Natural Language Processing and Chinese Computing

8th CCF International Conference, NLPCC 2019
Dunhuang, China, October 9–14, 2019
Proceedings, Part II
Contents – Part II

Text Mining

An Improved Class-Center Method for Text Classification Using Dependencies and WordNet .................................................. 3
  Xinhua Zhu, Qingting Xu, Yishan Chen, and Tianjun Wu

Dynamic Label Correction for Distant Supervision Relation Extraction via Semantic Similarity ........................................... 16
  Xinyu Zhu, Gongshen Liu, Bo Su, and Jan Pan Nee

Classification over Clustering: Augmenting Text Representation with Clusters Helps! ......................................................... 28
  Xiaoye Tan, Rui Yan, Chongyang Tao, and Mingrui Wu

Document-Based Question Answering Improves Query-Focused Multi-document Summarization .................................................. 41
  Weikang Li, Xingxing Zhang, Yunfang Wu, Furu Wei, and Ming Zhou

BSIL: A Brain Storm-Based Framework for Imbalanced Text Classification ................................................................. 53
  Jiachen Tian, Shizhan Chen, Xiaowang Zhang, and Zhiyong Feng

A Key-Phrase Aware End2end Neural Response Generation Model .......... 65
  Jun Xu, Haifeng Wang, Zhengyu Niu, Hua Wu, and Wanxiang Che

A Hierarchical Model with Recurrent Convolutional Neural Networks for Sequential Sentence Classification .......................... 78
  Xinyu Jiang, Bowen Zhang, Yunming Ye, and Zhenhua Liu

REKER: Relation Extraction with Knowledge of Entity and Relation ........ 90
  Hongtao Liu, Yian Wang, Fangzhao Wu, Pengfei Jiao, Hongyan Xu, and Xing Xie

IPRE: A Dataset for Inter-Personal Relationship Extraction ......................... 103
  Haitao Wang, Zhengqiu He, Jin Ma, Wentiang Chen, and Min Zhang

Shared-Private LSTM for Multi-domain Text Classification .................... 116
  Haiming Wu, Yue Zhang, Xi Jin, Yun Xue, and Ziwen Wang

Co-attention and Aggregation Based Chinese Recognizing Textual Entailment Model ......................................................... 129
  Pengcheng Liu, Lingling Mu, and Hongying Zan
A Knowledge-Gated Mechanism for Utterance Domain Classification 142
Zefeng Du, Peijie Huang, Yuhong He, Wei Liu, and Jiankai Zhu

Short Papers

Automatic Translating Between Ancient Chinese and Contemporary Chinese with Limited Aligned Corpora 157
Zhiyuan Zhang, Wei Li, and Qi Su

FlexNER: A Flexible LSTM-CNN Stack Framework for Named Entity Recognition 168
Hongyin Zhu, Wenpeng Hu, and Yi Zeng

Domain Adaptive Question Answering over Knowledge Base 179
Yulai Yang, Lei Hou, Haifong Jin, Peng Zhang, Juanzi Li, Yi Huang, and Min Hu

Extending the Transformer with Context and Multi-dimensional Mechanism for Dialogue Response Generation 189
Ruxin Tan, Jiahui Sun, Bo Su, and Gongshen Liu

Co-attention Networks for Aspect-Level Sentiment Analysis 200
Haihui Li, Yun Xue, Hongya Zhao, Xiaohui Hu, and Sancheng Peng

Hierarchical-Gate Multimodal Network for Human Communication Comprehension 210
Qiyuan Liu, Liangqing Wu, Yang Xu, Dong Zhang, Shoushan Li, and Guodong Zhou

End-to-End Model for Offline Handwritten Mongolian Word Recognition 220
Hongxi Wei, Cong Liu, Hui Zhang, Feilong Bao, and Guanglai Gao

A Recursive Information Flow Gated Model for RST-Style Text-Level Discourse Parsing 231
Longyin Zhang, Xin Tan, Fang Kong, and Guodong Zhou

Relation Classification in Scientific Papers Based on Convolutional Neural Network 242
Zhongbo Yin, Shuai Wu, Yi Yin, Wei Luo, Zhunchen Luo, Yushani Tan, Xiangyu Jiao, and Dong Wang

Effective Soft-Adaptation for Neural Machine Translation 254
Shuangzhi Wu, Dongdong Zhang, and Ming Zhou

Neural Machine Translation with Bilingual History Involved Attention 265
Haiyang Xue, Yang Feng, Di You, Wen Zhang, and Jingyu Li
Joint Modeling of Recognizing Macro Chinese Discourse Nuclearity and Relation Based on Structure and Topic Gated Semantic Network .......................... 276
    Feng Jiang, Peifeng Li, and Qiaoming Zhu

Attentional Neural Network for Emotion Detection in Conversations with Speaker Influence Awareness .............................................................. 287
    Jia Wei, Shi Feng, Daling Wang, Yifei Zhang, and Xiangju Li

Triple-Joint Modeling for Question Generation Using Cross-Task Autoencoder .................................................................................. 298
    Hongling Wang, Renshou Wu, Zhixu Li, Zhongqing Wang, Zhigang Chen, and Guodong Zhou

Deep Multi-task Learning with Cross Connected Layer for Slot Filling ........ 308
    Junsheng Kong, Yi Cai, Da Ren, and Zilu Li

A Sequence-to-Action Architecture for Character-Based Chinese Dependency Parsing with Status History ..................................................... 318
    Hang Liu, Yujie Zhang, Meng Chen, Jinan Xu, and Yufeng Chen

A New Fine-Tuning Architecture Based on Bert for Word Relation Extraction .................................................................................. 327
    Fanyu Meng, Junlan Feng, Danping Yin, and Min Hu

Rumor Detection with Hierarchical Recurrent Convolutional Neural Network .................................................................................. 338
    Xiang Lin, Xiangwen Liao, Tong Xu, Wenjing Pian, and Kam-Fai Wong

Automatic Proofreading in Chinese: Detect and Correct Spelling Errors in Character-Level with Deep Neural Networks ...................... 349
    Qiuqin Wang, Minghuan Liu, Weijia Zhang, Yuhang Guo, and Tianrui Li

Linguistic Feature Representation with Statistical Relational Learning for Readability Assessment ..................................................... 360
    Xinying Qiu, Dawei Lu, Yuming Shen, and Yi Cai

Neural Classifier with Statistic Information of User and Product for Sentiment Analysis .......................................................... 370
    Changliang Li, Jiayu Xie, and Yaoguang Xing

A New Algorithm for Component Decomposition and Type Recognition of Tibetan Syllable .......................................................... 381
    Jie Zhu, Shugen Wang, Yanru Wu, Meijing Guan, and Yang Feng

Event Temporal Relation Classification Based on Graph Convolutional Networks .................................................................................. 393
    Qianwen Dai, Fang Kong, and Qianying Dai
<table>
<thead>
<tr>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event Factuality Detection in Discourse</td>
<td>404</td>
</tr>
<tr>
<td>Rongtiao Huang, Bowei Zou, Hongling Wang, Peifeng Li, and Guodong Zhou</td>
<td></td>
</tr>
<tr>
<td>A Multi-pattern Matching Algorithm for Chinese-Hmong Mixed Strings</td>
<td>415</td>
</tr>
<tr>
<td>Shun-Ping He, Li-Ping Mo, and Di-Wen Kang</td>
<td></td>
</tr>
<tr>
<td>Constructing the Image Graph of Tang Poetry</td>
<td>426</td>
</tr>
<tr>
<td>Bihua Wang, Renfen Hu, and Lijiao Yang</td>
<td></td>
</tr>
<tr>
<td>Research on Fine-Grained Sentiment Classification</td>
<td>435</td>
</tr>
<tr>
<td>Zhihui Wang, Xiaodong Wang, Tao Chang, Shaohe Lv, and Xiaoting Guo</td>
<td></td>
</tr>
<tr>
<td>Question Generation Based Product Information</td>
<td>445</td>
</tr>
<tr>
<td>Kang Xiao, Xiaobing Zhou, Zhongqing Wang, Xiangyu Duan, and Min Zhang</td>
<td></td>
</tr>
<tr>
<td>Conversion and Exploitation of Dependency Treebanks with Full-Tree LSTM</td>
<td>456</td>
</tr>
<tr>
<td>Bo Zhang, Zhenghua Li, and Min Zhang</td>
<td></td>
</tr>
<tr>
<td>Analysis of Back-Translation Methods for Low-Resource Neural Machine Translation</td>
<td>466</td>
</tr>
<tr>
<td>Nuo Xu, Yinqiao Li, Chen Xu, Yanyang Li, Bei Li, Tong Xiao, and Jingbo Zhu</td>
<td></td>
</tr>
<tr>
<td>Multi-classification of Theses to Disciplines Based on Metadata</td>
<td>476</td>
</tr>
<tr>
<td>Jianling Li, Shiwen Yu, Shasha Li, and Jie Yu</td>
<td></td>
</tr>
<tr>
<td>Chinese Event Factuality Detection</td>
<td>486</td>
</tr>
<tr>
<td>Jiaxuan Sheng, Bowei Zou, Zhengxian Gong, Yu Hong, and Guodong Zhou</td>
<td></td>
</tr>
<tr>
<td>Research of Uyghur-Chinese Machine Translation System Combination Based on Semantic Information</td>
<td>497</td>
</tr>
<tr>
<td>YaJuan Wang, Xiao Li, YaTing Yang, Azmat Anwar, and Rui Dong</td>
<td></td>
</tr>
<tr>
<td>Subject Recognition in Chinese Sentences for Chatbots</td>
<td>508</td>
</tr>
<tr>
<td>Fangyuan Li, Huanhuan Wei, Qiayda Hao, Ruihong Zeng, Hao Shao, and Wenliang Chen</td>
<td></td>
</tr>
<tr>
<td>Research on Khalkha Dialect Mongolian Speech Recognition Acoustic Model Based on Weight Transfer</td>
<td>519</td>
</tr>
<tr>
<td>Linyan Shi, Feilong Bao, Yonghe Wang, and Guanglai Gao</td>
<td></td>
</tr>
</tbody>
</table>
GANCoder: An Automatic Natural Language-to-Programming Language Translation Approach Based on GAN ................................. 529
   Yabing Zhu, Yanfeng Zhang, Huili Yang, and Fangjing Wang

Automatic Chinese Spelling Checking and Correction Based on Character-Based Pre-trained Contextual Representations ........... 540
   Haihua Xie, Aolin Li, Yabo Li, Jing Cheng, Zhiyou Chen, Xiaqing Lyu, and Zhi Tang

A Context-Free Spelling Correction Method for Classical Mongolian. ........ 550
   Min Lu, Feilong Bao, and Guanglai Gao

Explainable AI Workshop

Explainable AI: A Brief Survey on History, Research Areas, Approaches and Challenges .................................................. 563
   Feiyu Xu, Hans Uszkoreit, Yangzhou Du, Wei Fan, Dongyan Zhao, and Jun Zhu

Interpretable Spatial-Temporal Attention Graph Convolution Network for Service Part Hierarchical Demand Forecast ..................... 575
   Wenli Ouyang, Yahong Zhang, Mingda Zhu, Xiuding Zhang, Hongye Chen, Yinghao Ren, and Wei Fan

Disentangling Latent Emotions of Word Embeddings on Complex Emotional Narratives ...................................................... 587
   Zhengxuan Wu and Yueyi Jiang

Modeling Human Intelligence in Customer-Agent Conversation Using Fine-Grained Dialogue Acts .......................................... 596
   Qicheng Ding, Guoguang Zhao, Penghui Xu, Yuancheng Jin, Yu Zhang, Changjian Hu, Qianying Wang, and Feiyu Xu

Feature-Less End-to-End Nested Term Extraction ........................................ 607
   Yuzhe Gao and Yu Yuan

Using Aspect-Based Analysis for Explainable Sentiment Predictions ........ 617
   Thiago De Sousa Silveira, Hans Uszkoreit, and Renlong Ai

Discrimination Assessment for Saliency Maps ..................................... 628
   Ruiyi Li, Yangzhou Du, Zhongchao Shi, Yang Zhang, and Zhiqiang He

A Modified LIME and Its Application to Explain Service Supply Chain Forecasting ......................................................... 637
   Haisheng Li, Wei Fan, Sheng Shi, and Qiang Chou

Cross-lingual Neural Vector Conceptualization .................................. 645
   Lisa Ratihel and Robert Schwarzenberg
Student Workshop

A Dynamic Word Representation Model Based on Deep Context .......... 655
Xiao Yuan, Xi Xiong, Shenggen Ju, Zhengwen Xie, and Jiawei Wang

A Category Detection Method for Evidence-Based Medicine .......... 665
Jingyan Wang, Shenggen Ju, Xi Xiong, Rui Zhang, and Ningning Liu

Gender Prediction Based on Chinese Name ......................... 676
Jizheng Jia and Qiyang Zhao

Explanation Chains Model Based on the Fine-Grained Data ........... 684
Fu-Yuan Ma, Wen-Qi Chen, Min-Hao Xiao, Xin Wang, and Ying Wang

Evaluation Workshop

Goal-Oriented Knowledge-Driven Neural Dialog Generation System ...... 701
Ao Luo, Chen Su, and Shengfeng Pan

BERT-Based Multi-head Selection for Joint Entity-Relation Extraction ...... 713
Weipeng Huang, Xingyi Cheng, Tai Feng Wang, and Wei Chu

Proactive Knowledge-Goals Dialogue System Based on Pointer Network .. 724
Han Zhou, Chong Chen, Hao Liu, Fei Qin, and Huaqing Liang

Multiple Perspective Answer Reranking for Multi-passage Reading
Comprehension ............................................................... 736
Muchen Ren, Heyan Huang, Ran Wei, Hongyu Liu, Yu Bai,
Yang Wang, and Yang Gao

A Sketch-Based System for Semantic Parsing .......................... 748
Zechang Li, Yuxuan Lai, Yuxi Xie, Yansong Feng, and Dongyan Zhao

Overview of the NLPCC 2019 Shared Task: Cross-Domain
Dependency Parsing ......................................................... 760
Xue Peng, Zhenghua Li, Min Zhang, Rui Wang, Yue Zhang, and Luo Si

A Transformer-Based Semantic Parser for NLPCC-2019 Shared Task 2 .... 772
Donglai Ge, Junhui Li, and Muhua Zhu

A Relation Proposal Network for End-to-End Information Extraction .... 782
Zhenhua Liu, Tianyi Wang, Wei Dai, Zehui Dai, and Guangpeng Zhang

DuIE: A Large-Scale Chinese Dataset for Information Extraction .......... 791
Shuangjie Li, Wei He, Yabin Shi, Wenbin Jiang, Haijin Liang, Ye Jiang,
Yang Zhang, Yajuan Lyu, and Yong Zhu
Domain Information Enhanced Dependency Parser .......................... 801
   Nan Yu, Zonglin Liu, Ranran Zhen, Tao Liu, Meishan Zhang,
   and Guohong Fu

Overview of the NLPCC 2019 Shared Task: Open Domain
Semantic Parsing............................................................... 811
   Nan Duan

An Overview of the 2019 Language and Intelligence Challenge ............ 818
   Quan Wang, Wenquan Wu, Yabing Shi, Hongyu Li, Zhen Guo, Wei He,
   Hongyu Liu, Ying Chen, Yajuan Lyu, and Hua Wu

Overview of the NLPCC 2019 Shared Task: Open Domain
Conversation Evaluation ..................................................... 829
   Ying Shan, Anqi Cui, Luchen Tan, and Kun Xiong

Cross-Domain Transfer Learning for Dependency Parsing ................ 835
   Zuchao Li, Junru Zhou, Hai Zhao, and Rui Wang

Author Index ................................................................. 845
Overview of the NLPCC 2019 Shared Task: Cross-Domain Dependency Parsing

Xue Peng¹, Zhenghua Li¹(✉), Min Zhang¹, Rui Wang², Yue Zhang², and Luo Si²

¹ School of Computer Science and Technology, Soochow University, Suzhou, China
20175227031@stu.suda.edu.cn, {zh1113, minzhang}@suda.edu.cn
² Alibaba Group, Hangzhou, China
{masi.wr, shiyu.zy, luo.si}@alibaba-inc.com

Abstract. This paper presents an overview of the NLPCC 2019 shared task on cross-domain dependency parsing, including (1) the data annotation process, (2) task settings, (3) methods, results, and analysis of submitted systems and our recent work (Li+19), (4) discussions on related works and future directions. Considering that unsupervised domain adaptation is very difficult and has made limited progress in the past decades, we for the first time setup semi-supervised subtasks that allow to use a few thousand target-domain labeled sentences for training. We provide about 17 K labeled sentences from a balanced corpus as the source domain (BC), and as three target domains 10 K sentences from product comments (PC), 8 K sentences from product blogs (PB), and 3 K sentences from the web fiction named “Zhuxian” (ZX). All information about this task can be found at http://hlt.suda.edu.cn/index.php/Nlpcc-2019-shared-task, including the data sharing agreement.

1 Introduction

With the surge of web data (or user generated content), cross-domain parsing has become the major challenge for applying syntactic analysis in realistic NLP systems. To meet the challenge of the lack of labeled data, we have manually annotated large-scale high-quality domain-aware datasets with a lot of effort in the past few years.¹ Figure 1 shows an example dependency tree.

For this shared task, we provide about 17 K sentences from a balanced corpus as the source domain (BC), and as three target domains 10 K sentences from product comments (PC), 8 K sentences from product blogs (PB), and 3 K sentences from the web fiction named “Zhuxian” (ZX). We setup four subtasks with two cross-domain scenarios, i.e., unsupervised domain adaptation (no target-domain training data) and semi-supervised (with target-domain training data), and two settings, i.e., closed and open.

¹ Webpage for our treebank annotation: http://hlt.suda.edu.cn/index.php/SUCDT.

Supported by National Natural Science Foundation of China (Grant No. 61876116, 61525205).

© Springer Nature Switzerland AG 2019
https://doi.org/10.1007/978-3-030-32236-6_60
shared task overview: cross-domain dependency parsing

fig. 1. An example from the PC domain. The English translation is “This looks very pretty with a white shirt.”

Please kindly note that semi-supervised in our domain adaptation scenario means that the model is also provided with labeled training data for the target domain. In contrast, traditional semi-supervised learning refers to approaches that make use of large-scale unlabeled data beside labeled data. Using target-domain unlabeled data is usually considered a must for domain adaptation, and thus we use semi-supervised to distinguish whether using target-domain labeled data, not unlabeled data.

There are totally 16 teams that sign in this shared task, and 2 other teams that only want to use the data for study. Finally, 7 teams submit their results, among which one team misunderstands the task settings and recalls their submission.

One team (SJ.SuperLZ) submits results for all four subtasks, and wins the first place in the semi-open subtask 4 among two submissions. Another 3 teams submit results for both closed (un/semi) subtasks, and SyntaxError wins the first place on both tracks. One team focuses on the semi-closed subtask 2 and another team focuses on the semi-open subtask 4. Only two teams participate in the open (un or semi) tracks, and the use of contextualized word representations such as ELMo [15] and BERT [4] is not common. Only SJ.SuperLZ builds their system upon BERT via fine-tuning.

In order to gain more insights on the state-of-the-art progress, we also experiment with the approaches with the same codes in our recent work (Li et al. 2019, abbr. as Li+19) [10] and compare with those submitted systems.

2 Related Shared Tasks

Due to space limitation, we only give a brief and incomplete introduction on previous shared tasks on dependency parsing, especially domain adaptations. Please refer to Li+19 for discussions on methods previously proposed for cross-domain dependency parsing.

As the first CoNLL shared task on dependency parsing, CoNLL-2006 on multi-Lingual dependency parsing releases benchmark datasets for 13 languages [1].

CoNLL-2007 focuses on both multilingual dependency parsing on 10 languages and domain adaptation on English [13]. For English, they release two
non-WSJ out-of-domain datasets, i.e., PCHEM (biomedical or chemical research abstracts) and CHILDES (the EVE corpus, only unlabeled dependency trees) datasets. They provide a combined development dataset for parameter selection and two separate test datasets for final evaluation. Although the organizers put a lot of effort to make the out-of-domain datasets be the consistent with the WSJ data in annotation schemes, the annotated trees still contains a lot of consistencies between source- and target-domains, which may be the reason behind the failure of many adaptation methods, as discussed in Dredze et al. [5].

CoNLL-2008 proposes the tasks of joint parsing of syntactic and semantic dependencies for English, with the aim of encouraging joint modeling of two related tasks, i.e., dependency parsing and word-based SRL [19]. They convert span-based SRL into word-based semantic dependencies based on head-finding rules.

CoNLL-2009 extends the previous-year task from monolingual English to multilingual, covering 7 languages [7]. For English, German, and Czech, they also prepare out-of-domain test datasets for final evaluation. However, the focus is not on domain adaptation without providing the dev data, though initial results are obtained on the outside domains.

CoNLL-2017/2018 proposes multilingual parsing from raw text to universal dependencies for many languages [23, 24], thanks to the development of the University Dependencies project [14]. Participants need to process raw text without gold-standard tokenization (or word segmentation) and morphological features such as part-of-speech tags. CoNLL-2018 covers 82 UD treebanks in 57 languages.

SANCL-2012. Petrov and McDonald [16] organized the “parsing the web” shared task and the first workshop on syntactic analysis of non-canonical language (SANCL), based on their newly-constructed Google English Web Treebank consisting of web texts of five sources, i.e., email, answers, weblogs, newsgroups, and reviews. Each source has 1–2 K test sentences manually annotated with constituent trees. The goal is to adapt WSJ-trained parsers (constituent- or dependency-based) to these new domains.

3 Data Annotation

Due to space limitation, we only introduce our data annotation process very briefly. Please refer to Li et al. [10] for more details.

Annotation Guideline. During the past few years, we have been developing and updating a detailed annotation guideline that on the one hand aims to fully capture Chinese syntax, and on the other hand tries to guarantee inter-annotator consistency and facilitate model learning. The current guideline has about 80 pages and contains 21 dependency labels. Please refer to the SUCDT webpage for the newest version.

Annotation Process. All sentences are labeled by two persons (annotation phase), and inconsistent submissions are discussed by senior annotators to
determine the final answer (review phase). Annotators are required to re-do the incorrect submissions according to the answer given in the annotation platform, so that they can gradually improve their annotation ability (study phase). However, they can also decide to make complaints in the annotation system if they are sure that the provided answer is wrong, so that more senior annotators can re-check the answer (complaint phase).

**Partial Annotation.** In order to reduce annotation cost, we adopt the active learning procedure based on partial annotation, where sentences that are most difficult for the model are first chosen, and then a certain percentage of most difficult words are then selected for manual annotation. Most sentences in our labeled data are partially annotated. However, since training with partial trees is not popular in parsing community, we automatically complete all partial trees into high-quality full trees [26] based on our CRF-enhanced biaffine parser [9] trained on the combination of all our labeled data.²

Table 1 shows data statistics for this shared task. The source domain is balanced corpus (BC), composed of 8K sentences from HIT-CDT [2] and 8K from Penn CTB [21], annotated according to our guideline.³ The first domain is product comments (PC) from Taobao. The product blog (PB) texts are crawled from the Taobao headline website, which contains articles written by users mainly on description and comparison of different commercial products. The third target domain is “Zhuxian” (ZX, also known as “Jade dynasty”), a web fiction previously annotated and used for cross-domain word segmentation [25].

The datasets are annotated by many different annotators, and the averaged sentence lengths and annotated words numbers per sentence also vary for different domains. Therefore, the consistency ratios can only give a rough idea on the annotation difficulty for texts of different domains.

All labeled datasets are shared by Soochow University and the PC/PB unlabeled datasets are provided by Alibaba Group. Participants need to sign corresponding data sharing agreements to obtain the data.

<table>
<thead>
<tr>
<th></th>
<th>BC</th>
<th>PC</th>
<th>PB</th>
<th>ZX</th>
</tr>
</thead>
<tbody>
<tr>
<td>train/dev/test (K)</td>
<td>16.3/1/2</td>
<td>6.6/1.3/2.6</td>
<td>5.1/1.3/2.6</td>
<td>1.6/0.5/1.1</td>
</tr>
<tr>
<td>consensus ratio (sent-wise)</td>
<td>45.88</td>
<td>39.20</td>
<td>35.88</td>
<td>46.20</td>
</tr>
<tr>
<td>consensus ratio (token-wise)</td>
<td>82.25</td>
<td>68.23</td>
<td>69.38</td>
<td>79.21</td>
</tr>
<tr>
<td>aver. sent len</td>
<td>13.82</td>
<td>13.95</td>
<td>12.12</td>
<td>21.25</td>
</tr>
<tr>
<td>aver. num of annotated words</td>
<td>3.80</td>
<td>3.33</td>
<td>5.09</td>
<td>5.07</td>
</tr>
<tr>
<td>unlabeled (K)</td>
<td>0</td>
<td>350</td>
<td>300</td>
<td>30</td>
</tr>
</tbody>
</table>

² The parser can be tried at [http://hit-la.suda.edu.cn](http://hit-la.suda.edu.cn).
³ Our major purpose for annotating these datasets is to support supervised treebank conversion.
4 Task Settings

- Subtask 1 (un-closed): unsupervised domain adaptation (closed)
- Subtask 2 (semi-closed): semi-supervised domain adaptation (closed)
- Subtask 3 (un-open): unsupervised domain adaptation (open)
- Subtask 4 (semi-open): semi-supervised domain adaptation (open)

Unsupervised domain adaptation assumes that there is no labeled training data for the target domain. For example, when the target domain is PC, in the unsupervised domain adaptation scenario, you cannot use PC-Train. However, PC-Dev/Unlabeled are allowed to use.

Semi-supervised domain adaptation means that there exists a labeled training dataset for the target domain. For example, when the target domain is PC, PC-Train/Dev/Unlabeled all can be used in semi-supervised domain adaptation scenario.

Closed means that (1) you can only use our provided data and information, including the our provided word segmentation, automatic part-of-speech (POS) tags, pre-trained word embeddings;4 (2) it is not allowed to use other resources such as dictionaries, labeled or unlabeled data for related tasks; (3) it is not allowed to use other tools to produce new information such as ELMo/BERT, POS tagger, etc.

Open has no restriction, and you can use any resource. However, it is strongly recommended that participants of this shared task and future researchers clearly describe in their system reports and papers what external resources are used and how parsing performance is affected.

Multi-source domain adaptation? NO. It is not allowed to use training data from other target domains. For example, when the target domain is PC, PB-Train and ZX-Train cannot be used in all four subtasks. However, after this shared task, researchers may explore this research line with our data.

Train with dev data? NO. It is not allowed to add dev data into training data. Dev data can only be used for parameter and model selection.

Evaluation Metrics. We use the standard unlabeled/labeled attachment score (UAS/LAS, percent of words that receive correct heads [and labels]). We do not complete partial trees into full ones for the dev and test datasets. It is straightforward to perform evaluation against partial trees by simply omitting the words without gold-standard heads. For any subtask, a participant team must submit results on all three target domains, so that we can obtain three LAS values. We average the three LAS directly to determine the final ranking.

5 Methods

5.1 Submitted Systems

We delete the detailed descriptions of the methods of the submitted systems, which are in the order of task registration SJ SuperLZ (Shanghai Jiao Tong

4 The word embeddings are obtained by training word2vec on the Chinese Gigaword 3 and all the target-domain unlabeled data.
University, Li et al. [11]), 14yhl9days (University of South China), Syntax-
Error (Heilongjiang University, Yu et al. [22]), AntNLP (East China Normal
University), BLCU.Parser (Beijing Language and Culture University), NNU
(Nanjing Normal University, Xia et al. [20]). Please refer to the long version of
this paper at Zhenghua’ homepage.

5.2 Summaries of the Systems from Different Aspects

**Basic Parsers.** Three teams (i.e., SJ_superLZ, SyntaxError, BLCU.Parser)
adopt the biaffine parser, or a similar graph-based framework, due to its sim-
plicity and state-of-the-art performance. AntNLP employs the transition-based
L2RPTR parser [6] and their recently proposed graph-based GNN parser [8], as
two different views for the co-training procedure. NNU adopts the transition-
based STACKPTR parser [12], and 14yhl9days uses the basic transition-based

Our preliminary experiments with the biaffine parser show that using charLSTM-based word representation besides word embeddings, also mentioned
by BLCU.Parser, leads to better performance than using POS tag embeddings.

**Handling Partially Annotated Data.** To control the influence of noisy arcs
in the completed trees, SyntaxError finds that it is helpful to simply removing
arcs with marginal probability less than 0.7 for both the auto-completed full trees
in the labeled training data and the auto-parsed 1-best trees in the unlabeled
data during loss computation.

After reading their system report, we also run experiments based on the
biaffine parser (w/ local loss) to verify this issue. We find that compared
with using all dependencies, LAS of the CONCAT method (semi-supervised)
decreases by about 4 points if using only manually labeled dependencies (prob-
bility being 1 in the provided data) on ZX-dev, and LAS increases by about 0.4
points if only using dependencies of probability higher than 0.7.

Our results show that (1) the completed full trees are actually of high-
quality, considering we use a CRF-enhanced biaffine parser trained on about 100
thousand labeled sentences for the completion task; (2) filtering low-probability
dependencies is helpful, but not so very helpful on the labeled training data
(maybe more useful for unlabeled data).

Our previous work demonstrates that using probabilistic CRF-based parser
can more effectively learn from partial trees (manually annotated or after filter-
ing) [26].

**Combination of Source- and Target-Domain Labeled Data.** For the semi
subtasks 2 and 4, it is a very important to combine and balance the source- and
target-domain labeled training data. Four different strategies are investigated.

(1) The **fine-tuning method** first trains the model on the source-domain
training and then fine-tunes the model with the target-domain training data.
This idea is used by SJ.superLZ, BLCU.Parser and NNU. This method may suffer from the limited use of the source-domain training data.

(2) The MTL method, used by BLCU.Parser, treats the source- and target-domain parsing as two independent tasks under the MTL framework. Since only the encoder parameters are shared, the contribution of the source-domain training data is limited, considering the two domains have the same annotation guideline.

(3) The concatenation (CONCAT) method directly merges the two training datasets together without corpus weighting. This method is intuitively more effective than the above two, and is adopted by SyntaxError and AntNLP.

(4) The domain embedding (DOEMB) method is similar to the CONCAT method but distinguishes whether the input sentence comes from the source or target domains by appending a domain id embedding for each word. This method is investigated in Li+19 and is shown to be more effective than the CONCAT method.

The corpus weighting strategy, as described in Li+19, may be useful for balancing the contributions of the source- and target-domain training data, especially when the scales of the two datasets have large gap.

Utilizing Unlabeled Data. It is very attractive to utilize the large-scale and easy-to-collect unlabeled target-domain data in the research area of domain adaptation. Self/co/tri-training are widely used as typical approaches in traditional semi-supervised learning. The basic idea is first using the current parser(s) to automatically parse the unlabeled data, and selecting high-confidence parse trees as extra training data, and re-training the parser(s), and so on, in a bootstrapping fashion. Both SyntaxError and SJ.superLZ use tri-training, and AntNLP adopts co-training.

Recently, the emergence of contextualized word representations, such as ELMo and Bert, has greatly advanced NLP research. The use of very-large-scale unlabeled data via language model loss, allows very deep neural networks to effectively learn extensively useful word and sentence representations. The improvement from using such representations is surprisingly large for a variety of NLP tasks, sometimes surpassing the total amount of improvement from research during the past decade. In some sense, it seems not enough to classify ELMo/BERT-enhanced approaches as traditional semi-supervised learning approaches. In fact, some researchers even base on ELMo and BERT as a new transferring learning (broader than domain adaptation) methodology [17].

In the open subtasks 3 and 4, the use of ELMo/BERT is rare so far, with the exception of SJ.superLZ. With many different ways to use ELMo/BERT for domain adaptation, we believe that it is very helpful to fine-tune ELMo/BERT on the unlabeled data of both the source and target domains. This simple method can build connections between the open-domain very-large-scale unlabeled texts

\[\text{Please notice again that semi-supervised in our domain adaptation scenario is about whether using target-domain labeled training data.}\]
for pre-training ELMo/BERT and the data for our tasks in hands, and between the source and target domains.

**Model Ensemble.** Model ensemble has been used as an extremely useful and popular technique in many shared tasks. Both SyntaxError and 14yhl9days use model ensemble. For traditional discrete-feature based parsing, it is usually needed to use several divergent models trained on the same training data or a homogeneous model trained on random subsets of the whole training data. For neural network based parsing, different models can usually be obtained by using different random seeds for parameter initialization, which is adopted by SyntaxError.

The arc/label probabilities of different models are averaged for finding the optimal tree via Viterbi decoding, as SyntaxError does. Another popular way is to use the vote counts of the 1-best output parses of different models as the arc/label score for Viterbi decoding [18].

6 Results and Analysis

6.1 Re-running Our Li+19 Codes

To understand the effectiveness of approaches investigated in Li+19, we run the same codes\(^6\) on the datasets of this shared task.

The data settings differ in two major aspects between this shared task and Li+19. First, an extra PC domain is used in this shared task. Second, the BC training data in Li+19 contains about 52K sentences from HIT-CDT, including about 8K manually labeled sentences and 44K sentences (noisy full trees) being converted from the HIT-CDT guideline through supervised treebank conversion [9]. Therefore, the BC training data in Li+19 is larger but more noisy.

We delete the results and discussions on the effect of **corpus weighting. Please refer to the long version of this paper at Zhenghua’s homepage.**

**Main results on the dev data.** Table 2 shows the results on the dev data for the methods investigated in Li+19, i.e., CONCAT, DOEMB, ELMo, and fine-tuned ELMo. Most observations are consistent with Li+19, though the BC training data differs. In the following, we briefly examine and discuss the results from different perspectives.

**Domain differences.** Training on BC-train produces best performance on ZX, and worst performance on PC, indicating that BC is most similar with ZX and most dissimilar with PC. Compared with training with BC-train, training on the corresponding target-domain training data greatly boosts parsing performance, especially for PC.

**CONCAT vs. DOEMB.** The results clearly show that for the semi-supervised scenario, DOEMB is consistently superior to CONCAT for combining the source- and target-domain data, increasing LAS by 1.5-2.2.

Table 2. Final results on the dev data. FT-ELMo means fine-tuned ELMo.

<table>
<thead>
<tr>
<th></th>
<th>PC</th>
<th></th>
<th>PB</th>
<th></th>
<th>ZX</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>UAS</td>
<td>LAS</td>
<td>UAS</td>
<td>LAS</td>
<td>UAS</td>
<td>LAS</td>
</tr>
<tr>
<td>Trained on the source-domain data (unsupervised)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BC-train</td>
<td>40.26</td>
<td>25.64</td>
<td>67.40</td>
<td>60.60</td>
<td>71.22</td>
<td>63.73</td>
</tr>
<tr>
<td>BC-train + ELMo</td>
<td>44.31</td>
<td>30.46</td>
<td>70.91</td>
<td>64.70</td>
<td>73.95</td>
<td>66.17</td>
</tr>
<tr>
<td>BC-train + FT-ELMo</td>
<td><strong>52.33</strong></td>
<td><strong>39.13</strong></td>
<td><strong>77.76</strong></td>
<td><strong>71.77</strong></td>
<td><strong>76.27</strong></td>
<td><strong>69.70</strong></td>
</tr>
</tbody>
</table>

| Trained on the target-domain data (semi-supervised) |       |       |       |       |       |       |
| PC-train        | 68.03 | 59.68 | 62.90 | 55.98 | 54.67 | 44.69 |
| PB-train        | 42.14 | 30.13 | 78.11 | 72.68 | 60.24 | 49.34 |
| ZX-train        | 34.71 | 20.40 | 58.36 | 50.75 | **75.99** | **70.70** |
| PC-train + FT-ELMo | **72.34** | **63.82** | 73.46 | 67.24 | 63.33 | 53.27 |
| PB-train + FT-ELMo | 54.89 | 43.03 | **82.18** | **77.10** | 71.34 | 61.20 |
| ZX-train + FT-ELMo | 44.94 | 29.68 | 70.39 | 62.95 | **80.92** | **75.83** |

| Trained on source- and target-domain data (semi-supervised) |       |       |       |       |       |       |
| CONCAT          | 68.31 | 59.42 | 78.11 | 72.80 | 79.44 | 74.35 |
| DOEMB           | 69.37 | 60.94 | 80.03 | 75.00 | 80.76 | 76.23 |
| DOEMB + ELMo    | **69.20** | **60.48** | 80.31 | 75.43 | 81.24 | 75.99 |
| DOEMB + FT-ELMo | **72.01** | **63.86** | **83.10** | **78.87** | **83.21** | **78.84** |

**Effect of un-fine-tuned ELMo.** We use the same ELMo with Li+19, which is trained on the general-purpose Chinese Gigaword 3 corpus. We have added un-fine-tuned ELMo for unsupervised "BC-train + ELMo" and semi-supervised "DOEMB + ELMo", leading to opposite effect. In the unsupervised scenario, using ELMo representations as extra input boosts parsing accuracy by large margin (2.4-4.8 in LAS). In contrast, in the semi-supervised case, un-fine-tuned ELMo has little effects, maybe due to the high baseline (DOEMB) performance using the target-domain training data. We do not conduct experiments using un-fine-tuned ELMo on other scenarios to save computation resource.

**Effect of fine-tuned ELMo.** Similar to Li+19, we fine-tune ELMo on the combined corpus of BC-train and (PC/PB/ZX)-(train/unlabeled). It is very clear that fine-tuned ELMo can effectively further boost parsing accuracy over un-fine-tuned cases, i.e., "BC-train + ELMo" by 2.5-8.7, and "DOEMB + ELMo" by 2.9-3.4. When training with only the target-domain data in the second major row (comparing the diagonals), fine-tuned ELMo can improve LAS by 4.1-5.1.

Comparing the performance of "PC-train + FT-ELMo" and "DOEMB + FT-ELMo" on PC-dev, we can see that the source-domain BC-train provides little help, indicating again that BC and PC diverges too much to be helpful for the latter. We leave the investigation of the deeper reason behind this for future.

Overall, it is clear that the major conclusions drawn in Li+19 still stand on the datasets of this shared task. First, the domain embedding approach with
Table 3. Final results on the test data. SJ\_superLZ reruns the experiments and updates their results at the posterior evaluation stage, due to incidental experimental mistakes for un-open subtask 3.

<table>
<thead>
<tr>
<th></th>
<th>PC</th>
<th>PB</th>
<th>ZK</th>
<th>Averaged</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>UAS</td>
<td>LAS</td>
<td>UAS</td>
<td>LAS</td>
</tr>
<tr>
<td><strong>Un-closed Subtask 1</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SyntaxError</td>
<td>49.82</td>
<td>36.86</td>
<td>71.48</td>
<td>65.43</td>
</tr>
<tr>
<td>AntNLP</td>
<td>43.64</td>
<td>30.47</td>
<td>70.50</td>
<td>63.83</td>
</tr>
<tr>
<td>Ours (Bioaffine)</td>
<td>38.80</td>
<td>25.87</td>
<td>66.83</td>
<td>60.28</td>
</tr>
<tr>
<td>SJ_superLZ</td>
<td>35.63</td>
<td>21.33</td>
<td>62.55</td>
<td>54.35</td>
</tr>
<tr>
<td>14yhi9days</td>
<td>26.72</td>
<td>10.92</td>
<td>41.50</td>
<td>27.38</td>
</tr>
<tr>
<td><strong>Semi-closed Subtask 2</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SyntaxError</td>
<td>72.18</td>
<td>64.12</td>
<td>82.57</td>
<td>77.83</td>
</tr>
<tr>
<td>AntNLP</td>
<td>72.23</td>
<td>63.96</td>
<td>82.94</td>
<td>77.27</td>
</tr>
<tr>
<td>Ours (DOEMB)</td>
<td>68.88</td>
<td>59.96</td>
<td>79.22</td>
<td>74.40</td>
</tr>
<tr>
<td>BLCU_Parser</td>
<td>70.89</td>
<td>61.72</td>
<td>79.30</td>
<td>74.25</td>
</tr>
<tr>
<td>SJ_superLZ</td>
<td>69.88</td>
<td>59.43</td>
<td>77.52</td>
<td>71.33</td>
</tr>
<tr>
<td>14yhi9days</td>
<td>47.38</td>
<td>26.26</td>
<td>47.26</td>
<td>32.88</td>
</tr>
<tr>
<td><strong>Un-open Subtask 3</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SJ_superLZ (BERT, Post-Eval)</td>
<td>60.50</td>
<td>49.49</td>
<td>81.61</td>
<td>76.77</td>
</tr>
<tr>
<td>Ours (+FT-ELMo)</td>
<td>53.04</td>
<td>39.48</td>
<td>77.15</td>
<td>71.54</td>
</tr>
<tr>
<td><strong>Semi-open Subtask 4</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SJ_superLZ (BERT, Post-Eval)</td>
<td>75.25</td>
<td>67.77</td>
<td>85.53</td>
<td>81.51</td>
</tr>
<tr>
<td>Ours (DOEMB+FT-ELMo)</td>
<td>73.16</td>
<td>64.33</td>
<td>83.05</td>
<td>78.57</td>
</tr>
<tr>
<td>SJ_superLZ (BERT)</td>
<td>72.40</td>
<td>64.37</td>
<td>80.79</td>
<td>76.13</td>
</tr>
<tr>
<td>NNU</td>
<td>70.97</td>
<td>61.82</td>
<td>80.59</td>
<td>75.85</td>
</tr>
</tbody>
</table>

Corpus weighting is the most effective way to combine the source- and target-domain training data in the semi-supervised scenario (maybe adversarial training can further improve). Second, fine-tuning ELMo on the target-domain unlabeled data is extremely helpful due to the effect of bridging the knowledge between general open-domain texts and target-domain texts.

6.2 Final Results on the Test Data

Table 3 shows the final results on the test datasets. SyntaxError dominates the un-closed subtask 1. Our Li+19 baseline biaffine parser ranks the third place. We guess we can further improve our result by (1) using charLSTM word representations; (2) performing model ensemble; (3) utilizing unlabeled data.

For the semi-closed subtask 2, though still being the best, SyntaxError outperforms AntNLP by a small margin. We do not know why the gap between them becomes much smaller than for the un-closed subtask 1. They used different techniques to handle the target-domain training data. SyntaxError uses CONCAT and adversarial training, whereas AntNLP directly combine source- and target-domain data (CONCAT).
For the un-open subtask 3, SJ\texttt{superLZ} is the only submission and their posterior evaluation results outperform our Li+19 best method by 7.4 in LAS.

For the semi-open subtask 4, SJ\texttt{superLZ} outperforms NNU. However, the DOEMB method with fine-tuned ELMo in Li+19 slightly outperforms SJ\texttt{superLZ}, which is very interesting because BERT-based systems usually achieve much higher performance than ELMo-based ones. After retraining their models at the posterior evaluation stage, their averaged LAS increases by nearly 4 points.

7 Conclusions and Future Works

Overall, we feel that the attention so far is far from extensive for our first-year organization of the shared task on cross-domain dependency parsing. However, we have strong confidence on the future, and plan to organize more shared tasks by gradually releasing more labeled data covering more genres. We require all organizations that use our data in their experiments to release their full codes when publishing papers, in order to promote replicability.

References