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

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

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

state-of-the-art unit for language modeling
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

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

how to interpret and understand it?
Sentence: “the movie is not that good”

Bag of words, TF-IDF

$\phi(x)$

Neural Bag-of-words (average embedding)

$\phi(x) \cdot M_{emb}$
Sentence: “the movie is not that good”

Ngram Kernel
(N=2)

not that
the movie
that good
is not
movie is

$\phi(x)$

CNNs

$\phi(x) \cdot M_{filter}$

Pre-activation as a dimension-reduction or projection of traditional methods
Sentence: “the movie is not that good”

String Kernel

\[
\begin{bmatrix}
0 \\
\lambda^0 \\
\lambda^2 \\
\vdots \\
\lambda^1 \\
0
\end{bmatrix}
\]

expands feature space \( \lambda \in (0, 1) \) penalize skips

Neural model inspired by this kernel method
Illustration

the movie is not that good
Illustration

The movie is not that good.
Illustration

the movie is not that good
Formulas

\[ c^{(1)}_t = \lambda_t \odot c^{(1)}_{t-1} + (1 - \lambda_t) \odot (W^{(1)}x_t) \]

\[ c^{(2)}_t = \lambda_t \odot c^{(2)}_{t-1} + (1 - \lambda_t) \odot (c^{(1)}_{t-1} + W^{(2)}x_t) \]

\[ h_t = \tanh(c^{(2)}_t) \]

aggregated 1-gram and 2-gram features
Formulas

\[c_t^{(1)} = \lambda_t \odot c_{t-1}^{(1)} + (1 - \lambda_t) \odot (W^{(1)}x_t)\]
\[c_t^{(2)} = \lambda_t \odot c_{t-1}^{(2)} + (1 - \lambda_t) \odot (c_{t-1}^{(1)} + W^{(2)}x_t)\]
\[h_t = \tanh(c_t^{(2)})\]

re-normalize to remove length bias

decay penalizing skip grams
Formulas

\[ c_t^{(1)} = \lambda_t \odot c_{t-1}^{(1)} + (1 - \lambda_t) \odot (W^{(1)}x_t) \]

\[ c_t^{(2)} = \lambda_t \odot c_{t-1}^{(2)} + (1 - \lambda_t) \odot (c_{t-1}^{(1)} + W^{(2)}x_t) \]

\[ h_t = \tanh(c_t^{(2)}) \]

\[ \lambda_t = 0 : \quad h_t = \tanh(W^{(1)}x_{t-1} + W^{(2)}x_t) \quad \text{(one-layer CNNs)} \]
Formulas

\[ c_t^{(1)} = \lambda_t \odot c_{t-1}^{(1)} + (1 - \lambda_t) \odot (W^{(1)} x_t) \]

\[ c_t^{(2)} = \lambda_t \odot c_{t-1}^{(2)} + (1 - \lambda_t) \odot (c_{t-1}^{(1)} \odot W^{(2)} x_t) \]

\[ h_t = \tanh(c_t^{(2)}) \]

*multiplicative mapping*
Formulas

\[ c_t^{(1)} = \lambda_t \odot c_{t-1}^{(1)} + (1 - \lambda_t) \odot (W^{(1)}x_t) \]

\[ c_t^{(2)} = \lambda_t \odot c_{t-1}^{(2)} + (1 - \lambda_t) \odot (c_{t-1}^{(1)} \odot W^{(2)}x_t) \]

... 

\[ c_t^{(n)} = \lambda_t \odot c_{t-1}^{(n)} + (1 - \lambda_t) \odot (c_{t-1}^{(n-1)} \odot W^{(n)}x_t) \]

\[ h_t = \tanh(c_t^{(n)}) \]

*can be generalized to n-grams*
String kernel counts shared patterns in sequences $x$ and $y$:

$$K_2(x, y) = \sum_{1 \leq i < j \leq |x|} \sum_{1 \leq k < l \leq |y|} \lambda^{|x| - i - 1} \lambda^{|y| - k - 1} [\mathbb{1}(x_i = y_k) \cdot \mathbb{1}(x_j = y_l)]$$

Written in vector form:

(i) multiplicative

$$\langle x_i, y_k \rangle \langle x_j, y_l \rangle$$

(ii) additive

$$\langle x_i, y_k \rangle + \langle x_j, y_l \rangle$$
From Kernel to Neural Model

String kernel counts shared patterns in sequences $\mathbf{x}$ and $\mathbf{y}$:

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

$$
= \left( \sum_{1 \leq i < j \leq |\mathbf{x}|} \lambda^{|\mathbf{x}| - i - 1} \mathbf{x}_i \otimes \mathbf{x}_j, \sum_{1 \leq k < l \leq |\mathbf{y}|} \lambda^{|\mathbf{y}| - k - 1} \mathbf{y}_k \otimes \mathbf{y}_l \right)
$$

$\phi(\mathbf{x})$ **underlying mapping**
From Kernel to Neural Model

Projecting $\phi(x)$ to hidden representation $c_t \in \mathbb{R}^d$

$$c_t^{(2)}[k] = \langle w_k^{(1)} \otimes w_k^{(2)}, \sum_{1 \leq i < j \leq t} \lambda^{|x_i - x_j|} x_i \otimes x_j \rangle$$

$k$-th filter

$\phi(x_{1:t})$

$$= \mathcal{K}_2 \left( w_k^{(1)} w_k^{(2)}, x_1 x_2 \cdots x_t \right)$$

can be seen as evaluating kernel functions;
naturally embeds sequence similarity computation
From Kernel to Neural Model

Projecting $\phi(x)$ to hidden representation $c_t \in \mathbb{R}^d$

$$c_t^{(2)}[k] = \langle w_k^{(1)} \otimes w_k^{(2)}, \sum_{1\leq i<j\leq t} \lambda^{||x||_{-i-1}} x_i \otimes x_j \rangle$$

$k$-th filter

$$\phi(x_{1:t})$$

**Efficient implementation to compute** $c_t$ *(dynamic programming)*

$$c_t^{(2)}[k] = \lambda \cdot c_{t-1}^{(2)}[k] + c_{t-1}^{(1)}[k] \cdot \langle w_k^{(2)}, x_t \rangle$$
Efficient implementation to compute $c_t$ (dynamic programming)

$$c_t^{(2)}[k] = \lambda \cdot c_{t-1}^{(2)}[k] + c_{t-1}[k] \cdot \langle w_{k}^{(2)}, x_t \rangle$$

- 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**

\[ \phi(x) = \phi_2(\phi_1(x)) \]

**Stacking multiple layers**

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

\[ \phi(x) = \sum_{i,j} \lambda^{t-i-1} \phi_1(x_{1:i}) \otimes \phi_1(x_{1:j}) \]
Choices of Decay

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

depends on x: \[ \lambda_t = \sigma(Ux_t + b) \]

depends on x and h: \[ \lambda_t = \sigma(Ux_t + Vh_{t-1} + b) \]
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

![Graph showing deeper models with better classification accuracy](image-url)
### Experiments: Classification

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

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

Test PPL of Small Networks (5m)

- CNNs: 99.0
- constants (0.8): 74
- constants (trained): 83
- adaptive (x): 91
- adaptive (x and h): 100
Experiments: Language Model

Test PPL of Small Networks (5m)

- CNNs: 99.0
- Constants (0.8): 84.3
- Constants (trained)
- Adaptive (x)
- Adaptive (x and h)
Experiments: Language Model

Test PPL of Small Networks (5m)

- CNNs: 99.0
- constants (0.8): 84.3
- constants (trained): 76.8
- adaptive (x): 
- adaptive (x and h): 

31
Experiments: Language Model

Test PPL of Small Networks (5m)

- CNNs: 99.0
- Constants (0.8): 84.3
- Constants (trained): 76.8
- Adaptive (x): 74.2
- Adaptive (x and h): 73.6
Experiments: Language Model

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

Comparison with state-of-the-art results can be improved with variational techniques.

Variational techniques have better regularized performance.
Experiments: Retrieval

Task: Find similar questions given the user’s input question
Experiments: Retrieval

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

Application to find duplicate MP3s [duplicate]

Possible Duplicate: How can I find duplicate songs?

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

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

\[ c_t^{(3)} = \lambda \odot c_{t-1}^{(3)} + (1 - \lambda) \odot (c_{t-1}^{(2)} + W_3 x_t) \]

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

(b) banshee crashes with `an unhandled exception was thrown` : "

```
banshee crashes with `unhandled exception was thrown` : "
```
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

<table>
<thead>
<tr>
<th>Look</th>
<th>5 stars</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aroma</td>
<td>2 stars</td>
</tr>
</tbody>
</table>

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

two modular components $gen()$ and $enc()$
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 ...

distribution over possible rationales $P(z|x)$

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 ...

0.8

0.02

0.1

0.05

0.01

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 ...

generator specifies the distribution of rationales
**Model Architecture**

**Generator** $\text{gen}(x)$

**Encoder** $\text{enc}(z)$

**distribution over possible rationales $P(z|x)$**

- 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 ...
- 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 ...
- 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 ...

**prediction $y$**

**encoder makes prediction given rationale**
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 ...

distribution over possible rationales $P(z|x)$

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 ...

prediction $y$

two components optimized jointly
Generator Implementations

binary selection $z$: 0 1 0 1 1

$P(z)$: 0 1 0 1 1

hidden states: 0 1 0 1 1

input words $x$: 0 1 0 1 1

independent selection, feedforward net
Generator Implementations

binary selection $z$: 0 1 0 1 1

$P(z)$:

hidden states:

input words $x$:

independent selection, bi-directional RNNs
Generator Implementations

binary selection \( z \):

\[
\begin{array}{cccc}
0 & 1 & 0 & 1 \\
\end{array}
\]

\( P(z) \):

\[
\begin{array}{cccc}
\text{hidden states:} \\
\end{array}
\]

\[
\begin{array}{cccc}
\text{input words} \ x: \\
0 & 1 & 0 & 1 \\
\end{array}
\]

dependent selection, bi-directional RNNs

choose networks based on the data/application
Training Objective

\[
\text{cost}(z, y) = \text{loss}(z, y) + \lambda_1 |z|_1 + \lambda_2 \sum_i |z_i - z_{i-1}|
\]

- **sufficiency**: correct prediction
- **sparsity**: rationale is short
- **coherency**: continuous selection

- receive this training signal after z is produced

**Minimizing expected cost:**

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

- intractable because summation over z is exponential
Learning Method

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

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

\[
\approx \frac{1}{N} \sum_{i=1}^{N} \text{cost}(z_i, y_i) \frac{\partial \log P(z_i|x_i)}{\partial \theta_g}
\]

where \(z_i\) are sampled rationales

- Stochastic gradient descent on sampled gradients
Learning as Policy Gradient Method

- **Generator** \(\text{gen}(x)\)

- **Encoder** \(\text{enc}(z)\)

- **input state**

- **policy function** \(P(z \mid x)\)

- **set of actions** \(z\)

- **cost (reward)**

- **prediction**

---

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 ...

a type of REINFORCE learning (Williams, 1992)
Experiments

Three real-world datasets and applications for evaluation:

- Predicting sentiment for product reviews
- Parsing medical pathology reports
- Finding similar posts on QA forum
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 $\text{gen()}$ and $\text{enc()}$

Metrics: precision:
percentage of selected words in correct sentences
mean squared error on sentiment prediction

Baselines: SVM classifier
attention-based RNN
Sentiment Prediction

\[
\text{test error} \quad \text{% selection}
\]

\[0\% \quad 25\% \quad 50\% \quad 75\% \quad 100\%\]

\[0.008 \quad 0.010 \quad 0.012 \quad 0.014 \quad 0.016\]

Various runs by changing sparsity & coherency

Full text
rationales getting close performance to full text
advantage of neural models over linear classifiers still clear
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.
Precision of Rationales

proper modeling leads to better rationale
Learning Curves

Learning curves of $\text{cost}(z)$ on dev and precision on test

**Aroma**

- Precision

**Palate**

- Precision

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 for $\text{gen}()$ and $\text{enc}()$
Evaluation: Parsing Pathology Report

**Category:**

**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 ...

**IDC**

**F-score:** 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 ...

**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
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:

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

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...

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. toasted malts hit on the nose. a little 'sweet chocolate follows'. big toasty character on the taste.

vision

• good looking
• heavy palate
• chocolate smell

improve training (variance reduction)

... ...