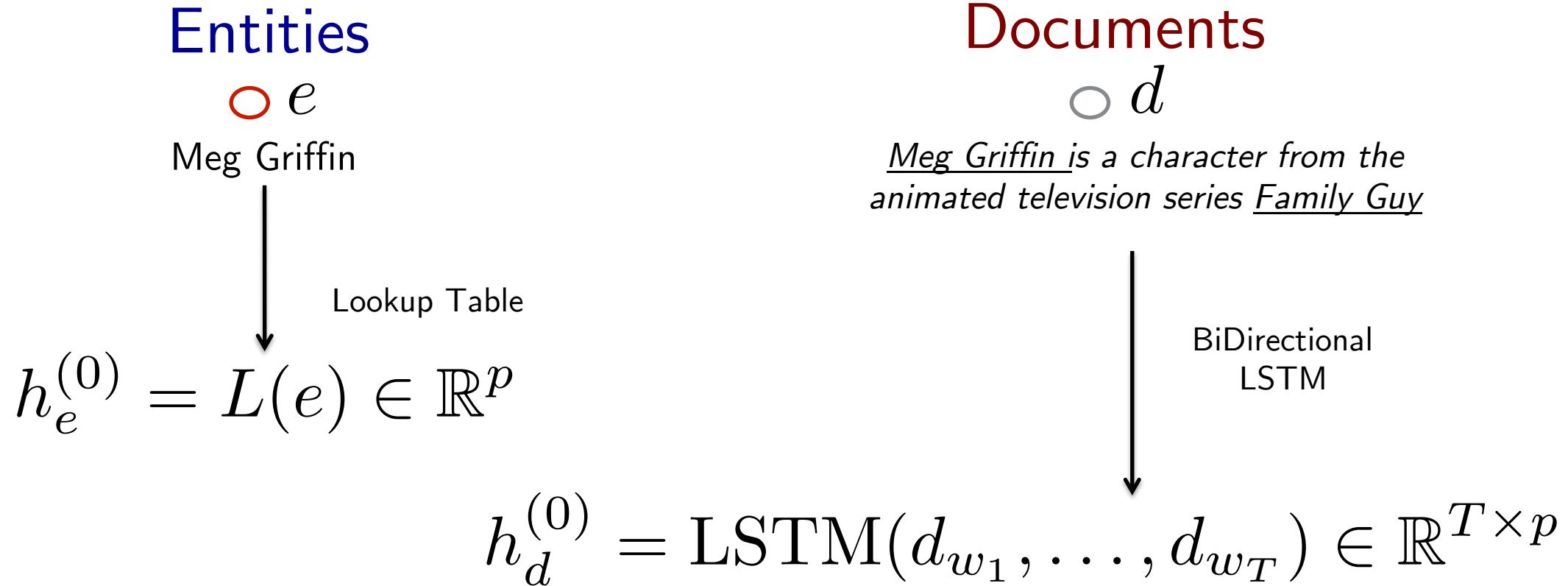
Relational Graph Convolution Network

Graphs with edge types



$$h_v^{(t)} = f \left(\sum_r W_1 h_v^{(t-1)} + W_2 \sum_{v' \in N_r(v)} \alpha_{v'} h_{v'}^{(t-1)} \right)$$

Graph Propagation / Graph Convolution



Graph Propagation / Graph Convolution

Entities



Meg Griffin

Documents



Meg Griffin is a character from the
animated television series Family Guy

$$h_d^{(t)} = \text{LSTM}(h_{d_1}^{(t-1)} || e_{w_1}^{(t-1)}, \dots, h_{d_T}^{(t-1)} || e_{w_T}^{(t-1)})$$



Meg Griffin



Meg Griffin is a character from the
animated television series Family Guy

$$h_e^{(t)} = f(W_1 h_e^{(t-1)} + \sum_r \sum_{v' \in N_r(v)} W_2^r h_{v'}^{(t-1)} + W_3 \sum_{d:e \in d} h_{d_w}^{(t-1)})$$

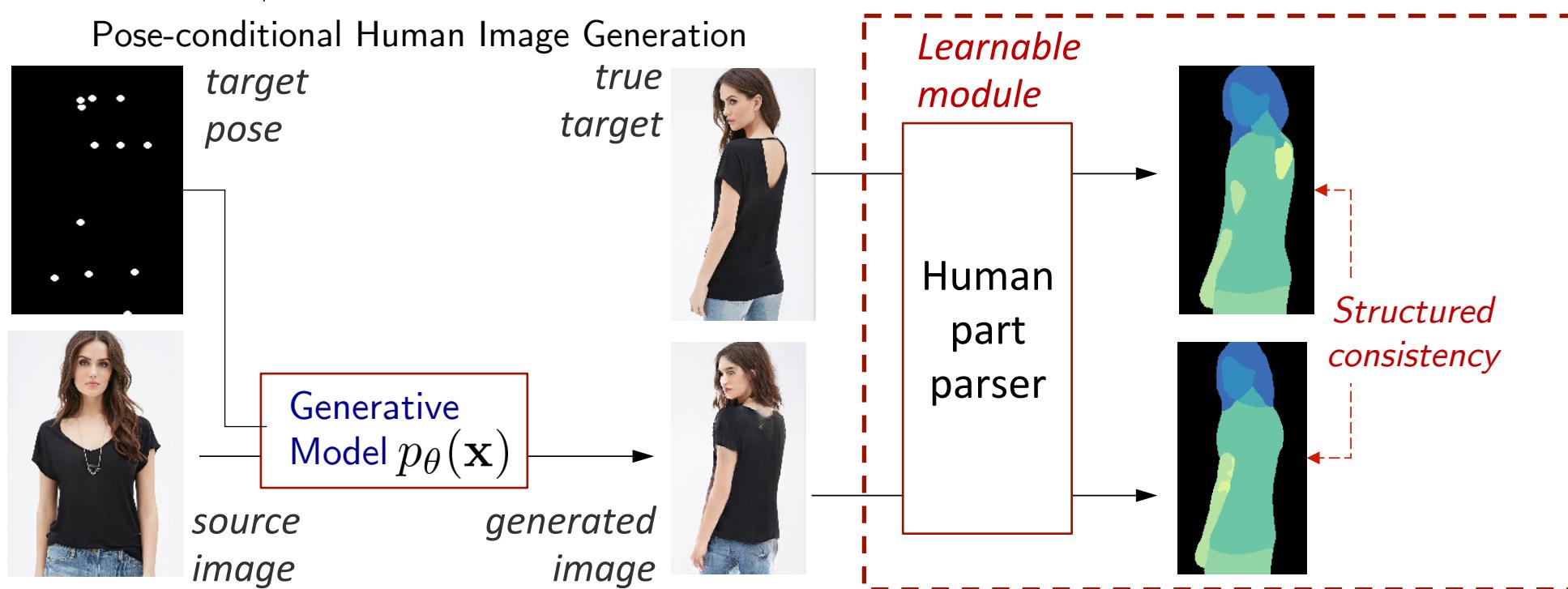
- Relational information via KB propagation

Domain knowledge

- ▶ We consider three classes of domain knowledge:
 - ▶ Relational
 - ▶ Logical (constraints)
 - ▶ Scientific

Learning with Constraints

- ▶ Consider a statistical model $\mathbf{x} \sim p_{\theta}(\mathbf{x})$
- ▶ Consider a constraint function, $f_{\phi}(\mathbf{x}) \in \mathbb{R}$ parameterized by ϕ
 - ▶ Higher $f_{\phi}(\mathbf{x})$ value, better \mathbf{x} w.r.t the knowledge



Learning with Constraints

- ▶ Consider a statistical model $\mathbf{x} \sim p_\theta(\mathbf{x})$
- ▶ Consider a constraint function, $f_\phi(\mathbf{x}) \in \mathbb{R}$ parameterized by ϕ
 - ▶ Higher $f_\phi(\mathbf{x})$ value, better \mathbf{x} w.r.t the knowledge
- ▶ Sentiment prediction:
 - ▶ This was a terrific movie, but the director could have done better
- ▶ Logical Rules:
 - ▶ Sentence S with structure A -but- B : \Rightarrow sentiment of B dominates

Learning with Constraints

- ▶ Consider a statistical model $\mathbf{x} \sim p_\theta(\mathbf{x})$
- ▶ Consider a constraint function, $f_\phi(\mathbf{x}) \in \mathbb{R}$ parameterized by ϕ
 - ▶ Higher $f_\phi(\mathbf{x})$ value, better \mathbf{x} w.r.t the knowledge
- ▶ One way to impose the constraint is to maximize: $\mathbb{E}_{p_\theta}[f_\phi(\mathbf{x})]$
- ▶ Objective:

$$\min_{\theta} (\mathcal{L}(\theta) - \alpha \mathbb{E}_{p_\theta}[f_\phi(\mathbf{x})])$$

Regular objective (e.g. cross-entropy loss, etc.)

Regularization: imposing constraints – **difficult to compute**

Posterior Regularization (Ganchev et al., 2010)

- ▶ Consider a statistical model $\mathbf{x} \sim p_\theta(\mathbf{x})$
- ▶ Consider a constraint function, $f_\phi(\mathbf{x}) \in \mathbb{R}$ parameterized by ϕ

$$\min_{\theta} (\mathcal{L}(\theta) - \alpha \underbrace{\mathbb{E}_{p_\theta}[f_\phi(\mathbf{x})]}_{\mathcal{L}(\theta, q) = \text{KL}(q(\mathbf{x}) || p_\theta(\mathbf{x})) - \lambda \mathbb{E}_q[f_\phi(\mathbf{x})]})$$

- ▶ Introduce variational distribution q , which is encouraged to stay close to p
- ▶ Objective:

$$\min_{\theta, q} (\mathcal{L}(\theta) + \alpha \mathcal{L}(\theta, q))$$

Posterior Regularization (Ganchev et al., 2010)

$$\min_{\theta, q} (\mathcal{L}(\theta) + \alpha \mathcal{L}(\theta, q))$$

$$\mathcal{L}(\theta, q) = \text{KL}(q(\mathbf{x}) || p_\theta(\mathbf{x})) - \lambda \mathbb{E}_q [f_\phi(\mathbf{x})]$$

- ▶ Optimal solution for q :

$$q^*(\mathbf{x}) = p_\theta(\mathbf{x}) \exp(\lambda f_\phi(\mathbf{x})) / \mathcal{Z}$$



Higher value -- higher probability
under q – “soft constraint”

- ▶ How do we fit our model parameters θ ?

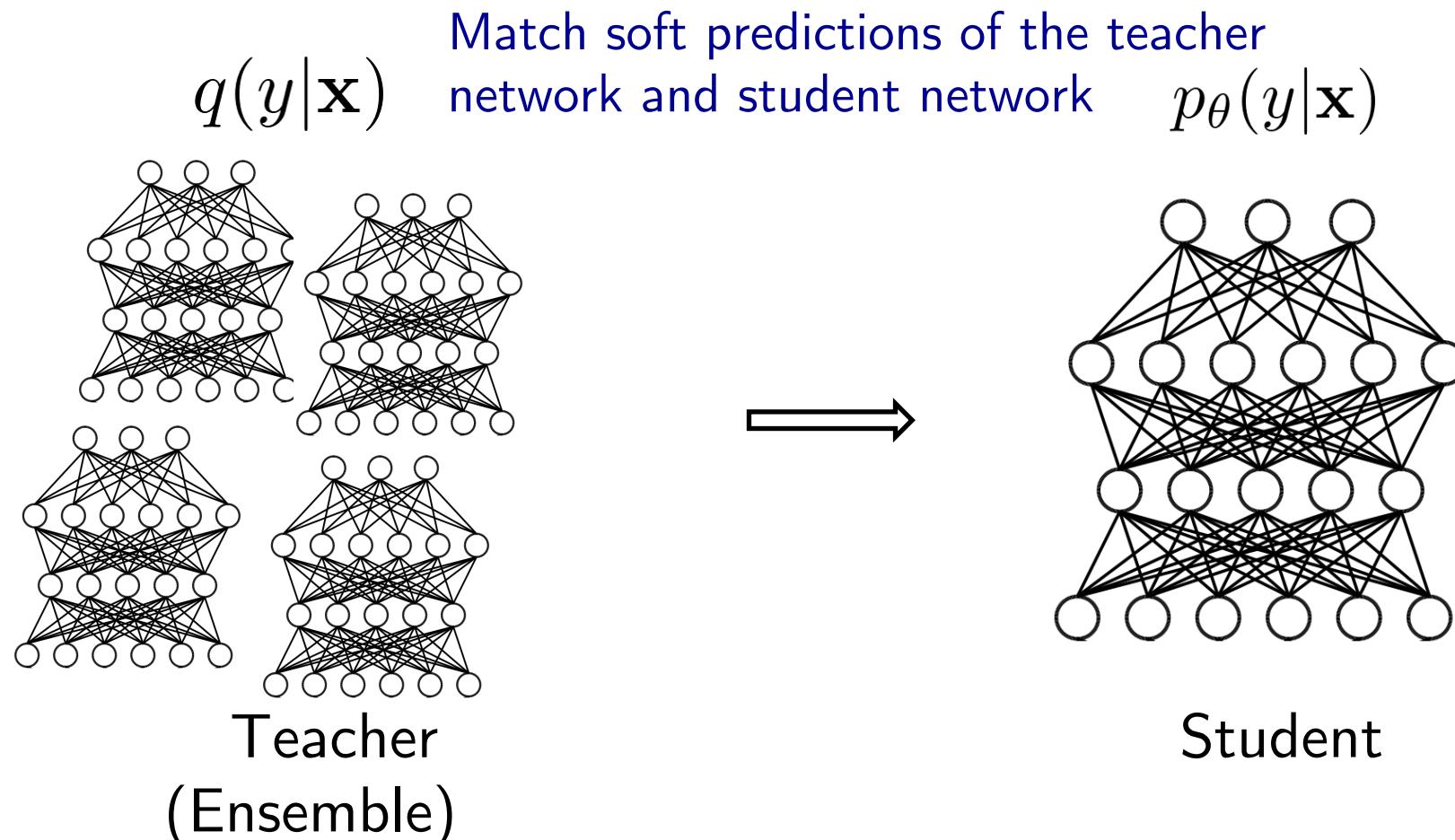
Logical Rule Formulation (Zhiting Hu et al., 2016)

- ▶ Consider a supervised learning: $p_\theta(y|\mathbf{x})$, e.g. deep neural network
- ▶ Input-Target space (X, Y)
- ▶ First-order logic rules: (r, λ)
 - ▶ $r(X, Y) \in [0, 1]$, could be soft
 - ▶ λ is the confidence level of the rule
- ▶ Within PR framework given l rules

$$q^*(y|\mathbf{x}) = p_\theta(y|\mathbf{x}) \exp\left(\sum_l \lambda_l r_l(y, \mathbf{x})\right) / \mathcal{Z}$$

- ▶ How to train a neural network: Knowledge Distillation [Hinton et al., 2015; Bucilu et al., 2006].

Knowledge Distillation



Knowledge Distillation [Hinton et al., 2015; Bucilu et al., 2006].

Rule Knowledge Distillation (Zhiting Hu et al., 2016)

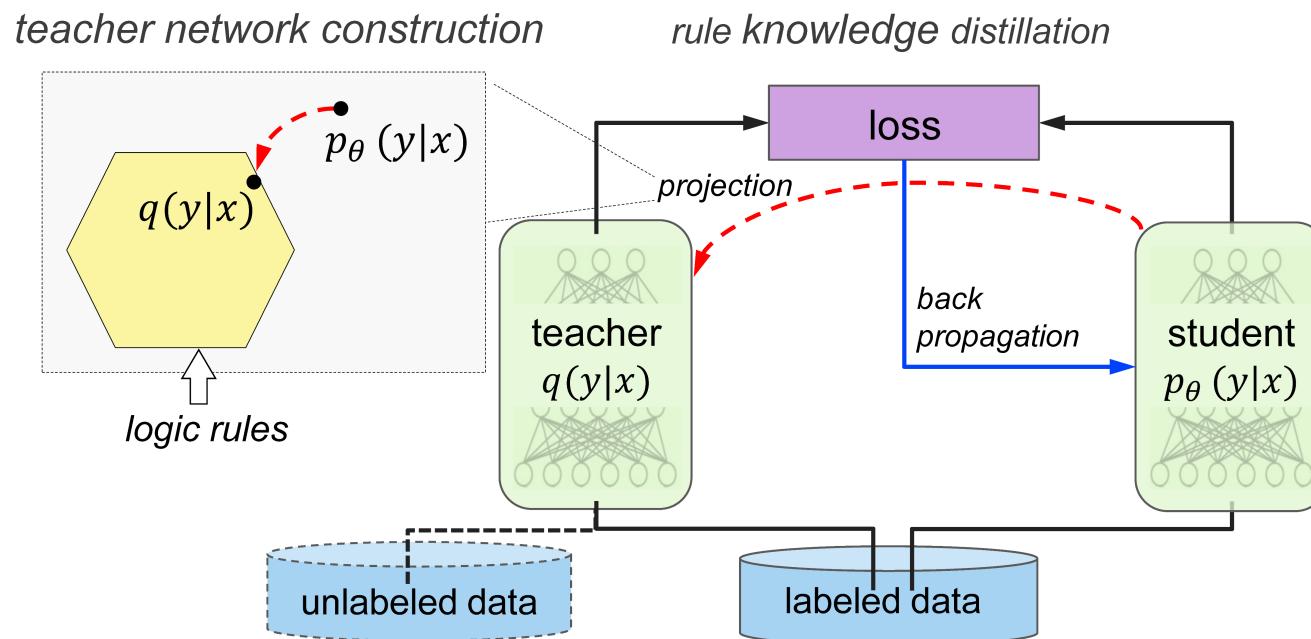
- ▶ Deep neural network $p_\theta(y|\mathbf{x})$
- ▶ Train to imitate the outputs of the rule-regularized teacher network
- ▶ At iteration t:

$$\theta^{(t+1)} = \operatorname{argmin}_\theta \frac{1}{N} \sum_{n=1}^N \ell(y_n, \sigma_\theta(\mathbf{x})) + \alpha \ell(s_n^{(t)}, \sigma_\theta(\mathbf{x}))$$

true hard
 label ↓ soft prediction of
 $p_\theta(y|\mathbf{x})$
 balancing parameter soft prediction of the
 teacher network q .
 $q^*(y|\mathbf{x}) = p_\theta(y|\mathbf{x}) \exp \left(\sum_l \lambda_l r_l(y, \mathbf{x}) \right) / \mathcal{Z}$

Rule Knowledge Distillation (Zhiting Hu et al., 2016)

- ▶ Deep neural network $p_\theta(y|x)$
- ▶ At each iteration:
 - ▶ Construct a teacher network $q(y|x)$ with “soft constraints”
 - ▶ Train DNN to emulate the teacher network



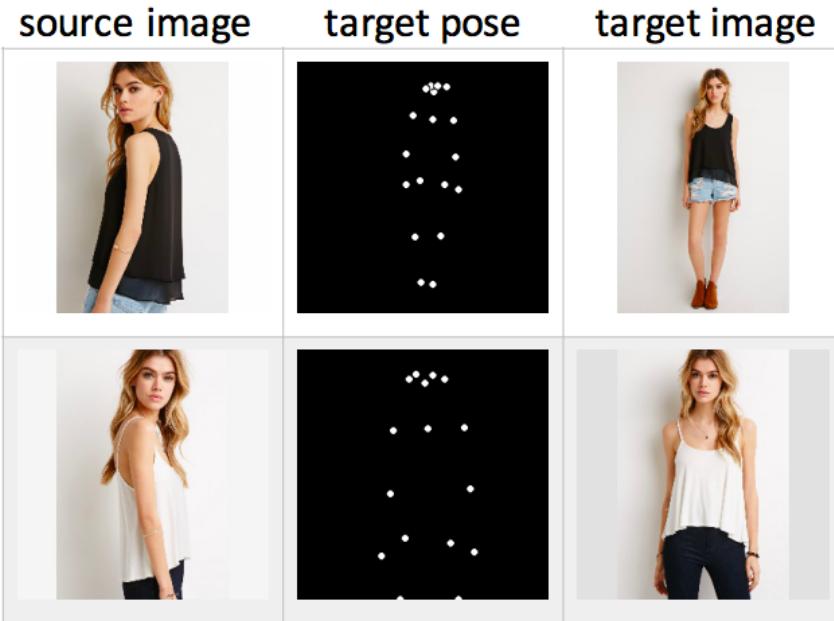
- ▶ Sentiment classification,
- ▶ Named entity recognition

Learning Rules / Constraints (Zhiting Hu et al., 2018)

$$q^*(y|\mathbf{x}) = p_\theta(y|\mathbf{x}) \exp \left(\sum_l \lambda_l r_l(y, \mathbf{x}) \right) / \mathcal{Z}$$

- ▶ We can also learn the "confidence" values λ_l for logical rules
- ▶ More generally, we can optimize parameters of the constraint function $f_\phi(\mathbf{x})$
$$q^*(\mathbf{x}) = p_\theta(\mathbf{x}) \exp (\lambda f_\phi(\mathbf{x})) / \mathcal{Z}$$
- ▶ Treat $f_\phi(\mathbf{x})$ as the reward function to be learned within the MaxEnt Inverse Reinforcement Learning

Pose-conditional Human Image Generation



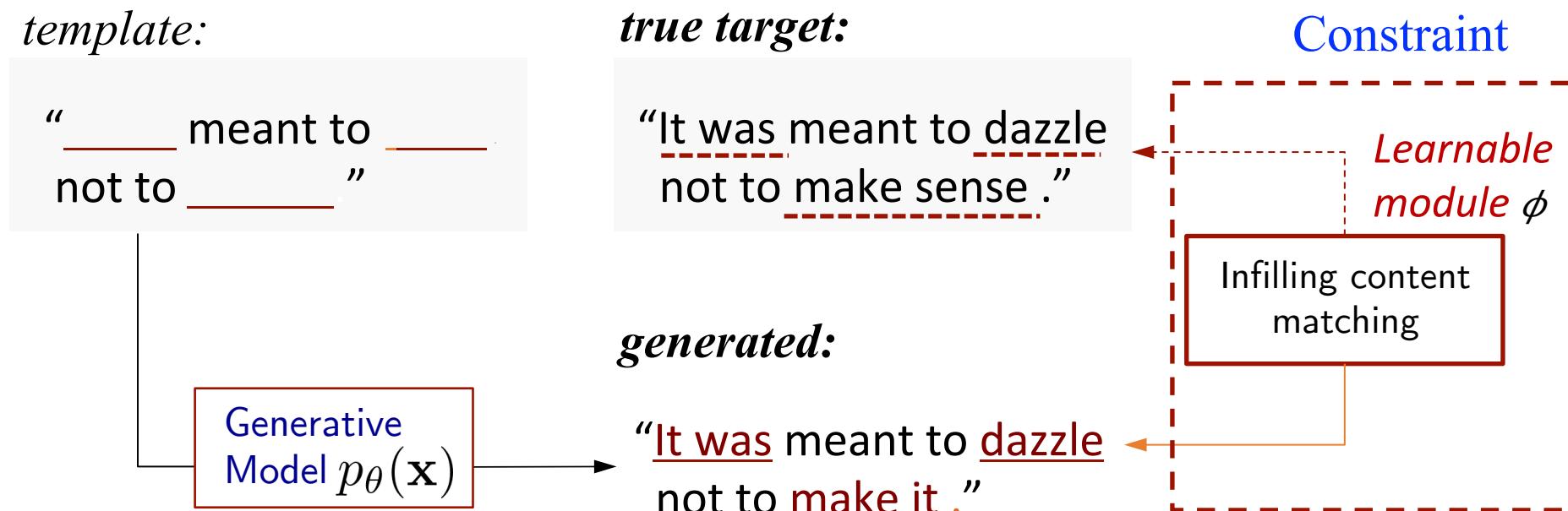
Samples generated by the models. Enforcing learned human part constraint generates correct poses and better preserves human body structure

	Method	SSIM	Human
1	Ma et al. [38]	0.614	—
2	Pumarola et al. [44]	0.747	—
3	Ma et al. [37]	0.762	—
4	Base model	0.676	0.03
5	With fixed constraint	0.679	0.12
6	With learned constraint	0.727	0.77

Results of image generation using Structural Similarity (SSIM) between generated and true images

Template-guided Sentence Generation

- ▶ Task: Given a template, generate a complete sentence following the template
- ▶ Constraint: force to match between infilling content of the generated sentence with the true content



Template-guided Sentence Generation

Model	Perplexity	Human
1 Base model	30.30	0.19
2 With binary D	30.01	0.20
3 With constraint updated in M-step (Eq.5)	31.27	0.15
4 With learned constraint	28.69	0.24

Samples by the full model are considered as of higher quality in 24% cases.

acting
the acting is the acting .
the acting is also very good .
out of 10 .
10 out of 10 .
I will give the movie 7 out of 10 .

Two test examples, including the template, the sample by the base model, and the sample by the constrained model.

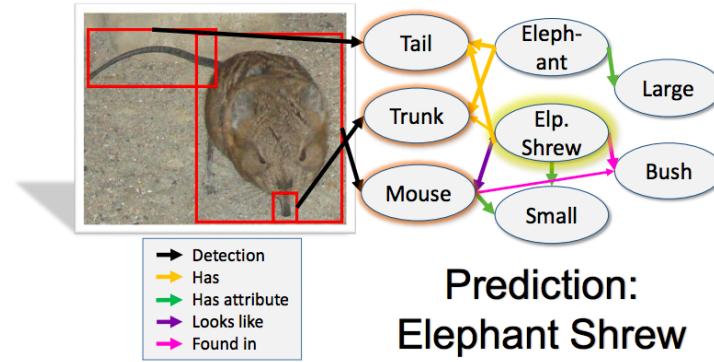
Conclusion

- ▶ **Limitations:** We considered very simple forms of domain knowledge: relational, logical, simple constraints
- ▶ **Human Knowledge:** abstract, fuzzy, build on high-level concepts
 - ▶ e.g. dogs have 4 legs



dog

#legs=4



Example of how semantic knowledge about the world aids classification.

Marino et al., CVPR 2017

- ▶ How do we encode this knowledge and how do we efficiently integrate this into deep learning models

Thank you