Generative Models for Discriminative Problems

Chris Dyer
DeepMind

ASRU 2017 December 19, 2017

Terminological clarification

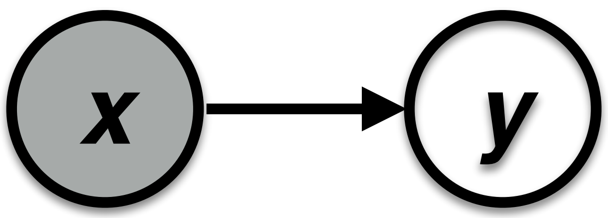
- A **discriminative problem**: for some input $\boldsymbol{x}$, find the most likely $\boldsymbol{y}$ in a set $\mathcal{Y}(\boldsymbol{x})$

Terminological clarification

- A **discriminative problem**: for some input $\boldsymbol{x}$, find the most likely $\boldsymbol{y}$ in a set $\mathcal{Y}(\boldsymbol{x})$

- A **discriminative model** directly models $p(\boldsymbol{y} \mid \boldsymbol{x})$

  logistic/linear/… regressions, MLPs, CRFs, MEMMs, seq2seq(+att)

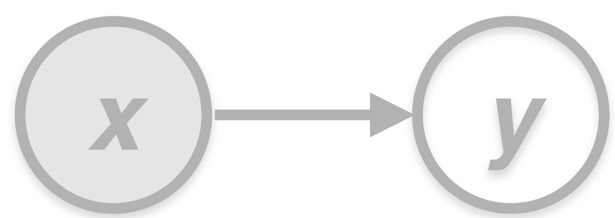


Terminological clarification

- A **discriminative problem**: for some input ***x***, find the most likely ***y*** in a set $\mathcal{Y}(x)$

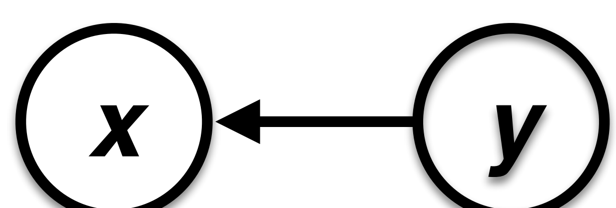
- A **discriminative model** directly models $p(\boldsymbol{y} \mid \boldsymbol{x})$

  logistic/linear/… regressions, MLPs, CRFs, MEMMs, seq2seq(+att)



- A **generative model for a discriminative problem** models $p(\boldsymbol{x}, \boldsymbol{y})$, often by breaking it into $p(\boldsymbol{y})p(\boldsymbol{x} \mid \boldsymbol{y})$

  Naive Bayes, GMMs, HMMs, PCFGs, the IBM translation models

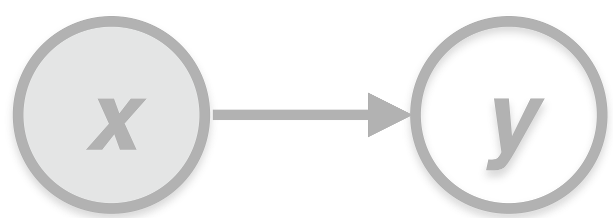


Terminological clarification

- A **discriminative problem**: for some input $\boldsymbol{x}$, find the most likely $\boldsymbol{y}$ in a set $\mathcal{Y}(x)$

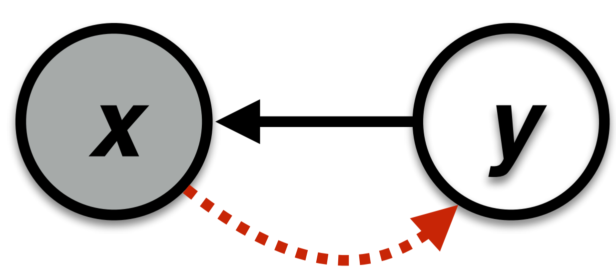
- A **discriminative model** directly models $p(\boldsymbol{y} \mid \boldsymbol{x})$

  logistic/linear/… regressions, MLPs, CRFs, MEMMs, seq2seq(+att)



- A **generative model for a discriminative problem** models $p(\boldsymbol{x}, \boldsymbol{y})$, often by breaking it into $p(\boldsymbol{y})p(\boldsymbol{x} \mid \boldsymbol{y})$

  Naive Bayes, GMMs, HMMs, PCFGs, the IBM translation models

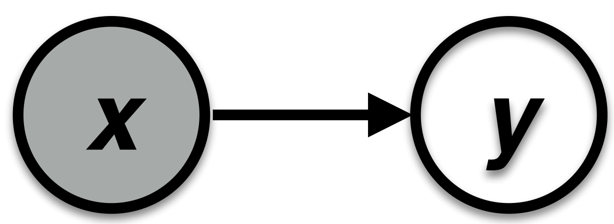


Terminological clarification

- A **discriminative problem**: for some input $\boldsymbol{x}$, find the most likely $\boldsymbol{y}$ in a set $\mathcal{Y}(\boldsymbol{x})$

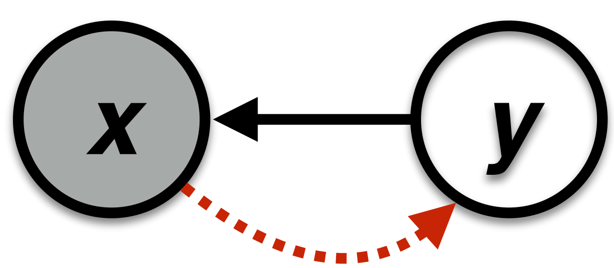
- A **discriminative model** directly models $p(\boldsymbol{y} \mid \boldsymbol{x})$

  logistic/linear/… regressions, MLPs, CRFs, MEMMs, seq2seq(+att)



- A **generative model for a discriminative problem** models $p(\boldsymbol{x}, \boldsymbol{y})$, often by breaking it into $p(\boldsymbol{y})p(\boldsymbol{x} \mid \boldsymbol{y})$

  Naive Bayes, GMMs, HMMs, PCFGs, the IBM translation models



system	BLEU	HTER	mTER
PBSY	25.3	28.0	21.8
HPB	24.6	29.9	23.4
SPB	25.8	29.0	22.7
NMT	31.1*	21.1*	16.2*

Table 2: Overall results on the HE Set: BLEU, computed against the original reference translation, and TER, computed with respect to the targeted post-edit (HTER) and multiple post-edits (mTER).

(Bentivogli et al., 2016)

But **why**?

Exp-ID	Model	Unidi	1st pass Model Size
E8	Proposed	**5.6**	**0.4 GB**
E9	Conventional LFR system	6.7	0.1 GB (AM) + 2.2 GB (PM) + 4.9 GB (LM) = 7.2GB

Table 5: The improved LAS outperforms the conventional LFR system while being more compact. Both models use second-pass rescoring.

(Chiu et al., last week)

Why generative models?
Five reasons

- "Human-like learning" looks more like **model building+inference** than **optimizing pattern recognition functions** (Lake et al., 2015)

Why generative models?
Five reasons

- “Human-like learning” looks more like **model building+inference** than **optimizing pattern recognition functions** (Lake et al., 2015)

- Generative models may be more **sample efficient** than equivalent discriminative models (Ng & Jordan, 2001)

  - In some domains, we can build (relatively) accurate models of data generation → even better sample efficiency

Why generative models?
Five reasons

- "Human-like learning" looks more like **model building+inference** than **optimizing pattern recognition functions** (Lake et al., 2015)

- Generative models may be more **sample efficient** than equivalent discriminative models (Ng & Jordan, 2001)

  - In some domains, we can build (relatively) accurate models of data generation → even better sample efficiency

- **Exploit alternative data/*variables***: zero shot learning, learning from unpaired samples, semisupervised learning, *exploit natural conditional independencies*

Why generative models?
Five reasons

- "Human-like learning" looks more like **model building+inference** than **optimizing pattern recognition functions** (Lake et al., 2015)
- Generative models may be more **sample efficient** than equivalent discriminative models (Ng & Jordan, 2001)
  - In some domains, we can build (relatively) accurate models of data generation → even better sample efficiency
- **Exploit alternative data/*variables***: zero shot learning, learning from unpaired samples, semisupervised learning, *exploit natural conditional independencies*
- **Reduce label bias** when producing sequential outputs

Why generative models?
Five reasons

- “Human-like learning” looks more like **model building+inference** than **optimizing pattern recognition functions** (Lake et al., 2015)
- Generative models may be more **sample efficient** than equivalent discriminative models (Ng & Jordan, 2001)
  - In some domains, we can build (relatively) accurate models of data generation → even better sample efficiency
- **Exploit alternative data/*variables***: zero shot learning, learning from unpaired samples, semisupervised learning, *exploit natural conditional independencies*
- **Reduce label bias** when producing sequential outputs
- **Safety considerations**: model introspection by sampling, generative models “know what they know”

Why generative models?
Five reasons

- “Human-like learning” looks more like **model building+inference** than **optimizing pattern recognition functions** (Lake et al., 2015)
- Generative models may be more **sample efficient** than equivalent discriminative models (Ng & Jordan, 2001)
  - In some domains, we can build (relatively) accurate models of data generation → even better sample efficiency
- **Exploit alternative data/*variables***: zero shot learning, learning from unpaired samples, semisupervised learning, *exploit natural conditional independencies*
- **Reduce label bias** when producing sequential outputs
- **Safety considerations**: model introspection by sampling, generative models “know what they know”

But didn’t we use generative models and give them up for some reason?

Why not generative models?

- To use “generative models for discriminative problems” we must **model complex distributions** (sentences, documents, speech, images)
  - Complex distributions → lots of bad independence assumptions (*naive Bayes, n-grams, HMMs, statistical translation models*)

Why not generative models?

- To use “generative models for discriminative problems” we must **model complex distributions** (sentences, documents, speech, images)
  - Complex distributions → lots of bad independence assumptions (*naive Bayes, n-grams, HMMs, statistical translation models*)
  - **But**: neural networks let the learner figure out their own independence assumptions!

Why not generative models?

- To use “generative models for discriminative problems” we must **model complex distributions** (sentences, documents, speech, images)
  - Complex distributions → lots of bad independence assumptions (*naive Bayes, n-grams, HMMs, statistical translation models*)
  - **But**: neural networks let the learner figure out their own independence assumptions!
- Using generative models require solving **difficult inference problems**
  - Inference problems are especially difficult when you get rid of the “bad independence assumptions”!

Why not generative models?

- To use “generative models for discriminative problems” we must **model complex distributions** (sentences, documents, speech, images)
  - Complex distributions → lots of bad independence assumptions (*naive Bayes, n-grams, HMMs, statistical translation models*)
  - **But**: neural networks let the learner figure out their own independence assumptions!
- Using generative models require solving **difficult inference problems**
  - Inference problems are especially difficult when you get rid of the “bad independence assumptions”!
- You aren’t “**optimizing the task**”!

Why not generative models?

- To use “generative models for discriminative problems” we must **model complex distributions** (sentences, documents, speech, images)
  - Complex distributions → lots of bad independence assumptions (*naive Bayes, n-grams, HMMs, statistical translation models*)
  - **But**: neural networks let the learner figure out their own independence assumptions!
- Using generative models require solving **difficult inference problems**
  - Inference problems are especially difficult when you get rid of the “bad independence assumptions”!
- You aren’t “**optimizing the task**”!

Case studies

- **Text categorization**

$x =$ 

US surrounds new London embassy with a moat

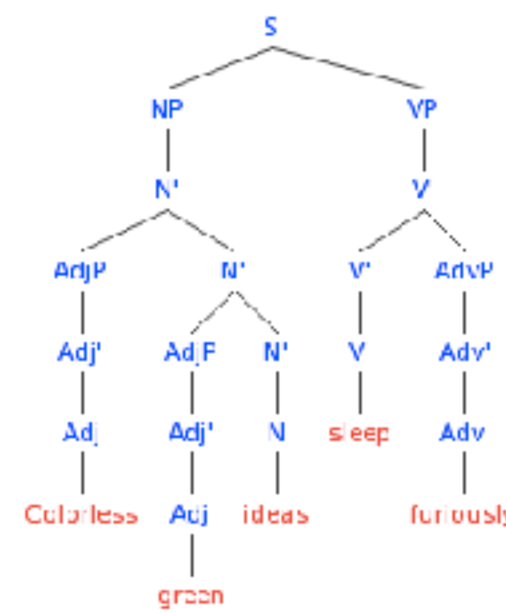
It is central London's first new moated building since the medieval era. The new US embassy in Nine Elms is a paradox, a heavily defended delicate glass box. Its architect calls it a "crystalline radiant beacon"; in fact, it resembles a corporate cube.

It is also one of the world's most expensive embassies, costing a cool $1bn. Remarkably, not a cent of US taxpayer money has been spent. Speaking at the press launch on Wednesday, Ambassador William Moser, principal deputy director of the Bureau of US Overseas Buildings Operations, confirmed that the new building "was entirely funded from the proceeds of real estate sales".

$y =$ POLITICS

- **Syntactic parsing**

$x =$ Colorless green ideas sleep furiously

$y =$ 

- **Sequence to sequence transduction**

$x =$ Welcome to Okinawa $\quad y =$ 沖縄へようこそ。

Case studies

- **Text categorization**

$x =$ 

US surrounds new London embassy with a moat

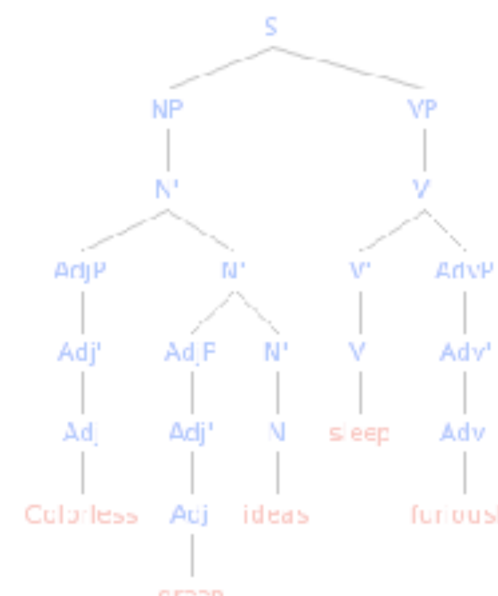
It is central London's first new moated building since the medieval era. The new US embassy in Nine Elms is a paradox, a heavily defended delicate glass box. Its architect calls it a "crystalline radiant beacon"; in fact, it resembles a corporate cube.

It is also one of the world's most expensive embassies, costing a cool $1bn. Remarkably, not a cent of US taxpayer money has been spent. Speaking at the press launch on Wednesday, Ambassador William Moser, principal deputy director of the Bureau of US Overseas Buildings Operations, confirmed that the new building "was entirely funded from the proceeds of real estate sales".

$y =$ POLITICS

- **Syntactic parsing**

$x =$ Colorless green ideas sleep furiously

$y =$ 

- **Sequence to sequence transduction**

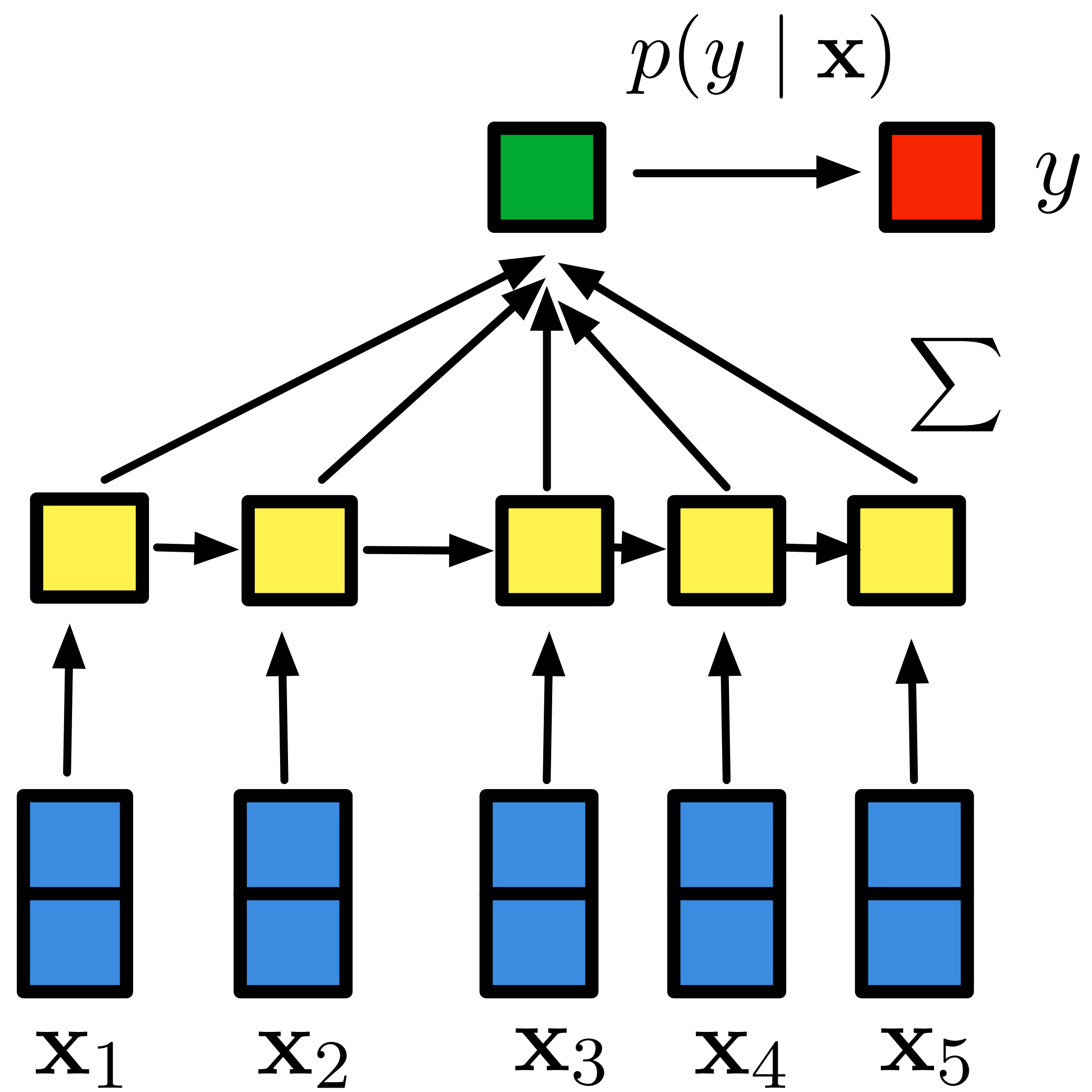
$x =$ Welcome to Okinawa $\quad y =$ 沖縄へようこそ。

Experimental setup
Text categorization

- **Supervised classification**
  - Sample efficiency of a generative-discriminative pair (Ng and Jordan, 2001)
  - How well do generative models do on standard datasets "at scale"?
  - How well do generative models do across a range of data conditions?

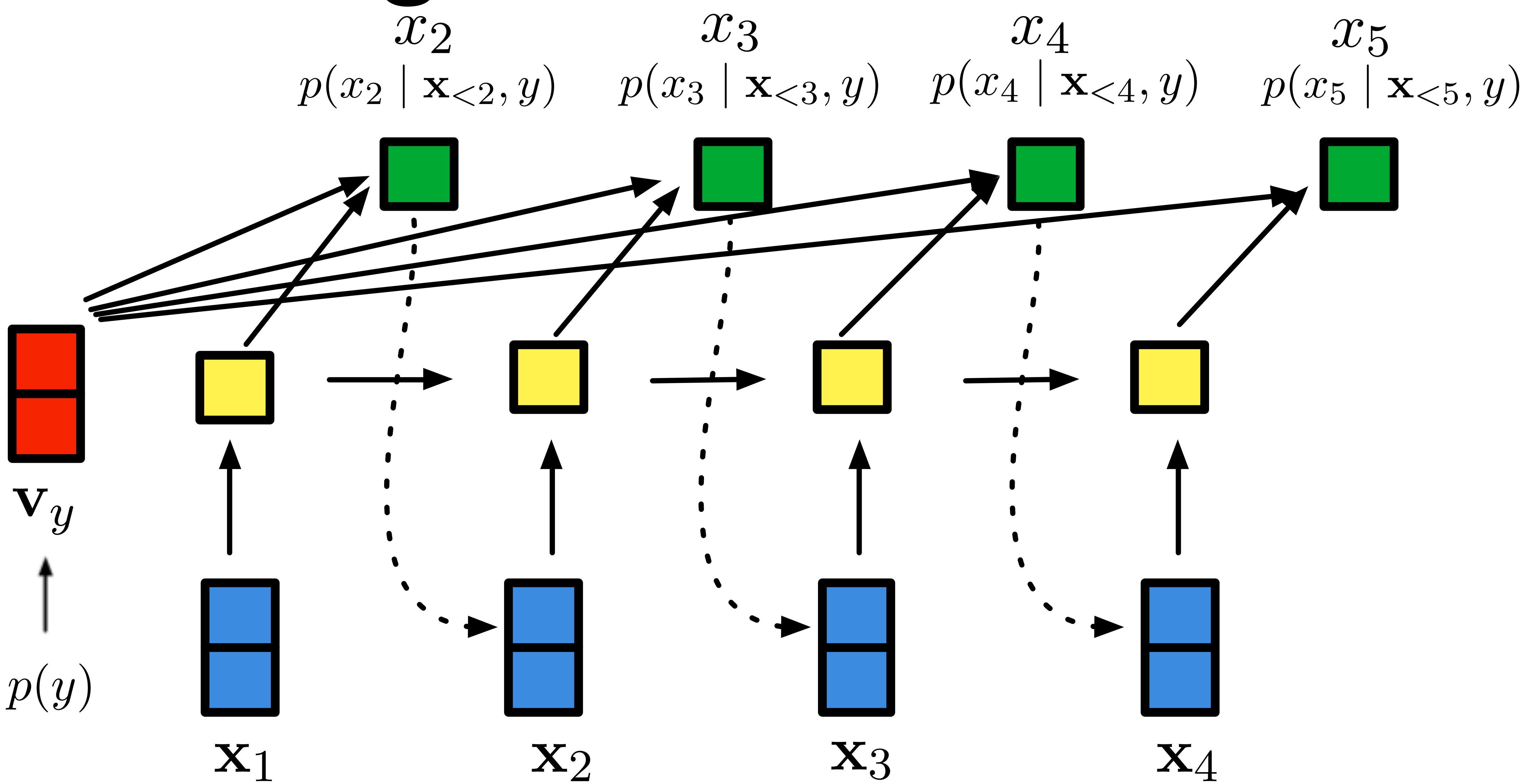
(Yogatama, **D**, et al., arXiv 2017)

Discriminative model
Text categorization



$$\mathcal{L}(\mathbf{W}) = \sum_i \log p(y_i \mid \mathbf{x}_i; \mathbf{W})$$

Generative model
Text categorization



$$\mathcal{L}(\mathbf{W}) = \sum_i \log p(\mathbf{x}_i \mid y_i) p(y_i)$$

Supervised text categorization

	AGNews	DBPedia	Yahoo	Yelp Binary
Bag of Words (Zhang et al., 2015)	88.8	96.6	68.9	92.2
char-CRNN (Xiao and Cho, 2016)	**91.4**	98.6	71.7	94.5
very deep CNN (Conneau et al., 2016)	91.3	**98.7**	**73.4**	**95.7**

Supervised text categorization

	AGNews	DBPedia	Yahoo	Yelp Binary
Bag of Words (Zhang et al., 2015)	88.8	96.6	68.9	92.2
char-CRNN (Xiao and Cho, 2016)	91.4	98.6	71.7	94.5
very deep CNN (Conneau et al., 2016)	91.3	**98.7**	73.4	**95.7**
Discriminative LSTM	**92.1**	**98.7**	**73.7**	92.6

Supervised text categorization

	AGNews	DBPedia	Yahoo	Yelp Binary
Bag of Words (Zhang et al., 2015)	88.8	96.6	68.9	92.2
char-CRNN (Xiao and Cho, 2016)	91.4	98.6	71.7	94.5
very deep CNN (Conneau et al., 2016)	91.3	**98.7**	73.4	**95.7**
Discriminative LSTM	**92.1**	**98.7**	**73.7**	92.6
Naive Bayes	**90.0**	**96.0**	68.7	**86.0**
Kneser-Ney Bayes	89.3	95.4	**69.3**	81.8

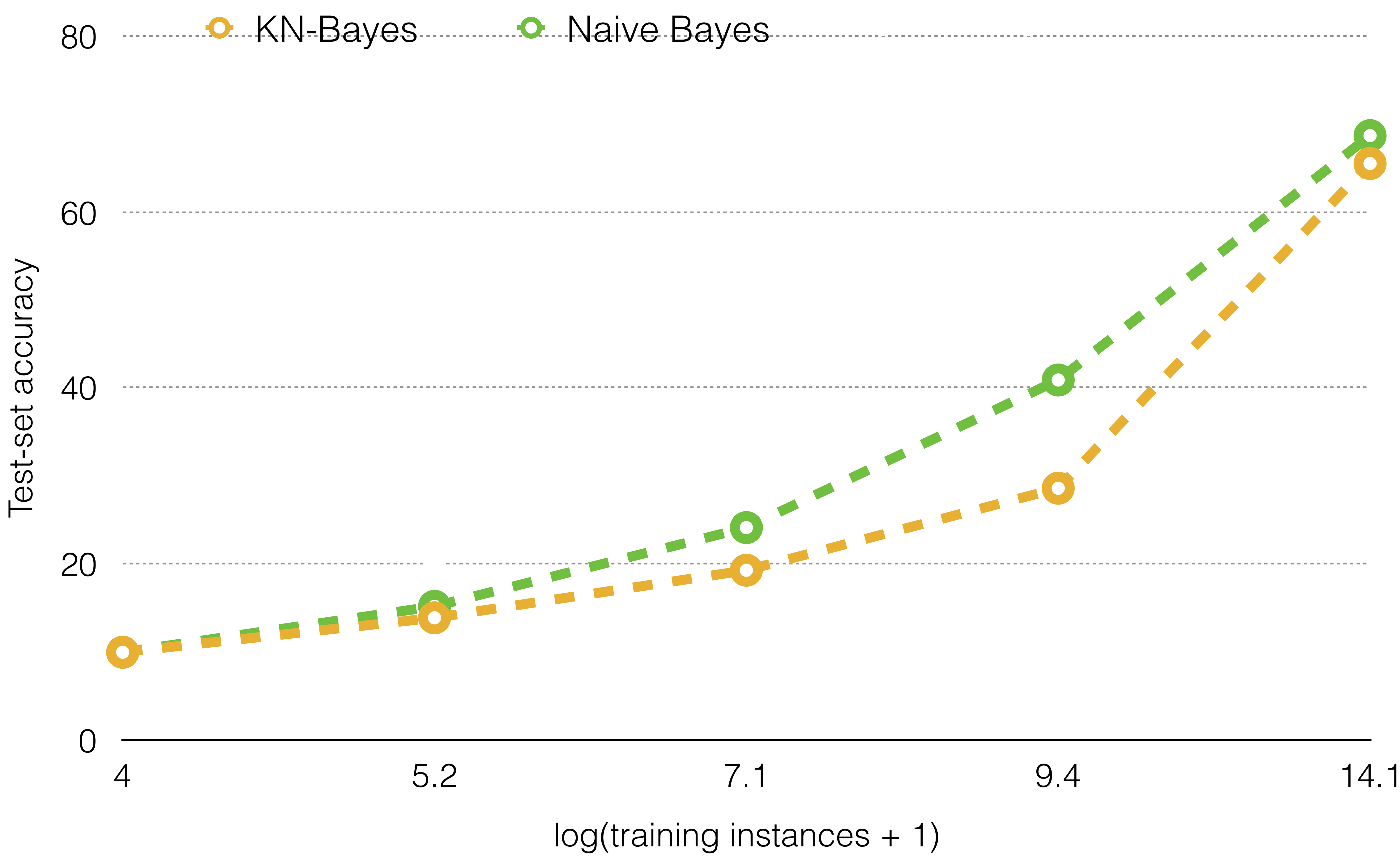
Supervised text categorization

	AGNews	DBPedia	Yahoo	Yelp Binary
Bag of Words (Zhang et al., 2015)	88.8	96.6	68.9	92.2
char-CRNN (Xiao and Cho, 2016)	91.4	98.6	71.7	94.5
very deep CNN (Conneau et al., 2016)	91.3	**98.7**	73.4	**95.7**
Discriminative LSTM	**92.1**	**98.7**	**73.7**	92.6
Naive Bayes	90.0	**96.0**	68.7	86.0
Kneser-Ney Bayes	89.3	95.4	69.3	81.8
Generative LSTM	**90.7**	94.8	**70.5**	**90.0**

Supervised text categorization

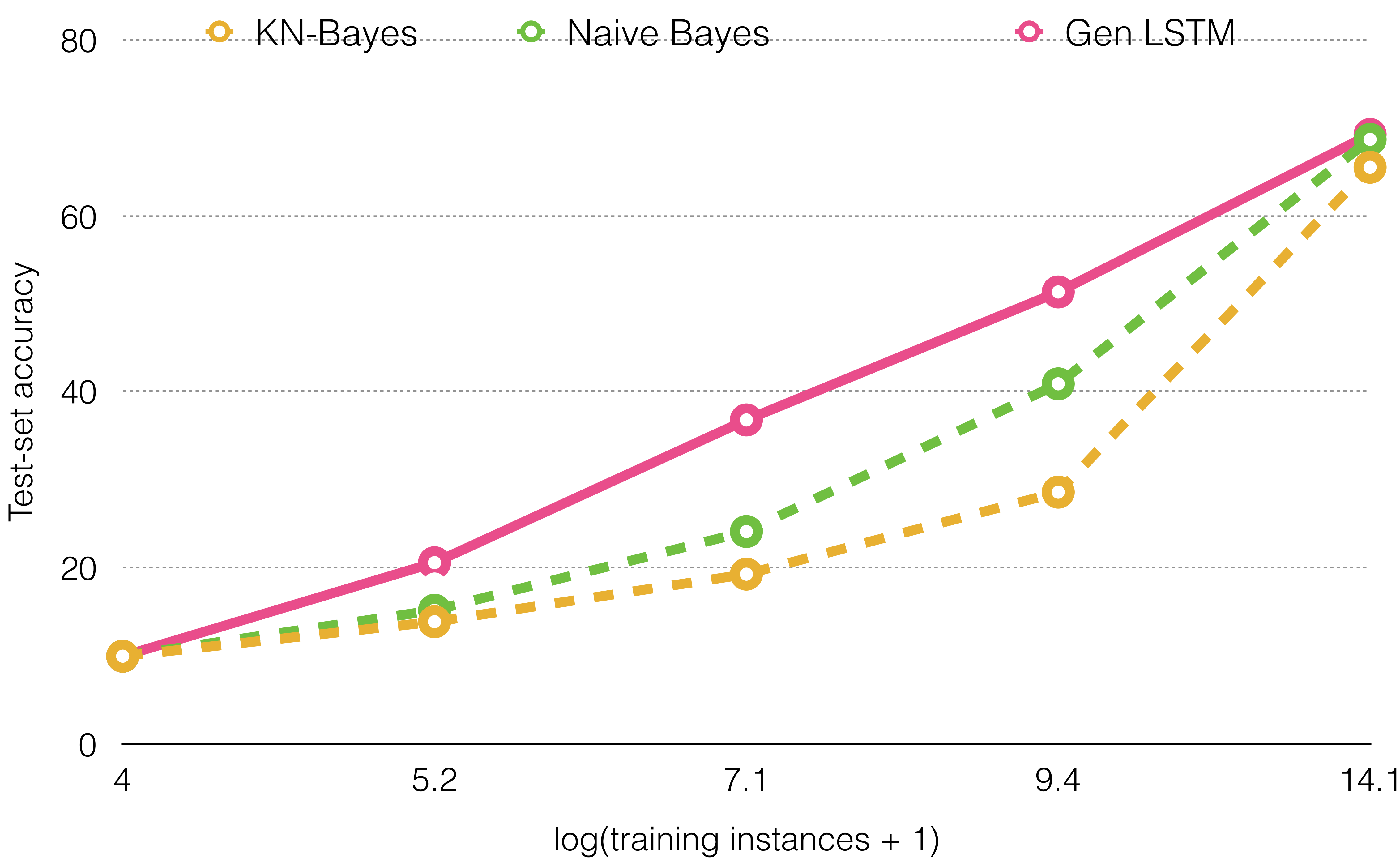
	AGNews	DBPedia	Yahoo	Yelp Binary
Bag of Words (Zhang et al., 2015)	88.8	96.6	68.9	92.2
char-CRNN (Xiao and Cho, 2016)	91.4	98.6	71.7	94.5
very deep CNN (Conneau et al., 2016)	91.3	**98.7**	73.4	**95.7**
Discriminative LSTM	**92.1**	**98.7**	**73.7**	92.6
Naive Bayes	90.0	96.0	68.7	86.0
Kneser-Ney Bayes	89.3	95.4	69.3	81.8
Generative LSTM	90.7	94.8	70.5	90.0

Sample efficiency



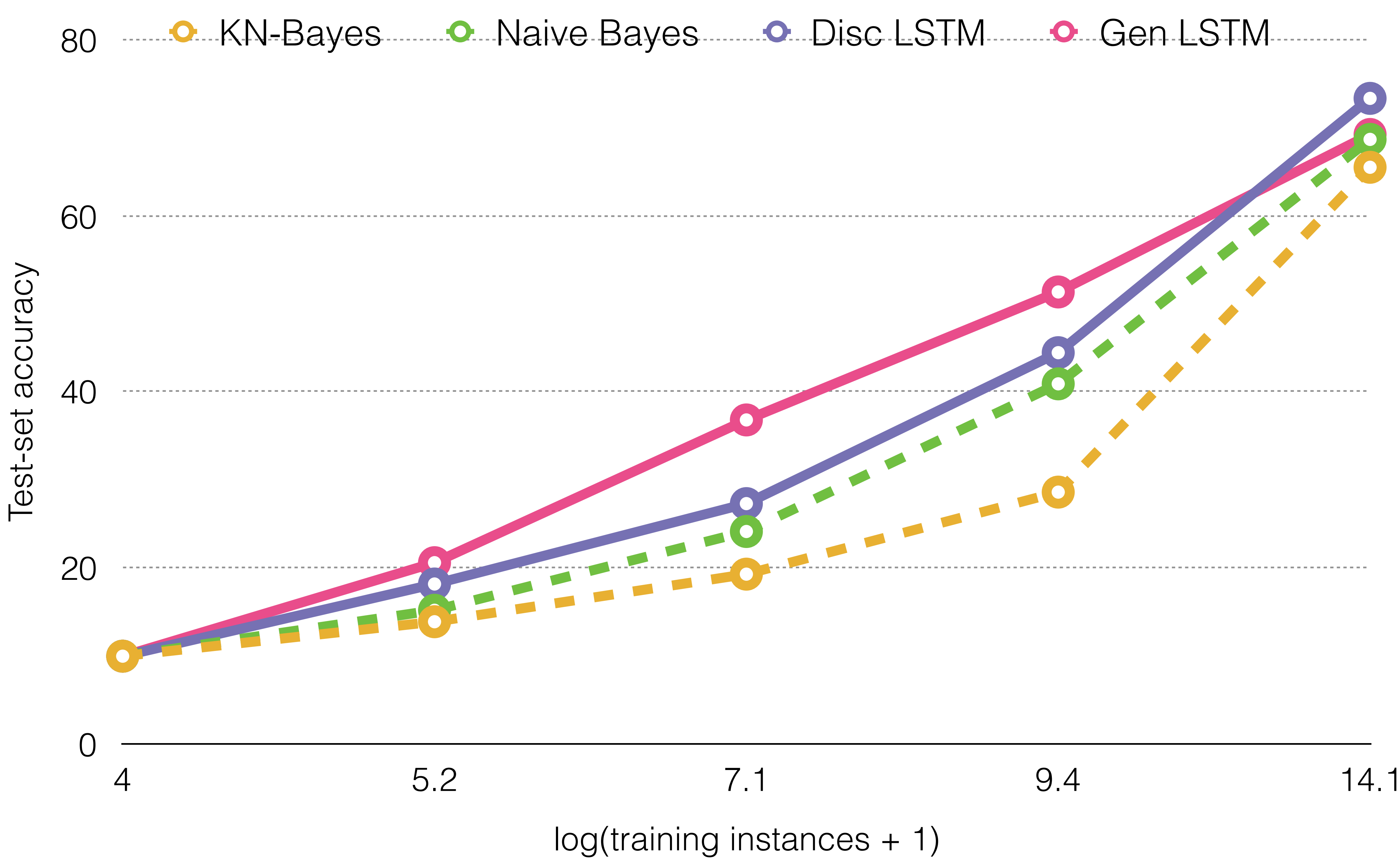
Yahoo! Answers data: 1,395,000 instances / 10 classes

Sample efficiency



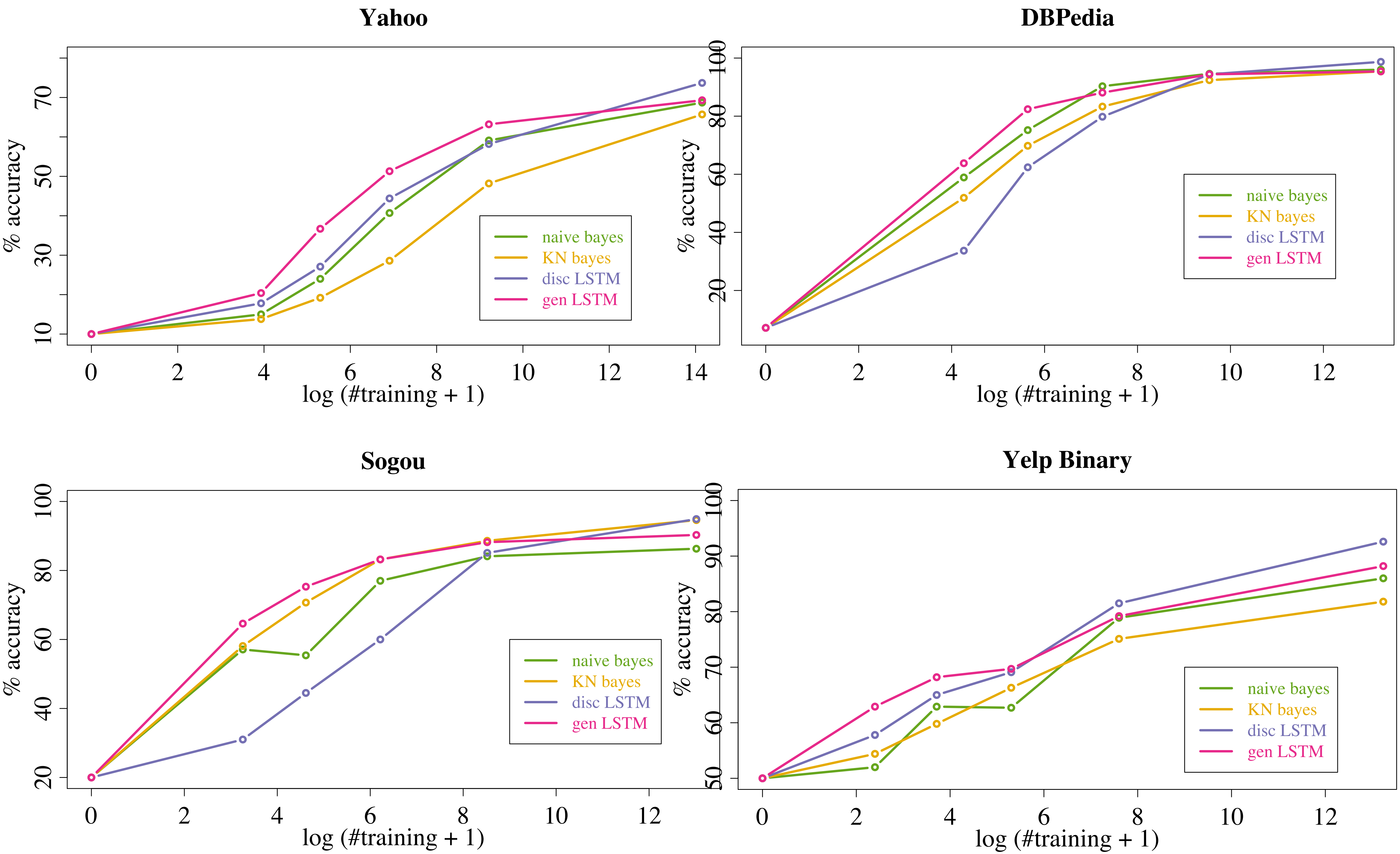
Yahoo! Answers data: 1,395,000 instances / 10 classes

Sample efficiency



Yahoo! Answers data: 1,395,000 instances / 10 classes

Sample efficiency



Discussion

- **Generative** models of text **approach** their **asymptotic errors** more **rapidly** (better in small-data regime).

- **Discriminative** models of text have **lower asymptotic errors**, **faster training and inference time**, **and a good estimate of** $p(\mathbf{x})$

- The downside is **inference is expensive**. We have to evaluate the likelihood of the document for every class!

Case studies

- **Text categorization**

$x =$

US surrounds new London embassy with a moat

It is central London's first new moated building since the medieval era. The new US embassy in Nine Elms is a paradox, a heavily defended delicate glass box. Its architect calls it a "crystalline radiant beacon"; in fact, it resembles a corporate cube.

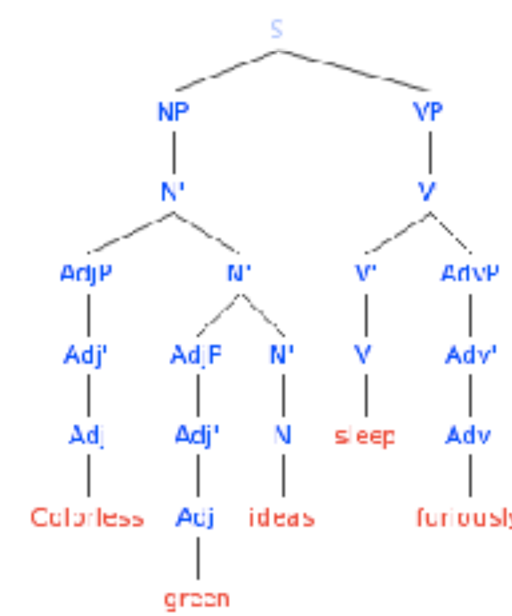
It is also one of the world's most expensive embassies, costing a cool $1bn. Remarkably, not a cent of US taxpayer money has been spent. Speaking at the press launch on Wednesday, Ambassador William Moser, principal deputy director of the Bureau of US Overseas Buildings Operations, confirmed that the new building "was entirely funded from the proceeds of real estate sales".

$y =$ POLITICS

- **Syntactic parsing**

$\boldsymbol{x} =$ Colorless green ideas sleep furiously

$\boldsymbol{y} =$



- **Sequence to sequence transduction**

$x =$ Welcome to Okinawa $\quad y =$ 沖縄へようこそ。

Syntactic parsing

Recurrent Neural Net **Grammars**

(**D**, et al., ACL 2016; Kuncoro, **D**, et al., EACL 2017)

Syntactic parsing
Recurrent Neural Net Grammars

- Generate **symbols** sequentially using an **RNN**

(**D**, et al., ACL 2016; Kuncoro, **D**, et al., EACL 2017)

Syntactic parsing
Recurrent Neural Net Grammars

- Generate **symbols** sequentially using an **RNN**
- Add some **control symbols** to rewrite the history occasionally
  - Occasionally **compress** a sequence into a **constituent**
  - RNN predicts next terminal/control symbol based on the history of compressed elements and non-compressed terminals

(**D**, et al., ACL 2016; Kuncoro, **D**, et al., EACL 2017)

Syntactic parsing
Recurrent Neural Net Grammars

- Generate **symbols** sequentially using an **RNN**
- Add some **control symbols** to rewrite the history occasionally
    - Occasionally **compress** a sequence into a **constituent**
    - RNN predicts next terminal/control symbol based on the history of compressed elements and non-compressed terminals
- This is a **top-down, left-to-right generation** of a tree+sequence (other traversal orders are possible)

(**D**, et al., ACL 2016; Kuncoro, **D**, et al., EACL 2017)

Example derivation

The hungry cat meows loudly

stack	action	probability
	NT(S)	$p(\textsc{nt}(\mathrm{S}) \mid \textsc{top})$
(S	**NT(NP)**	$p(\textsc{nt}(\mathrm{NP}) \mid (\mathrm{S})$
(S (NP	**GEN(*The*)**	$p(\textsc{gen}(\mathit{The}) \mid (\mathrm{S}, (\mathrm{NP})$
(S (NP *The*	**GEN(*hungry*)**	$p(\textsc{gen}(\mathit{hungry}) \mid (\mathrm{S}, (\mathrm{NP}, \mathit{The})$
(S (NP *The* *hungry*	**GEN(*cat*)**	$p(\textsc{gen}(\mathit{cat}) \mid \ldots)$
(S (NP *The* *hungry* *cat*	**REDUCE**	$p(\textsc{reduce} \mid \ldots)$
(S (NP *The* *hungry* *cat*)		
(S (NP *The hungry cat*)		

Compress "The hungry cat" into a single composite symbol

stack	action	probability
	NT(S)	$p(\textsc{nt}(\text{S}) \mid \textsc{top})$
(S	**NT**(NP)	$p(\textsc{nt}(\text{NP}) \mid (\text{S})$
(S (NP	**GEN**(*The*)	$p(\textsc{gen}(\textit{The}) \mid (\text{S}, (\text{NP})$
(S (NP *The*	**GEN**(*hungry*)	$p(\textsc{gen}(\textit{hungry}) \mid (\text{S}, (\text{NP}, \textit{The})$
(S (NP *The* *hungry*	**GEN**(*cat*)	$p(\textsc{gen}(\textit{cat}) \mid \ldots)$
(S (NP *The* *hungry* *cat*	**REDUCE**	$p(\textsc{reduce} \mid \ldots)$
(S (NP *The hungry cat*)	**NT**(VP)	$p(\textsc{nt}(\text{VP}) \mid (\text{S}, (\text{NP}\ \textit{The hungry cat})$
(S (NP *The hungry cat*) (VP	**GEN**(*meows*)	
(S (NP *The hungry cat*) (VP *meows*	**REDUCE**	
(S (NP *The hungry cat*) (VP *meows*)	**GEN**(.)	
(S (NP *The hungry cat*) (VP *meows*) .	**REDUCE**	
(S (NP *The hungry cat*) (VP *meows*) .)		

Deriving the model

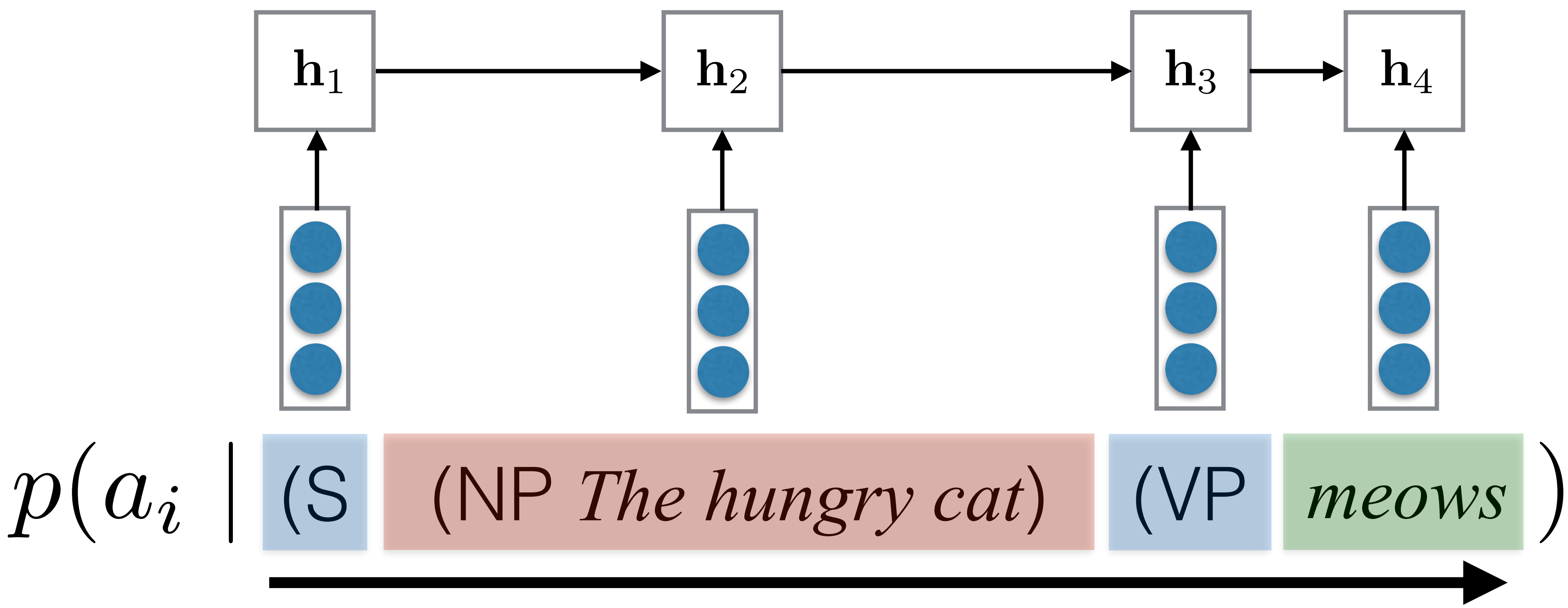
- Valid (***tree***, ***string***) pairs are in bijection to valid sequences of actions (specifically, the DFS, left-to-right traversal of the trees)

- Every stack configuration perfectly encodes the complete history of actions.

- Therefore, the probability decomposition is justified by the chain rule, i.e.

$$
\begin{aligned}
p(\boldsymbol{x}, \boldsymbol{y}) &= p(actions(\boldsymbol{x}, \boldsymbol{y})) && \text{(prop 1)} \\
p(actions(\boldsymbol{x}, \boldsymbol{y})) &= \prod_i p(a_i \mid \boldsymbol{a}_{<i}) && \text{(chain rule)} \\
&= \prod_i p(a_i \mid stack(\boldsymbol{a}_{<i})) && \text{(prop 2)}
\end{aligned}
$$

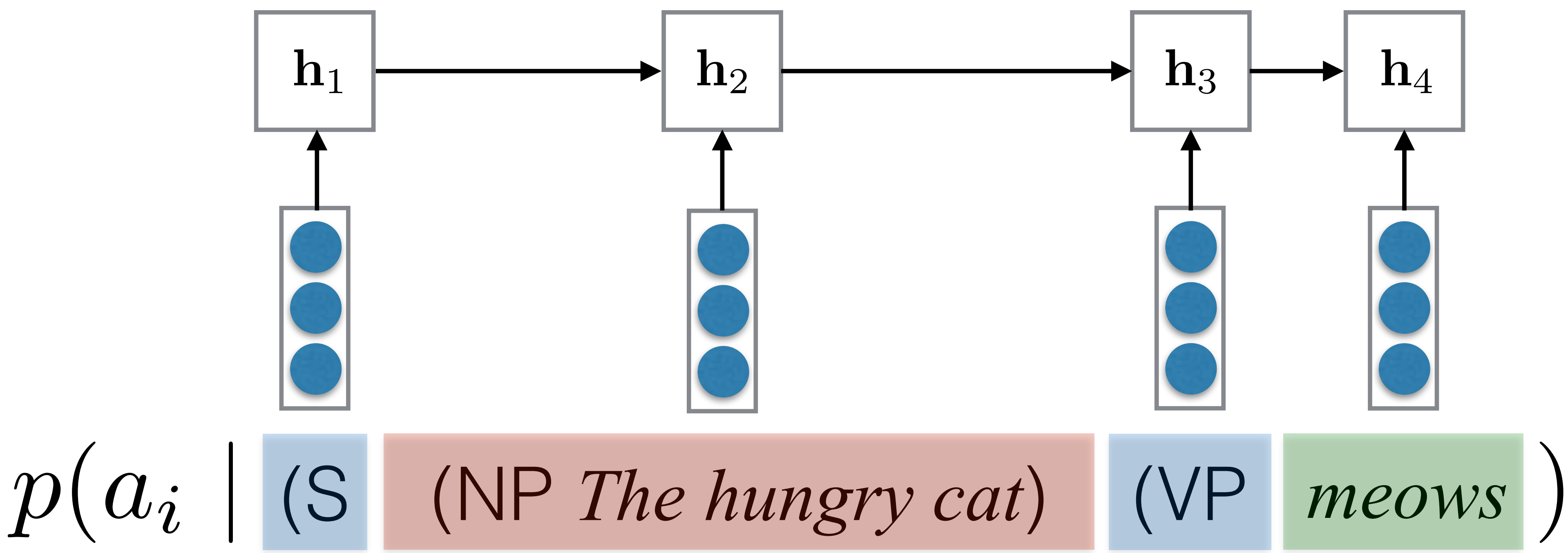
Modeling the next action



1. unbounded depth

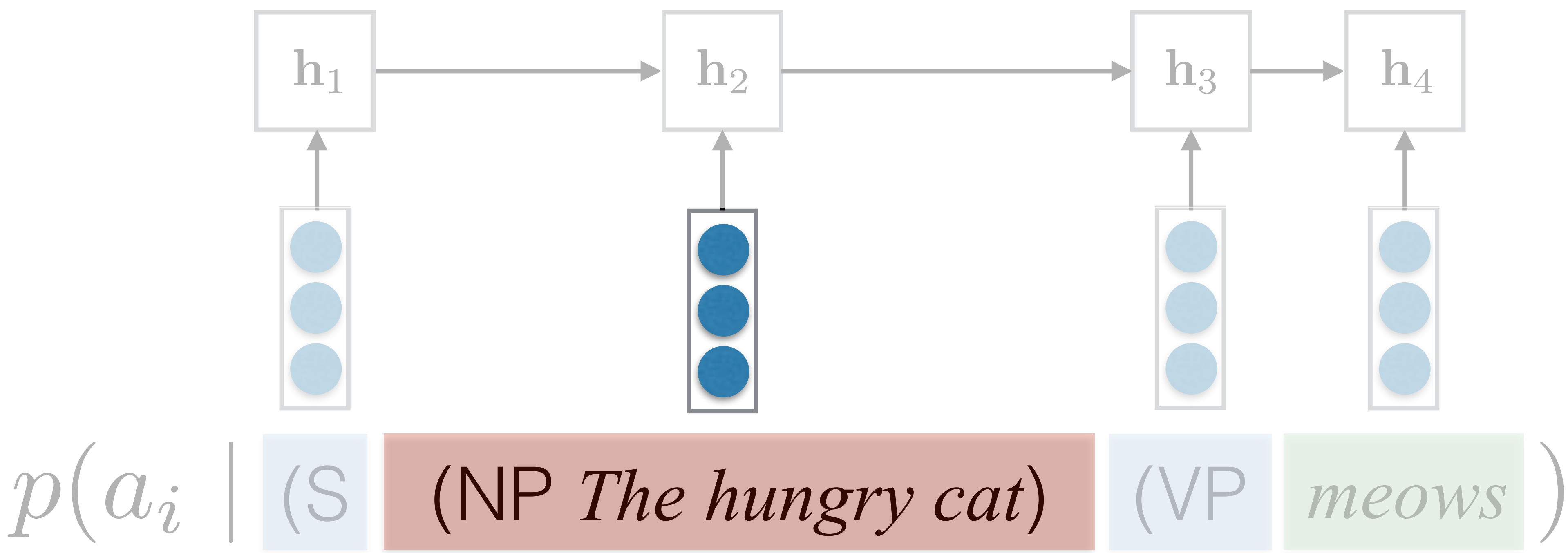
1. Unbounded depth → recurrent neural nets

Modeling the next action



1. Unbounded depth → recurrent neural nets

Modeling the next action

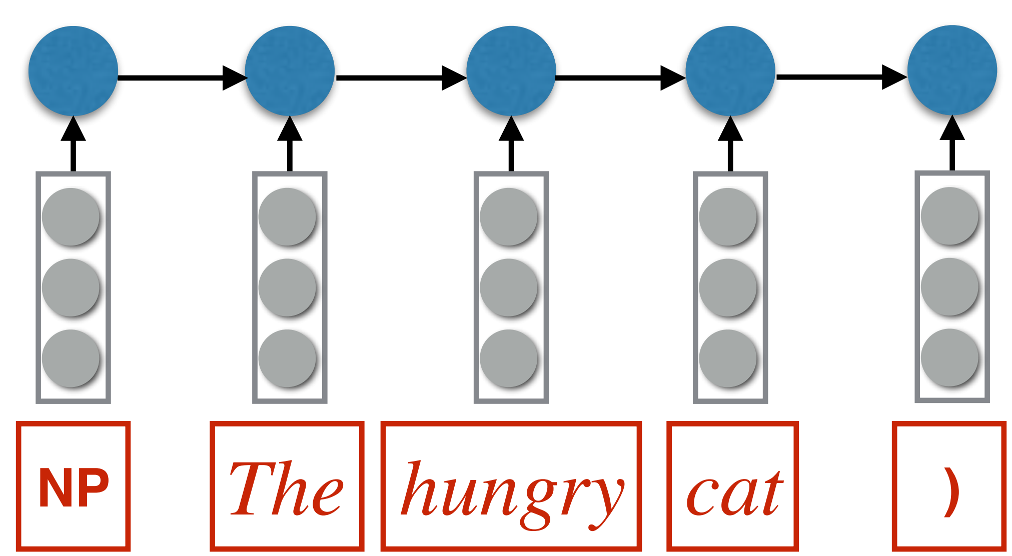


2. arbitrarily complex trees

1. Unbounded depth → recurrent neural nets
2. Arbitrarily complex trees → recursive neural nets

Syntactic composition

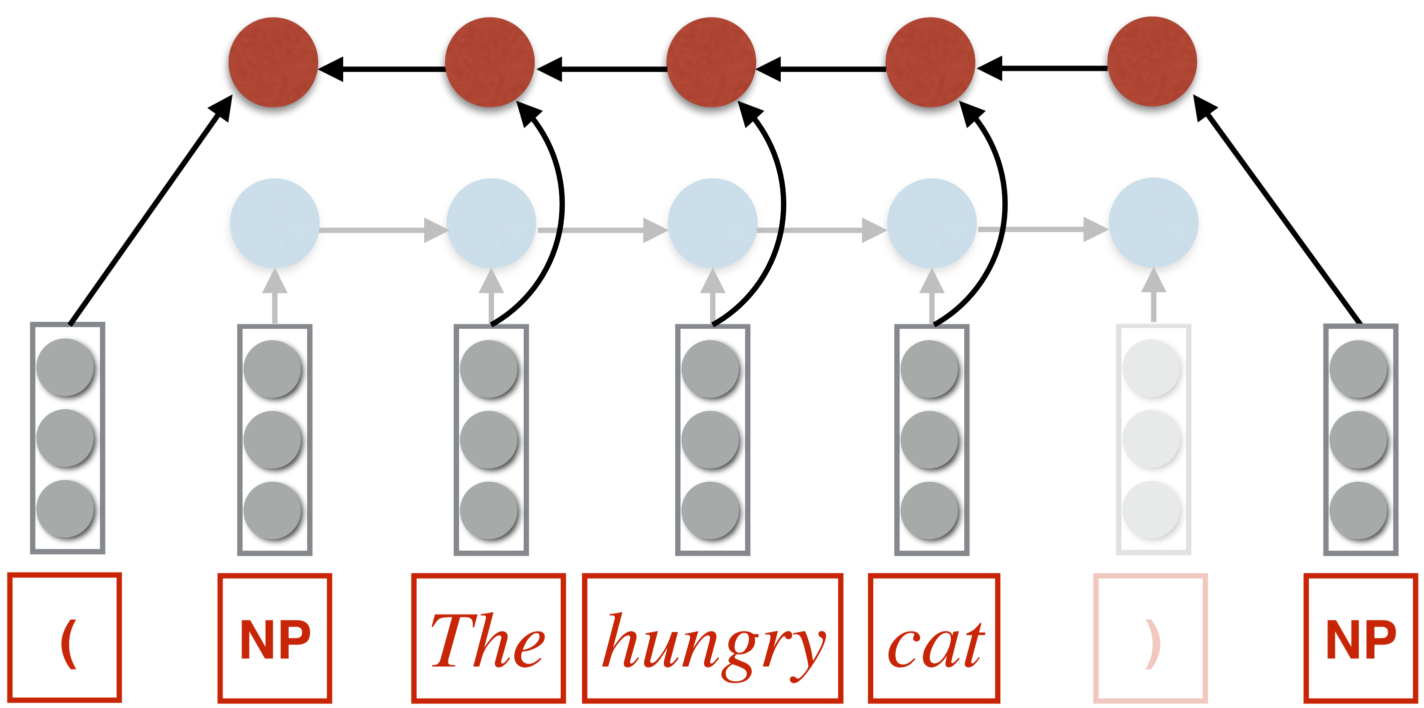
Need representation for: (NP *The hungry cat*)



What head type?

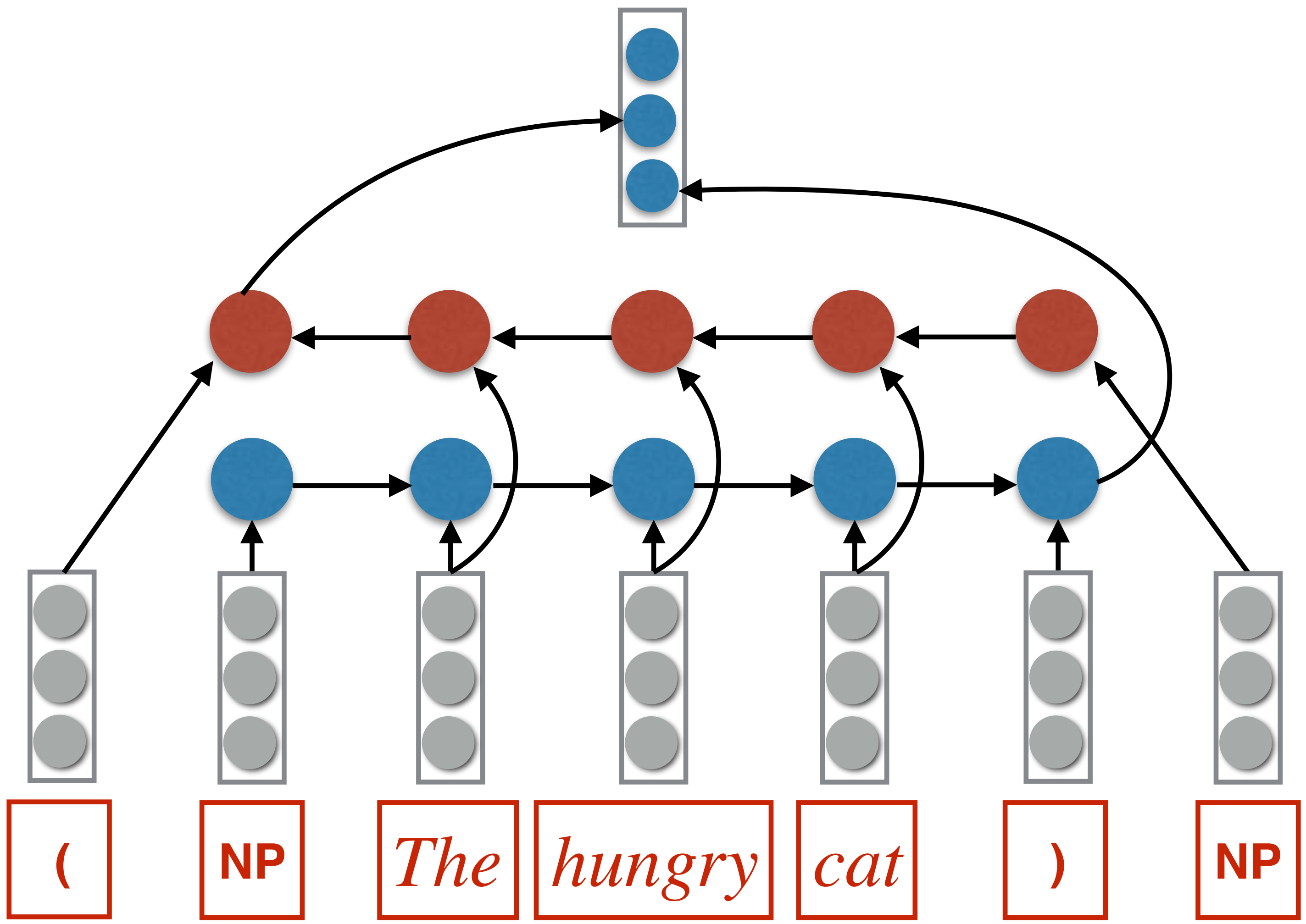
Syntactic composition

Need representation for: (NP *The hungry cat*)



Syntactic composition

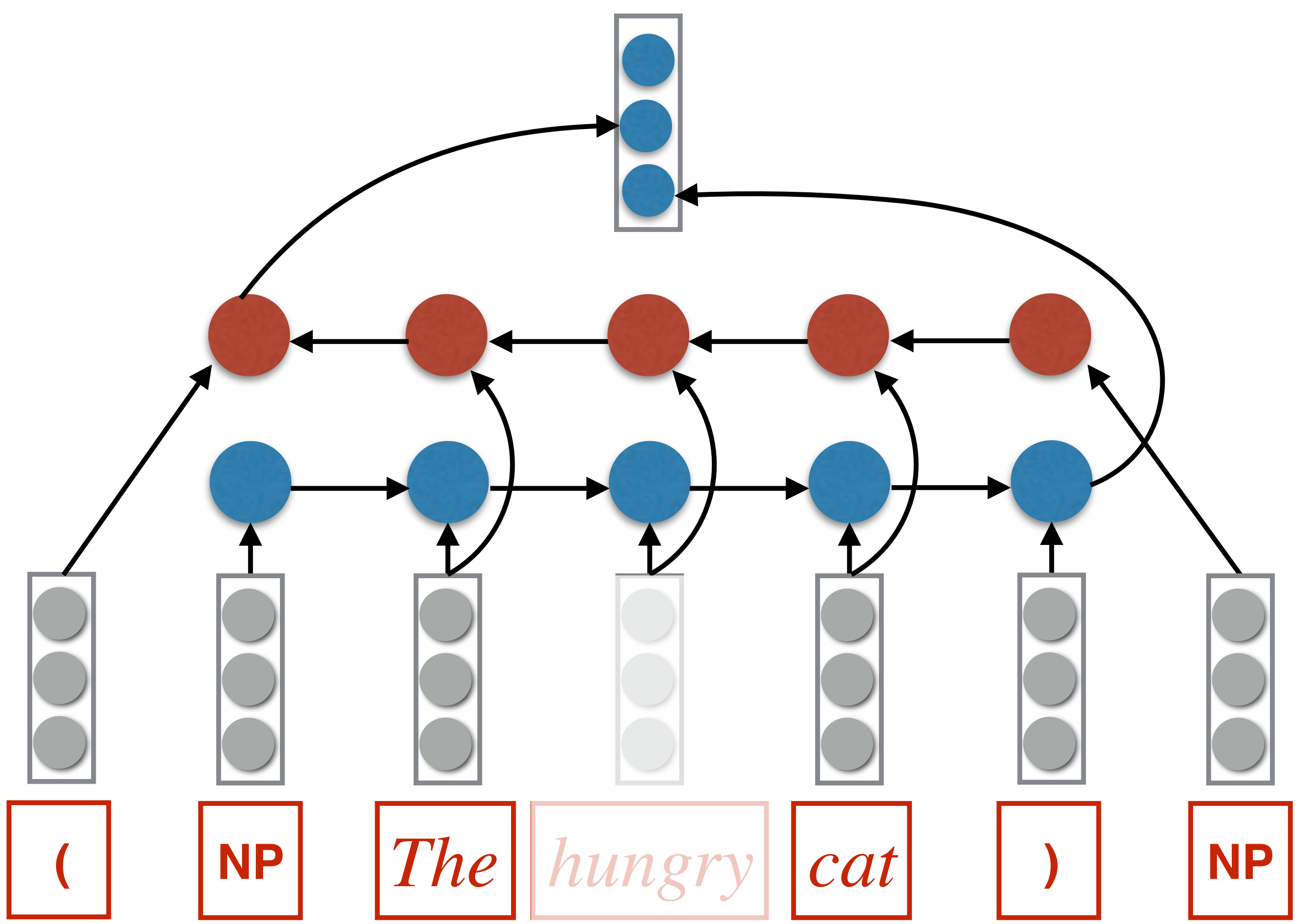
Need representation for: (NP *The hungry cat*)



Syntactic composition
Recursion

Need representation for: (NP *The hungry cat*)

(NP *The* (ADJP *very hungry*) *cat*)

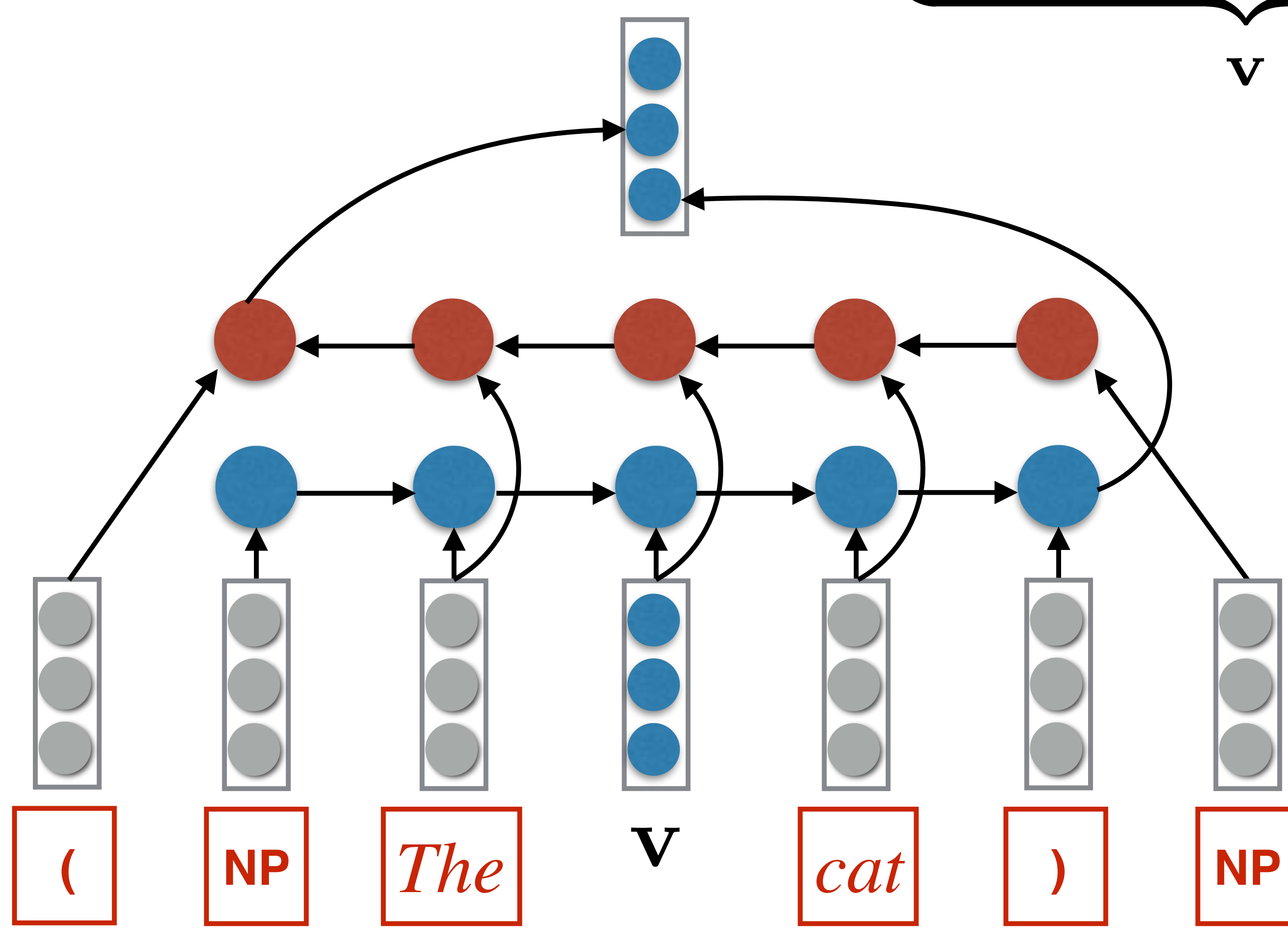


Syntactic composition
Recursion

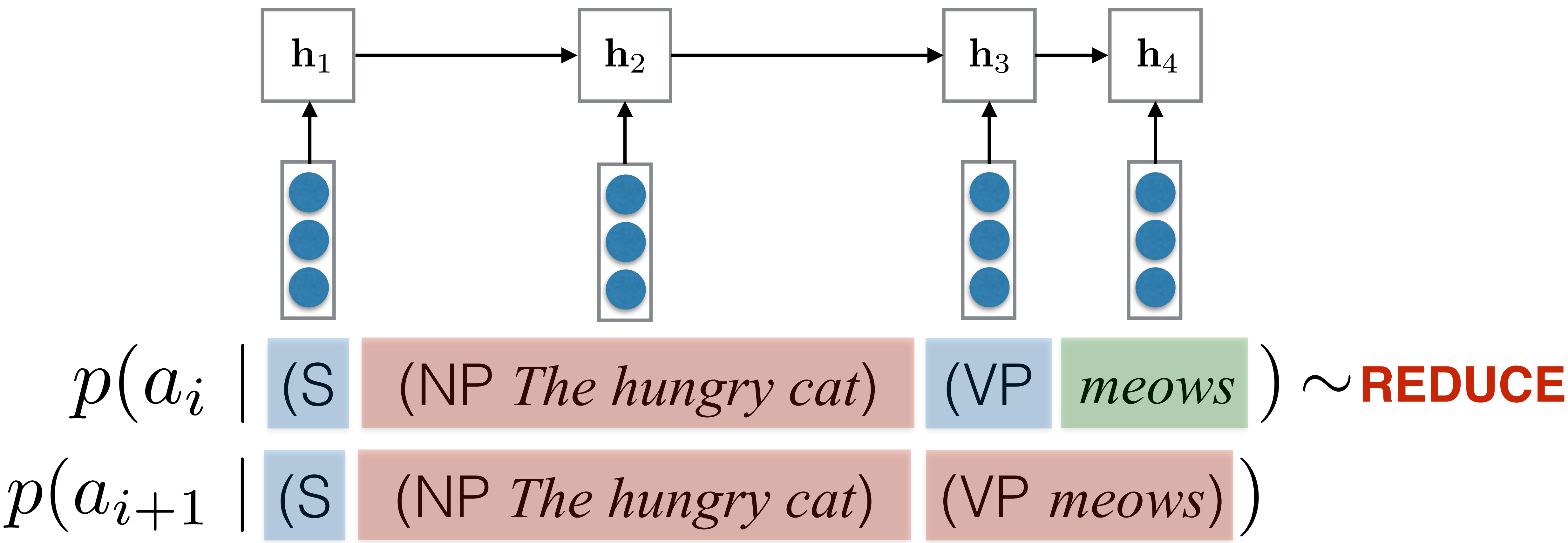
Need representation for: (NP *The hungry cat*)

(NP *The* (ADJP *very hungry*) *cat*)

$\mathbf{v}$



Modeling the next action



3. limited updates

1. Unbounded depth → recurrent neural nets
2. Arbitrarily complex trees → recursive neural nets
3. Limited updates to state → stack RNNs

(**D**, et al., ACL 2015; Ballesteros, **D**, et al., EMNLP 2015)

Inference

- In text categorization, it was not really a problem to exhaustively evaluate all candidate *y*'s.
- Here, we can't do that — we have $O(2^{|\boldsymbol{x}|})$ candidates!
- Outline of the solution
  - Learn a tractable instrumental distribution, $q(\boldsymbol{y} \mid \boldsymbol{x})$, which approximates the posterior over trees
  - Use **importance sampling** to solve the inference problems (maximization, marginalization) we care about

Results: **Parsing**

	Type	F1
Petrov and Klein (2007)	Gen	90.1
Shindo et al (2012) Single model	Gen	91.1
Vinyals et al (2015) PTB only	Disc	90.5
Shindo et al (2012) Ensemble	Gen+*Ensemble*	92.4
Vinyals et al (2015) Semisupervised	Disc+*SemiSup*	92.8
Discriminative PTB only	Disc	91.7
Generative PTB only	Gen	93.6

Results: **Parsing**

	Type	F1
Petrov and Klein (2007)	Gen	90.1
Shindo et al (2012) Single model	Gen	91.1
Vinyals et al (2015) PTB only	Disc	90.5
Shindo et al (2012) Ensemble	Gen+*Ensemble*	92.4
Vinyals et al (2015) Semisupervised	Disc+*SemiSup*	92.8
Discriminative PTB only	Disc	91.7
Generative PTB only	Gen	93.6
Choe and Charniak (2016) Semisupervised	Gen *+SemiSup*	**93.8**
Fried et al. (2017)	Gen+*Semi* *+Ensemble*	**94.7**

Discussion

- RNNGs are effective both for **modeling language** and **parsing**
- Generative parser outperforms discriminative parser
  - Expectation: the discriminative model would do better with more data
  - We are in the “generative” regime!

Case studies

- **Text categorization**

$x =$ 

US surrounds new London embassy with a moat

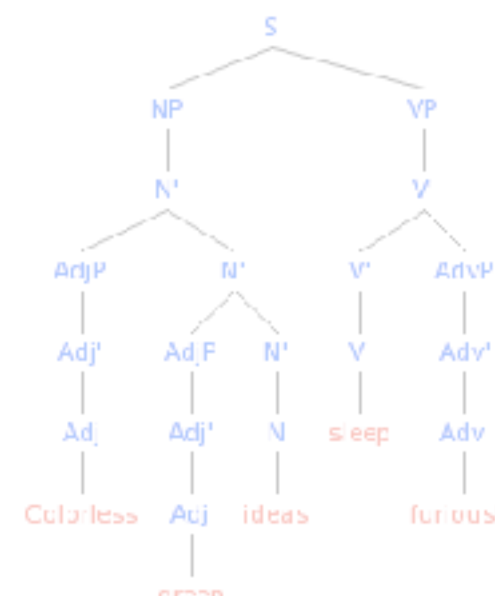
It is central London's first new moated building since the medieval era. The new US embassy in Nine Elms is a paradox, a heavily defended delicate glass box. Its architect calls it a "crystalline radiant beacon"; in fact, it resembles a corporate cube.

It is also one of the world's most expensive embassies, costing a cool $1bn. Remarkably, not a cent of US taxpayer money has been spent. Speaking at the press launch on Wednesday, Ambassador William Moser, principal deputy director of the Bureau of US Overseas Buildings Operations, confirmed that the new building "was entirely funded from the proceeds of real estate sales".

$y =$ POLITICS

- **Syntactic parsing**

$x =$ Colorless green ideas sleep furiously

$\boldsymbol{y} =$ 

- **Sequence to sequence transduction**

$\boldsymbol{x} =$ Welcome to Okinawa $\boldsymbol{y} =$ 沖縄へようこそ。

Seq2Seq Modeling
Direct model

$$p(\boldsymbol{y} \mid \boldsymbol{x}) = \text{ConditionalRNNLM}(\boldsymbol{x})$$
$$= \prod_i p(y_i \mid \boldsymbol{x}, \boldsymbol{y}_{<i})$$

- State of the art performance in most applications
- Two serious problems that concern us:
  - Nontrivial to use "unpaired" samples of ***x*** or ***y*** to train the model
  - "Explaining away effects" - models like this learn to ignore "inconvenient" inputs (i.e., ***x***), in favor of high probability continuations of an output prefix ($\boldsymbol{y}_{<i}$)

(Yu, **D**, et al., ICLR 2017)

Seq2Seq Modeling
What is label bias?

Label bias is a species of “explaining away” that causes trouble in directed (locally normalized) models.

a	b	c	→	x	y	z
a	b	c’	→	x	y	z
a	b’	c	→	x	y	z
		d	→	w		
a	b’	d	→	x	y	z

Seq2Seq Modeling
Generative model

$$p(\boldsymbol{y} \mid \boldsymbol{x}) \propto p(\boldsymbol{y}) \times p(\boldsymbol{x} \mid \boldsymbol{y})$$

Seq2Seq Modeling
Generative model

$$p(\boldsymbol{y} \mid \boldsymbol{x}) \propto \underbrace{p(\boldsymbol{y})}_{\text{“Source model”}} \times \underbrace{p(\boldsymbol{x} \mid \boldsymbol{y})}_{\text{“Channel model”}}$$

Seq2Seq Modeling
Generative model

$$p(\boldsymbol{y} \mid \boldsymbol{x}) \propto \underbrace{p(\boldsymbol{y})}_{\text{“Source model”}} \times \underbrace{p(\boldsymbol{x} \mid \boldsymbol{y})}_{\text{“Channel model”}}$$

$$\boldsymbol{y} \sim p(\boldsymbol{y})$$

The world is colorful because of the Internet.

Seq2Seq Modeling
Generative model

$$p(\boldsymbol{y} \mid \boldsymbol{x}) \propto \underbrace{p(\boldsymbol{y})}_{\text{“Source model”}} \times \underbrace{p(\boldsymbol{x} \mid \boldsymbol{y})}_{\text{“Channel model”}}$$

$$\boldsymbol{y} \sim p(\boldsymbol{y})$$

The world is colorful because of the Internet.

Seq2Seq Modeling
Generative model

$$p(\boldsymbol{y} \mid \boldsymbol{x}) \propto p(\boldsymbol{y}) \times p(\boldsymbol{x} \mid \boldsymbol{y})$$

"Source model" **"Channel model"**

$$\boldsymbol{y} \sim p(\boldsymbol{y})$$

The world is colorful because of the Internet.

$$\boldsymbol{x} \sim p(\boldsymbol{x} \mid \boldsymbol{y})$$

世界はインターネットのためにカラフルです。

Seq2Seq Modeling
Generative model

$$p(\boldsymbol{y} \mid \boldsymbol{x}) \propto p(\boldsymbol{y}) \times p(\boldsymbol{x} \mid \boldsymbol{y})$$

"Source model" **"Channel model"**

$$\boldsymbol{y} \sim p(\boldsymbol{y})$$

The world is colorful because of the Internet.

$$\boldsymbol{x} \sim p(\boldsymbol{x} \mid \boldsymbol{y})$$

世界はインターネットのためにカラフルです。

👍 Source model can be estimated from unpaired ***y***'s

Seq2Seq Modeling
Generative model

$$p(\boldsymbol{y} \mid \boldsymbol{x}) \propto p(\boldsymbol{y}) \times p(\boldsymbol{x} \mid \boldsymbol{y})$$

"Source model" **"Channel model"**

$$\boldsymbol{y} \sim p(\boldsymbol{y})$$

The world is colorful because of the Internet.

$$\boldsymbol{x} \sim p(\boldsymbol{x} \mid \boldsymbol{y})$$

世界はインターネットのためにカラフルです。

$$\hat{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}} p(\boldsymbol{y}) p(\boldsymbol{x} \mid \boldsymbol{y})$$

👍 Inference model form avoids explaining away of inputs ("label bias").

Seq2Seq Modeling
Generative model

- Question: Can we use **neural network component models** without bad independence assumptions?
    - **Training — straightforward**
    - **Decoding — challenging**

Decoding

- Some bad initial results
  - The IS algorithm we proposed hurt us unless the number of samples (k) was massive
  - Reranking an k-best list from a direct model didn't help unless k was even bigger
- Question: **can we develop a left-to-right decoder for a noisy channel MT model?**

Decoding
Direct vs. generative

Direct model:

$$\begin{aligned}
&\text{while } y_i \neq \text{STOP}: \\
&\qquad \hat{y}_i = \arg\max_{y} p(y \mid \boldsymbol{x}, \hat{\boldsymbol{y}}_{<i}) \\
&\qquad i \leftarrow i + 1
\end{aligned}$$

Decoding
Direct vs. generative

Direct model:

$$
\begin{aligned}
&\text{while } y_i \neq \text{STOP}: \\
&\quad \hat{y}_i = \arg\max_y \boxed{p(y \mid \boldsymbol{x}, \hat{\boldsymbol{y}}_{<i})} \\
&\quad i \leftarrow i + 1
\end{aligned}
$$

👍 Chain rule!

Decoding
Direct vs. generative

Direct model:

$$
\begin{aligned}
&\text{while } y_i \neq \text{STOP}: \\
&\qquad \hat{y}_i = \arg\max_{y} \boxed{p(y \mid \boldsymbol{x}, \hat{\boldsymbol{y}}_{<i})} \\
&\qquad i \leftarrow i + 1
\end{aligned}
$$

👍 Chain rule!

Not perfect, but $\hat{\boldsymbol{y}} \approx \arg\max_{\boldsymbol{y}} p(\boldsymbol{y} \mid \boldsymbol{x})$

(Compare to using greedy decoding with MEMMs)

Decoding
Direct vs. generative

Generative model (naive):

$$
\begin{aligned}
&\text{while } y_i \neq \text{STOP}: \\
&\qquad \hat{y}_i = \arg\max_{y} p(y \mid \hat{\boldsymbol{y}}_{<i})\, p(\boldsymbol{x} \mid \hat{\boldsymbol{y}}_{<i}, y) \\
&\qquad i \leftarrow i + 1
\end{aligned}
$$

Decoding
Direct vs. generative

Generative model (naive):

$$\begin{aligned}&\text{while } y_i \neq \text{STOP}:\\&\quad \hat{y}_i = \arg\max_y \boxed{p(y \mid \hat{\boldsymbol{y}}_{<i})}\, p(\boldsymbol{x} \mid \hat{\boldsymbol{y}}_{<i}, y)\\&\quad i \leftarrow i + 1\end{aligned}$$

👍 Chain rule!

Decoding
Direct vs. generative

Generative model (naive):

$$\begin{aligned}&\text{while } y_i \neq \text{STOP}:\\&\quad \hat{y}_i = \arg\max_y p(y \mid \hat{\boldsymbol{y}}_{<i}) \boxed{p(\boldsymbol{x} \mid \hat{\boldsymbol{y}}_{<i}, y)}\\&\quad i \leftarrow i + 1\end{aligned}$$

👎 Probability doesn't work like this.

Decoding
Direct vs. generative

Outline of solution:

Introduce a latent variable ***z*** that determines when enough of the conditioning context has been read to generate another symbol

$$p(\boldsymbol{x} \mid \boldsymbol{y}) = \sum_{\boldsymbol{z}} p(\boldsymbol{x}, \boldsymbol{z} \mid \boldsymbol{y})$$

$$p(\boldsymbol{x}, \boldsymbol{z} \mid \boldsymbol{y}) \approx \prod_{j=1}^{|\boldsymbol{x}|} \underbrace{p(\boxed{z_j} \mid z_{j-1}, \boldsymbol{y}_1^{z_j}, \boldsymbol{x}_1^{j-1})}_{\text{alignment probability}} \underbrace{p(x_j \mid \boldsymbol{y}_1^{z_j}, \boldsymbol{x}_1^{j-1})}_{\text{word probability}}$$

How much of ***y*** do we need to read to model the j^{th} token of ***x***?

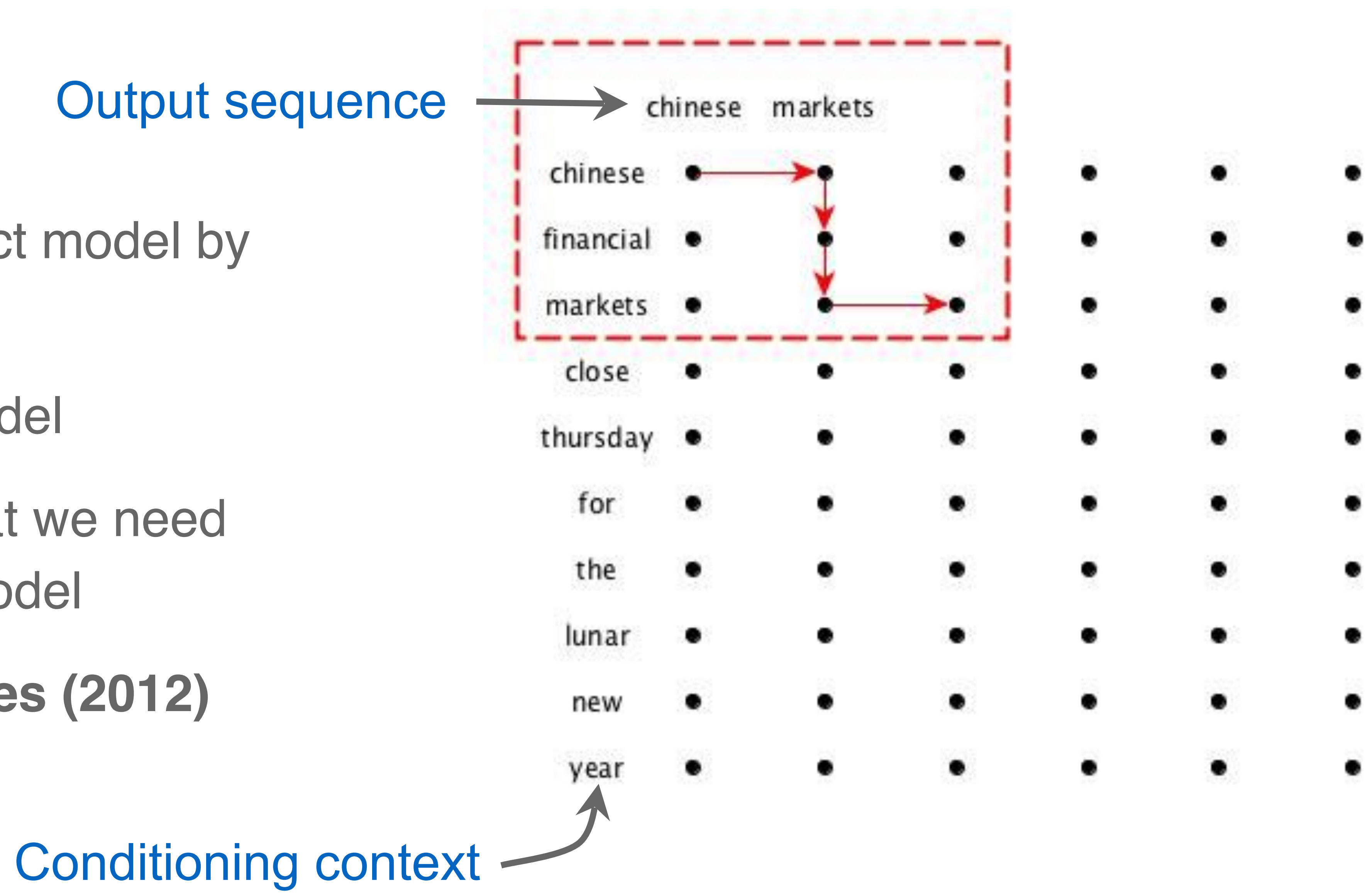
The Segment to Segment Model

Introduced as a direct model by **Yu et al. (2016)**

It's a good direct model

It also is exactly what we need for the channel model

Similar model: **Graves (2012)**

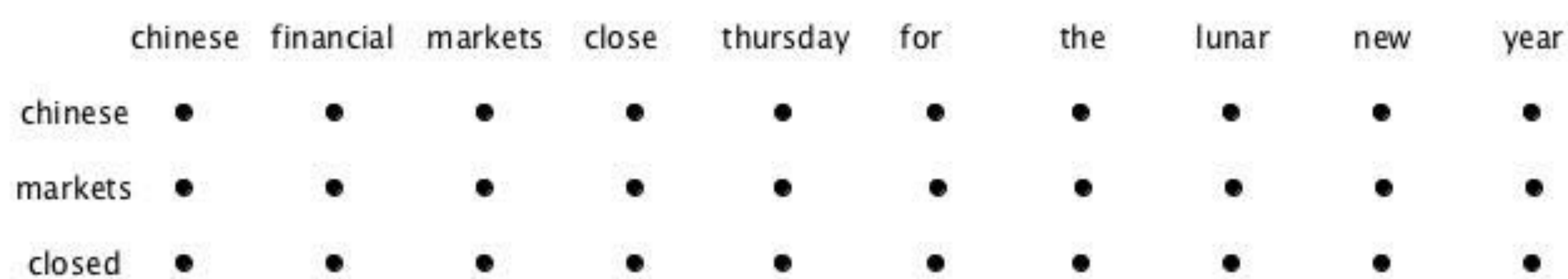


Decoding with an auxiliary model

Expensive to go through every token y_j in the vocabulary and calculate

$$p(\boldsymbol{x}_{1:i}|\boldsymbol{y}_{1:j})p(\boldsymbol{y}_{1:j})$$

Use an auxiliary direct model $q(\boldsymbol{y}, \boldsymbol{z} \mid \boldsymbol{x})$ to guide the search.



Decoding with an auxiliary model

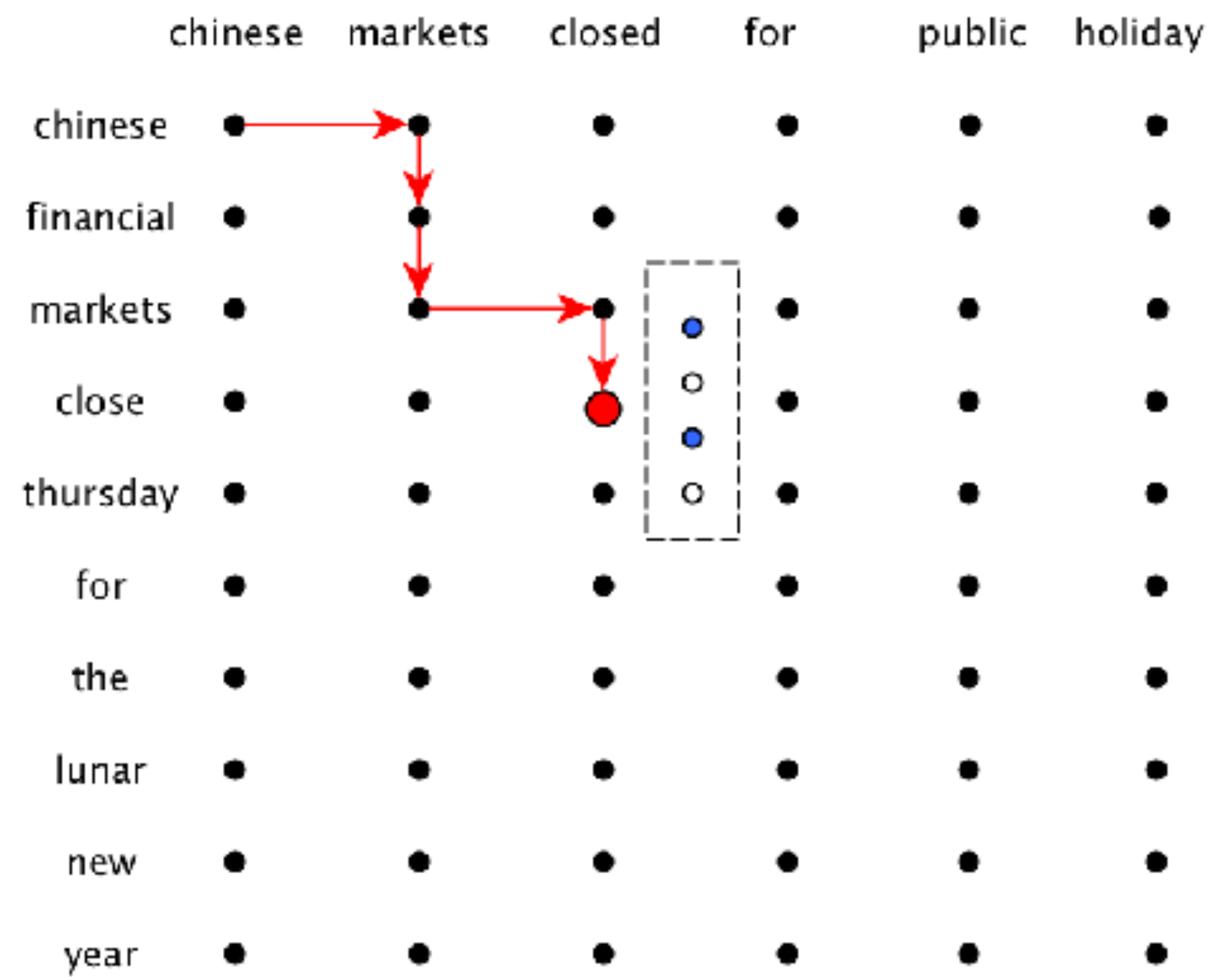
Possible proposals:

Chinese markets open

Chinese markets closed

Market close

Financial markets



Decoding with an auxiliary model

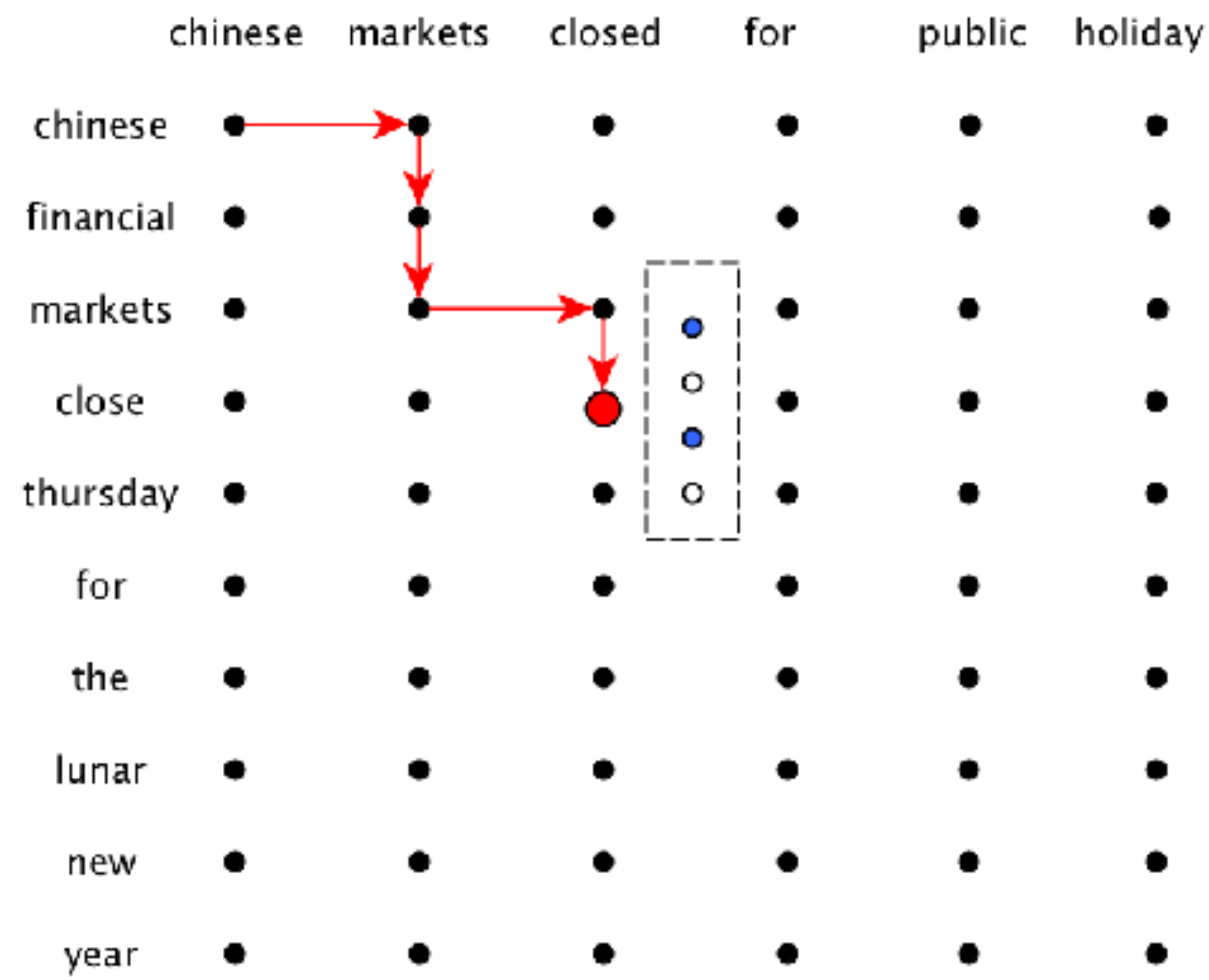
Possible proposals:

Chinese markets open

Chinese markets closed

Market close

Financial markets



Expanded objective

$$O_{\boldsymbol{x}_1^i, \boldsymbol{y}_1^j} = \lambda_1 \log p(\boldsymbol{y}_1^j \mid \boldsymbol{x}_1^i) + \lambda_2 \log p(\boldsymbol{x}_1^i \mid \boldsymbol{y}_1^j) + \lambda_3 \log p(\boldsymbol{y}_1^j) + \lambda_4 |\boldsymbol{y}_1^j|.$$

Experiments
Machine translation

- Medium-sized Chinese-English news parallel data
- Large LSTM language model trained on English news + target side of parallel data
- Evaluation using BLEU-4 (higher is better)

Experiments
Machine translation

	Model	**BLEU**
Discriminative	Seq2seq with attention	25.27
	Direct model (*q* by itself)	23.33
	Direct + LM + bias	23.33
Gen	Channel + LM + bias	26.28
	Direct + channel + LM + bias	26.44

Experiments

Machine translation

	Model	BLEU
Discriminative	Seq2seq with attention	25.27
	Direct model (*q* by itself)	23.33
	Direct + LM + bias	23.33
Gen	Channel + LM + bias	26.28
	Direct + channel + LM + bias	26.44

Experiments
Machine translation

	Model	BLEU
Discriminative	Seq2seq with attention	25.27
Discriminative	Direct model (*q* by itself)	23.33
Discriminative	Direct + LM + bias	23.33
Gen	Channel + LM + bias	26.28
Gen	Direct + channel + LM + bias	26.44

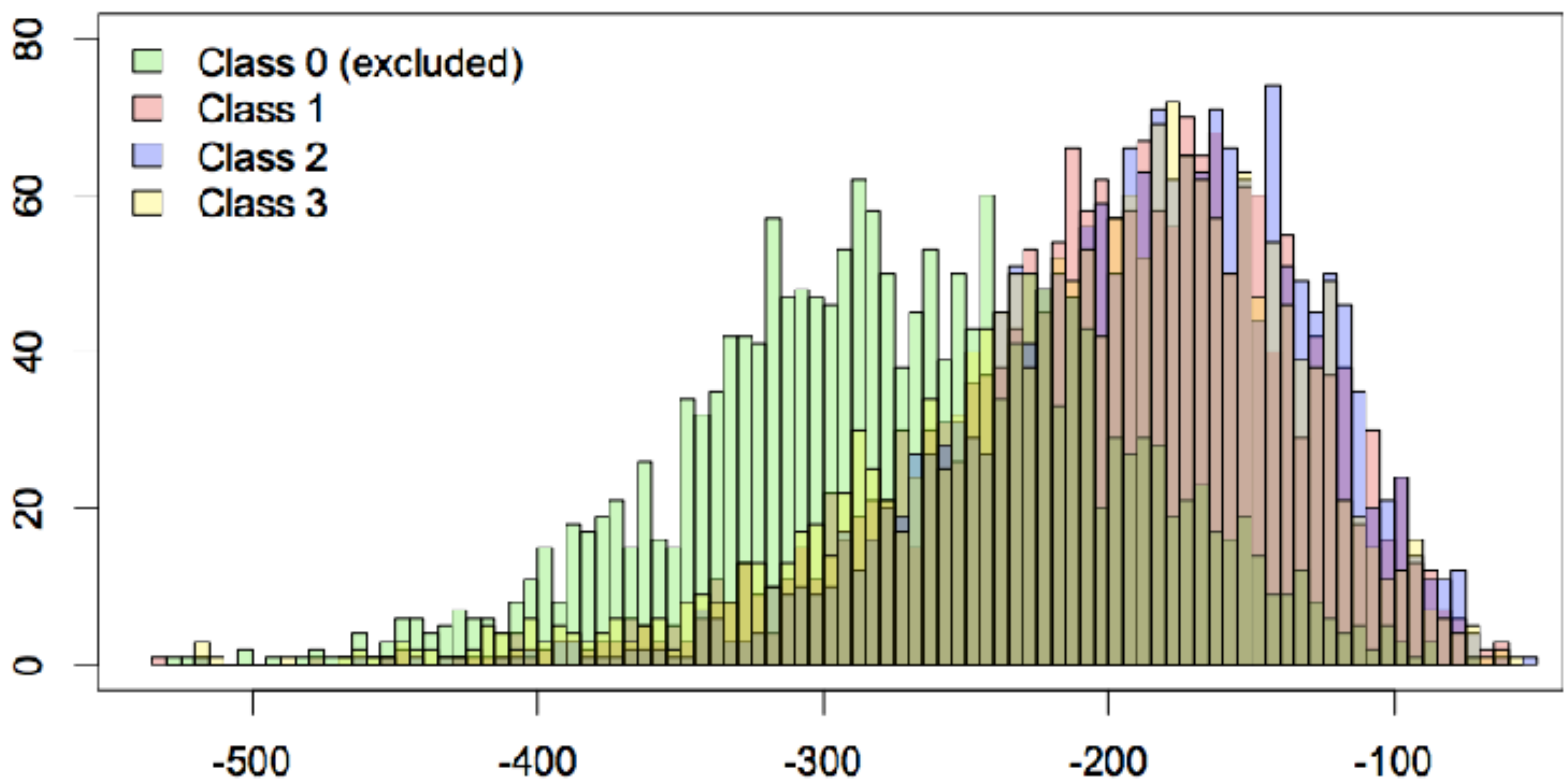
Conclusions

- **Generative can be used well for “discriminative problems”**
  - Especially in data-restricted scenarios
  - Especially with neural nets, which let us define great generative models
- **Open questions**
  - Inference is hard, but there are lots of exciting possibilities for learning to do inference
  - Is there a theoretical account for when a particular dataset is in the “generative” vs. “discriminative” regime and where the crossover point is?

Thank you!

Outlier detection

- Generative models also provide an estimate of p(**x**)
  - The likelihood of the input is a good estimate of “what the model knows”. Examples that fall out of this are a good indication that the model should stop what it’s doing and get help.



Zero-shot learning

- **Train** on $n-1$ classes
- **Predict** for **all** classes
- Learn **(label) concepts**, to be used as class embeddings $\mathbf{v}_y$ from an **auxiliary task**
    - For example, from a large unannotated corpus, learn standard **word embeddings** and use them **as class embeddings**
- **Fix** the class embeddings **during training**
- When we see a **new class**, use the word embedding for the class

Zero-shot learning

Class	Precision	Recall	Accuracy
company	98.9	46.6	93.3
educational institution	99.2	49.5	92.8
athlete	96.5	90.1	94.6
means of transportation	96.5	74.3	94.2
building	99.9	37.7	92.1
natural place	98.9	88.2	95.4
village	99.9	68.1	93.8
animal	99.7	68.1	93.8
plant	99.2	76.9	94.3
film	99.4	73.3	94.5
written work	93.8	26.5	91.3
AVERAGE	98.3	63.6	93.6

Inference
Importance sampling

Assume we've got a conditional distribution $q(\boldsymbol{y} \mid \boldsymbol{x})$

s.t. (i) $p(\boldsymbol{x}, \boldsymbol{y}) > 0 \implies q(\boldsymbol{y} \mid \boldsymbol{x}) > 0$

(ii) $\boldsymbol{y} \sim q(\boldsymbol{y} \mid \boldsymbol{x})$ is tractable and

(iii) $q(\boldsymbol{y} \mid \boldsymbol{x})$ is tractable

Let the importance weights $w(\boldsymbol{x}, \boldsymbol{y}) = \dfrac{p(\boldsymbol{x}, \boldsymbol{y})}{q(\boldsymbol{y} \mid \boldsymbol{x})}$

$$p(\boldsymbol{x}) = \sum_{\boldsymbol{y} \in \mathcal{Y}(\boldsymbol{x})} p(\boldsymbol{x}, \boldsymbol{y}) = \sum_{\boldsymbol{y} \in \mathcal{Y}(\boldsymbol{x})} w(\boldsymbol{x}, \boldsymbol{y}) q(\boldsymbol{y} \mid \boldsymbol{x})$$

$$= \mathbb{E}_{\boldsymbol{y} \sim q(\boldsymbol{y} \mid \boldsymbol{x})} w(\boldsymbol{x}, \boldsymbol{y})$$

Inference

Importance sampling

$$p(\boldsymbol{x}) = \sum_{\boldsymbol{y} \in \mathcal{Y}(\boldsymbol{x})} p(\boldsymbol{x}, \boldsymbol{y}) = \sum_{\boldsymbol{y} \in \mathcal{Y}(\boldsymbol{x})} w(\boldsymbol{x}, \boldsymbol{y}) q(\boldsymbol{y} \mid \boldsymbol{x})$$
$$= \mathbb{E}_{\boldsymbol{y} \sim q(\boldsymbol{y} \mid \boldsymbol{x})} w(\boldsymbol{x}, \boldsymbol{y})$$

Replace this expectation with its Monte Carlo estimate.

$$\boldsymbol{y}^{(i)} \sim q(\boldsymbol{y} \mid \boldsymbol{x}) \quad \text{for } i \in \{1, 2, \ldots, N\}$$

$$\mathbb{E}_{q(\boldsymbol{y} \mid \boldsymbol{x})} w(\boldsymbol{x}, \boldsymbol{y}) \overset{\mathrm{MC}}{\approx} \frac{1}{N} \sum_{i=1}^{N} w(\boldsymbol{x}, \boldsymbol{y}^{(i)})$$

Results: **Language modeling**

	Perplexity
5-gram IKN	169.3
LSTM LM	113.4
Generative (IS)	**102.4**