Building Natural Language Processing Applications with EasyNLP

Chengyu Wang, Minghui Qiu, Jun Huang
Alibaba Group, Hangzhou, China
Main Contents

✓ Knowledge-enhanced Pre-training

✓ Deploying Large Pre-trained Models
  • Prompt-based Few-shot Learning
  • Knowledge Distillation for Large Pre-trained Models

✓ Multi-modal Pre-trained Models

✓ Overview of EasyNLP
**Development and Challenges for Pre-trained Models**

Larger pre-trained models often lead to better performance.

<table>
<thead>
<tr>
<th>Rank</th>
<th>Name</th>
<th>Model</th>
<th>URL</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Liam Fedus</td>
<td>SS-MoE</td>
<td></td>
<td>91.0</td>
</tr>
<tr>
<td>2</td>
<td>Microsoft Alexander v-team</td>
<td>Turing NLR v5</td>
<td></td>
<td>90.9</td>
</tr>
<tr>
<td>3</td>
<td>ERNIE Team - Baidu</td>
<td>ERNIE 3.0</td>
<td></td>
<td>90.6</td>
</tr>
<tr>
<td>4</td>
<td>Zhirui Wang</td>
<td>T5 + UDG, Single Model (Google Brain)</td>
<td></td>
<td>90.4</td>
</tr>
<tr>
<td>5</td>
<td>DeBERTa Team - Microsoft</td>
<td>DeBERTa / TuringNLRv4</td>
<td></td>
<td>90.3</td>
</tr>
</tbody>
</table>

Yet, it is not easy to apply large pre-trained models to real-world, industrial applications.

a. Large models are black boxes, which are prone to anti-common sense errors. The prediction performance in specific domains is also poor.

b. The low inference speed of large models make them hard to be deployed online.

c. Large models are easy to overfit, and are difficult to train with little training data.

Big Model & Small Labeled Data = OVERFITTING
Main Features of DKPLM

• DKPLM only uses knowledge graphs in pre-training, which is easy to tune and deploy during fine-tuning and inference.

• It effectively protects the knowledge graph data and avoids leakage for cloud service.

• The structure of DKPLM is compatible with BERT and can be directly used by the open-source community.

DKPLM for Knowledge-enhanced Pre-training

Framework of DKPLM

Key Techniques

- **Knowledge injection for long-tail entities**
  - Avoiding learning too much redundant knowledge

- **No additional parameters**
  - Making the backbone fully aligned with BERT

- **Relation-based knowledge decoding**
  - Decoding the injected triple knowledge as one of the pre-training tasks

\[ \mathcal{L}_{total} = \lambda_1 \mathcal{L}_{MLM} + (1 - \lambda_1) \mathcal{L}_{De} \]
Evaluation Results

Our medical DKPLM

<table>
<thead>
<tr>
<th></th>
<th>DKPLM</th>
<th>BERT</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMedQANER (NER)</td>
<td>84.79</td>
<td>81.43</td>
</tr>
<tr>
<td>CHIP20 (RE)</td>
<td>77.13</td>
<td>73.05</td>
</tr>
<tr>
<td>CMedMRC (MRC)</td>
<td><strong>EM=67.18</strong></td>
<td><strong>F1=85.33</strong></td>
</tr>
</tbody>
</table>

Our financial DKPLM

<table>
<thead>
<tr>
<th></th>
<th>DKPLM</th>
<th>BERT</th>
</tr>
</thead>
<tbody>
<tr>
<td>FinNER (NER)</td>
<td>87.81</td>
<td>77.56</td>
</tr>
<tr>
<td>FinSent (Sentence Classification)</td>
<td>85.75</td>
<td>83.68</td>
</tr>
<tr>
<td>FinMatch (Sentence Matching)</td>
<td>92.81</td>
<td>91.99</td>
</tr>
<tr>
<td>FinNegReview (Sentence Classification)</td>
<td>93.81</td>
<td>92.50</td>
</tr>
</tbody>
</table>

Hugging Face Models

- [alibaba-pai/pai-dkplm-medical-base-zh](#)
- [alibaba-pai/pai-dkplm-financial-base-zh](#)

**Hosted inference API**

- Fill-Mask

Mask token: [MASK]

Compute

Computation time on cpu: 0.077 s

<table>
<thead>
<tr>
<th>Text</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>药</td>
<td>0.938</td>
</tr>
<tr>
<td></td>
<td>0.012</td>
</tr>
<tr>
<td>的</td>
<td>0.009</td>
</tr>
<tr>
<td></td>
<td>0.007</td>
</tr>
<tr>
<td></td>
<td>0.006</td>
</tr>
</tbody>
</table>

<JSON Output>
Main Contents

- Knowledge-enhanced Pre-training
- **Deploying Large Pre-trained Models**
  - Prompt-based Few-shot Learning
  - Knowledge Distillation for Large Pre-trained Models
- Multi-modal Pre-trained Models
- Overview of EasyNLP
Why Prompt-based Few-shot Learning?

✓ **Fine-tuning**: requires sufficient labeled training data, hard to obtain in some real-world applications

✓ **Prompt-based Fine-tuning**: a new paradigm for few-shot learning

Current Problems of Prompt-based Fine-tuning

• Manually designed prompts and verbalizers

• Unstable results with different prompts

Contrastive Prompt Tuning (CP-Turing)

- **Improvement of Prompts**
  - Using continuous prompt embeddings in input

- **Improvement of Verbalizers**
  - Replacing verbalizer mapping with Contrastive Learning

- **Loss function of CP-Tuning**

\[
\mathcal{L}(i) = \mathcal{L}_{PCCL}(i) + \lambda \mathcal{L}_{MLM}(i)
\]

## Evaluation Results of CP-Tuning

<table>
<thead>
<tr>
<th>Backbone</th>
<th>Method</th>
<th>Sentiment Analysis</th>
<th>Sentence Matching</th>
<th>NLI</th>
<th>Subjectivity</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>SST-2</td>
<td>MR</td>
<td>CR</td>
<td>MRPC</td>
<td>QQP</td>
</tr>
<tr>
<td><strong>RoBERTa</strong></td>
<td>Standard Fine-tuning</td>
<td>78.62</td>
<td>76.17</td>
<td>72.48</td>
<td>64.40</td>
<td>63.01</td>
</tr>
<tr>
<td></td>
<td>PET</td>
<td>92.06</td>
<td>87.13</td>
<td>87.13</td>
<td>66.23</td>
<td>70.34</td>
</tr>
<tr>
<td></td>
<td>LM-BFF (Auto T)</td>
<td>90.60</td>
<td>87.57</td>
<td>90.76</td>
<td>66.72</td>
<td>65.25</td>
</tr>
<tr>
<td></td>
<td>LM-BFF (Auto L)</td>
<td>90.55</td>
<td>85.51</td>
<td>91.11</td>
<td>67.75</td>
<td>70.92</td>
</tr>
<tr>
<td></td>
<td>LM-BFF (Auto T+L)</td>
<td>91.42</td>
<td>86.84</td>
<td>90.40</td>
<td>66.81</td>
<td>61.61</td>
</tr>
<tr>
<td></td>
<td>P-tuning</td>
<td>91.42</td>
<td>87.41</td>
<td>90.90</td>
<td>71.23</td>
<td>66.77</td>
</tr>
<tr>
<td></td>
<td>WARP</td>
<td>58.80</td>
<td>55.25</td>
<td>55.55</td>
<td>65.74</td>
<td>65.80</td>
</tr>
<tr>
<td><strong>CP-Tuning</strong></td>
<td></td>
<td><strong>93.35</strong></td>
<td><strong>89.43</strong></td>
<td><strong>91.57</strong></td>
<td><strong>72.60</strong></td>
<td><strong>73.56</strong></td>
</tr>
<tr>
<td><strong>ALBERT</strong></td>
<td>Standard Fine-tuning</td>
<td>63.98</td>
<td>64.90</td>
<td>71.50</td>
<td>56.78</td>
<td>59.32</td>
</tr>
<tr>
<td></td>
<td>PET</td>
<td>87.11</td>
<td>81.47</td>
<td>88.32</td>
<td>57.21</td>
<td>66.16</td>
</tr>
<tr>
<td></td>
<td>LM-BFF (Auto T)</td>
<td>82.60</td>
<td>83.23</td>
<td>88.48</td>
<td>64.04</td>
<td>60.28</td>
</tr>
<tr>
<td></td>
<td>LM-BFF (Auto L)</td>
<td>86.83</td>
<td>83.02</td>
<td>89.12</td>
<td>63.43</td>
<td>59.49</td>
</tr>
<tr>
<td></td>
<td>LM-BFF (Auto T+L)</td>
<td>84.40</td>
<td>82.75</td>
<td>89.52</td>
<td>62.48</td>
<td>56.48</td>
</tr>
<tr>
<td></td>
<td>P-tuning</td>
<td>85.42</td>
<td>84.32</td>
<td>82.35</td>
<td>58.76</td>
<td>57.46</td>
</tr>
<tr>
<td></td>
<td>WARP</td>
<td>66.63</td>
<td>65.59</td>
<td>72.34</td>
<td>63.48</td>
<td>58.20</td>
</tr>
<tr>
<td><strong>CP-Tuning</strong></td>
<td></td>
<td><strong>89.63</strong></td>
<td><strong>84.68</strong></td>
<td><strong>90.39</strong></td>
<td><strong>63.52</strong></td>
<td><strong>71.05</strong></td>
</tr>
</tbody>
</table>
Meta Knowledge Distillation (Meta-KD)

- Goal: Improving the effectiveness of knowledge distillation across domains

### Analogy Analysis

Students who master common knowledge in math and physics can have a better grasp of specific problems in math and physics.

### All-purpose

Science Teacher -> Meta Leaner

---

Model Architecture of Meta-KD

- **Core idea:** **Selectively transferring** cross-domain, transferable knowledge from Meta Teacher to Student

![Diagram of Meta-KD Model](image)

- Domain 1
- Domain 2
- Core idea: Selectively transferring cross-domain, transferable knowledge from Meta Teacher to Student
# Experimental Results of Meta-KD

- Compared to original BERT, the small model obtained by Meta-KD reduces accuracy by 1.5% only. (#Para. 109M->14.5M)

<table>
<thead>
<tr>
<th>Methods</th>
<th>Fiction</th>
<th>Government</th>
<th>Slate</th>
<th>Telephone</th>
<th>Travel</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT-single</td>
<td>82.2</td>
<td>84.2</td>
<td>76.7</td>
<td>82.4</td>
<td>84.2</td>
<td>81.9</td>
</tr>
<tr>
<td>BERT-mix</td>
<td>84.8</td>
<td>87.2</td>
<td>80.5</td>
<td>83.8</td>
<td>85.5</td>
<td>84.4</td>
</tr>
<tr>
<td>BERT-mlt</td>
<td>83.7</td>
<td>87.1</td>
<td>80.6</td>
<td>83.9</td>
<td>85.8</td>
<td>84.2</td>
</tr>
<tr>
<td>Meta-teacher</td>
<td>85.1</td>
<td>86.5</td>
<td>81.0</td>
<td>83.9</td>
<td>85.5</td>
<td>84.4</td>
</tr>
<tr>
<td>BERT-single → TinyBERT</td>
<td>78.8</td>
<td>83.2</td>
<td>73.6</td>
<td>78.8</td>
<td>81.9</td>
<td>79.3</td>
</tr>
<tr>
<td>BERT-mix → TinyBERT</td>
<td>79.6</td>
<td>83.3</td>
<td>74.8</td>
<td>79.0</td>
<td>81.5</td>
<td>79.6</td>
</tr>
<tr>
<td>BERT-mlt → TinyBERT</td>
<td>79.7</td>
<td>83.1</td>
<td>74.2</td>
<td>79.3</td>
<td>82.0</td>
<td>79.7</td>
</tr>
<tr>
<td>Multi-teachers → MTN-KD</td>
<td>77.4</td>
<td>81.1</td>
<td>72.2</td>
<td>77.2</td>
<td>78.0</td>
<td>77.2</td>
</tr>
<tr>
<td>Meta-teacher → TinyBERT</td>
<td>80.3</td>
<td>83.0</td>
<td>75.1</td>
<td>80.2</td>
<td>81.6</td>
<td>80.0</td>
</tr>
<tr>
<td>Meta-teacher → Meta-distillation (ours)</td>
<td><strong>80.5</strong></td>
<td><strong>83.7</strong></td>
<td><strong>75.0</strong></td>
<td><strong>80.5</strong></td>
<td><strong>82.1</strong></td>
<td><strong>80.4</strong></td>
</tr>
</tbody>
</table>
Main Contents

✓ Knowledge-enhanced Pre-training

✓ Deploying Large Pre-trained Models
  • Prompt-based Few-shot Learning
  • Knowledge Distillation for Large Pre-trained Models

✓ Multi-modal Pre-trained Models

✓ Overview of EasyNLP
CLIP-style Models for Text-image Retrieval

✅ EasyNLP supports Chinese CLIP-style Models

Jiaxi Gu, Xiaojun Meng, Guansong Lu, Lu Hou, Minzhe Niu, Hang Xu, Xiaodan Liang, Wei Zhang, Xin Jiang, Chunjing Xu. Wukong: 100 Million Large-scale Chinese Cross-modal Pre-training Dataset and A Foundation Framework. arXiv
**CLIP-style Models for Text-image Retrieval**

- EasyNLP supports SOTA English CLIP-style Models for fashion

### Evaluation Results on Fashion-Gen

<table>
<thead>
<tr>
<th>Model</th>
<th>Rank@1</th>
<th>Rank@5</th>
<th>Rank@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>FashionBERT</td>
<td>26.75</td>
<td>46.48</td>
<td>55.74</td>
</tr>
<tr>
<td>KaleidoBERT</td>
<td>33.9</td>
<td>60.5</td>
<td>68.6</td>
</tr>
<tr>
<td>CLIP</td>
<td>36.8</td>
<td>58.9</td>
<td>67.6</td>
</tr>
<tr>
<td>CommerceMM</td>
<td>39.6</td>
<td><strong>61.5</strong></td>
<td>72.7</td>
</tr>
<tr>
<td>EL-CLIP</td>
<td>28.4</td>
<td>57.1</td>
<td>69.4</td>
</tr>
<tr>
<td>pai-clip-commercial-base-en</td>
<td>39.5</td>
<td><strong>61.5</strong></td>
<td>70.0</td>
</tr>
<tr>
<td>pai-clip-commercial-large-en</td>
<td><strong>54.6</strong></td>
<td><strong>75.1</strong></td>
<td><strong>81.4</strong></td>
</tr>
</tbody>
</table>

---

**Image Encoding**

- **Text Encoder**
  - Input: `T1, T2, ..., TN`

- **Image Encoder**
  - Input: `I1, I2, ..., IN`

- Example: Pepper the aussie pup

---

**Diagram**

- Visual representation of text and image encoders with respective inputs and outputs.
<table>
<thead>
<tr>
<th>Text</th>
<th>Top-1 result of our CLIP</th>
<th>Top-1 result of OpenCLIP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Canvas slip-on sandals in black. Fringed edges throughout. Open</td>
<td><img src="image1" alt="Sandals" /></td>
<td><img src="image2" alt="Sandals" /></td>
</tr>
<tr>
<td>round toe. Leather lining in beige. Round block heel. Tonal leather</td>
<td><img src="image3" alt="Shoe" /></td>
<td><img src="image4" alt="Shoe" /></td>
</tr>
<tr>
<td>sole. Tonal stitching. Approx. 3” heel.</td>
<td><img src="image5" alt="Shoe" /></td>
<td><img src="image6" alt="Shoe" /></td>
</tr>
<tr>
<td>Long sleeve cotton-blend jersey henley in heather 'medium' grey.</td>
<td><img src="image7" alt="Shirt" /></td>
<td><img src="image8" alt="Shirt" /></td>
</tr>
<tr>
<td>Crewneck collar. Three-button placket. Rib knit cuffs. Tonal stitching.</td>
<td><img src="image9" alt="Shirt" /></td>
<td><img src="image10" alt="Shirt" /></td>
</tr>
<tr>
<td>Jersey skirt in black. Elasticized waistband. Shirring at front</td>
<td><img src="image11" alt="Skirt" /></td>
<td><img src="image12" alt="Skirt" /></td>
</tr>
<tr>
<td>waist. Drop-tail hem. Fully lined. Tonal stitching.</td>
<td><img src="image13" alt="Skirt" /></td>
<td><img src="image14" alt="Skirt" /></td>
</tr>
</tbody>
</table>
DALLE-style Text-to-image Generation

EasyNLP Text-to-image Generation Models

- Specific for the Chinese language
- VQGAN for image generation
- Transformer for converting texts to image tokens
- Moderate model size (<100M parameters)

一只俏皮的狗正跑过草地
A playful dog is running across the grass

一片水域的景色以日落为背景
A view of water with sunset in the background
风阁水帘今在眼，
且来先看早梅红

见说春风偏有贺，
露花千朵照庭闱

Red plum blossom

Thousands of flowers in spring
Chinese Painting Generation

Floating mist in quiet night

Seeing mountain view in a sad mood
Main Contents

✓ Knowledge-enhanced Pre-training
✓ Deploying Large Pre-trained Models
  • Prompt-based Few-shot Learning
  • Knowledge Distillation for Large Pre-trained Models
✓ Multi-modal Pre-trained Models
✓ Overview of EasyNLP
Overview of the EasyNLP Toolkit

✓ History
  • In 2021, we started building the EasyNLP toolkit.
  • EasyNLP has supported over 10 BUs in Alibaba Group since 2021.
  • Starting from May 2022, EasyNLP goes open-sourced in GitHub.

✓ Features of EasyNLP
  • Easy to use and highly customizable
  • Compatible with open-source libraries
  • Knowledge-enhanced pre-training
  • Deploying large pre-trained models (knowledge distillation and few-shot learning)
  • Multi-modal pre-trained models

Chengyu Wang, Minghui Qiu, Taolin Zhang, Tingting Liu, Lei Li, Jianing Wang, Ming Wang, Jun Huang, Wei Lin. EasyNLP: A Comprehensive and Easy-to-use Toolkit for Natural Language Processing. EMNLP 2022
# EasyNLP Framework

## Environments
- **Local Environment**
  - Text Tagging
  - Spam Detection

## Solutions
- **Text Tagging**
- **Spam Detection**
- **Text Retrieval**
- **User-defined Solutions**

## Basic Modules
- **AppZoo**
  - Text Classification
  - Text Matching
  - Sequence Labeling
  - Language Modeling
  - Feature Vectorization
  - Text Generation
  - Text-Image Matching
  - Text-to-Image Generation
  - Other Apps

## Model Zoo
- GPT
- mT5
- CLIP
- ARTIST
- BERT
- ALBERT
- DKPLM
- Other Models

## IO Modules
- **Table Reader**
- **ODPS Reader**
- **CSV Reader**
- **WebDataset**
- **Datasets**
- **Layers**
- **Losses**
- **Optimizers**

## Backend
- **PyTorch**

## Datasets
- **Model Zoo**
  - Text-Image Matching
  - Text-to-Image Generation
  - Other Apps

## Knowledge Distillation
- **Vanilla KD**
- **MetaKD**
- **Data Augmentation**

## Few-shot Learning
- **PET**
- **P-Tuning**
- **CP-Tuning**
API Examples – AppZoo Mode

✓ Black-box commands for model training, evaluation and prediction

easyNlp

  --mode=train
  --worker_gpu=1
  --tables=train.tsv,dev.tsv
  --input_schema=sent:str:1,label:str:1
  --first_sequence=sent
  --label_name=label
  --label Enumerate_Values=0,1
  --checkpoint_dir=./classification_model
  --epoch_num=1
  --sequence_length=128
  --app_name=text_classify
  --user_defined_parameters='pretrain_model_name_or_path=bert-small-uncased'

Command for text classification
API Examples – Python Mode

✓ Python APIs for model training, evaluation and prediction

```python
from easynlp.dataset import load_dataset, GeneralDataset

# load dataset
dataset = load_dataset('clue', 'tnews')['train']

# parse data into classification model input
coded = GeneralDataset(dataset, 'chinese-bert-base')

# load model
model = SequenceClassification('chinese-bert-base')
trainer = Trainer(model, coded)

# start to train
trainer.train()
```

Load dataset

Data Pre-processing

Load Model

Begin Training
Future Roadmap

✓ Knowledge Pre-training
  • Releasing more knowledge pre-trained models to improve the models’ understanding abilities of knowledge

✓ Multi-modal Pre-training
  • Releasing better multi-modal pre-trained models for various tasks

✓ Pre-trained Models for Closed-domains
  • Supporting various NLP and multi-modal tasks for closed-domains domains (e.g., medicine, finance)

✓ Better support for cloud products
  • Providing better support on the cloud
Key References

- Chengyu Wang, Minghui Qiu, Taolin Zhang, Tingting Liu, Lei Li, Jianing Wang, Ming Wang, Jun Huang, Wei Lin. EasyNLP: A Comprehensive and Easy-to-use Toolkit for Natural Language Processing. EMNLP 2022


- Jiaxi Gu, Xiaojun Meng, Guansong Lu, Lu Hou, Minzhe Niu, Hang Xu, Xiaodan Liang, Wei Zhang, Xin Jiang, Chunjing Xu. Wukong: 100 Million Large-scale Chinese Cross-modal Pre-training Dataset and A Foundation Framework. arXiv

Following Our GitHub Project

EasyNLP

https://github.com/alibaba/EasyNLP
THANKS

-------- Q&A Section --------