Massachusetts Institute of Technology

CSAIL

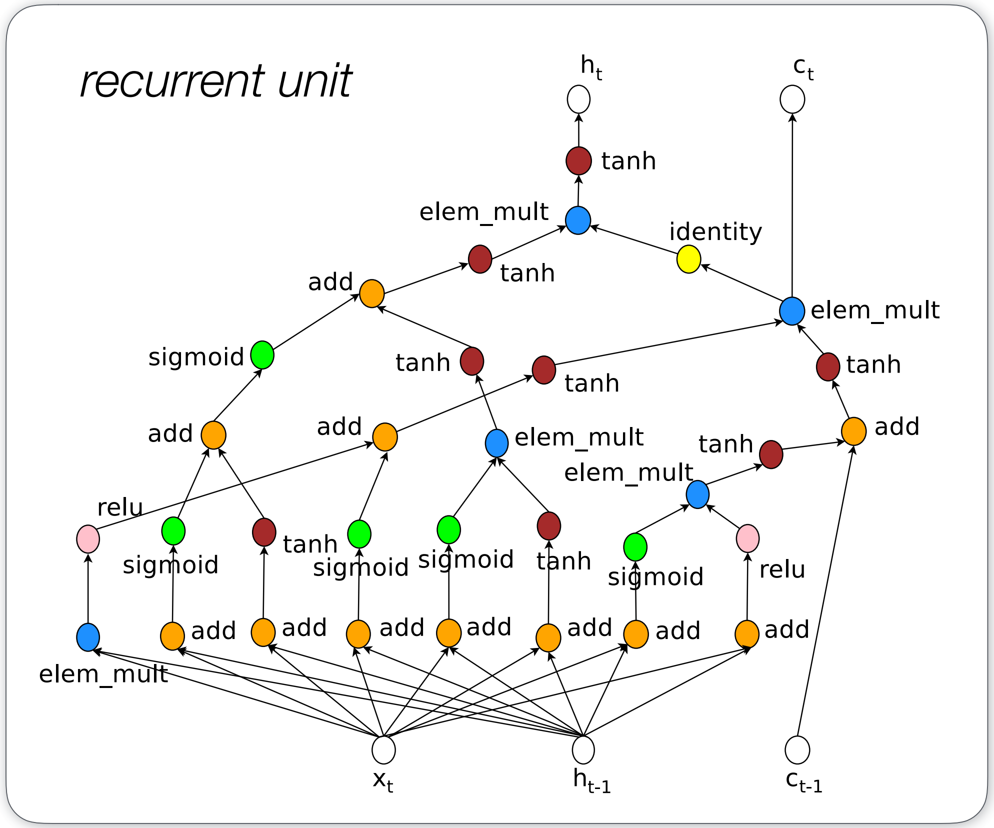
Interpretable Neural Models for NLP

Tao Lei

Jan 19, 2017

Motivation

- Deep learning enables very flexible model exploration
- Often leads to state-of-the-art performance



state-of-the-art unit for language modeling

8 tanh(), 5 sigmoid() and 2 ReLU()

- *why this unit?*
- *what's happening inside?*
- *why this prediction?*
- *what if I change this operator?*
- *...*

Our Goal

Design neural methods ***better*** for NLP applications

- *Performance*

  being able to achieve top accuracy

- *Interpretability*

  being able to explain the model's design

  being able to explain the model's decision

Outlines (i)

- *From (deep) kernel to (deep) neural model*
  - a class of neural operator for text / sequence
  - can be derived from traditional sequence kernel
  - encodes an efficient algorithm as its central part of computation

Example of Proposed Component

$$\mathbf{c}_t^{(1)} = \lambda_t \odot \mathbf{c}_{t-1}^{(1)} + (1 - \lambda_t) \odot (\mathbf{W}^{(1)} \mathbf{x}_t)$$

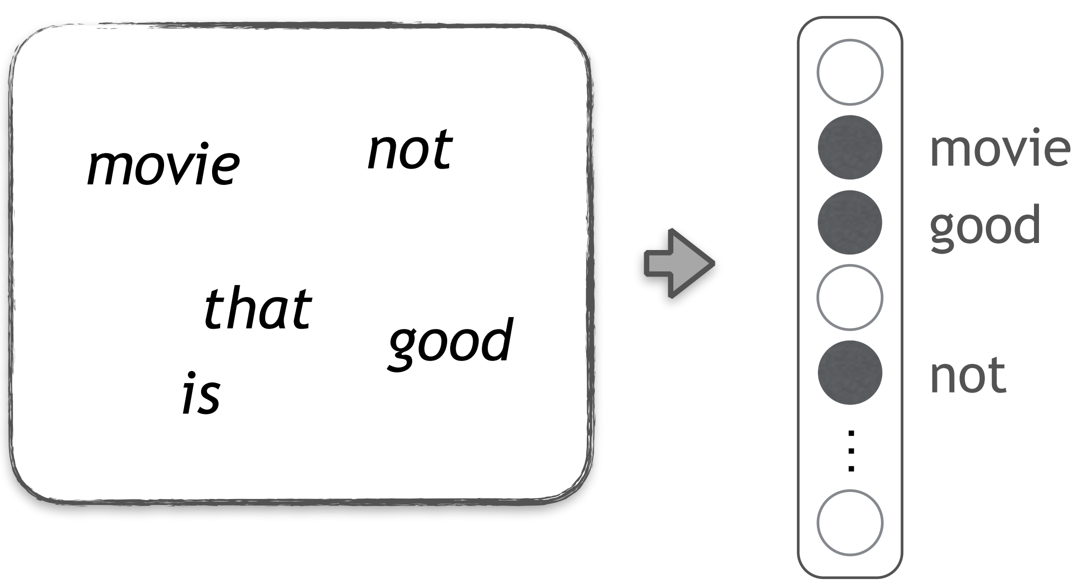
$$\mathbf{c}_t^{(2)} = \lambda_t \odot \mathbf{c}_{t-1}^{(2)} + (1 - \lambda_t) \odot (\mathbf{c}_{t-1}^{(1)} + \mathbf{W}^{(2)} \mathbf{x}_t)$$

$$\mathbf{h}_t = \tanh(\mathbf{c}_t^{(2)})$$

how to interpret and understand it?

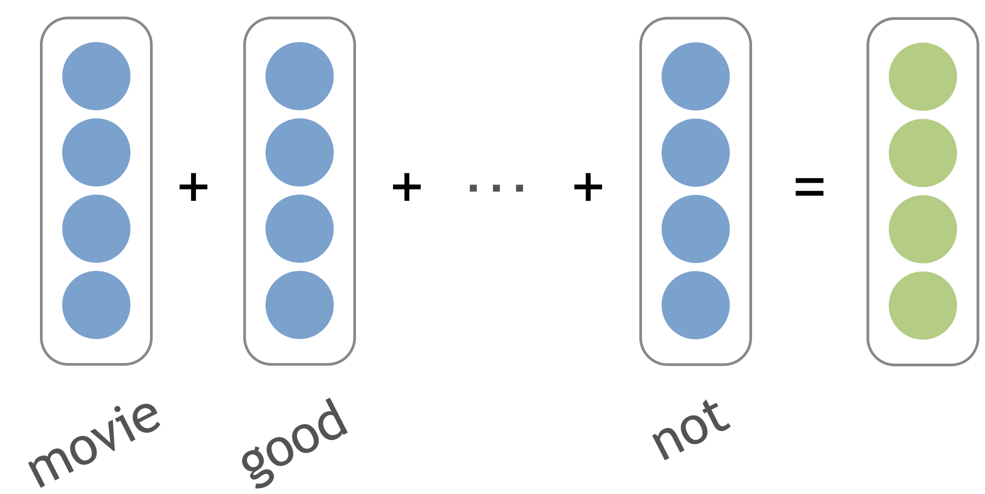
Sentence: *“the movie is not that good”*

Bag of words, TF-IDF



$\phi(\mathbf{x})$

Neural Bag-of-words
(average embedding)

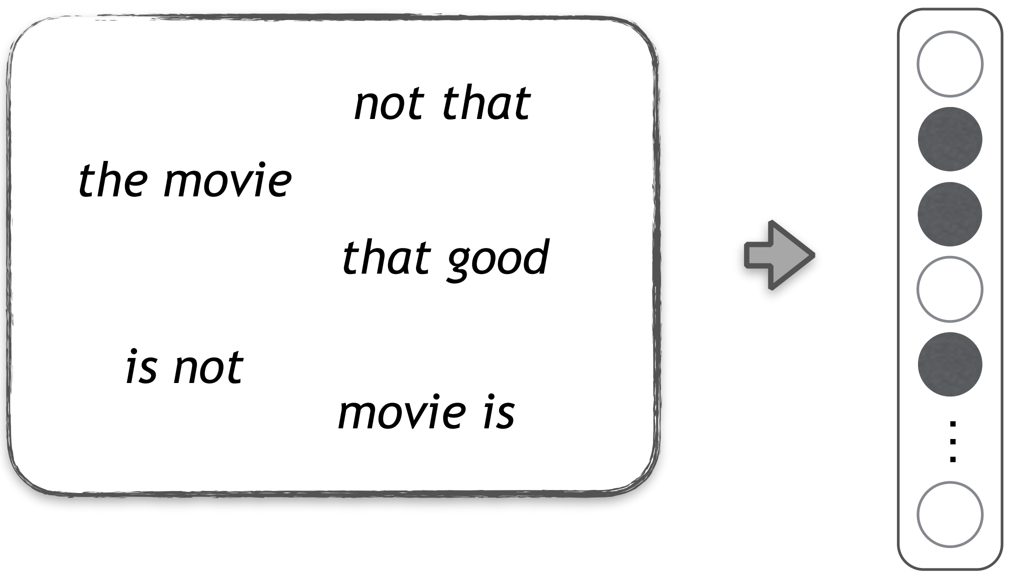


$\phi(\mathbf{x}) \cdot \mathbf{M}_{\text{emb}}$

Sentence: *"the movie is not that good"*

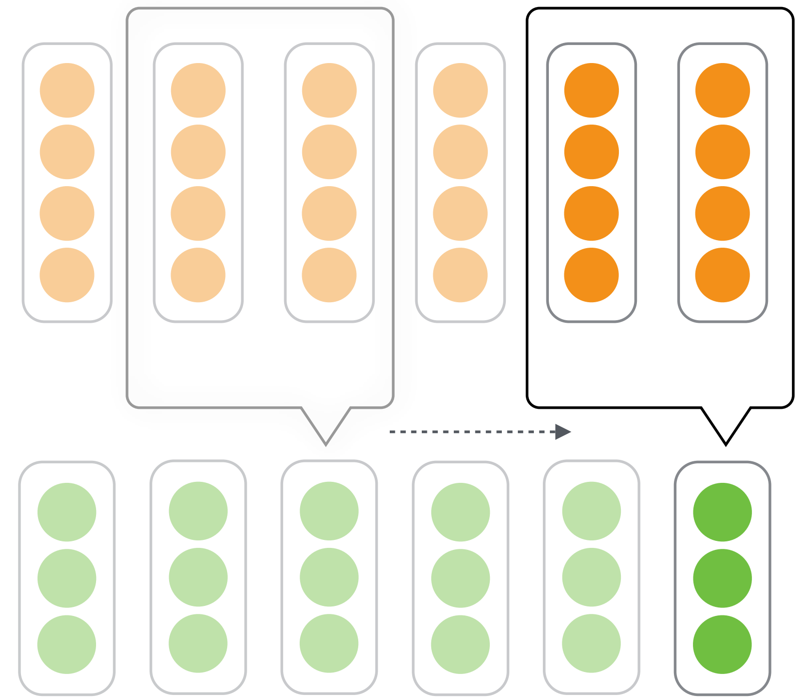
Ngram Kernel

(N=2)



$\phi(\mathbf{x})$

CNNs

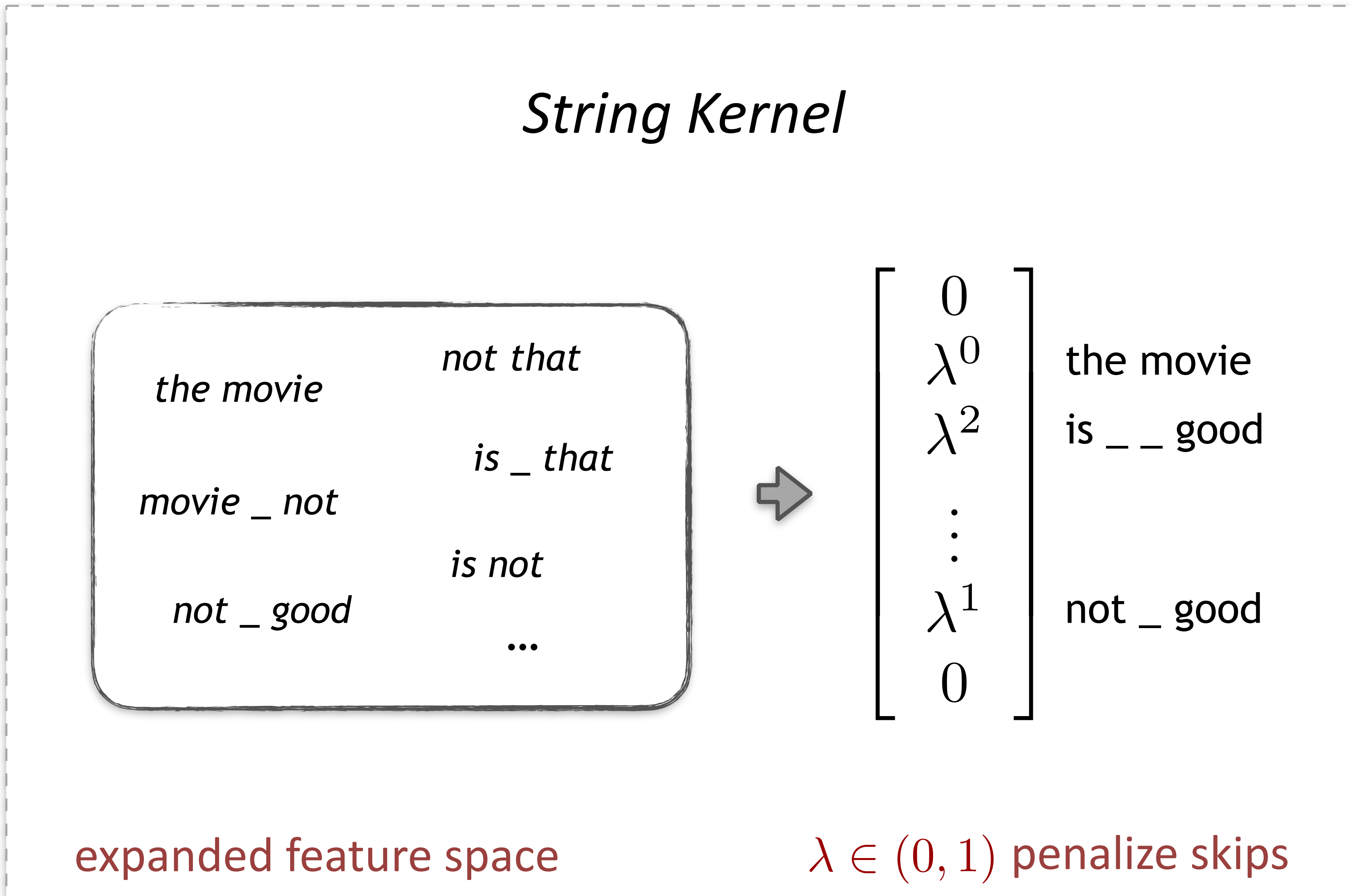


$\phi(\mathbf{x}) \cdot \mathbf{M}_{\text{filter}}$

Pre-activation as a dimension-reduction or projection

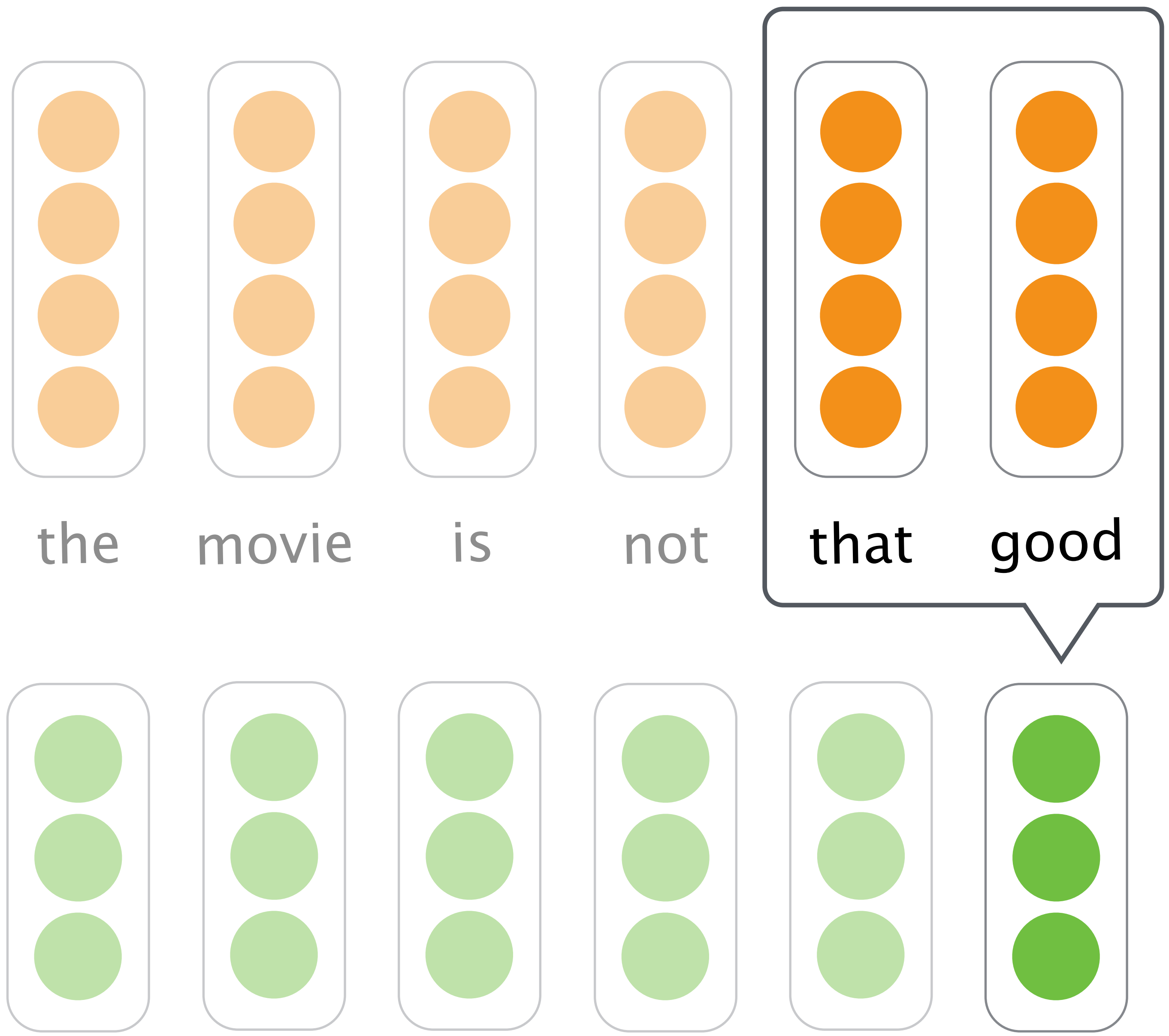
of traditional methods

Sentence: *“the movie is not that good”*

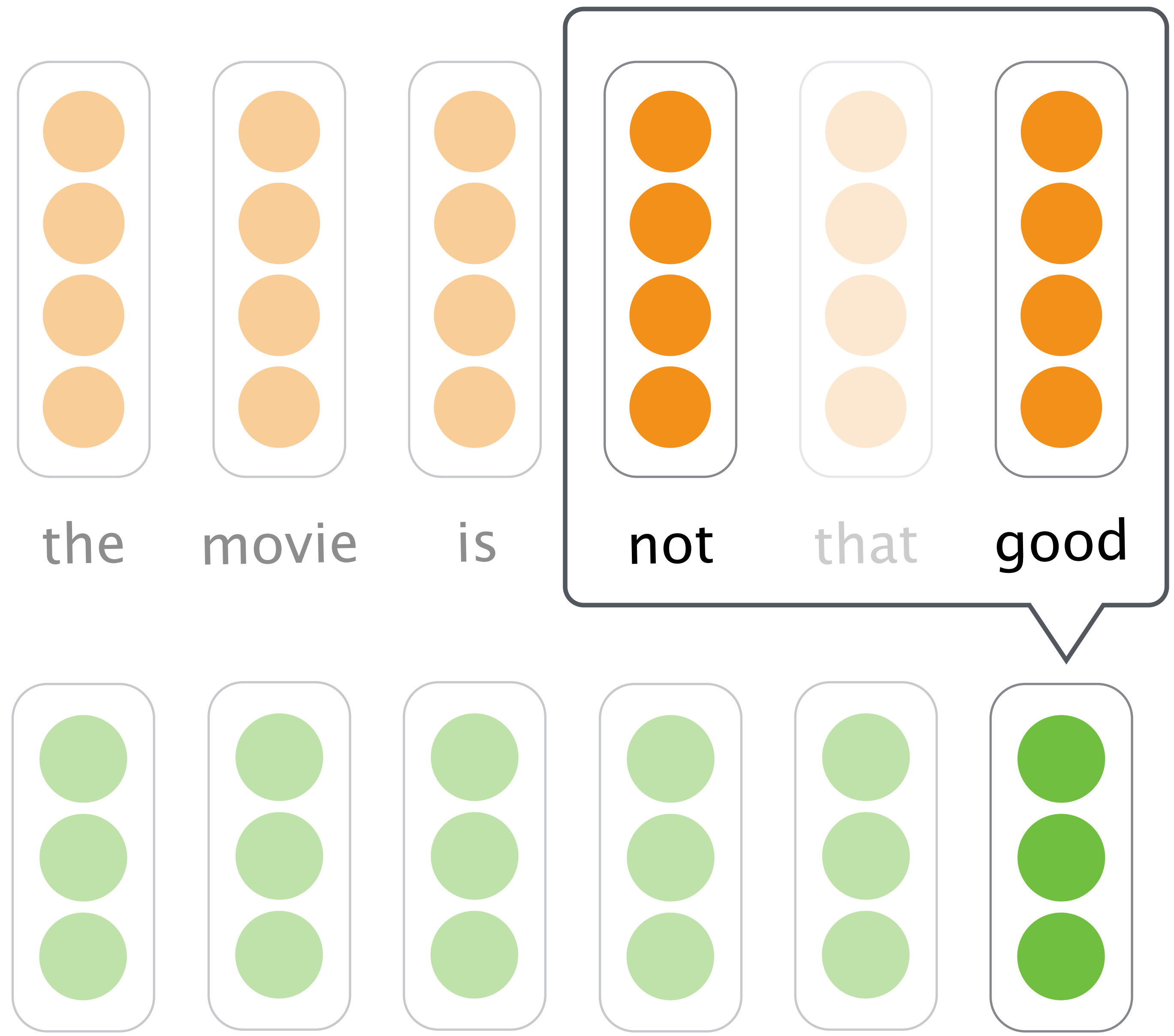


Neural model inspired by this kernel method

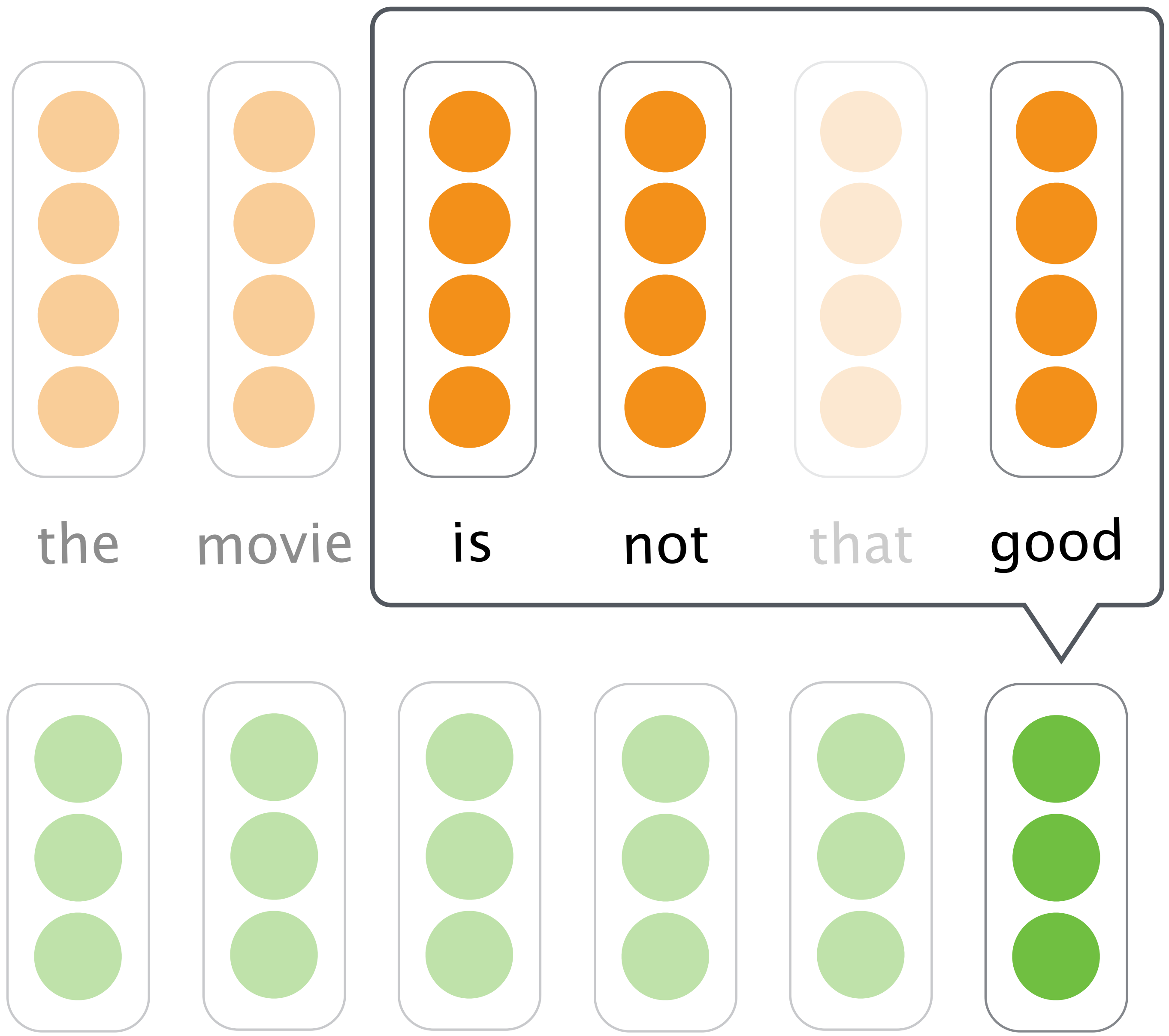
Illustration



Illustration



Illustration



Formulas

$$\begin{aligned}\mathbf{c}_t^{(1)} &= \lambda_t \odot \mathbf{c}_{t-1}^{(1)} + (1-\lambda_t) \odot (\mathbf{W}^{(1)}\mathbf{x}_t) \\ \mathbf{c}_t^{(2)} &= \lambda_t \odot \mathbf{c}_{t-1}^{(2)} + (1-\lambda_t) \odot (\mathbf{c}_{t-1}^{(1)} + \mathbf{W}^{(2)}\mathbf{x}_t) \\ \mathbf{h}_t &= \tanh(\mathbf{c}_t^{(2)})\end{aligned}$$

aggregated 1-gram and 2-gram features

Formulas

$$
\begin{aligned}
\mathbf{c}_t^{(1)} &= \lambda_t \odot \mathbf{c}_{t-1}^{(1)} + (1 - \lambda_t) \odot (\mathbf{W}^{(1)} \mathbf{x}_t) \\
\mathbf{c}_t^{(2)} &= \lambda_t \odot \mathbf{c}_{t-1}^{(2)} + (1 - \lambda_t) \odot (\mathbf{c}_{t-1}^{(1)} + \mathbf{W}^{(2)} \mathbf{x}_t) \\
\mathbf{h}_t &= \tanh(\mathbf{c}_t^{(2)})
\end{aligned}
$$

re-normalize to remove length bias

decay penalizing skip grams

Formulas

$$\mathbf{c}_t^{(1)} = \lambda_t \odot \mathbf{c}_{t-1}^{(1)} + (1 - \lambda_t) \odot (\mathbf{W}^{(1)}\mathbf{x}_t)$$

$$\mathbf{c}_t^{(2)} = \lambda_t \odot \mathbf{c}_{t-1}^{(2)} + (1 - \lambda_t) \odot (\mathbf{c}_{t-1}^{(1)} + \mathbf{W}^{(2)}\mathbf{x}_t)$$

$$\mathbf{h}_t = \tanh(\mathbf{c}_t^{(2)})$$

$\lambda_t = 0:$ $\quad \mathbf{h}_t = \tanh(\mathbf{W}^{(1)}\mathbf{x}_{t-1} + \mathbf{W}^{(2)}\mathbf{x}_t)$ $\quad$ (one-layer CNNs)

Formulas

$$
\begin{aligned}
\mathbf{c}_t^{(1)} &= \lambda_t \odot \mathbf{c}_{t-1}^{(1)} + (1-\lambda_t) \odot (\mathbf{W}^{(1)}\mathbf{x}_t) \\
\mathbf{c}_t^{(2)} &= \lambda_t \odot \mathbf{c}_{t-1}^{(2)} + (1-\lambda_t) \odot (\mathbf{c}_{t-1}^{(1)} \odot \mathbf{W}^{(2)}\mathbf{x}_t) \\
\mathbf{h}_t &= \tanh(\mathbf{c}_t^{(2)})
\end{aligned}
$$

multiplicative mapping

Formulas

$$
\begin{aligned}
\mathbf{c}_t^{(1)} &= \lambda_t \odot \mathbf{c}_{t-1}^{(1)} + (1-\lambda_t) \odot (\mathbf{W}^{(1)}\mathbf{x}_t) \\
\mathbf{c}_t^{(2)} &= \lambda_t \odot \mathbf{c}_{t-1}^{(2)} + (1-\lambda_t) \odot (\mathbf{c}_{t-1}^{(1)} \odot \mathbf{W}^{(2)}\mathbf{x}_t) \\
&\cdots \\
\mathbf{c}_t^{(n)} &= \lambda_t \odot \mathbf{c}_{t-1}^{(n)} + (1-\lambda_t) \odot (\mathbf{c}_{t-1}^{(n-1)} \odot \mathbf{W}^{(n)}\mathbf{x}_t) \\
\mathbf{h}_t &= \tanh(\mathbf{c}_t^{(n)})
\end{aligned}
$$

can be generalized to n-grams

From Kernel to Neural Model

*String kernel counts shared patterns in sequences **x** and **y**:*

$$\mathcal{K}_2(\mathbf{x},\mathbf{y}) = \sum_{1\le i<j\le|\mathbf{x}|} \sum_{1\le k<l\le|\mathbf{y}|} \lambda^{|\mathbf{x}|-i-1}\lambda^{|\mathbf{y}|-k-1} \boxed{[\mathbb{1}(\mathbf{x}_i = \mathbf{y}_k)\cdot\mathbb{1}(\mathbf{x}_j = \mathbf{y}_l)]}$$

$$\mathbf{x}_i\mathbf{x}_j = \mathbf{y}_k\mathbf{y}_l$$

Written in vector form:

(i) multiplicative $\langle\mathbf{x}_i,\mathbf{y}_k\rangle\,\langle\mathbf{x}_j,\mathbf{y}_l\rangle$

(ii) additive $\langle\mathbf{x}_i,\mathbf{y}_k\rangle + \langle\mathbf{x}_j,\mathbf{y}_l\rangle$

From Kernel to Neural Model

String kernel counts shared patterns in sequences ***x*** *and* ***y*****:**

$$\mathcal{K}_2(\mathbf{x},\mathbf{y}) = \sum_{1\le i<j\le|\mathbf{x}|}\ \sum_{1\le k<l\le|\mathbf{y}|} \lambda^{|\mathbf{x}|-i-1}\lambda^{|\mathbf{y}|-k-1}\langle\mathbf{x}_i,\mathbf{y}_k\rangle\langle\mathbf{x}_j,\mathbf{y}_l\rangle$$

$$= \left\langle \boxed{\sum_{1\le i<j\le|\mathbf{x}|}\lambda^{|\mathbf{x}|-i-1}\mathbf{x}_i\otimes\mathbf{x}_j,}\ \sum_{1\le k<l\le|\mathbf{y}|}\lambda^{|\mathbf{y}|-k-1}\mathbf{y}_k\otimes\mathbf{y}_l \right\rangle$$

$\phi(\mathbf{x})$ ***underlying mapping***

From Kernel to Neural Model

Projecting $\phi(\mathbf{x})$ *to hidden representation* $\mathbf{c}_t \in \mathbb{R}^d$

$$\mathbf{c}_t^{(2)}[k] = \left\langle \underbrace{\mathbf{w}_k^{(1)} \otimes \mathbf{w}_k^{(2)}}_{\text{k-th filter}}, \underbrace{\sum_{1 \le i < j \le t} \lambda^{|\mathbf{x}|-i-1}\, \mathbf{x}_i \otimes \mathbf{x}_j}_{\phi(\mathbf{x}_{1:t})} \right\rangle$$

$$= \mathcal{K}_2\left(\mathbf{w}_k^{(1)}\mathbf{w}_k^{(2)},\ \mathbf{x}_1\mathbf{x}_2\cdots\mathbf{x}_t\right)$$

can be seen as evaluating kernel functions;

naturally embeds sequence similarity computation

From Kernel to Neural Model

Projecting $\phi(\mathbf{x})$ *to hidden representation* $\mathbf{c}_t \in \mathbb{R}^d$

$$\mathbf{c}_t^{(2)}[k] = \left\langle \underbrace{\mathbf{w}_k^{(1)} \otimes \mathbf{w}_k^{(2)}}_{\textit{k-th filter}},\ \underbrace{\sum_{1 \le i < j \le t} \lambda^{|\mathbf{x}|-i-1}\, \mathbf{x}_i \otimes \mathbf{x}_j}_{\phi(\mathbf{x}_{1:t})} \right\rangle$$

Efficient implementation to compute $\mathbf{c}_t$ *(dynamic programming)*

$$\mathbf{c}_t^{(2)}[k] \;=\; \lambda \cdot \mathbf{c}_{t-1}^{(2)}[k] \;+\; \mathbf{c}_{t-1}^{(1)}[k] \cdot \left\langle \mathbf{w}_k^{(2)}, \mathbf{x}_t \right\rangle$$

From Kernel to Neural Model

Efficient implementation to compute $\mathbf{c}_t$ *(dynamic programming)*

$$\mathbf{c}_t^{(2)}[k] \;=\; \lambda \cdot \boxed{\mathbf{c}_{t-1}^{(2)}[k]} + \boxed{\mathbf{c}_{t-1}^{(1)}[k] \cdot \left\langle \mathbf{w}_k^{(2)}, \mathbf{x}_t \right\rangle}$$

all 2-grams up to position ***t***

all 2-grams up to position ***t-1***

2-grams end exactly at position ***t***

Interpreting Other Operations

Applying non-linear activation

*can be seen as **function composition** between **string kernel** and **the dual kernel of the activation function***

$$\phi(\mathbf{x}) = \phi_2(\phi_1(\mathbf{x}))$$

Stacking multiple layers

*can be seen as **recursive kernel construction** using the kernel of the previous layer as the **base kernel***

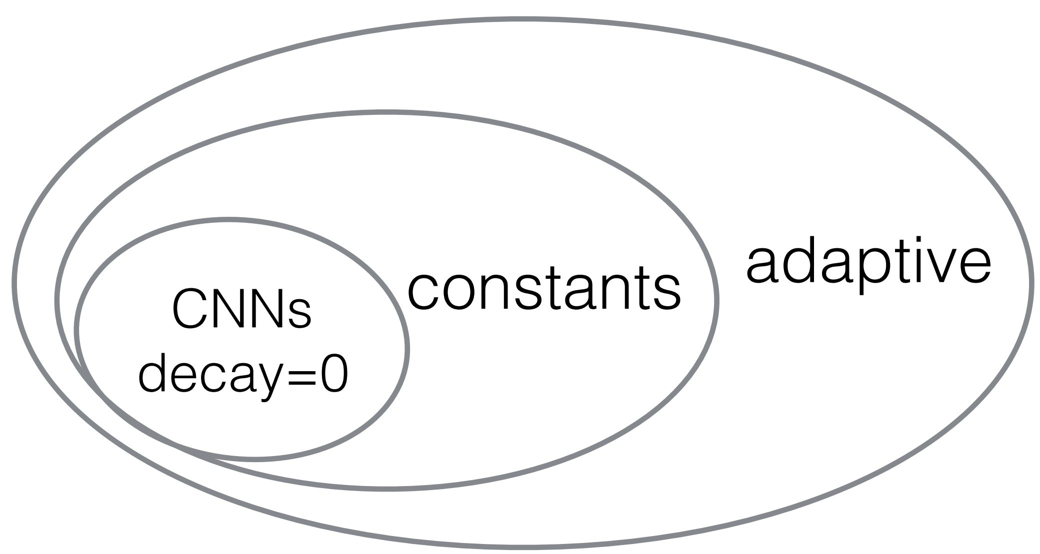
$$\phi(\mathbf{x}) = \sum_{i,j} \lambda^{t-i-1} \; \phi_1(\mathbf{x}_{1:i}) \otimes \phi_1(\mathbf{x}_{1:j})$$

Choices of Decay

constants: $\lambda_t = [u_1, u_2, \cdots, u_d]$

depends on x: $\lambda_t = \sigma(\mathbf{U}\mathbf{x}_t + \mathbf{b})$

depends on x and h: $\lambda_t = \sigma(\mathbf{U}\mathbf{x}_t + \mathbf{V}\mathbf{h}_{t-1} + \mathbf{b})$



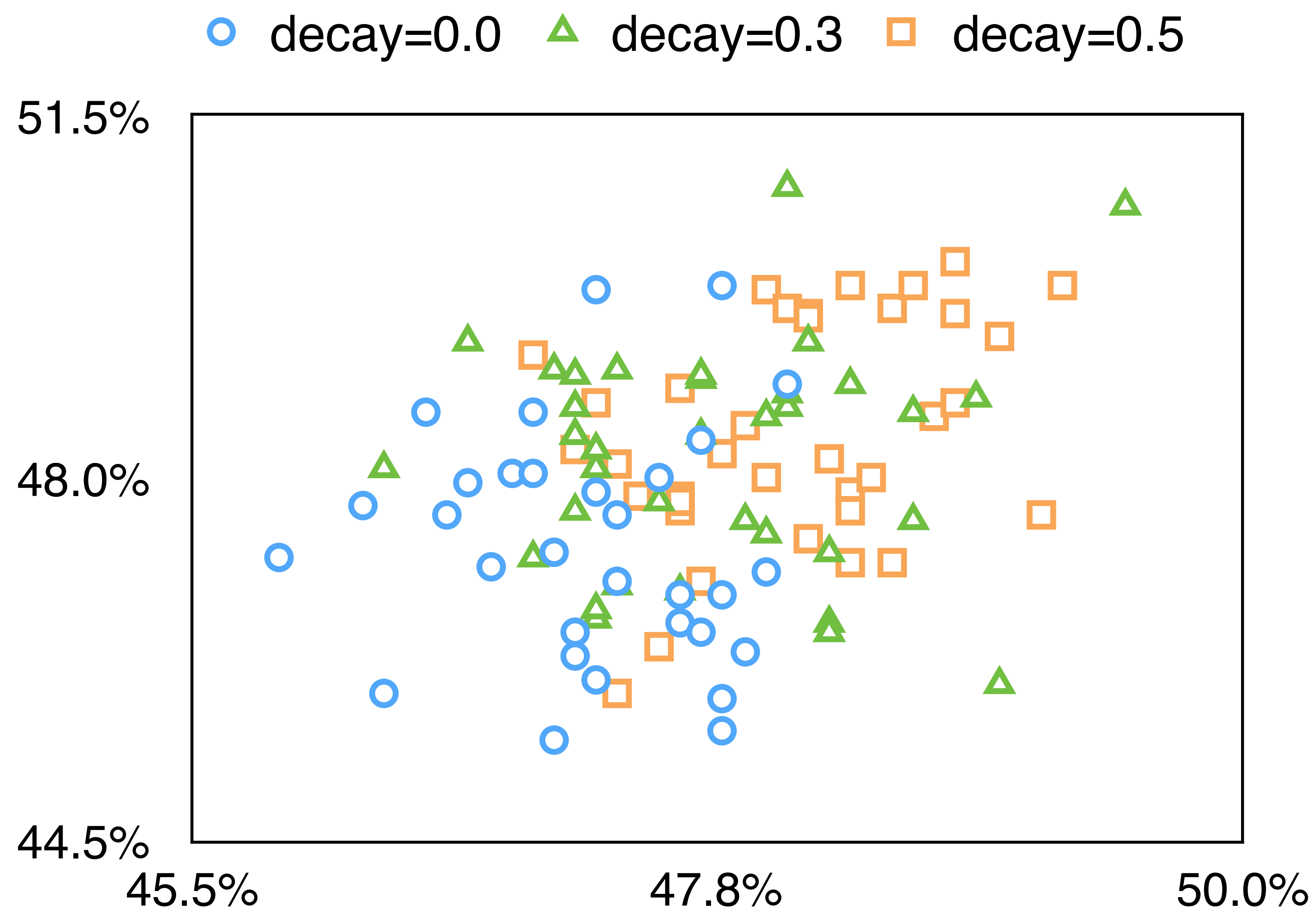
Experiments: Classification

Task: Predict the sentiment given a sentence in a review

Data: Stanford sentiment treebank

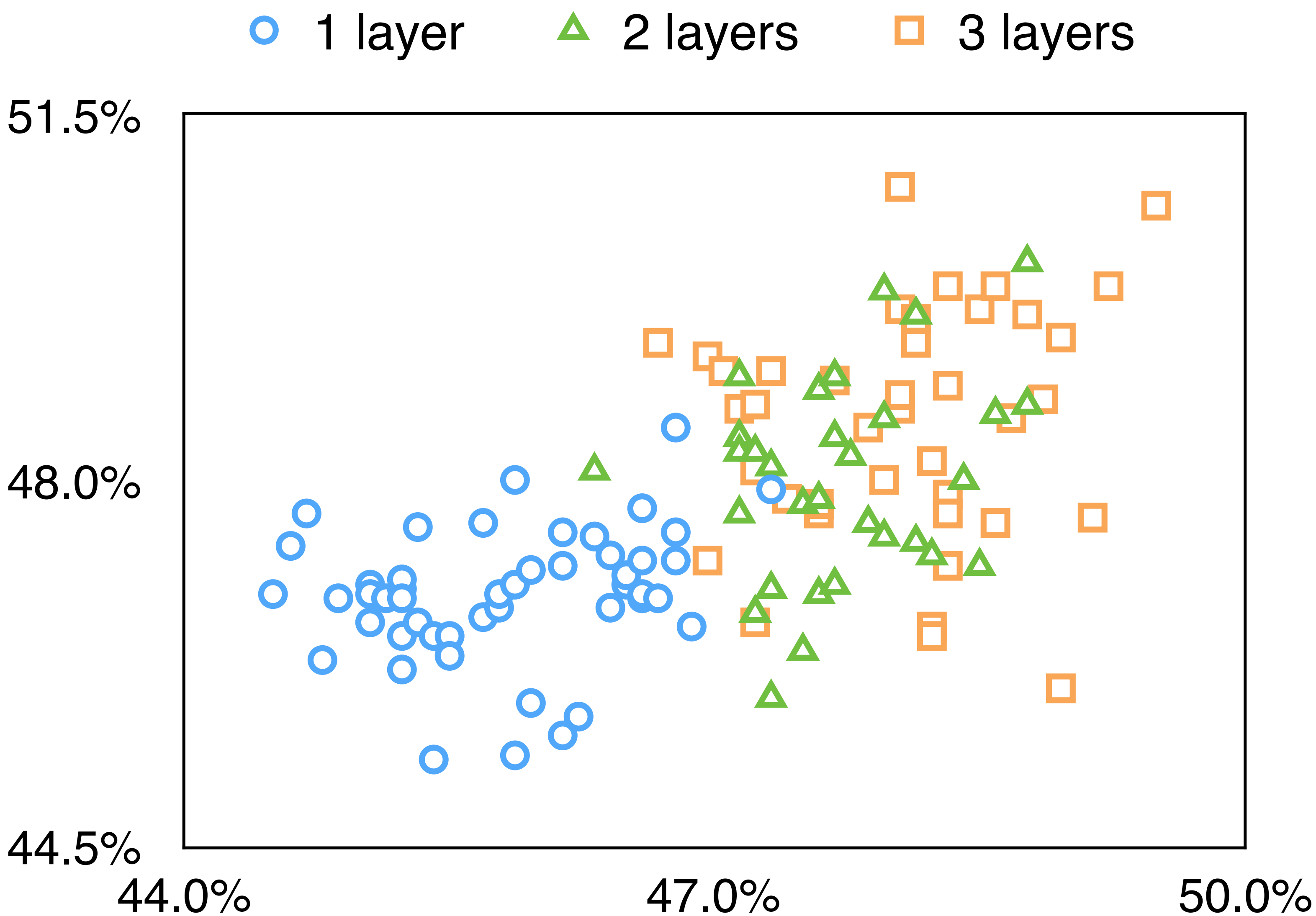
Experiments: Classification

Does it help to model non-consecutive patterns?



Experiments: Classification

Deeper model exhibits better representational power



Experiments: Classification

Model	5-class	Binary
CNNs (Kalchbrener et al. 2014)	48.5	86.9
CNNs (Kim 2014)	47.4	88.1
Bi-LSTMs (Tai et al. 2015)	49.1	87.5
RLSTMs (Tai et al. 2015)	51.0	88.0
Dynamic MemNet (Kumar et al. 2016)	52.1	88.6
Constant (0.5)	51.2	88.6
Adaptive (depends on x)	51.4	89.2
Adaptive (depends on x and h)	53.2	89.9

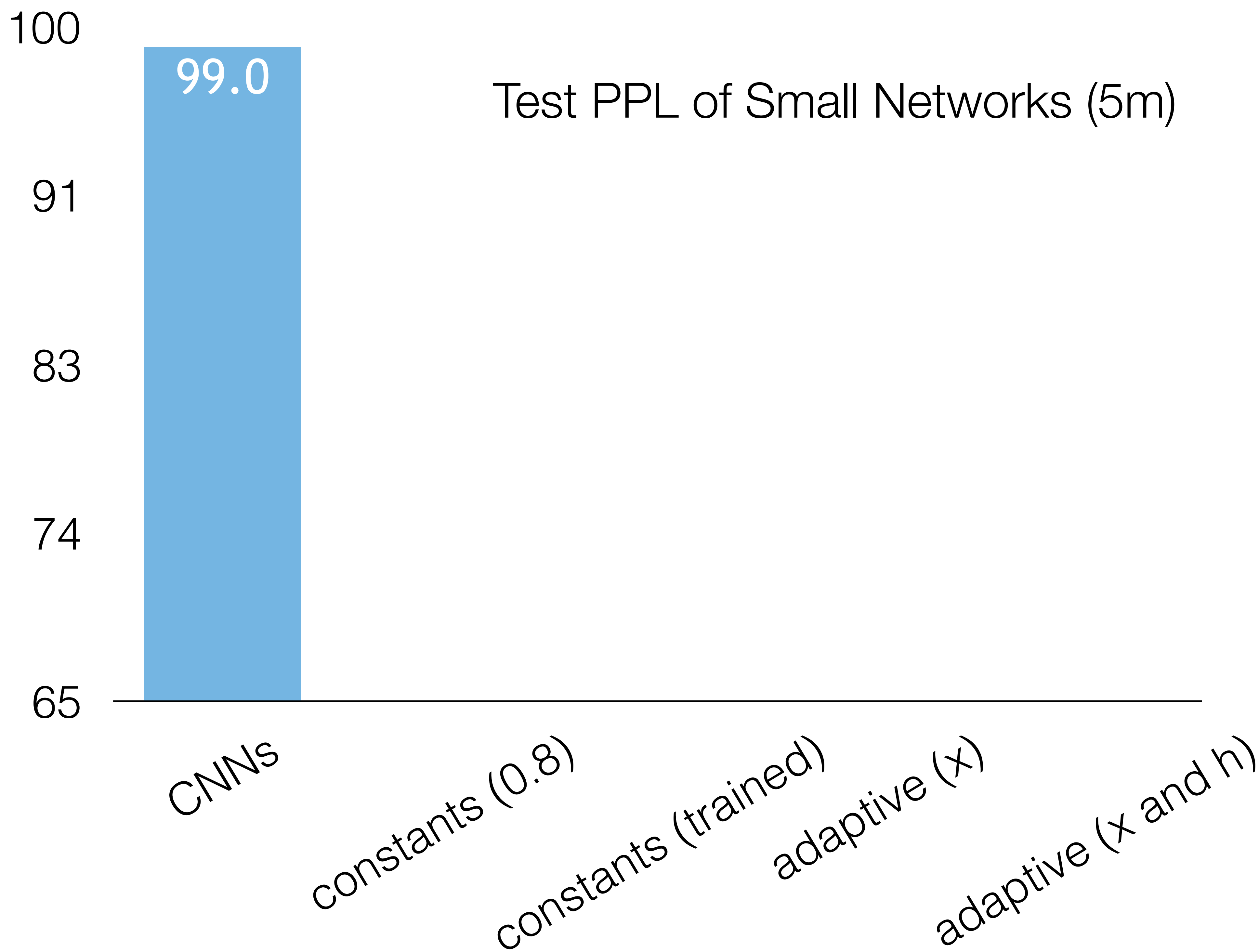
Test Results on Stanford Sentiment Treebank

Experiments: Language Model

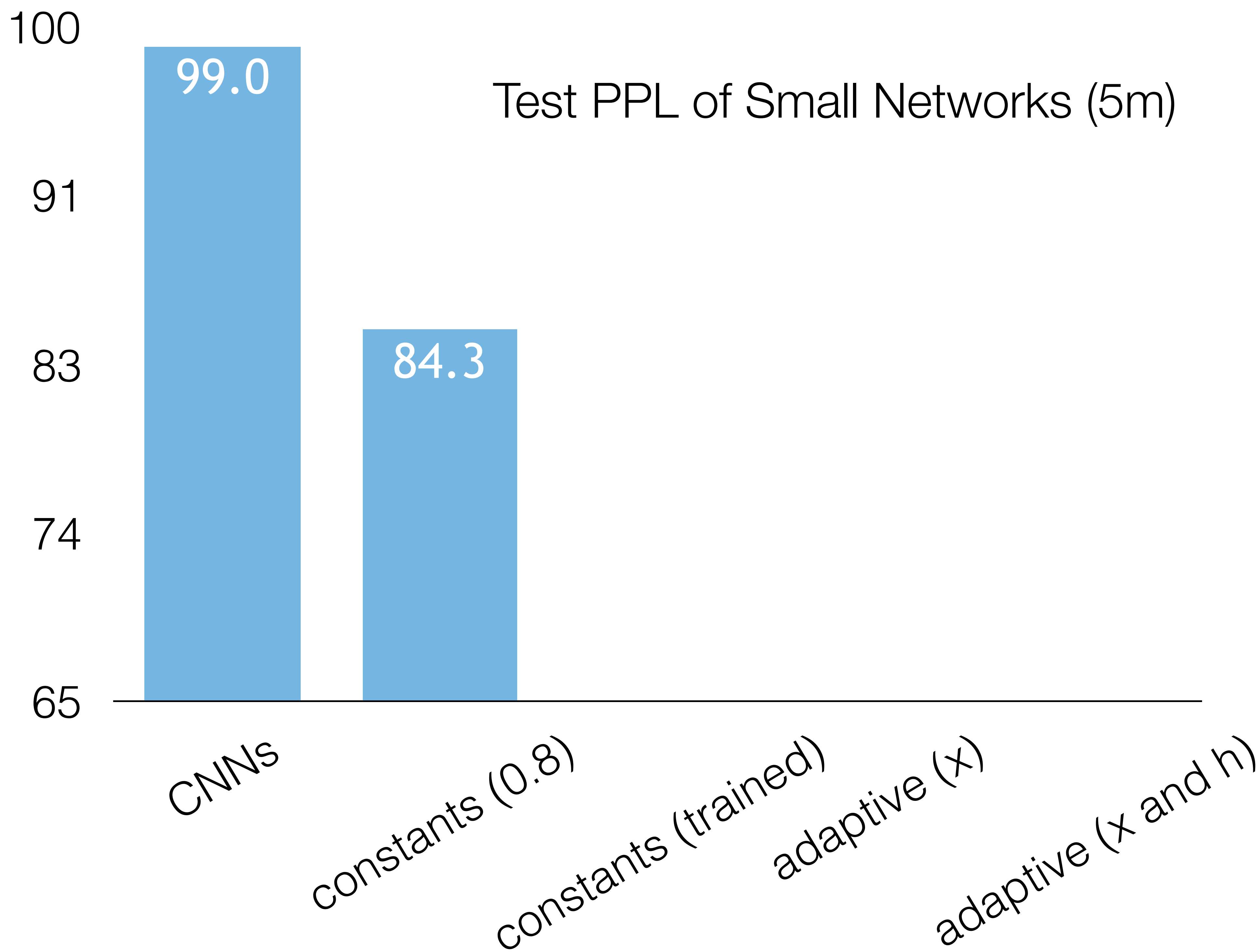
Task: Predict the next word given previous words

Data: Penn treebank (Wall street journal corpus)

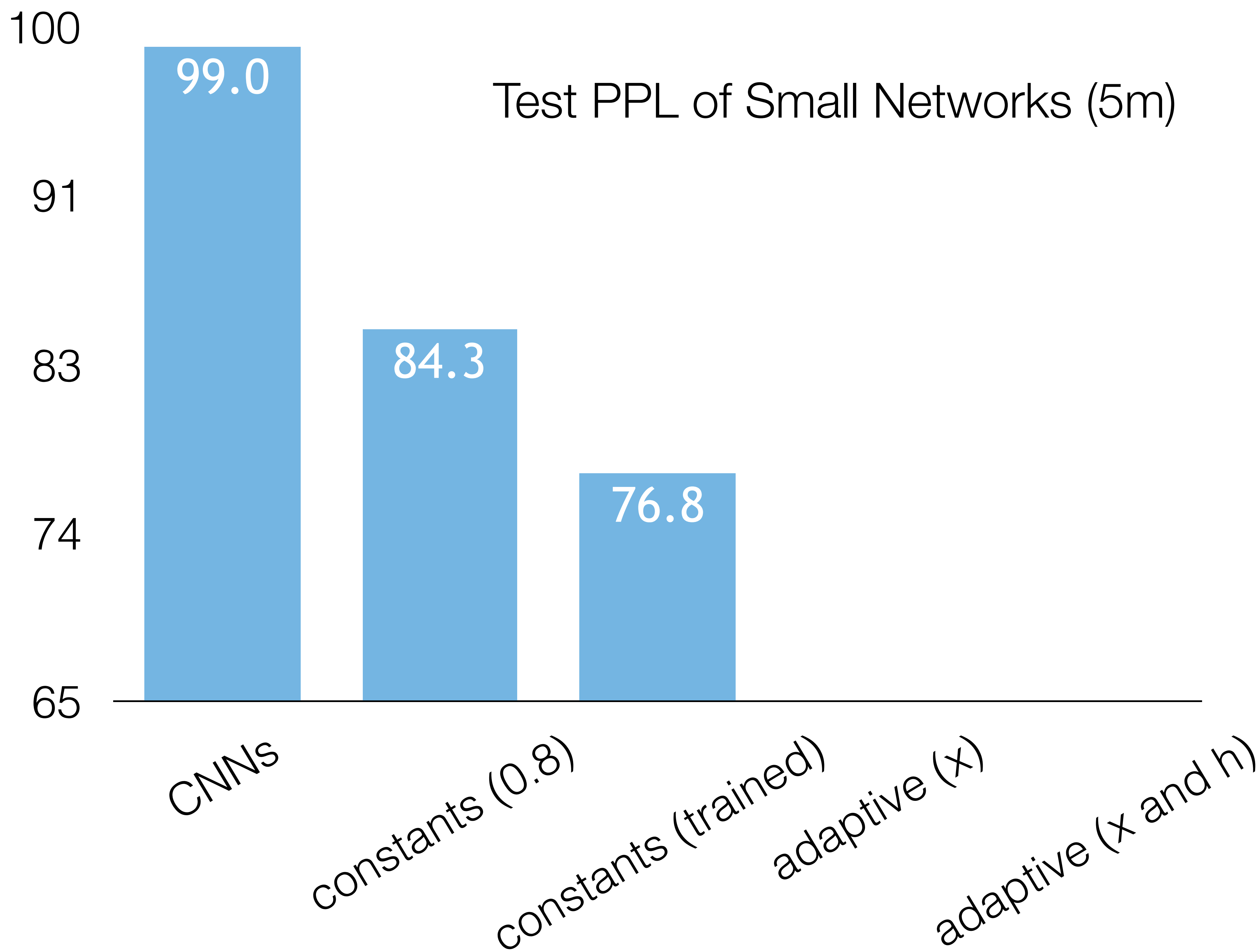
Experiments: Language Model



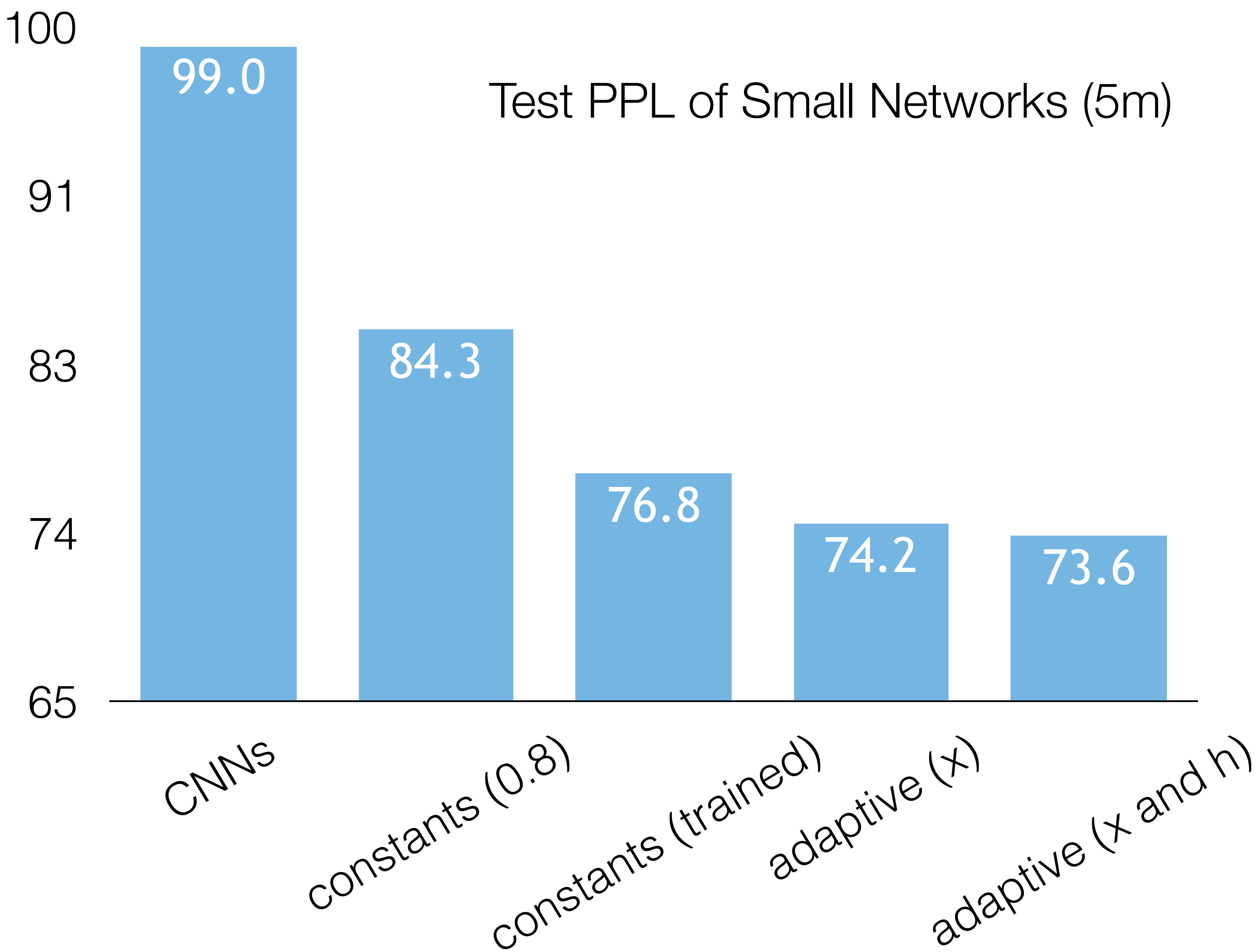
Experiments: Language Model



Experiments: Language Model



Experiments: Language Model



Experiments: Language Model

Model	Size	Perplexity
Character CNNs	19m	78.9
LSTM (large)	66m	78.4
Variational LSTM (medium)	20m	78.6
Variational LSTM (large)	51m	73.2
Pointer Sentinel LSTM	21m	70.9
Variational Deep Highway RNN	**24m**	**66.0**
Neural Net Search	**25m**	**64.0**
Ours (adaptive on x)	**20m**	**70.9**
Ours (adaptive on x and h)	**20m**	**69.2**

better regularized

can be improved w/ variational techniques

Comparison with state-of-the-art results

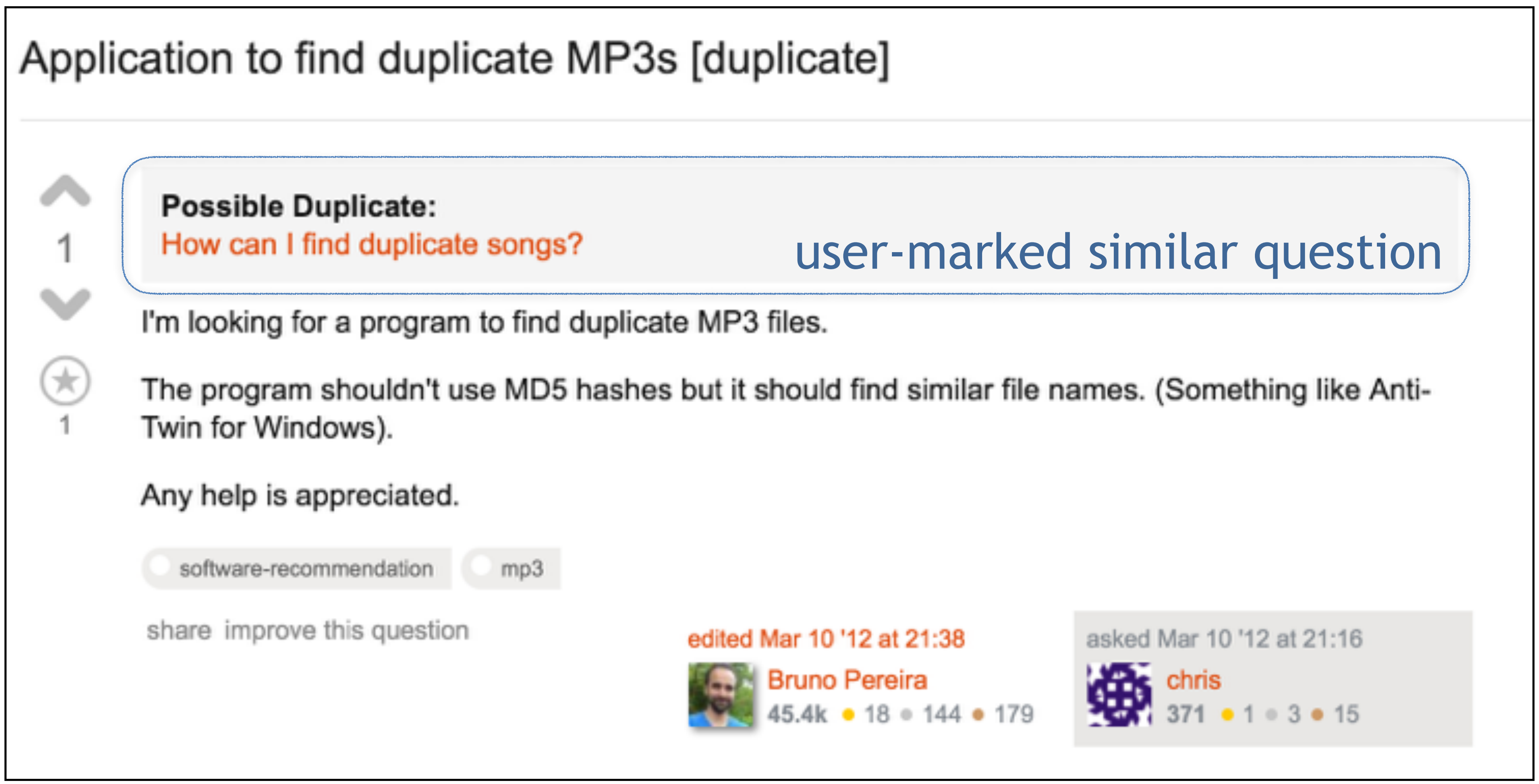
Experiments: Retrieval

Task: Find similar questions given the user's input question

question from Stack Exchange AskUbuntu

Experiments: Retrieval

Task: Find similar questions given the user's input question



Application to find duplicate MP3s [duplicate]

1

1

Possible Duplicate:
How can I find duplicate songs?

user-marked similar question

I'm looking for a program to find duplicate MP3 files.

The program shouldn't use MD5 hashes but it should find similar file names. (Something like Anti-Twin for Windows).

Any help is appreciated.

software-recommendation mp3

share improve this question

edited Mar 10 '12 at 21:38
Bruno Pereira
45.4k • 18 • 144 • 179

asked Mar 10 '12 at 21:16
chris
371 • 1 • 3 • 15

question from Stack Exchange AskUbuntu

Experiments: Retrieval

Dataset: AskUbuntu 2014 dump

pre-train on 167k, fine-tune on 16k

evaluate using 8k pairs (50/50 split for dev/test)

Baselines: TF-IDF, BM25 and SVM reranker

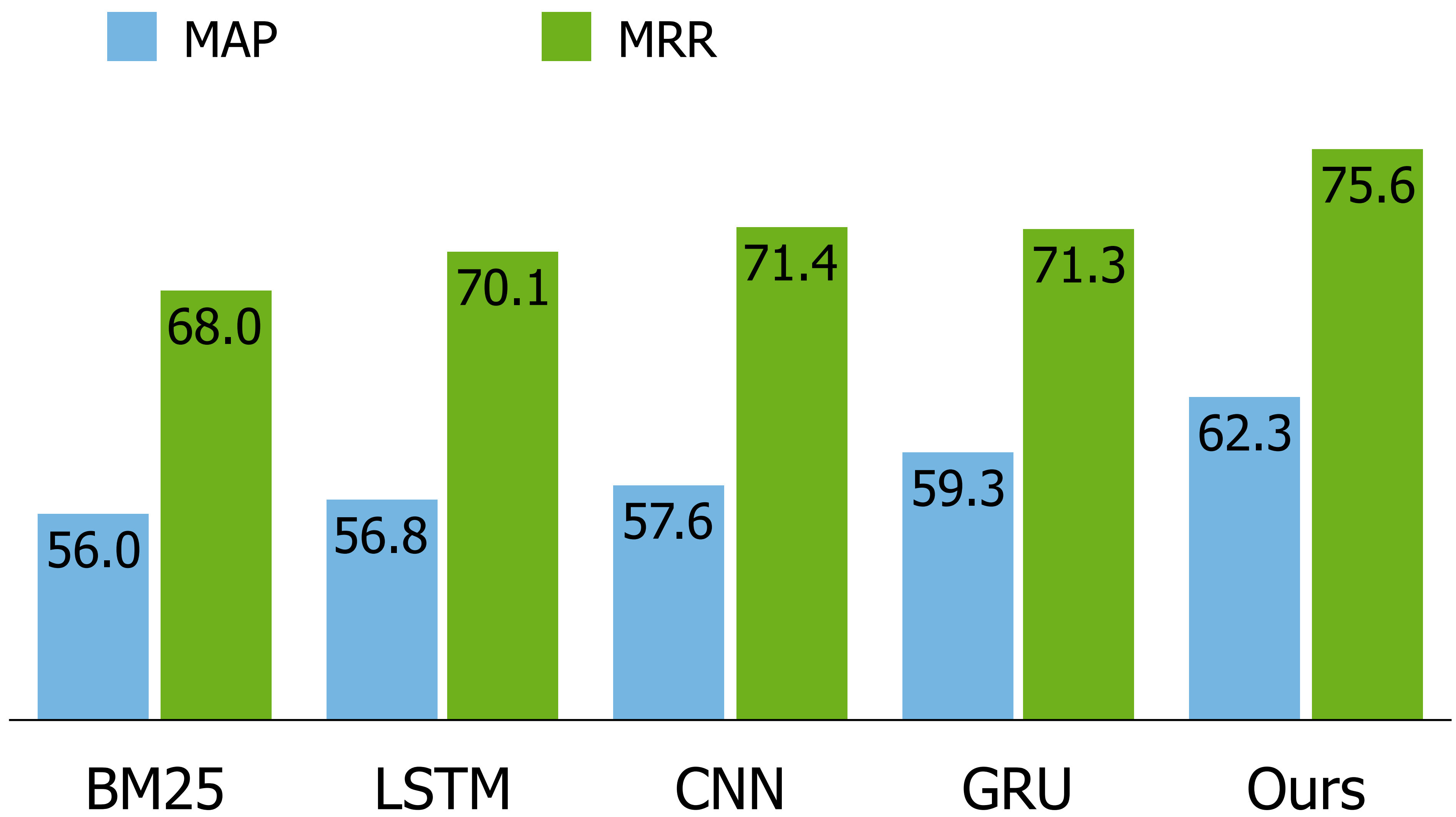
CNNs, LSTMs and GRUs

Grid-search: learning rate, dropout, pooling, filter size, pre-training, …

5 independent runs for each config.

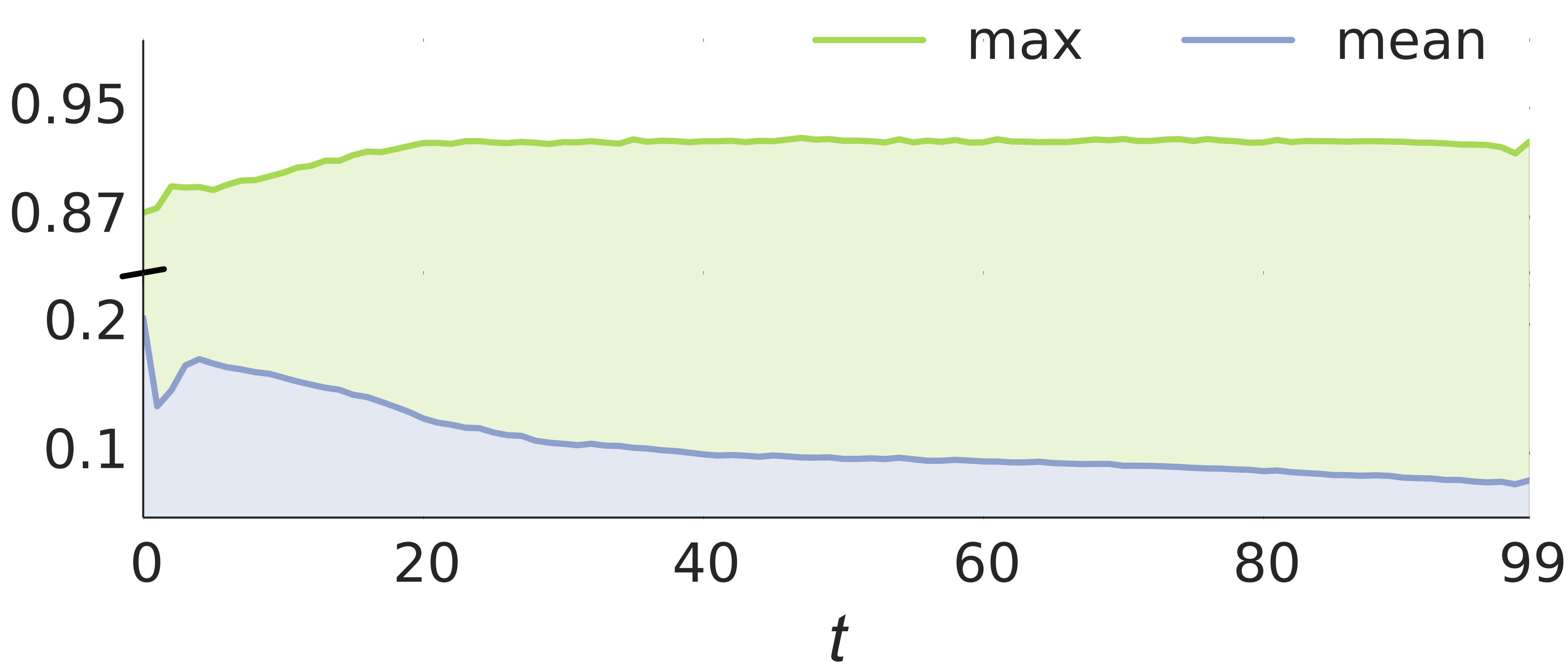
> 500 runs in total

Experiments: Retrieval



Our improvement is significant

Experiments: Retrieval

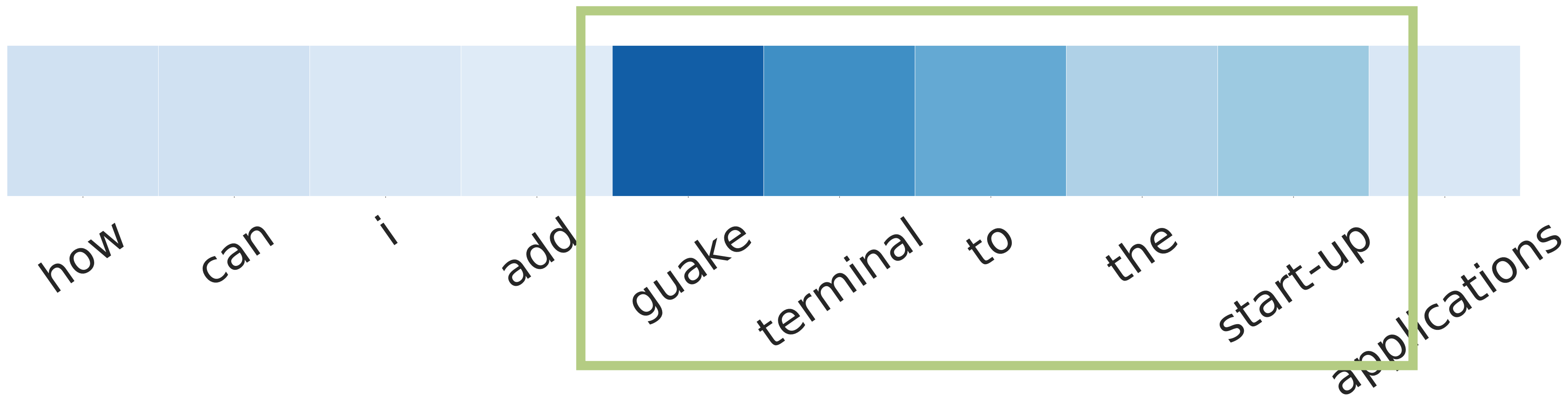


$$\mathbf{c}_t^{(3)} = \lambda \odot \mathbf{c}_{t-1}^{(3)} + \boxed{(1-\lambda)} \odot \left(\mathbf{c}_{t-1}^{(2)} + \mathbf{W}_3 \mathbf{x}_t\right)$$

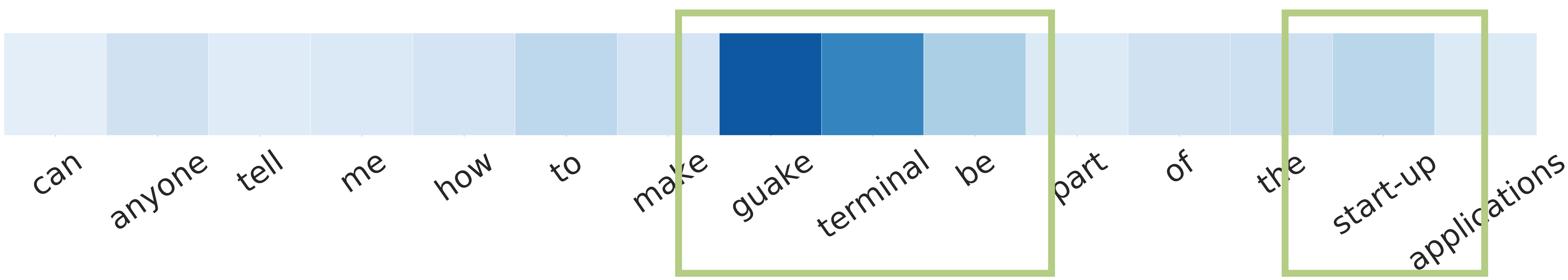
Analyze the weight vector over time

Experiments: Retrieval

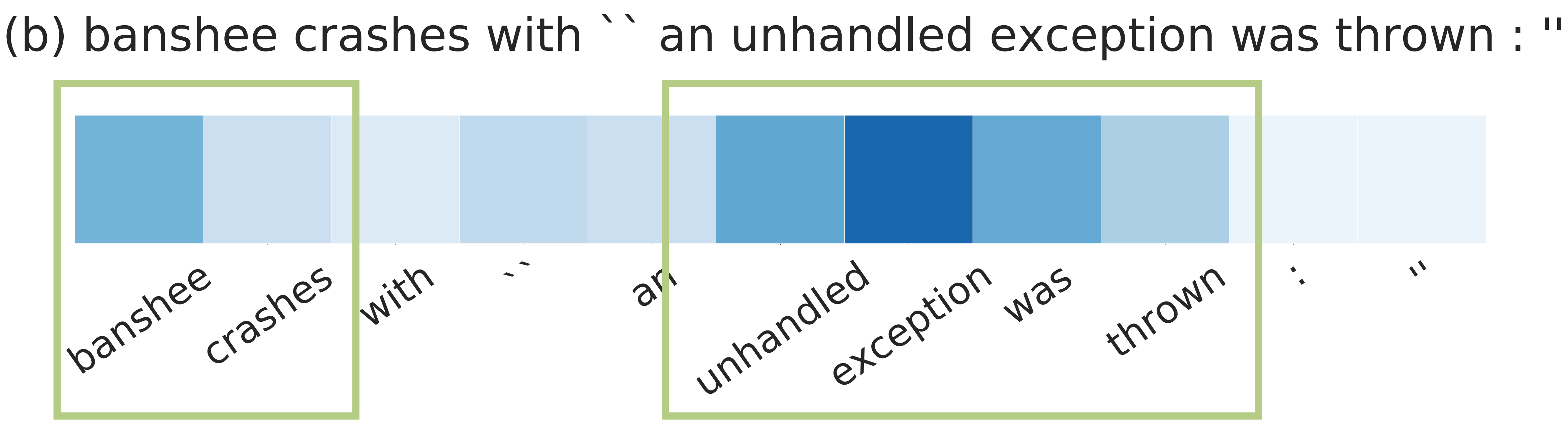
(a) how can i add guake terminal to the start-up applications



(f) can anyone tell me how to make guake terminal be part of the start-up applications



Experiments: Retrieval



Outlines (ii)

- *Rationalizing neural predictions*
  - a framework for understanding/justifying predictions
  - rationales are extracted from input as “supporting evidence”
  - can be optimized in RL w/o rationale annotations

Motivation

- Complex (neural) models come at the cost of interpretability
- Applications often need interpretable justifications — rationales.

this beer pours ridiculously clear with tons of carbonation that forms a rather impressive rocky head that settles slowly into a fairly dense layer of foam. this is a real good lookin' beer, unfortunately it gets worse from here ... first, the aroma is kind of bubblegum-like and grainy. next, the taste is sweet and grainy with an unpleasant bitterness in the finish. ... ... overall, the fat weasel is good for a fairly cheap buzz, but only if you like your beer grainy and bitter .

Ratings

Look: 5 stars

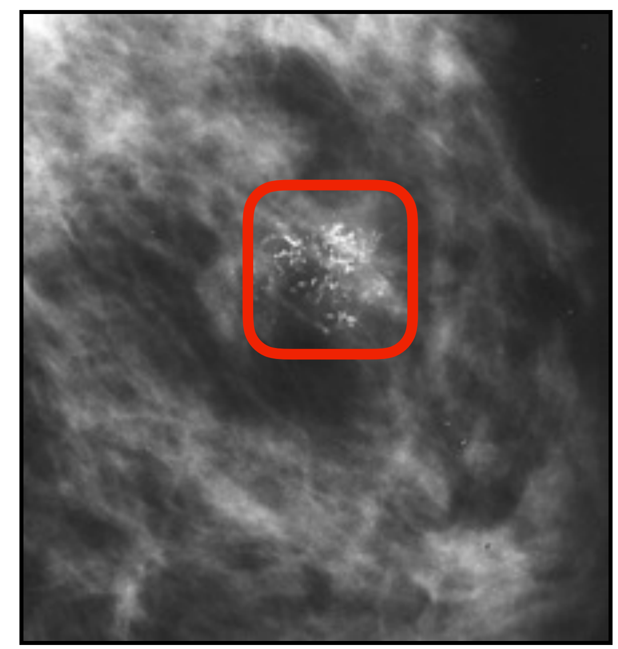
Aroma: 2 stars

review with rationales

Motivation

- Complex (neural) models come at the cost of interpretability
- Applications often need interpretable justifications — rationales.

There is no evidence of extranodal extension.
BREAST (RIGHT), EXCISIONAL BIOPSY:
INVASIVE DUCTAL CARCINOMA (SEE TABLE #1). DUCTAL CARCINOMA IN-SITU, GRADE 1. ATYPICAL DUCTAL HYPERPLASIA. LOBULAR NEOPLASIA (ATYPICAL LOBULAR HYPERPLASIA). TABLE OF PATHOLOGICAL FINDINGS #1 INVASIVE CARCINOMA
... ...



prediction: high risk of recurring cancer

Doctors won't trust machines, unless evidence is provided

Motivation

- Complex (neural) models come at the cost of interpretability
- Applications often need interpretable justifications — rationales.

Our goal: make powerful models more interpretable by learning rationales behind the prediction

Problem Setup

Interpretability via providing concise evidence from input

Rationales (evidence) should be:

- short and coherent pieces
- sufficient for correct prediction

Rationales are not provided during training

in contrast to *(Zaidan et al., 2007; Marshall et al.,2015; Zhang et al., 2016)*

Use powerful neural nets to avoid accuracy loss

in contrast to *(Thrun, 1995; Craven and Shavlik, 1996; Ribeiro et al., 2016)*

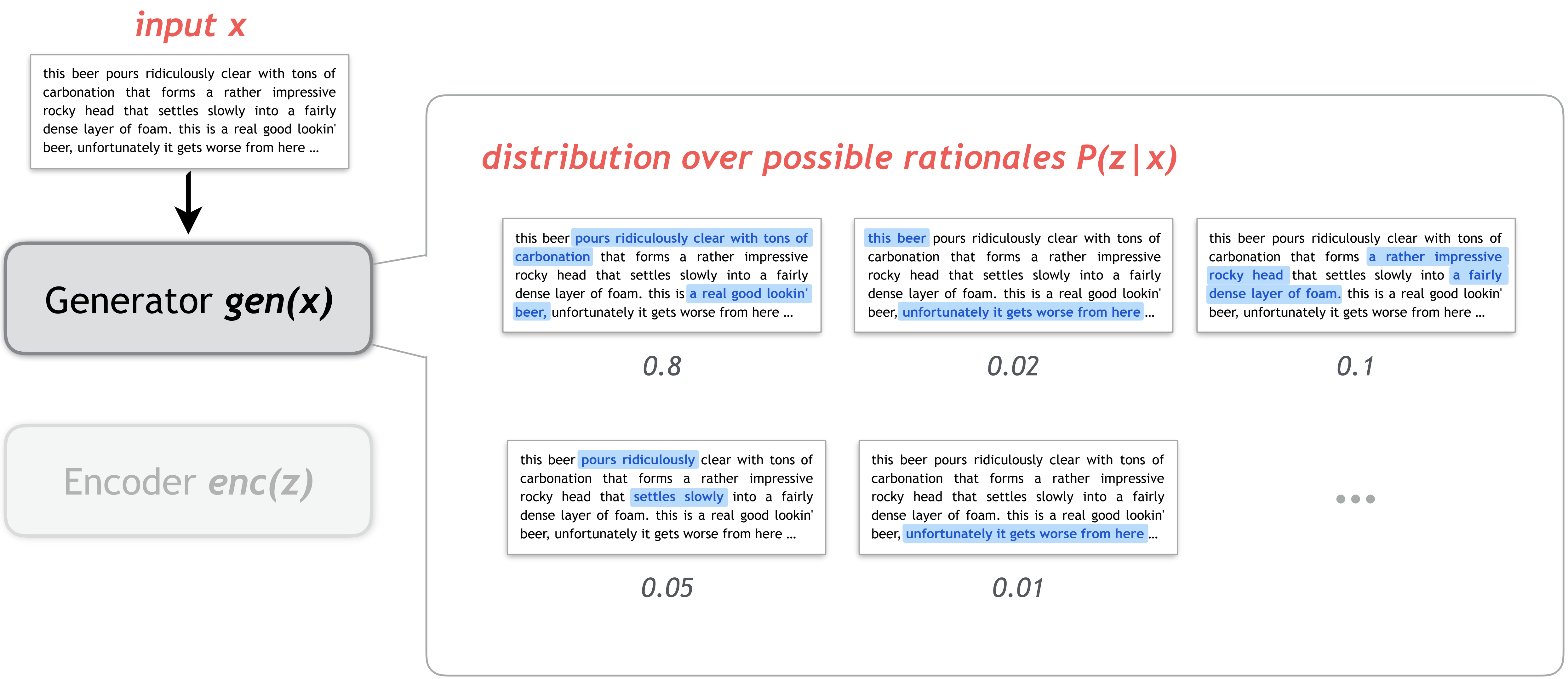
Model Architecture

Generator ***gen(x)***

Encoder ***enc(z)***

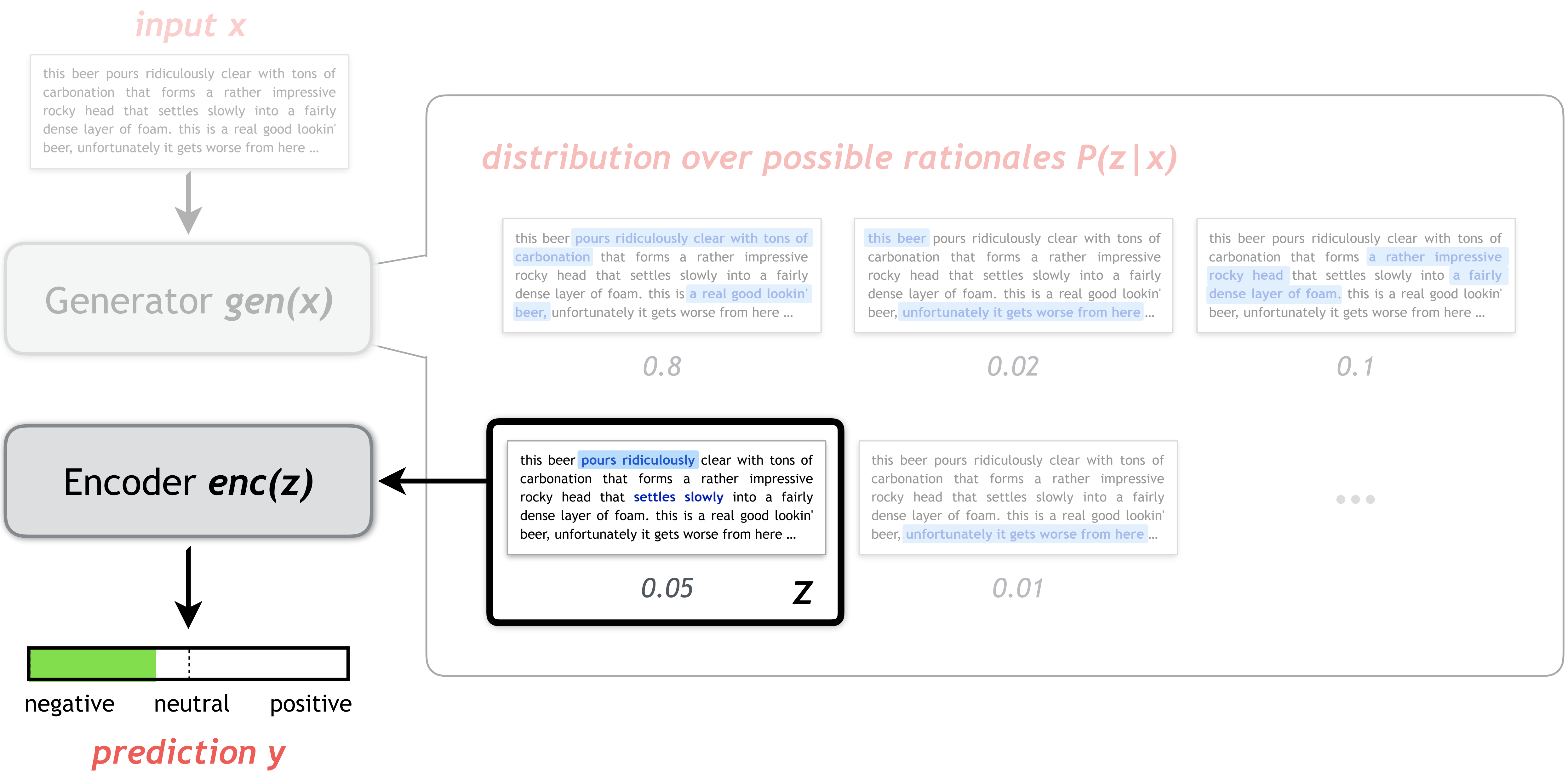
two modular components ***gen()*** and ***enc()***

Model Architecture



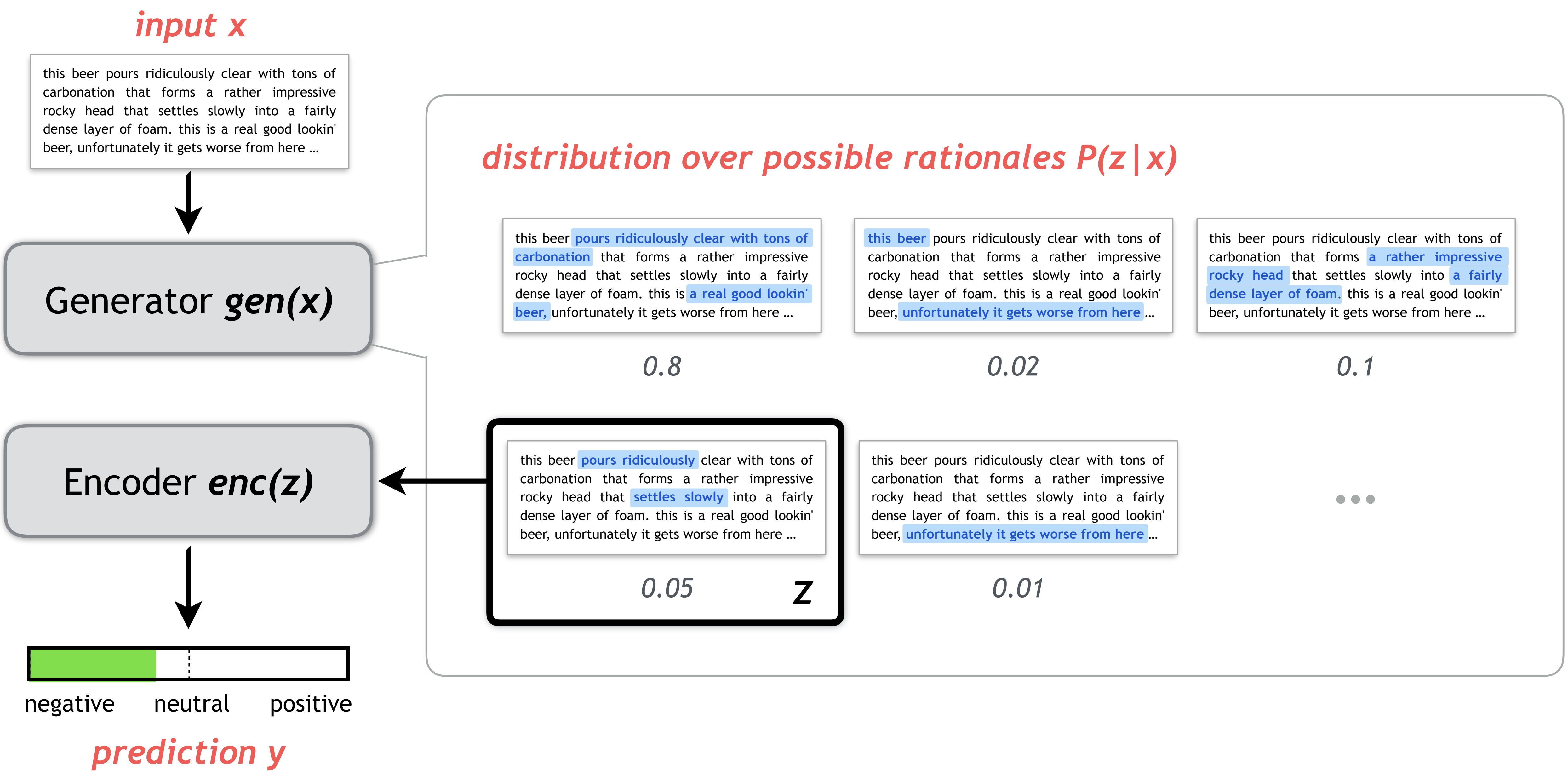
generator specifies the distribution of rationales

Model Architecture



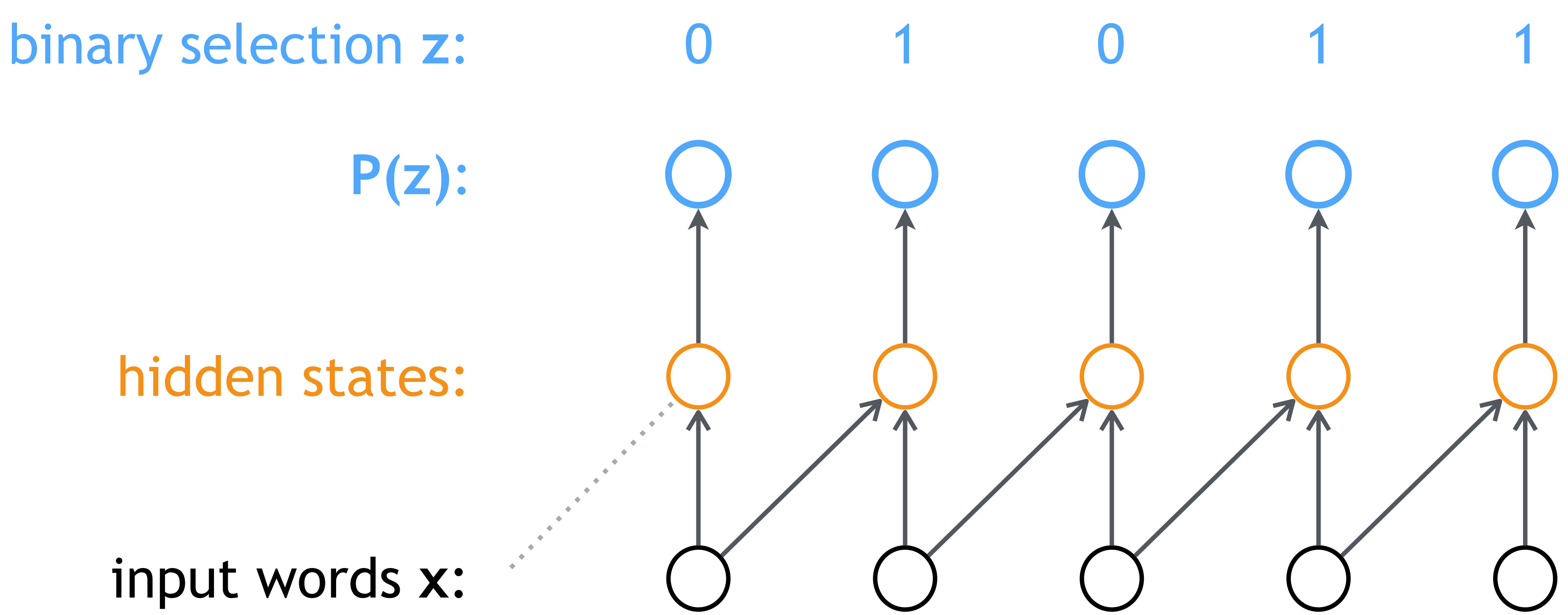
encoder makes prediction given rationale

Model Architecture



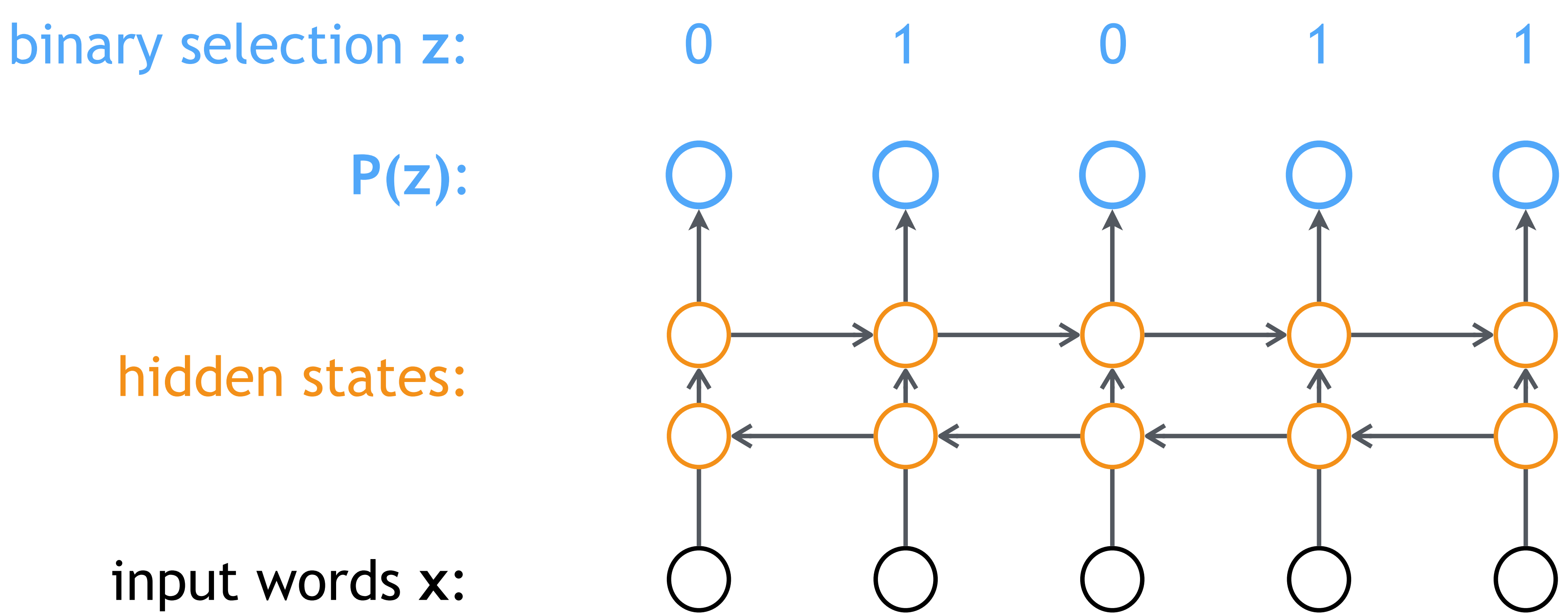
two components optimized jointly

Generator Implementations



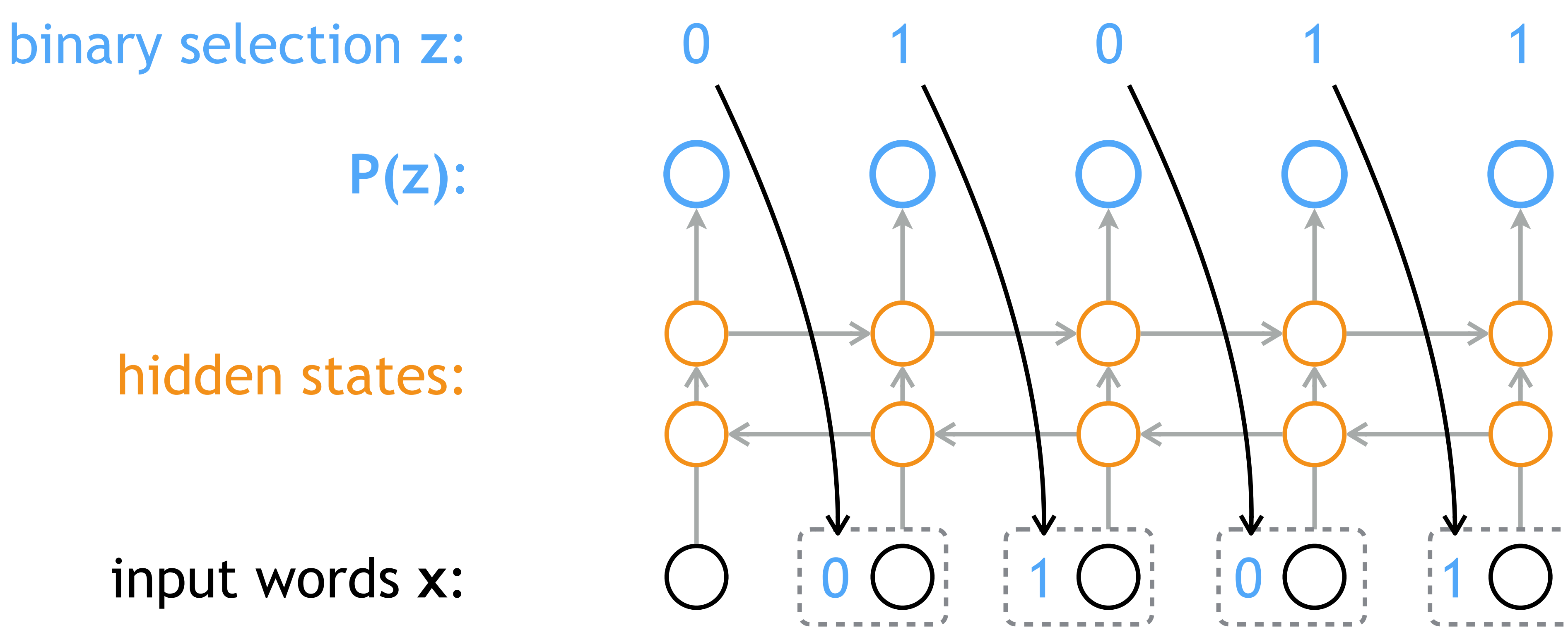
independent selection, feedforward net

Generator Implementations



independent selection, bi-directional RNNs

Generator Implementations



dependent selection, bi-directional RNNs

choose networks based on the data/application

Training Objective

$$\mathrm{cost}(\mathbf{z},\mathbf{y}) \;=\; \boxed{\mathrm{loss}(\mathbf{z},\mathbf{y})} + \boxed{\lambda_1 |\mathbf{z}|_1} + \boxed{\lambda_2 \sum_i |\mathbf{z}_i - \mathbf{z}_{i-1}|}$$

sufficiency *correct prediction*

sparsity *rationale is short*

coherency *continuous selection*

- receive this training signal after z is produced

Minimizing expected cost:

$$\min_{\theta} \sum_{(\mathbf{x},\mathbf{y})\in D} \mathbb{E}_{\mathbf{z}\sim \mathrm{gen}(\mathbf{x})} \left[\, \mathrm{cost}(\mathbf{z},\mathbf{y}) \,\right]$$

- intractable because summation over z is exponential

Learning Method

- Possible to sample the gradient, e.g.:

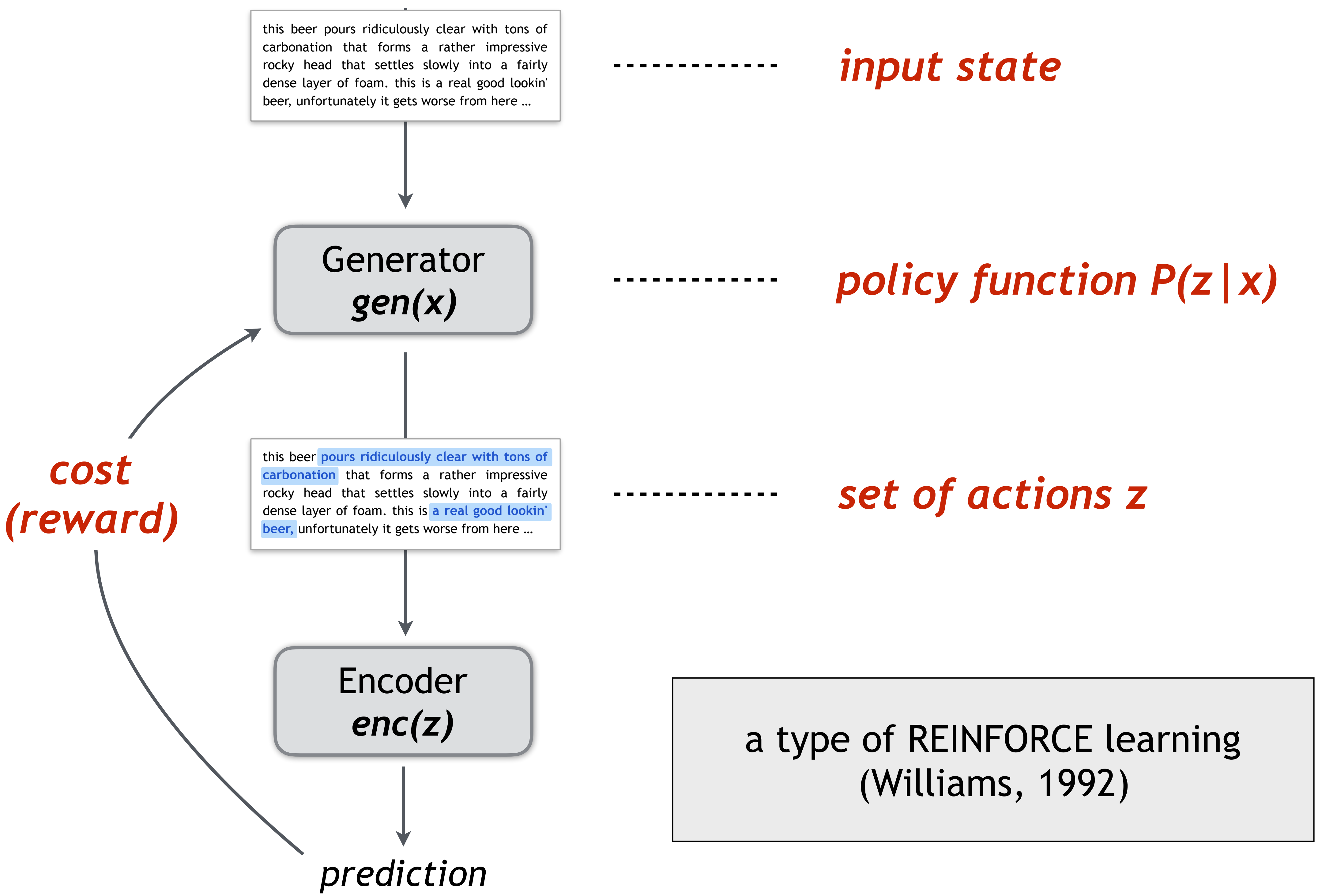
$$\mathbb{E}_{\mathbf{z}\sim \mathrm{gen}(\mathbf{x})}\left[\mathrm{cost}(\mathbf{z},\mathbf{y})\;\frac{\partial \log P(\mathbf{z}|\mathbf{x})}{\partial \theta_g}\right]$$

$$\approx \;\frac{1}{N}\sum_{i=1}^{N}\mathrm{cost}(\mathbf{z}_i,\mathbf{y}_i)\frac{\partial \log P(\mathbf{z}_i|\mathbf{x}_i)}{\partial \theta_g}$$

where z_i are sampled rationales

- Stochastic gradient decent on sampled gradients

Learning as Policy Gradient Method



Experiments

Three real-world datasets and applications for evaluation:

Predicting sentiment for product reviews

Parsing medical pathology reports

Finding similar posts on QA forum

Evaluation: Product Review

Dataset: multi-aspect beer reviews from *BeerAdvocate* (McAuley et al, 2012) 1.5m in total

1,000 reviews annotated at sentence level with aspect label (used only for evaluation)

Task: predict ratings and rationales for each aspect

this beer **pours ridiculously clear with tons of carbonation** that forms a rather impressive rocky head that settles slowly into a fairly dense layer of foam. **this is a real good lookin' beer**, unfortunately it gets worse from here ... first, **the aroma is kind of bubblegum-like and grainy.** next, the taste is sweet and grainy with an unpleasant bitterness in the finish. ... ... overall, the fat weasel is good for a fairly cheap buzz, but only if you like your beer grainy and bitter .

Ratings

Look: 5 stars

Aroma: 2 stars

Evaluation: Product Review

Set-up: ratings are fractional; treat the task as **regression** following (McAuley et al, 2012)

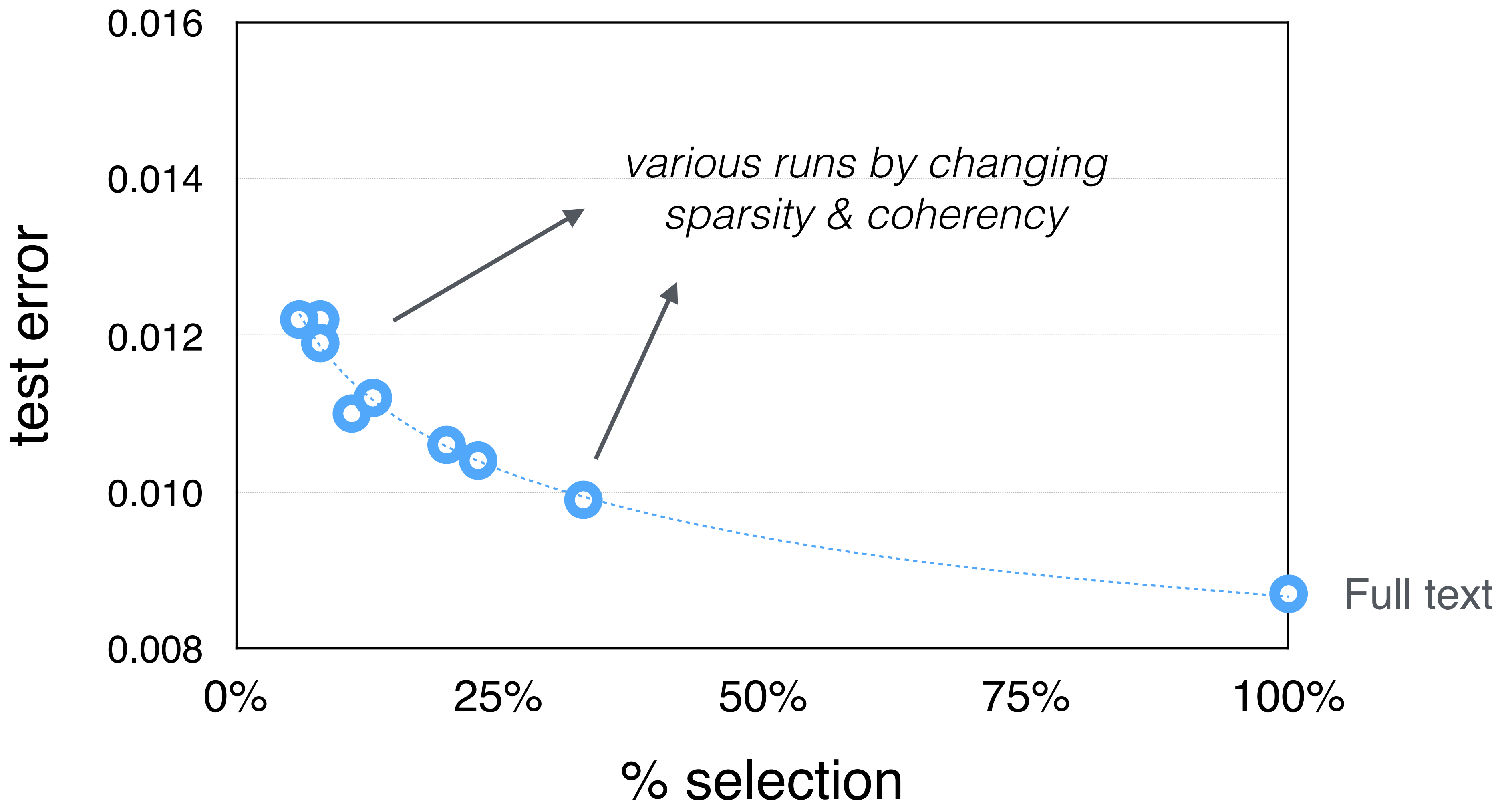
use recurrent networks for ***gen()*** and ***enc()***

Metrics: **precision**:
percentage of selected words in correct sentences

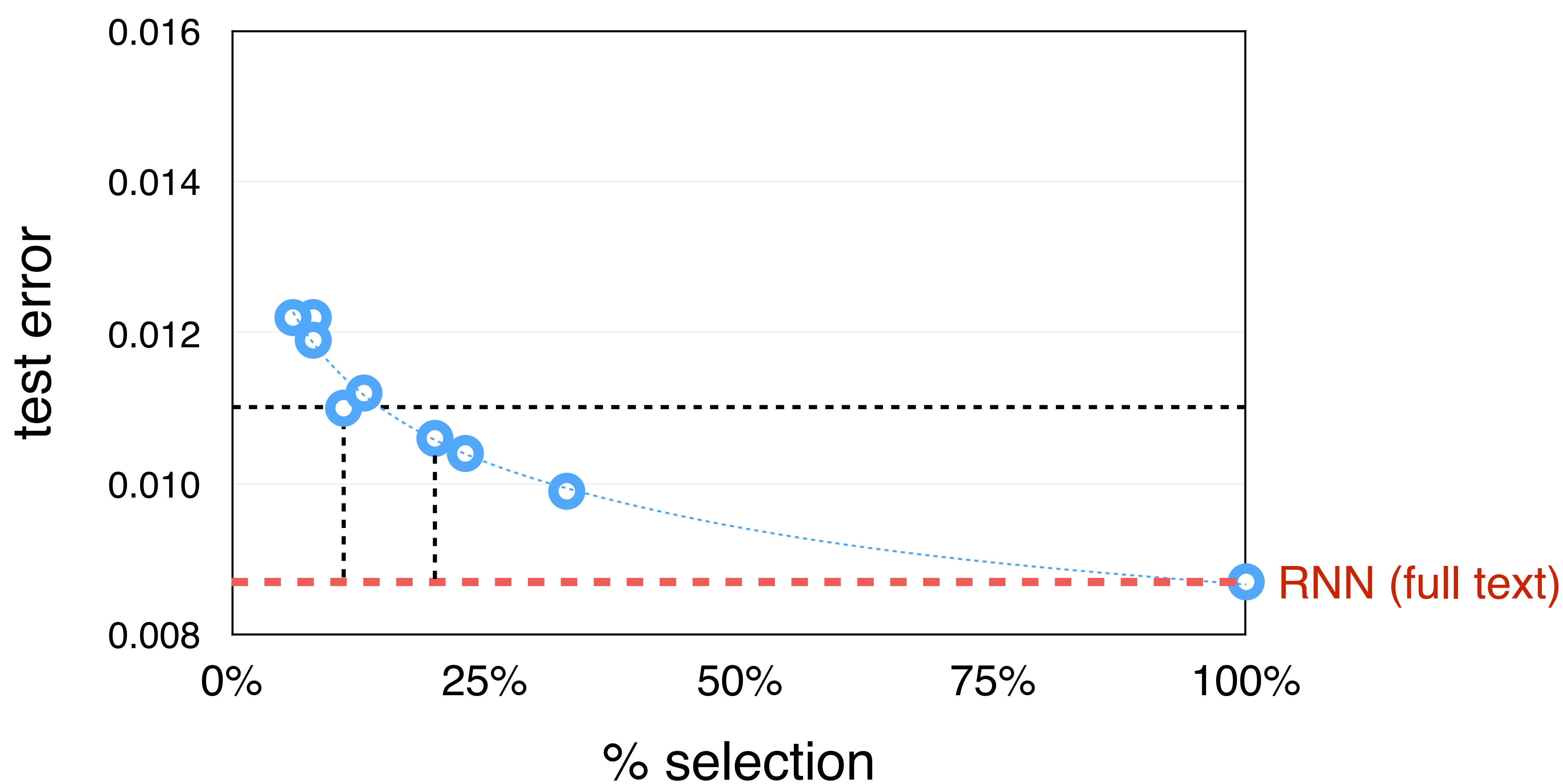
mean squared error on sentiment prediction

Baselines: SVM classifier
attention-based RNN

Sentiment Prediction

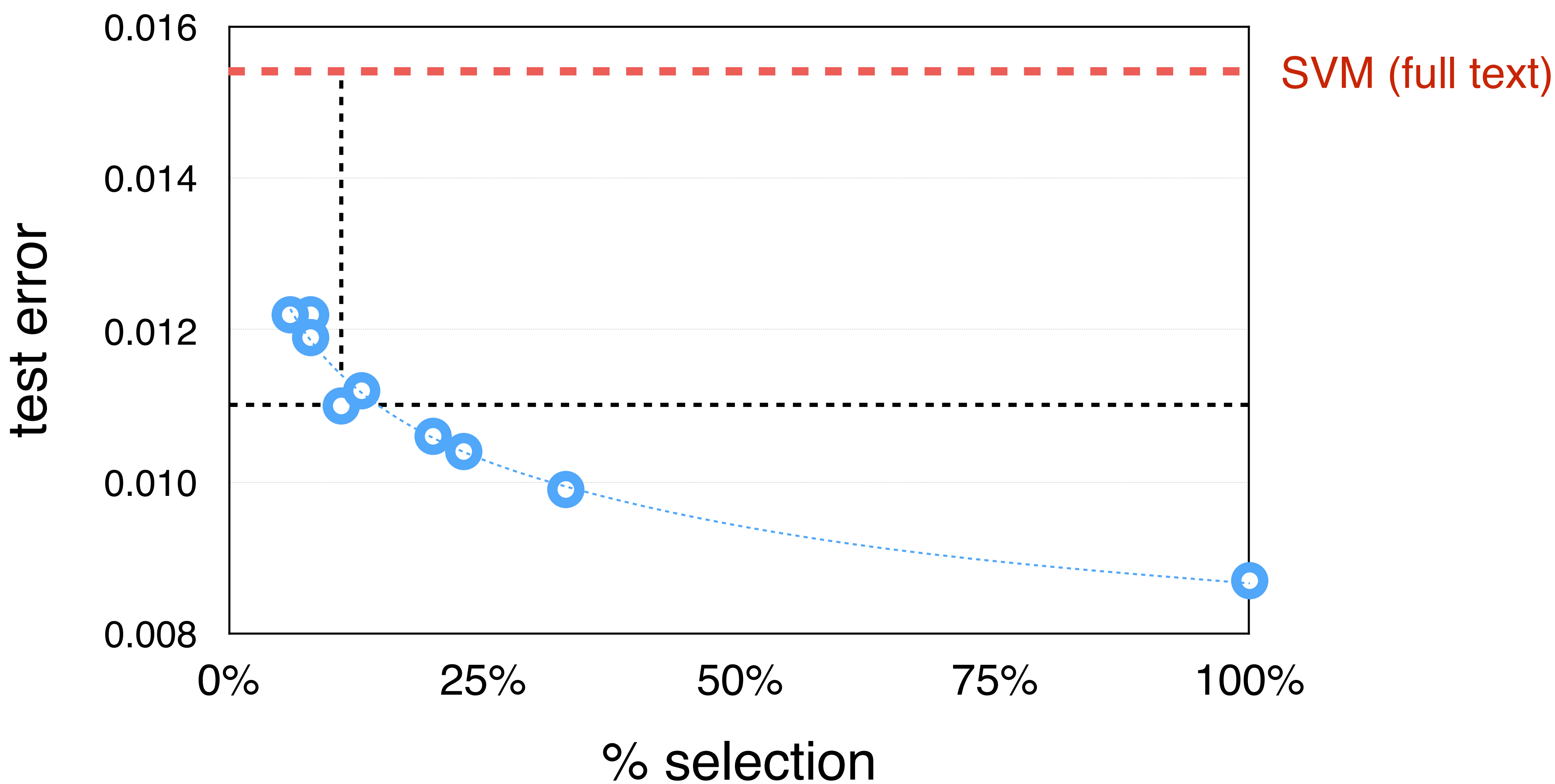


Sentiment Prediction



rationales getting close performance to full text

Sentiment Prediction



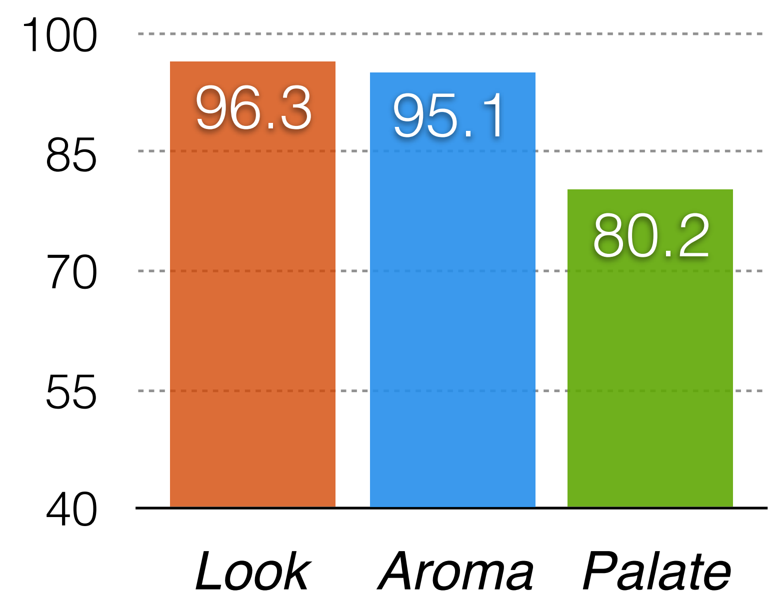
advantage of neural models over linear classifiers still clear

Precision of Rationales

Examples and precisions of rationales

a beer that is not sold in my neck of the woods , but managed to get while on a roadtrip . poured into an imperial pint glass with a generous head that sustained life throughout . nothing out of the ordinary here , but a good brew still . body was kind of heavy , but not thick . the hop smell was excellent and enticing . very drinkable

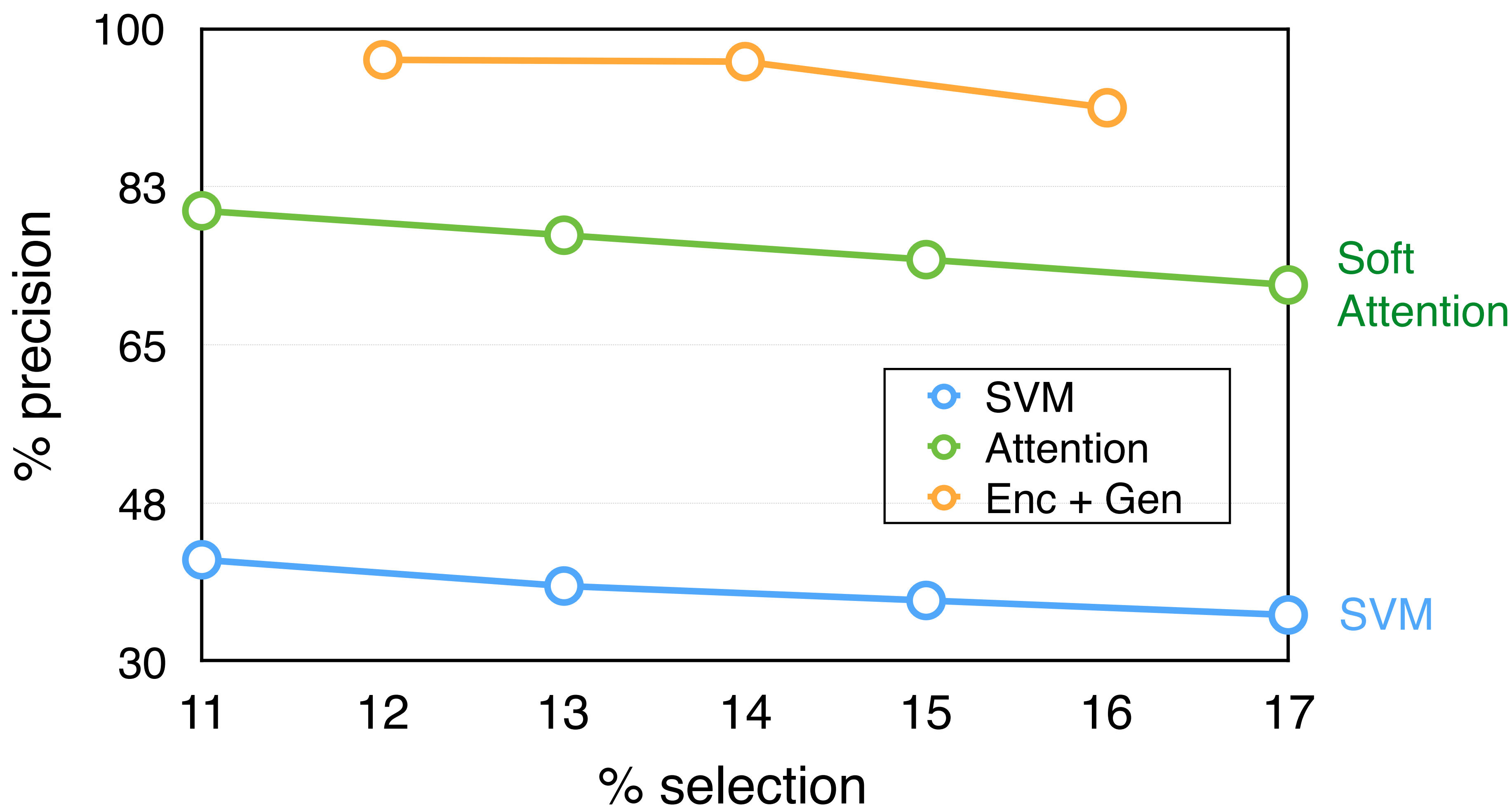
poured into a snifter . produces a small coffee head that reduces quickly . black as night . pretty typical imp . roasted malts hit on the nose . a little sweet chocolate follows . big toasty character on the taste . in between i 'm getting plenty of dark chocolate and some bitter espresso . it finishes with hop bitterness . nice smooth mouthfeel with perfect carbonation for the style . overall a nice stout i would love to have again , maybe with some age on it .



more examples available at

https://github.com/taolei87/rcnn/tree/master/code/rationale

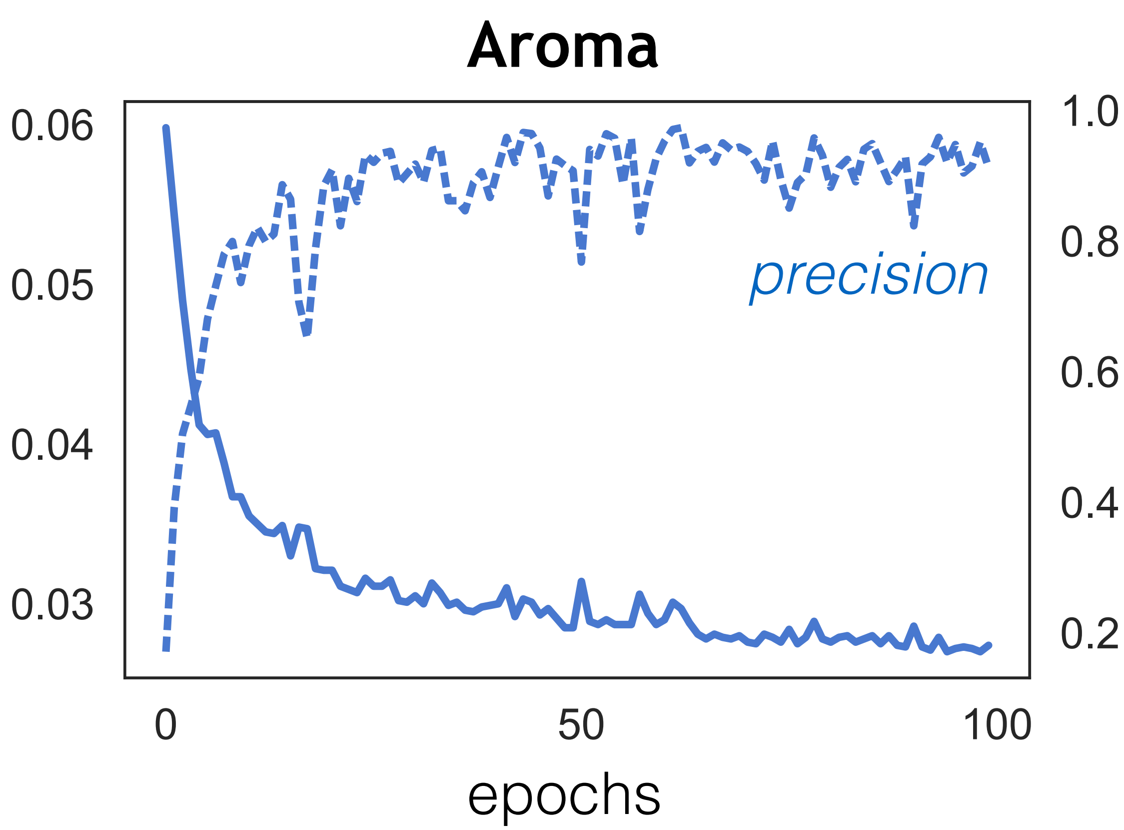
Precision of Rationales

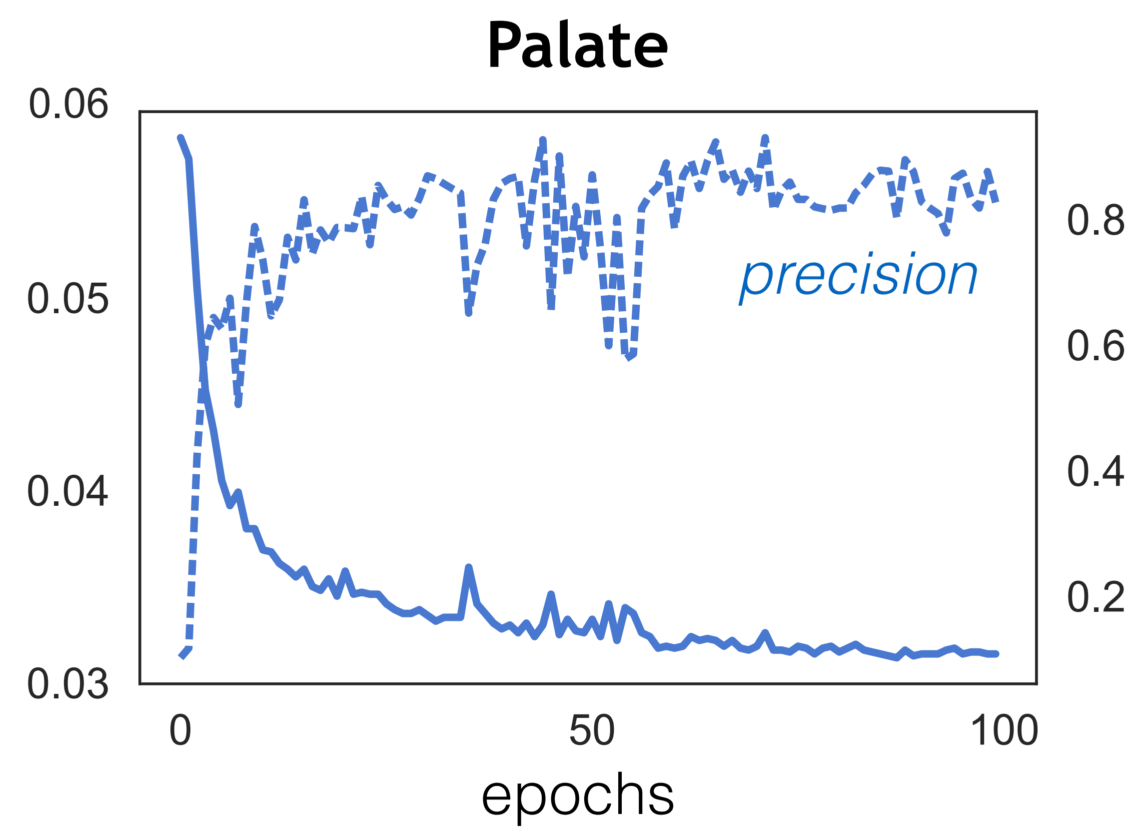


proper modeling leads to better rationale

Learning Curves

Learning curves of **cost(z)** on dev and precision on test





find good rationales after epochs of exploration

Evaluation: Parsing Pathology Report

Dataset: patients' pathology reports from hospitals such as MGH

Task: check if a disease/symptom is positive in text

binary classification for each category

Statistics: several thousand report for each category

pathology report is long (>1000 words) but structured

Model: use CNNs fro ***gen()*** and ***enc()***

Evaluation: Parsing Pathology Report

Category:		*F-score:*
IDC	*Accession Number* <unk> *Report Status* Final *Type* Surgical Pathology … *Pathology Report:* LEFT BREAST ULTRASOUND GUIDED CORE NEEDLE BIOPSIES … **INVASIVE DUCTAL CARCINOMA poorly differentiated modified Bloom** Richardson grade III III measuring at least 0 7cm in this limited specimen Central hyalinization is present within the tumor mass but no necrosis is noted No lymphovascular invasion is identified No in situ carcinoma is present Special studies were performed at an outside institution with the following results not reviewed ESTROGEN RECEPTOR NEGATIVE PROGESTERONE RECEPTOR NEGATIVE …	98%
LCIS	… **Extensive** LCIS DCIS **Invasive** carcinoma of left breast FINAL DIAGNOSIS BREAST **LEFT LOBULAR CARCINOMA IN SITU PRESENT** ADJACENT TO PREVIOUS BIOPSY SITE SEE NOTE CHRONIC INFLAMMATION ORGANIZING HEMORRHAGE AND FAT NECROSIS BIOPSY SITE NOTE There is a second area of focal lobular carcinoma in situ noted with pagetoid spread into ducts No vascular invasion is seen The margins are free of tumor No tumor seen in 14 lymph nodes examined BREAST left breast is a <unk> gram 25 x 28 x 6cm left …	97%
LVI	FINAL DIAGNOSIS BREAST RIGHT EXCISIONAL BIOPSY INVASIVE DUCTAL CARCINOMA DUCTAL CARCINOMA IN SITU SEE TABLE 1 MULTIPLE LEVELS EXAMINED TABLE OF PATHOLOGICAL FINDINGS 1 INVASIVE CARCINOMA Tumor size <unk> X <unk> X 1 3cm Grade 2 **Lymphatic vessel invasion Present Blood vessel invasion Not identified** Margin of invasive carcinoma Invasive carcinoma extends to less than 0 2cm from the inferior margin of the specimen in one focus Location of ductal carcinoma in situ …	84%

Evaluation: Question Retrieval

Dataset: question posts from *AskUbuntu* forum (dos Santos et al., 2015; Lei et al., 2016)

question pairs annotated as similar by users

Task: optimize neural representations such that distance between similar questions is small

Rationales:

underlined texts are question titles

what is the easiest way to **install all the media codec available** for ubuntu ? i am having issues with multiple applications prompting me to install codecs before they can play my files . how do i install **media codecs** ?
please any one give the solution for this whenever i try to **convert the rpm file to deb** file i always get this problem error : <unk> : not an **rpm package** (or package **manifest**) error executing `` lang=c rpm -qp -- queryformat % { name } <unk> ' '' : at <unk> line 489 thanks . converting **rpm file** to debian file

Conclusion

Explain model's design:

- We derive better justified (recurrent) neural architectures that are inspired by traditional kernel methods;
- We show model with better intuition and understanding can lead to better performance

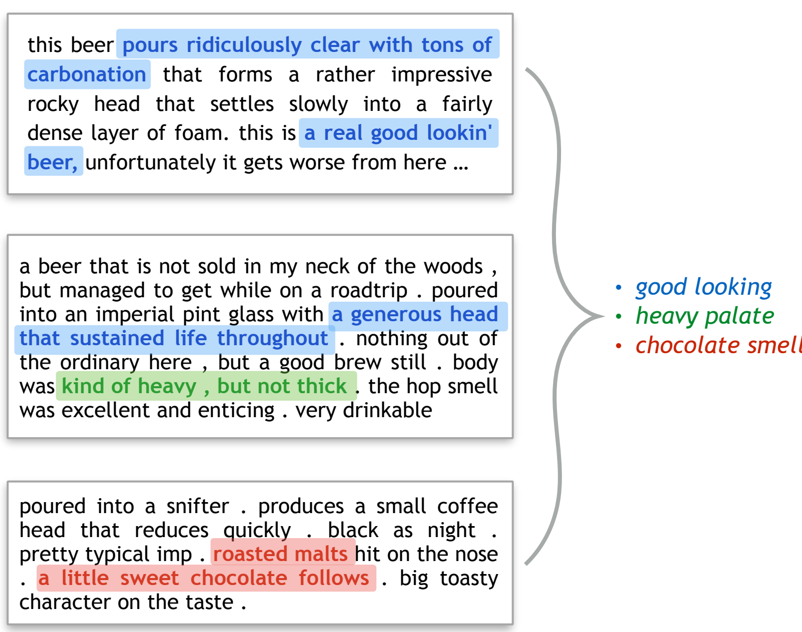
Explain model's prediction:

- We present a prototype framework for rationalizing model predictions, and evaluate it quantitatively and qualitatively on various applications

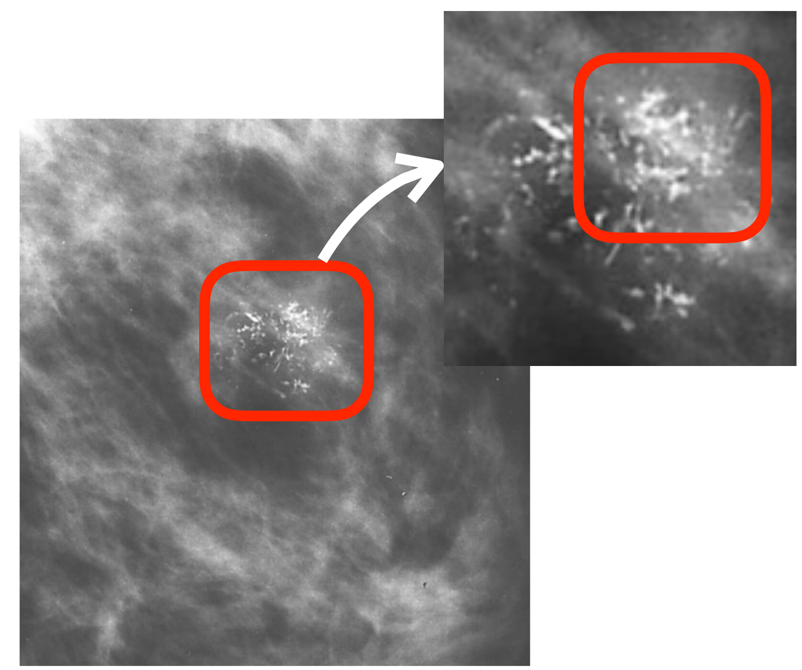
Future Work

interpretable components for trees and graphs

aggregation



vision



improve training (variance reduction)

… …