Self-Supervised Learning in NLP

Minlie Huang
Tsinghua University
aihuang@tsinghua.edu.cn
Why SSL?

- Yann Lecun:
  - Human and animal babies learning by observations

The Future is Self-Supervised

Yann LeCun
NYU - Courant Institute & Center for Data Science
Facebook AI Research
Learning Paradigms

- **Unsupervised learning**: $P(X)$
  - Autoencoder, VAE, Boltzmann Machine

- **Supervised learning**: $P(y|X)$
  - SVM, NB, DT, MLP, CNN, RNN

- **Semi-supervised learning**: labeled data + unlabeled data
  - Self-training
  - Self-supervised learning (sometimes)
Key Concepts

- Pretraining
- Self-training
- Self-supervised learning
Pretraining + Fine-tuning

- Pretrained on ImageNet (labeled), fine-tuned on image classification, detection, segmentation
- Pretrained on large text corpora (unlabeled), fine-tuned for NLU tasks: BERT, GPT, etc.
Self-training

- Labeled data
- Teacher models
- Noisy labeling
- Unlabeled data
- Noisy student models
- Pseudo labels
- Retraining loop
Self-training with Noisy Student

- Adding noise to the student (augmentation, dropout, stochastic depth)
- Using student models that are not smaller than the teacher

\[
\frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f_{\text{noised}}(x_i, \theta^t))
\]

\[
\tilde{y}_i = f(\tilde{x}_i, \theta^t_*) \quad \forall i = 1, \ldots, m
\]

\[
\frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f_{\text{noised}}(x_i, \theta^s)) + \frac{1}{m} \sum_{i=1}^{m} \ell(\tilde{y}_i, f_{\text{noised}}(\tilde{x}_i, \theta^s))
\]

Self-training with Noisy Student improves ImageNet classification.
Self-Supervised Learning (SSL)

- Data augmentation
- Data corruption
- Negative sampling

Labeled/unlabeled data → Trained models

Training

Supervision signals
SSL: predicting future from past

Sequence/stream data

Context (human written): In a shocking finding, scientists discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.
SSL: predicting future from past

Representation Learning with Contrastive Predictive Coding. Van den Oord et al. 2018
SSL: recovering from corruption/perturbation

Objectives:
- Maximize prob. \( P(y|X) \)
- Predict 0 degrees rotation (\( y=0 \))
- Maximize prob. \( P(y'|X') \)
- Predict 90 degrees rotation (\( y=1 \))
- Maximize prob. \( P(y''|X'') \)
- Predict 180 degrees rotation (\( y=2 \))
- Maximize prob. \( P(y'''|X''') \)
- Predict 270 degrees rotation (\( y=3 \))

(g) Cutout
(h) Gaussian noise
(i) Gaussian blur
SSL in natural language processing

Self-supervised Learning

Generative
- Auto-encoding (AE): **Masked language modeling** from BERT-style models
- Auto-regressive (AR): **Language modeling** from GPT-style models
- AE+AR: **Permutation language modeling** from XLNet

Contrastive
- Context-instance / Global-local contrast: **Text order prediction** from BERT-style models
- Context-context / Global-global contrast: **Sentence distance prediction** from CONPONO

Generative-contrastive / Adversarial: **Replaced token detection** from ELECTRA
SSL in NLP: language modeling

- Estimating $P(\text{present}|\text{context})$

$$\text{Max: } \log P(w_t|w_{t-2}w_{t-1}w_{t+1}w_{t+2})$$

$$\text{Max: } \sum_j \log P(w_{t+j}|w_t)$$

- CBOW

- SkipGram

$$\text{Max: } \sum_t \log P(w_t|w_1w_2 \ldots w_{t-1})$$

OpenAI GPT
SSL in NLP: masked language modeling

Recovering masked words in the input text

The top probability words corresponding to the masked word ‘perched’

BERT for Masked Language Model

Input:
[CLS] the cat perched on the mat [SEP]

Sentence S with masked word ‘perched’

Output:
[CLS] the cat perched on the mat [SEP]
SSL in NLP: masked language modeling

\[
\mathcal{L}(\text{football}) = \mathcal{L}_{\text{MLM}}(\text{football}) + \mathcal{L}_{\text{SBO}}(\text{football}) \\
= -\log P(\text{football} \mid \mathbf{x}_7) - \log P(\text{football} \mid \mathbf{x}_4, \mathbf{x}_9, \mathbf{p}_3)
\]

Span length distribution

SSL in NLP: masked language modeling

- Machine reading comprehension

- Masking strategies

![Bar chart showing performance metrics for different tasks and models.](chart.png)
SSL in NLP: permutation language modeling

- XLNet: Permutation Language Model

\[
\max_{E_{z \sim \mathcal{Z}_T}} \mathbb{E}_{z \sim \mathcal{Z}_T} \left[ \sum_{t=1}^{T} \log p(x_{z_t} \mid x_{z_{<t}}) \right]
\]

\[ \mathcal{L}_{BERT} = - \left( \log P(\text{deep}|I \text{ like } \text{[MASK]} \text{ [MASK] very much}) + \log P(\text{learning}|I \text{ like } \text{[MASK]} \text{ [MASK] very much}) \right) \]

\[ \mathcal{L}_{XLNet} = - \left( \log P(\text{learning}|I \text{ like } \text{[MASK]} \text{ learning very much}) + \log P(\text{deep}|I \text{ like } \text{[MASK]} \text{ very much}) - \right) \]

\[ - \left( \log P(\text{deep}|I \text{ like } \text{[MASK]} \text{ [MASK] very much}) + \log P(\text{learning}|I \text{ like } \text{deep } \text{[MASK] very much}) \right) \]

\[ \mathbf{Z}_1: P(1)*P(2 \mid 1)*P(3 \mid 1,2)*P(4 \mid 1,2,3) \]

\[ \mathbf{Z}_2: P(3)*P(1 \mid 3)*P(2 \mid 3,1)*P(4 \mid 3,1,2) \]

\[ \mathbf{Z}_3: P(2)*P(4 \mid 2)*P(3 \mid 2,4)*P(1 \mid 2,4,3) \]
SSL in NLP: permutation language modeling

- Permutation language modeling outperforms MLM significantly (XLNet vs. BERT)
- The model structure Transformer-XL performs better than vanilla Transformer (XLNet vs. -memory)
- NSP seems to degrade the performance of XLNet (XLNet vs. +next-sent pred)

<table>
<thead>
<tr>
<th>#</th>
<th>Model</th>
<th>RACE</th>
<th>SQuAD2.0</th>
<th>MNLI</th>
<th>SST-2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>F1</td>
<td>EM</td>
<td>m/mm</td>
</tr>
<tr>
<td>1</td>
<td>BERT-Base</td>
<td>64.3</td>
<td>76.30</td>
<td>73.66</td>
<td>84.34/84.65</td>
</tr>
<tr>
<td>2</td>
<td>DAE + Transformer-XL</td>
<td>65.03</td>
<td>79.56</td>
<td>76.80</td>
<td>84.88/84.45</td>
</tr>
<tr>
<td>3</td>
<td>XLNet-Base ($K = 7$)</td>
<td>66.05</td>
<td>81.33</td>
<td>78.46</td>
<td>85.84/85.43</td>
</tr>
<tr>
<td>4</td>
<td>XLNet-Base ($K = 6$)</td>
<td>66.66</td>
<td>80.98</td>
<td>78.18</td>
<td>85.63/85.12</td>
</tr>
<tr>
<td>5</td>
<td>- memory</td>
<td>65.55</td>
<td>80.15</td>
<td>77.27</td>
<td>85.32/85.05</td>
</tr>
<tr>
<td>6</td>
<td>- span-based pred</td>
<td>65.95</td>
<td>80.61</td>
<td>77.91</td>
<td>85.49/85.02</td>
</tr>
<tr>
<td>7</td>
<td>- bidirectional data</td>
<td>66.34</td>
<td>80.65</td>
<td>77.87</td>
<td>85.31/84.99</td>
</tr>
<tr>
<td>8</td>
<td>+ next-sent pred</td>
<td><strong>66.76</strong></td>
<td><strong>79.83</strong></td>
<td><strong>76.94</strong></td>
<td><strong>85.32/85.09</strong></td>
</tr>
</tbody>
</table>
SSL in NLP: text order prediction

- **Bert: Next Sentence Prediction**
- **Negative sampling**

---

**Sentence 1**
I am going outside.

**Sentence 2**
You know nothing John snow.

**Next Sentence?**

- **YES**
  - I will be back after 6.
- **NO**
  - I am going outside.
SSL in NLP: text order prediction

Does next sentence prediction (NSP) work well?

<table>
<thead>
<tr>
<th>Model</th>
<th>SQuAD 1.1/2.0</th>
<th>MNLI-m</th>
<th>SST-2</th>
<th>RACE</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Our reimplementation (with NSP loss):</em></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SEGMENT-PAIR</td>
<td>90.4/78.7</td>
<td>84.0</td>
<td>92.9</td>
<td>64.2</td>
</tr>
<tr>
<td>SENTENCE-PAIR</td>
<td>88.7/76.2</td>
<td>82.9</td>
<td>92.1</td>
<td>63.0</td>
</tr>
<tr>
<td><em>Our reimplementation (without NSP loss):</em></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FULL-SENTENCES</td>
<td>90.4/79.1</td>
<td>84.7</td>
<td>92.5</td>
<td>64.8</td>
</tr>
<tr>
<td>DOC-SENTENCES</td>
<td>90.6/79.7</td>
<td>84.7</td>
<td>92.7</td>
<td>65.6</td>
</tr>
</tbody>
</table>

◆ This 2-class classification task may be too easy for BERT to learn.
◆ The input format of two sentence segments may not be consistent with downstream tasks.

SSL in NLP: text order prediction

- **StructBERT**: word shuffle in subsequence

---

SSL in NLP: text order prediction

StructBERT: sentence-order type prediction
SSL in NLP: text order prediction

- Fine-grained text order prediction tasks at the word level and the sentence level outperform vanilla NSP in BERT.

<table>
<thead>
<tr>
<th>Task</th>
<th>CoLA (Acc)</th>
<th>SST-2 (Acc)</th>
<th>MNLI (Acc)</th>
<th>SNLI (Acc)</th>
<th>QQP (Acc)</th>
<th>SQuAD (F1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>StructBERTBase</td>
<td>85.8</td>
<td>92.9</td>
<td>85.4</td>
<td>91.5</td>
<td>91.1</td>
<td>90.6</td>
</tr>
<tr>
<td>-word structure</td>
<td>81.7</td>
<td>92.7</td>
<td>85.2</td>
<td>91.6</td>
<td>90.7</td>
<td>90.3</td>
</tr>
<tr>
<td>-sentence structure</td>
<td>84.9</td>
<td>92.9</td>
<td>84.1</td>
<td>91.1</td>
<td>90.5</td>
<td>89.1</td>
</tr>
<tr>
<td>BERTBase</td>
<td>80.9</td>
<td>92.7</td>
<td>84.1</td>
<td>91.3</td>
<td>90.4</td>
<td>88.5</td>
</tr>
</tbody>
</table>
SSL in NLP: sentence distance prediction

- **CONPONO**: Distance Prediction between Sentences

\[ t_{i+k} = g_\theta(s_i, s_{i+k}), c_i = g_\theta(s_i) \]

\[ \mathcal{L}_k = -\mathbb{E}_S[\log \frac{\exp(t_{i+k}^T W_k c_i)}{\sum_{s_j \in S} \exp(t_j^T W_k c_i)}] \]

**Objective:**
select candidate in \( S \) which has \( k \)-distance to \( s_i \)

**Negative samples:**
1) in the same document, but distance is not \( k \) to \( s_i \)
2) randomly sampled from other documents
**SSL in NLP: sentence distance prediction**

- CONPONO performs better than BERT-style models in the discourse-level representation tasks.

<table>
<thead>
<tr>
<th>Model</th>
<th>SP</th>
<th>BSO</th>
<th>DC</th>
<th>SSP</th>
<th>PDTB-E</th>
<th>PDTB-I</th>
<th>RST-DT</th>
<th>avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT-Base</td>
<td>53.1</td>
<td>68.5</td>
<td>58.9</td>
<td>80.3</td>
<td>41.9</td>
<td>42.4</td>
<td>58.8</td>
<td>57.7</td>
</tr>
<tr>
<td>BERT-Large</td>
<td>53.8</td>
<td>69.3</td>
<td>59.6</td>
<td>80.4</td>
<td>44.3</td>
<td>43.6</td>
<td>59.1</td>
<td>58.6</td>
</tr>
<tr>
<td>RoBERTa-Base</td>
<td>38.7</td>
<td>58.7</td>
<td>58.4</td>
<td>79.7</td>
<td>39.4</td>
<td>40.6</td>
<td>44.1</td>
<td>51.4</td>
</tr>
<tr>
<td>BERT-Base BSO</td>
<td>53.7</td>
<td>72.0</td>
<td>71.9</td>
<td>80.0</td>
<td>42.7</td>
<td>40.5</td>
<td>63.8</td>
<td>60.6</td>
</tr>
<tr>
<td>CONPONO <em>isolated</em></td>
<td>50.2</td>
<td>57.9</td>
<td>63.2</td>
<td>79.9</td>
<td>35.8</td>
<td>39.6</td>
<td>48.7</td>
<td>53.6</td>
</tr>
<tr>
<td>CONPONO <em>uni-encoder</em></td>
<td>59.9</td>
<td>74.6</td>
<td>72.0</td>
<td>79.6</td>
<td>40.0</td>
<td>43.9</td>
<td>61.9</td>
<td>61.7</td>
</tr>
<tr>
<td>CONPONO (k=2)</td>
<td><strong>60.7</strong></td>
<td><strong>76.8</strong></td>
<td><strong>72.9</strong></td>
<td><strong>80.4</strong></td>
<td><strong>42.9</strong></td>
<td><strong>44.9</strong></td>
<td>63.1</td>
<td><strong>63.0</strong></td>
</tr>
<tr>
<td>CONPONO std.</td>
<td>±.3</td>
<td>±.1</td>
<td>±.3</td>
<td>±.1</td>
<td>±.7</td>
<td>±.6</td>
<td>±.2</td>
<td>-</td>
</tr>
</tbody>
</table>

- Different settings of k (window size) may work in different tasks.
  - k>1 seems key to downstream tasks, because there is more variation farther from the anchor sentence.
  - Larger distances (k>2) from the anchor sentence lead to more ambiguity.
SSL in NLP: replaced token detection

- ELECTRA: Replaced Token Detection
  \[ \mathcal{L} = \mathcal{L}_{MLM}(x, \theta_G) + \lambda \mathcal{L}_{Disc}(x, \theta_D) \]

Kevin Clark, Minh-Thang Luong, Quoc V. Le, Christopher D. Manning: ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. ICLR 2020.
SSL in NLP: replaced token detection

- Replaced token detection consistently outperforms language modeling (GPT) and masked language model (BERT, RoBERTa) given the same compute budget.
SSL in NLP: other tasks

- Dialog modeling (Wu et al. ACL2019)
- Sequence-to-sequence generation (He et al. ICLR 2020)
- Machine reading comprehension (Niu et al. ACL 2020; Klein and Nabi ACL 2020)
- Text classification (5+ papers)
Evidence finding in MRC

Q: Did a little boy write the note?
D: ...This note is from a little girl. She wants to be your friend. If you want to be her friend, ...
A: No

Q: Is she carrying something?
D: ...On the step, I find the elderly Chinese lady, small and slight, holding the hand of a little boy. In her other hand, she holds a paper carrier bag. ...
A: Yes

Unreferenced evaluation of NLG

Leading Context
Jack was at the bar.

Reference By Human
He noticed a phone on the floor. He was going to take it to lost and found. But it started ringing on the way. Jack answered it and returned it to the owner’s friends.

Sample 1 (Reasonable, B=0.29, M=0.49, U=1.00)
On the way out he noticed a phone on the floor. He asked around if anybody owned it. Eventually he gave it to the bartender. They put it into their lost and found box.

Sample 2 (Reasonable, B=0.14, M=0.27, U=1.00)
He had a drinking problem. He kept having more beers. After a while he passed out. When he woke up, he was surprised to find that he lost over a hundred dollars.

Sample 3 (Unreasonable, B=0.20, M=0.35, U=0.00)
He was going to get drunk and get drunk. The bartender told him it was already time to leave. Jack started drinking. Jack wound up returning but cops came on the way home.

Summary

- Self-Supervised Learning (SSL) is learning dependencies:
  - Pixel-level, patch-level, word-level, sentence-level, discourse-level, etc.
  - Vector-level: making learned representations more predictive
  - Task-level: encoding task-agnostic information vs. task-specific information

- For NLP:
  - Data augmentation is hard (label-preserving)
  - (Strong) Negative samples are hard to collect
  - Data perturbation seems to be very effective in many tasks
Thanks for your attention

- Recruiting post-docs, PhDs, & interns

- Minlie Huang, Tsinghua University
- aihuang@tsinghua.edu.cn
- http://coai.cs.tsinghua.edu.cn/hml