TRANSFER LEARNING FOR SEQUENCE TAGGING WITH HIERARCHICAL RECURRENT NETWORKS

Zhilin Yang, Ruslan Salakhutdinov & William W. Cohen
ICLR 2017

YUAN SHUAI
SMILE LAB
- Background knowledge
- Basic model
- Transfer learning based on the model
Background knowledge

- What is transfer learning?
  - 转移学习基本概念
    - 域(Domain)：由数据特征和特征分布组成，是学习的主体
      - Source domain (源域)：已有知识的域
      - Target domain (目标域)：要进行学习的域
    - 任务(Task)：由目标函数和学习结果组成，是学习的结果
  - 形式化
    - 条件：给定一个源域 $D_s$ 和源域上的学习任务 $T_s$，目标域 $D_t$ 和目标域上的学习任务 $T_T$
    - 目标：利用 $D_s$ 和 $T_s$ 学习在目标域上的预测函数 $f(\cdot)$。
    - 限制条件：$D_s \neq D_t$ 或 $T_s \neq T_T$
Word Embedding

- Mapping words or phrases from the vocabulary to vectors of real numbers.

LSTM & GRU

Forget gate
Input gate
Output gate

Reset gate
Update gate

http://colah.github.io/posts/2015-08-Understanding-LSTMs/
In this topic, we only focus on Linear Chain CRFs

A framework for building probabilistic models to segment and label sequence data

Sequence tagging: POS tagging, text chunking, NER (named entity recognition)

Conditional random fields: Probabilistic models for segmenting and labeling sequence data

http://blog.echen.me/2012/01/03/introduction-to-conditional-random-fields/
**Definition.** Let $G = (V, E)$ be a graph such that $Y = (Y_v)_{v \in V}$, so that $Y$ is indexed by the vertices of $G$. Then $(X, Y)$ is a conditional random field in case, when conditioned on $X$, the random variables $Y_v$ obey the Markov property with respect to the graph: $p(Y_v | X, Y_w, w \neq v) = p(Y_v | X, Y_w, w \sim v)$, where $w \sim v$ means that $w$ and $v$ are neighbors in $G$.

**Hammersley & Clifford Theorem:**

$$P(Y) = \frac{1}{Z} \prod_C \Psi_C(Y_C)$$

$$Z = \sum_Y \prod_C \Psi_C(Y_C)$$

**Linear Chain CRF:**

$$P(I | O) = \frac{1}{Z(O)} e^{\sum_i \sum_{k} \lambda_k f_k(O, I_{i-1}, I_i, i)} = \frac{1}{Z(O)} e^{[\sum_i \sum_j \lambda_j t_j(O, I_{i-1}, I_i, i) + \sum_i \sum_l \mu_l s_l(O, I_i, i) + \sum_i \sum_j \lambda_j t_j(O, I_{i-1}, I_i, i)]}$$
Part-of-Speech Tagging:

In POS tagging, the goal is to label a sentence (a sequence of words or tokens) with tags like ADJECTIVE, NOUN, PREPOSITION, VERB, ADVERB, ARTICLE.

For example

“Bob drank coffee at Starbucks” -> “Bob (n.) drank (v.) coffee (n.) at (prep.) Starbucks (n.)”

[n. v. n. prep. n.] is called label list 1.

Another one may be [n. v. v. prep. n.]

We only want the best one, so we need feature functions to score each label list.
Feature Functions in a CRF

In a CRF, each **feature function** is a function that takes in as input:

- a sentence $s$
- the position $i$ of a word in the sentence
- the label $l_i$ of the current word
- the label $l_{i-1}$ of the previous word

and outputs a real-valued number (though the numbers are often just either 0 or 1).

Some examples:

$$f_1(s, i, l_i, l_{i-1}) = 1 \text{ if } l_i = \text{ADVERB and the ith word ends in "-ly";} \ 0 \text{ otherwise.}$$

$$f_2(s, i, l_i, l_{i-1}) = 1 \text{ if } i = 1, l_i = \text{VERB, and the sentence ends in a question mark;}$$

$$f_4(s, i, l_i, l_{i-1}) = 1 \text{ if } l_{i-1} = \text{PREPOSITION and } l_i = \text{PREPOSITION.}$$
Next, assign each feature function $f_j$ a weight $\lambda_j$ (I’ll talk below about how to learn these weights from the data). Given a sentence $s$, we can now score a labeling $l$ of $s$ by adding up the weighted features over all words in the sentence:

$$score(l|s) = \sum_{j=1}^{m} \sum_{i=1}^{n} \lambda_j f_j(s, i, l, l_{i-1})$$

(The first sum runs over each feature function $j$, and the inner sum runs over each position $i$ of the sentence.)

Finally, we can transform these scores into probabilities $p(l|s)$ between 0 and 1 by exponentiating and normalizing:

$$p(l|s) = \frac{\exp(score(l|s))}{\sum_{l'} \exp(score(l'|s))} = \frac{\exp[\sum_{j=1}^{m} \sum_{i=1}^{n} \lambda_j f_j(s, i, l, l_{i-1})]}{\sum_{l'} \exp[\sum_{j=1}^{m} \sum_{i=1}^{n} \lambda_j f_j(s, i, l', l'_{i-1})]}$$

Smells like Logistic Regression?
Basic model

Embedding + Bi-LSTM + CRF

This model is so prevailing.
Why embedding?

- To replace traditional one-hot inputs of CRF with word vectors which include more semantic information

  e.g. $v(\text{queen}) - v(\text{king}) = v(\text{woman}) - v(\text{man})$
Why embedding in character level?

- Some proper nouns have no trained word embeddings.

- Introducing morphological information
e.g. capital letter - ‘Apple’

- Extracting more semantic information
e.g. Prefix – ‘re’-> ‘back’
  Suffix – ‘-ed’ -> past tenses
  Root
Both of the word-level layer and the character-level layer can be implemented as convolutional neural networks (CNNs) or recurrent neural networks (RNNs) (e.g. GRU/LSTM).

“Named Entity Recognition with Bidirectional LSTM-CNNs”
Word NN and Char NN

- Aiming to gain contextual information
- Usually using Bi-LSTM (why?)

Input: embedding vector
Output: contextual representation
Transfer models

T-A: used for cross-domain transfer where label mapping is possible.

e.g. POS tags in the Genia biomedical corpus & Penn Treebank tags (Barrett & Weber-Jahnke, 2014)

In this situation, we share all the model parameters and feature representation in the neural networks, including the word and character embedding.
Transfer models

T-B: used for cross-domain transfer with disparate label sets, and cross-application transfer.

- Cross-domain transfer with disparate label sets:
  - POS tags in the Genia biomedical corpus & POS tags in Twitter (e.g., “URL”)
- Cross-application transfer:
  - POS tagging, chunking and named entity recognition

In this situation, we untie the parameter sharing in the CRF layer—i.e., each task learns a separate CRF layer.
T-C: used for cross-lingual transfer.

It is very difficult for transfer learning between languages with disparate alphabets (e.g., English and Chinese).

In this situation, we share the character embeddings and the character-level layer between different languages for transfer learning.
Experiment

(a) Transfer from PTB to Genia.
(b) Transfer from CoNLL 2003 NER to Genia.
(c) Transfer from Spanish NER to Genia.
(d) Transfer from PTB to Twitter POS tagging.
(e) Transfer from CoNLL 2003 to Twitter NER.
Experiment

(f) Transfer from CoNLL 2003 NER to PTB POS tagging.

(g) Transfer from PTB POS tagging to CoNLL 2000 chunking.

(h) Transfer from PTB POS tagging to CoNLL 2003 NER.

(i) Transfer from CoNLL 2003 English NER to Spanish NER.

(j) Transfer from Spanish NER to CoNLL 2003 English NER.
## Experiment

Table 1: Dataset statistics.

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Task</th>
<th>Language</th>
<th># Training Tokens</th>
<th># Dev Tokens</th>
<th># Test Tokens</th>
</tr>
</thead>
<tbody>
<tr>
<td>PTB 2003</td>
<td>POS Tagging</td>
<td>English</td>
<td>912,344</td>
<td>131,768</td>
<td>129,654</td>
</tr>
<tr>
<td>CoNLL 2000</td>
<td>Chunking</td>
<td>English</td>
<td>211,727</td>
<td>-</td>
<td>47,377</td>
</tr>
<tr>
<td>CoNLL 2003</td>
<td>NER</td>
<td>English</td>
<td>204,567</td>
<td>51,578</td>
<td>46,666</td>
</tr>
<tr>
<td>CoNLL 2002</td>
<td>NER</td>
<td>Dutch</td>
<td>202,931</td>
<td>37,761</td>
<td>68,994</td>
</tr>
<tr>
<td>CoNLL 2002</td>
<td>NER</td>
<td>Spanish</td>
<td>207,484</td>
<td>51,645</td>
<td>52,098</td>
</tr>
<tr>
<td>Genia</td>
<td>POS Tagging</td>
<td>English</td>
<td>400,658</td>
<td>50,525</td>
<td>49,761</td>
</tr>
<tr>
<td>Twitter</td>
<td>POS Tagging</td>
<td>English</td>
<td>12,196</td>
<td>1,362</td>
<td>1,627</td>
</tr>
<tr>
<td>Twitter</td>
<td>NER</td>
<td>English</td>
<td>36,936</td>
<td>4,612</td>
<td>4,921</td>
</tr>
</tbody>
</table>
Experiment

Table 2: Improvements with transfer learning under multiple low-resource settings (%). “Dom”, “app”, and “ling” denote cross-domain, cross-application, and cross-lingual transfer settings respectively. The numbers following the slashes are labeling rates (chosen such that the number of labeled examples are of the same scale).

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Model</th>
<th>Setting</th>
<th>Transfer</th>
<th>No Transfer</th>
<th>Delta</th>
</tr>
</thead>
<tbody>
<tr>
<td>PTB</td>
<td>Twitter/0.1</td>
<td>T-A</td>
<td>dom</td>
<td>83.65</td>
<td>74.80</td>
<td>8.85</td>
</tr>
<tr>
<td>CoNLL03</td>
<td>Twitter/0.1</td>
<td>T-A</td>
<td>dom</td>
<td>43.24</td>
<td>34.65</td>
<td>8.59</td>
</tr>
<tr>
<td>PTB</td>
<td>CoNLL03/0.01</td>
<td>T-B</td>
<td>app</td>
<td>74.92</td>
<td>68.64</td>
<td>6.28</td>
</tr>
<tr>
<td>PTB</td>
<td>CoNLL00/0.01</td>
<td>T-B</td>
<td>app</td>
<td>86.73</td>
<td>83.49</td>
<td>3.24</td>
</tr>
<tr>
<td>CoNLL03</td>
<td>PTB/0.001</td>
<td>T-B</td>
<td>app</td>
<td>87.47</td>
<td>84.16</td>
<td>3.31</td>
</tr>
<tr>
<td>Spanish</td>
<td>CoNLL03/0.01</td>
<td>T-C</td>
<td>ling</td>
<td>72.61</td>
<td>68.64</td>
<td>3.97</td>
</tr>
<tr>
<td>CoNLL03</td>
<td>Spanish/0.01</td>
<td>T-C</td>
<td>ling</td>
<td>60.43</td>
<td>59.84</td>
<td>0.59</td>
</tr>
<tr>
<td>PTB</td>
<td>Genia/0.001</td>
<td>T-A</td>
<td>dom</td>
<td>92.62</td>
<td>83.26</td>
<td>9.36</td>
</tr>
<tr>
<td>CoNLL03</td>
<td>Genia/0.001</td>
<td>T-B</td>
<td>dom&amp;app</td>
<td>87.47</td>
<td>83.26</td>
<td>4.21</td>
</tr>
<tr>
<td>Spanish</td>
<td>Genia/0.001</td>
<td>T-C</td>
<td>dom&amp;app&amp;ling</td>
<td>84.39</td>
<td>83.26</td>
<td>1.13</td>
</tr>
<tr>
<td>PTB</td>
<td>Genia/0.001</td>
<td>T-B</td>
<td>dom</td>
<td>89.77</td>
<td>83.26</td>
<td>6.51</td>
</tr>
<tr>
<td>PTB</td>
<td>Genia/0.001</td>
<td>T-C</td>
<td>dom</td>
<td>84.65</td>
<td>83.26</td>
<td>1.39</td>
</tr>
</tbody>
</table>
Experiment

Table 3: Comparison with state-of-the-art results (%).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Collobert et al. (2011)</td>
<td>94.32</td>
<td>89.59</td>
<td>–</td>
<td>–</td>
<td>97.29</td>
</tr>
<tr>
<td>Passos et al. (2014)</td>
<td>–</td>
<td>90.90</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Luo et al. (2015)</td>
<td>–</td>
<td>91.2</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Huang et al. (2015)</td>
<td>94.46</td>
<td>90.10</td>
<td>–</td>
<td>–</td>
<td>97.55</td>
</tr>
<tr>
<td>Gillick et al. (2015)</td>
<td>–</td>
<td>86.50</td>
<td>82.95</td>
<td>82.84</td>
<td>–</td>
</tr>
<tr>
<td>Ling et al. (2015)</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>97.78</td>
</tr>
<tr>
<td>Lample et al. (2016)</td>
<td>–</td>
<td>90.94</td>
<td>85.75</td>
<td>81.74</td>
<td>–</td>
</tr>
<tr>
<td>Ma &amp; Hovy (2016)</td>
<td>–</td>
<td>91.21</td>
<td>–</td>
<td>–</td>
<td>97.55</td>
</tr>
<tr>
<td>Ours w/o transfer</td>
<td>94.66</td>
<td>91.20</td>
<td>84.69</td>
<td>85.00</td>
<td>97.55</td>
</tr>
<tr>
<td>Ours w/ transfer</td>
<td><strong>95.41</strong></td>
<td><strong>91.26</strong></td>
<td><strong>85.77</strong></td>
<td><strong>85.19</strong></td>
<td><strong>97.55</strong></td>
</tr>
</tbody>
</table>
Factors are crucial for the performance transfer learning approach:

a) label abundance for the target task,
b) relatedness between the source and target tasks
c) the number of parameters that can be shared.
Conclusions

Contributions:
1) Achieving significant improvement on various datasets under low-resource conditions, as well as new state-of-the-art results on some of the benchmarks
2) Proposing three neural network architectures for the settings of cross-domain, cross-application, and cross-lingual transfer and drawing some valuable experiment conclusions.

Shortages:
1) It is not clear why choosing GRU for embedding, instead of LSTM.
2) Only discussing model-based transfer

Improvements:
1) A mixed structure of LSTM and GRU
2) Combining model-based transfer with resource-based transfer for cross-lingual transfer learning.
Thanks!