Learning with Latent Linguistic Structure

Graham Neubig
@ BlackBoxNLP 11/1/2018

with: Junxian He, Pengcheng Yin, Chunting Zhou, Taylor Berg-Kirkpatrick
How to Achieve Interpretability in Neural Nets?

Black Box

Analysis

Syntax

Semantics

Glass Box!

Syntax

Semantics
Research Problems

- Fundamentally highly interpretable models (e.g. discrete HMMs) are not sufficiently powerful
- How can we harness the power of neural networks, with underlying interpretable representations?
- How can we learn them on unlabeled data?
e.g. Syntactic Analysis

Dependency:

Parts-of-speech: DT NN VBD IN DT JJ NN

The cat sat on a green wall
Supervised Approach

John passes the ball upfield to Peter, who shoots for the goal. The shot is deflected by Mary and the ball goes out of bounds.
Supervised Learning

\[ X \quad \theta \quad Y \]

- John passes the ball upfield to Peter, who shoots for the goal. The shot is deflected by Mary and the ball goes out of bounds.

- John passes the ball upfield to Peter, who shoots for the goal. The shot is deflected by Mary and the ball goes out of bounds.

- John passes the ball upfield to Peter, who shoots for the goal. The shot is deflected by Mary and the ball goes out of bounds.

John passes the ball upfield to Peter, who shoots for the goal. The shot is deflected by Mary and the ball goes out of bounds.
Supervised Approach

Supervised Learning

\[ X \quad \theta \quad Y \]

John passes the ball upfield to Peter, who shoots for the goal. The shot is deflected by Mary and the ball goes out of bounds.

John passes the ball upfield to Peter, who shoots for the goal. The shot is deflected by Mary and the ball goes out of bounds.

John passes the ball upfield to Peter, who shoots for the goal. The shot is deflected by Mary and the ball goes out of bounds.

John passes the ball upfield to Peter, who shoots for the goal. The shot is deflected by Mary and the ball goes out of bounds.
Latent Variable Approach

Unsupervised Learning

\[ X \xrightarrow{\theta} Y \]

\[ X \]

\[ Y \]

\[ ? \]
Latent Variable Approach

Unsupervised Learning

$X$  $\theta$  $Y$

$X$  $\theta$  $Y$

Carnegie Mellon University
Multi-space Variational Encoder-Decoders

Chunting Zhou and Graham Neubig
(ACL 2017)
Features of Words

- **Syntax:**
  - What syntactic features does the word have?
  - Closed-class, generally enumerable for a specific language.

- **Meaning/Symbol:**
  - What is the meaning of the word, how is it spelled/pronounced?
  - Open-class, complicated regularities and relationships.

- Can we create a model that elegantly models both?
Background: Variational Auto-encoder (Kingma et al., 2014, Bowman et al., 2016)

Maximize the Variational lower bound:

$$\log p_\theta(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} \left[ \log p_\theta(\mathbf{x}|\mathbf{z}) \right] - KL(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))$$

$\ell$: Recognition model/Encoder

$p_\theta(\mathbf{x}|\mathbf{z})$: Generation model/Decoder
Proposed Model: Multi-space Variational Encoder-Decoder

- Modeling complicated higher-level structure (e.g. meaning or symbol of the word): incorporation of continuous latent variables
- Modeling closed-class and interpretable features (e.g. syntax): incorporation of discrete latent variables

plays, played, playing

- How can we learn in a un- or semi-supervised way?
**Variable Definitions**

- $x^{(s)}$: a source sequence
- $x^{(t)}$: a target sequence
- $y^{(t)} = [y_1^{(t)}, y_2^{(t)}, \cdots, y_K^{(t)}]$: discrete labels for each target sequence
- $z$: continuous latent variable

**Diagram**

```
  x^{(s)}: a source sequence  Lemma  x^{(t)}: a target sequence
  p l a y s <s>                   p l a y e d <s>
  +PP  +PAST  +VERB
```

**Lemma**

```
+PP
+PAST
+VERB
```
Supervised Learning: Labeled Multi-space Variational Autoencoders

\[ p(y|x, z) = f(x|y, z; \theta) \]

Decoder: \( p_\theta(x|y, z) = f(x|y, z; \theta) \)

(with label attention)

\[ q_\phi(z|x) = \mathcal{N}(z|\mu_\phi(x), \text{diag}(\sigma_\phi^2(x))) \]

Discrete Labels

``play''

``play''

\[ \mathcal{D}(x, y) = \mathbb{E}_{(x,y) \sim p_l(x,y)}[ - \log q_\phi(y|x)] \]

Maximize: \( \mathcal{U}(x) = \text{Variational Lower Bound of } \log p(x, y) \)
Unsupervised Learning: Unlabeled Multi-space Variational Auto-encoders

Training Data: surface form only

Maximize : $U(x) = \text{Variational Lower Bound of } \log p(x, y)$
Labeled Sequence-to-sequence Training: Multi-space Variational Encoder-Decoders

Training Data:
- two surface forms
- + labels

Latent Continuous

``play''

Discrete Labels

Maximize: $\mathcal{L}_l(x^{(t)}, y^{(t)}|x^{(s)}) = \text{Variational Lower Bound of } \log p(x^{(t)}, y^{(t)}|x^{(s)})$
Learning MSVED

- **Learning Continuous Latent Variables:**
  Reparameterization trick (Kingma et al., 2014):

  \[
  \epsilon \sim \mathcal{N}(0, 1), \quad z = \mu_\phi(x) + \sigma_\phi(x) \circ \epsilon
  \]

- **Learning Discrete Latent Variables:**
  Gumbel-Softmax (Maddison et al., 2017)

  \[
  \hat{y}_{ij} = \frac{\exp((\log(\pi_{ij}) + g_{ij})/\tau)}{\sum_{k=1}^{N_i} \exp((\log(\pi_{ik}) + g_{ik})/\tau)}
  \]

- **Training tricks** (Bowman et al. 2016):
  - KL-divergence Annealing
  - Input dropout in the decoder
Experimental Setup

**Task:** Morphology re-inflection

**Dataset:** SIGMORPHON 2016 task 3
- source word: communicated
- target word: communicates
- target labels: V;3;SG;PRS

**Language:** Turkish, Arabic, Maltese, Finnish, Spanish, German, Hungarian, Navajo, Georgian, Russian
Results and Analysis

Single-directional supervised model (SD-Sup);
Bidirectional supervised model (BD-Sup);
Semi-supervised model (Semi-Sup)

Competitive w/ standard attention

Helped by bi-directional training

Helped by semi-supervised training

Avg. Acc over 10 Languages

Models

Proposed MSVED
Baseline MED
Analysis on Tag Attention

case=NOM
mood=None
pos=ADJ
per=None
poss=None
num=PL
tense=None
aspect=None
voice=None
gend=FEM
def=DEF
Visualization of Latent Continuous Variables

- Clusters colored by actual lemma:

  ![Turkish Clusters](image)
  ![Maltese Clusters](image)

Pengcheng Yin, Chunting Zhou, Junxian He, Graham Neubig
(ACL 2018)
What About More Complicated Structure?

**Semantic Parsing:** Transducing natural language utterances (e.g., queries) into machine-executable formal meaning representations (e.g., logical form, source code)

**Domain-Specific Meaning Representations**

- *Show me flights from Pittsburgh to Washington*
  - `lambda $0 e (and (flight $0) (from $0 san_Francisco:ci) (to $0 washington:ci))`
  - lambda-calculus logical form

**General-Purpose Programming Languages**

- *Sort my_list in descending order*
  - `sorted(my_list, reverse=True)`
  - Python
Data Collection is Costly

Copy the content of file 'file.txt' to file 'file2.txt'
```
shutil.copy('file.txt','file2.txt')
```

Get a list of words `words` of a file 'myfile'
```
words = open('myfile').read().split()
```

Check if all elements in list `mylist` are the same
```
len(set(mylist)) == 1
```

Collecting parallel training data costs and

[Yin et al., 2018] 1700 USD for 3K Python code generation examples
[Berant et al., 2013] 3000 USD for 5.7K question-to-logical form examples
Semi-supervised Semantic Parsing

### Limited Amount of Labeled Data
- Sort my_list in descending order
  ```python
  sorted(my_list, reverse=True)
  ```
- Copy the content of file `file.txt` to file `file2.txt`
  ```python
  shutil.copy('file.txt', 'file2.txt')
  ```
- Check if all elements in list `mylist` are the same
  ```python
  len(set(mylist)) == 1
  ```

### Extra Unlabeled Utterances
- Get a list of words `words` of a file `myfile`
- Convert a list of integers into a single integer
- Format a datetime object `when` to extract date only
- Swap values in a tuple/list in list `mylist`
- BeautifulSoup search string `Elsie` inside tag `'a'`
- Convert string to lowercase
Meaning Representations as Tree-structured Latent Variables

Structured Latent Semantic Space

Prior

Inference Model

Reconstruction Model

Posterior inference corresponds to semantic parsing

Sort my_list in descending order
Semi-supervised Learning with StructVAE

Supervised Objective
\[ \sum \log q_{\phi}(\cdot | \cdot) \]
\( \cdot \) Labeled Data

Unsupervised Objective
\[ \sum \log p(\cdot) \]
\( \cdot \) Unlabeled Data

Structured Latent Semantic Space

Prior
\[ p(\cdot) \]

Inference Model
\[ q_{\phi}(\cdot | \cdot) \]

Reconstruction Model
\[ p_{\theta}(\cdot | \cdot) \]

Sort my_list in descending order

\[ p(\cdot) = \int p(\cdot | \cdot) p(\cdot) \]
StructVAE: VAEs with Tree-structured Latent Variables

Inference Model
$q_\phi(\mathcal{T} \mid \mathbf{y})$
Neural semantic parser

Reconstruction Model
$p_\psi(\mathbf{y} \mid \mathcal{T})$
Neural sequence-to-sequence model

Prior
$p(\mathcal{T})$
Neural Language Model

(use linearized trees as inputs)

Unsupervised Objective
$\sum \log p(\mathcal{T})$
Unlabeled Data

Variational approximation of the marginal likelihood

$$\log p(\mathbf{y}) \geq \sum \log p_\theta(\mathbf{y} \mid \mathcal{T}^\prime) - KL_Divergence[q_\phi(\mathcal{T} \mid \mathbf{y}) \parallel p(\mathcal{T})]$$

[Miao and Blunsom, 2016]
How does extra unlabeled data help learning?

\[ \nabla = \sum \frac{\partial \log q_\phi(\text{unlabeled} \mid \text{labeled})}{\partial \phi} \]

[Training Examples]
How does extra unlabeled data help learning?

Unsupervised Objective

$$\sum \log p(\text{Unlabeled Data})$$

The learning signal

$$\nabla \propto \sum_{\text{Sampled}} \times \frac{\partial q_{\phi}(\cdot\mid \cdot)}{\partial \phi}$$

Learning signal acts as the tuning weights of gradients received by different sampled latent meaning representations from the inference model.

Reference: [Miao and Blunsom, 2016]
How does extra unlabeled data help learning?

- Faithfully encode the semantics of the utterance -> high reconstruction score
- Are succinct and natural -> high prior probability

Sort my_list in descending order

```python
sorted(my_list, reverse=True)
```

Learning favors sampled latent meaning representations that both:

1. `sorted(my_list)`
2. `sorted(my_list, reverse=True)`
3. `sorted(my_list, descending=True)`
A transition-based parser that transduces natural language utterances into Abstract Syntax Trees

**Grammar Specification**

\[
\text{stmt} \mapsto \text{FunctionDef}(\text{identifier name, arguments args, stmt* body}) \\
| \text{Expr(\text{expr value})} \\
\text{expr} \mapsto \text{Call(\text{expr func, expr* args, keyword* keywords})} \\
| \text{Name(\text{identifier id})} \\
| \text{Str(\text{string id})}
\]

**Input Utterance**

Sort my_list in descending order

[Inference Model: a Transition-based Parser]

[Yin and Neubig, 2017; Rabinovich et al. 2017]
Datasets

**Django Python Code Generation Task**

Call the function `_generator`, join the result into a string, return the result

```python
return ''.join(_generator())
```

**ATIS Semantic Parsing Task**

Show me flights from San Francisco to Washington

```python
lambda $0 e
    (and (flight $0)
        (from $0 san_Francisco:ci)
        (to $0 washington:ci))
```
Research Questions

• **RQ1** Does StructVAE outperforms purely supervised semantic parsers with extra unlabeled data?

• **RQ2** Can we get some empirical evidence about why StructVAE works?
StructVAE v.s. Baselines

- Inference model as supervised parser
- Self Training (semi-supervised baseline)
- StructVAE

The gap is much more obvious when we use a mediocre parser 😊

---

all available training utterances as unlabeled data
Why does StructVAE work?

- For each unlabeled utterance, compute the learning signal for gold samples and other (imperfect) samples.

![Graph showing learning signals for gold and other samples with averages labeled as Avg.=-5.12 and Avg.=2.59.](image)
Case Studies

Join p and cmd into a file path, substitute it for f

- **f = os.path.join(p, cmd)**
  - Parser Score: -1.00
  - Prior: -24.33
  - Reconstruction Score: -2.00
  - Learning Signal: 9.14

- **p = path.join(p, cmd)**
  - Parser Score: -8.12
  - Prior: -27.89
  - Reconstruction Score: -20.96
  - Learning Signal: -9.47

Split string pks by ‘,’ , substitute the result for primary_keys

- **primary_keys = pks.split(‘,’)**
  - Parser Score: -2.38
  - Prior: -10.24
  - Reconstruction Score: -11.39
  - Learning Signal: 2.05

- **primary_keys = pks.split + ‘,’**
  - Parser Score: -1.83
  - Prior: -20.41
  - Reconstruction Score: -14.87
  - Learning Signal: -2.60
Unsupervised Learning of Syntactic Structure w/ Invertible Neural Projections

Junxian He, Graham Neubig, Taylor Berg-Kirkpatrick
(EMNLP 2018)
HMM for Part-of-Speech Induction

The cat sat
Gaussian HMM for POS Induction

\[ x_i \sim \mathcal{N}(\mu_{z_i}, \Sigma_{z_i}) \]

[Lin et al. 2015]
Latent Embeddings w/ Neural Projection

\[ z_i \sim \text{Markov Structure} \]

\[ e_i \sim \mathcal{N}(\mu_{z_i}, \Sigma_{z_i}) \]

Neural Projector

\[ x_i = f_{\phi}(e_i) \]

\[ x_i \sim \text{Point mass at } f_{\phi}(e_i) \]
Dependency Model with Valence

[The cat stopped a dog in Paris.]

[Klein and Manning 2004]
Dependency Model with Valence

[Klein and Manning 2004]
Dependency Parse Induction from POS
Grammar Induction from Raw Text
Grammar Induction from Raw Text

The cat stopped a dog in Paris
$z_i \sim \text{Syntax Model}$

$e_i \sim \mathcal{N}(\mu_{z_i}, \Sigma_{z_i})$

Neural Projector

$x_i = f_\phi(e_i)$

$x_i \sim \text{Point mass at } f_\phi(e_i)$

Latent Embeddings w/ Neural Projection

Markov prior

DMV prior
Learning and Inference

\[ p(x_i | z_i; \eta, \phi) \]
Learning and Inference

$p(x_i | z_i; \eta, \phi)$

Gaussian embedding parameters
Learning and Inference

\[ p(x_i | z_i; \eta, \phi) \]

Projection parameters
Learning and Inference

dim(x) = dim(e) and $f$ is invertible

$$p(x_i | z_i; \eta, \phi)$$

$$= p(f^{-1}_\phi(x_i) | z_i; \eta) \left| \det \frac{\partial f^{-1}}{\partial x_i} \right|$$
Learning and Inference

\[
\text{dim}(x) = \text{dim}(e) \text{ and } f \text{ is invertible}
\]

\[
p(x_i | z_i ; \eta, \phi) = p(f^{-1}_\phi(x_i) | z_i ; \eta) \left| \det \frac{\partial f^{-1}}{\partial x_i} \right|
\]

Determinant of Jacobian matrix
Learning and Inference

\[ \dim(x) = \dim(e) \text{ and } f \text{ is invertible} \]

\[ p(x_i | z_i; \eta, \phi) = p(f_{\phi}^{-1}(x_i) | z_i; \eta) \left| \det \frac{\partial f^{-1}}{\partial x_i} \right| \]

Gaussian distribution

Determinant of Jacobian matrix
Learning and Inference

Example of Markov prior

\[ \log p(\mathbf{x}) = \log p_{\text{GHMM}}(f^{-1}_\phi(\mathbf{x})) + \sum \log \left| \det \frac{\partial f^{-1}_\phi}{\partial \mathbf{x}_i} \right| \]

\(-\infty\) when \(f\) is not invertible

\[ p(f^{-1}_\phi(\mathbf{x}_i)|z_i; \eta) \left| \det \frac{\partial f^{-1}_\phi}{\partial \mathbf{x}_i} \right| \]
Why Invertible

Example of Markov prior

\[
\max \log p_{\text{GHMM}}(f_\phi(x))
\]

Information Loss
Learning with Inverse Projection

\[ e_i \xrightarrow{f} f^{-1}_\phi(x_i) \xrightarrow{\text{Inverse Projection}} x_i \]
Learning with Inverse Projection

\[ h_{i,l}^{(1)} = x_{i,l} \]

\[ h_{i,r}^{(1)} = x_{i,r} + g(x_{i,l}) \]

[Dinh et al. 2014]
Learning with Inverse Projection

\[ h_{i,l}^{(1)} = x_{i,l} \]

\[ h_{i,r}^{(1)} = x_{i,r} + g(x_{i,l}) \]

[Dinh et al. 2014]
Experiments

• Dataset: English Penn Treebank

• POS tagging

  Trained and tested on whole PTB

• Grammar induction

  Trained on sentences of length <= 10 in section 2-21

  Tested on sentences in section 23
Part-of-speech Induction

Outperform feature-based SOTA
Dependency Parse Induction

Directed % on len ≤ 10

- DMV: 49.6
- Gaussian DMV: 55.4
- Neural Projection: 60.2

Oracle POS 55.1

Directed % on all

- DMV: 35.8
- Gaussian DMV: 43.1
- Neural Projection: 47.9

Oracle POS 39.7
Original Embedding Space

- adjective
- adverb
- noun singular
- noun proper
- noun plural
- verb base
- verb gerund
- verb past tense
- verb past participle
- verb 3rd singular
- cardinal number
Projected Embedding Space w/ Markov Prior
Projected Embedding Space w/ DMV Prior

- adjective
- adverb
- noun singular
- noun proper
- noun plural
- verb base
- verb gerund
- verb past tense
- verb past participle
- verb 3rd singular
- cardinal number
Projected Embedding Space w/ DMV Prior

- adjective
- adverb
- noun singular
- noun proper
- noun plural
- verb base
- verb gerund
- verb past tense
- verb past participle
- verb 3rd singular
- cardinal number
Projected Embedding Space w/ DMV Prior

- adjective
- adverb
- noun singular
- noun proper
- noun plural
- verb base
- verb gerund
- verb past tense
- verb past participle
- verb 3rd singular
- cardinal number

- smokers
- advertisers
- parents
- riders
- performers
- issuers
- foes
- principals
- staff
- furriers
- aides
- professionals
- technicians
- specialists
- authorities

- aide
- resident
- owner
- politician
- associate
- examiner
- attorney
- actress
- commissioner
- singer
Conclusion
Learning with Latent Linguistic Structure

- How can we harness the power of neural networks?
  - NN-based learning on top of latent structured representations
- How can we learn on unlabeled data?
  - Structured variational auto-encoders for semi-supervised learning
  - Structured priors and invertible transformations for unsupervised learning