

A Universal Framework for Inductive Transfer Parsing across Multi-typed Treebanks

Jiang Guo[♯], Wanxiang Che[♯], Haifeng Wang[♯] and Ting Liu[♯]

[♯]Center for Social Computing and Information Retrieval, Harbin Institute of Technology, China

[♯]Baidu Inc., China

{jguo, car, tliu}@ir.hit.edu.cn
wanghaifeng@baidu.com

Abstract

Various treebanks have been released for dependency parsing. Despite that treebanks may belong to different languages or have different annotation schemes, they contain common syntactic knowledge that is potential to benefit each other. This paper presents a universal framework for transfer parsing across multi-typed treebanks with deep multi-task learning. We consider two kinds of treebanks as source: the *multilingual universal treebanks* and the *monolingual heterogeneous treebanks*. Knowledge across the source and target treebanks are effectively transferred through multi-level parameter sharing. Experiments on several benchmark datasets in various languages demonstrate that our approach can make effective use of arbitrary source treebanks to improve target parsing models.

1 Introduction

As a long-standing central problem in natural language processing (NLP), dependency parsing has been dominated by data-driven approaches for decades. The foundation of data-driven parsing is the availability and scale of annotated training data (i.e., *treebanks*). Numerous efforts have been made towards the construction of treebanks which established the benchmark research on dependency parsing, such as the widely-used Penn Treebank (Marcus et al., 1993). However, the heavy cost of treebanking typically limits the existing treebanks in both scale and coverage of languages.

To address the problem, a variety of authors have proposed to exploit existing heterogeneous treebanks with different annotation schemes via grammar conversion (Niu et al., 2009), quasi-synchronous grammar features (Li et al., 2012) or shared feature representations (Johansson, 2013) for the enhancement of parsing models. Despite their effectiveness in specific datasets, these methods typically lack the scalability of exploiting richer source treebanks, such as cross-lingual treebanks.

In this paper, we aim at developing a universal framework for transfer parsing that can exploit multi-typed source treebanks to improve parsing of a target treebank. Specifically, we will consider two kinds of source treebanks, that are *multilingual universal treebanks* and *monolingual heterogeneous treebanks*.

Cross-lingual supervision has proven highly beneficial for parsing low-resource languages (Hwa et al., 2005; McDonald et al., 2011), implying that different languages have a great deal of common ground in grammars. But unfortunately, linguistic inconsistencies also exist in both typologies and lexical representations across languages. Figure 1(a) illustrates two sentences in German and English with universal dependency annotations. The typological differences (*subject-verb-object* order) results in the opposite directions of the *dobj* arcs, while the rest arcs remain consistent.

Similar problems also come with monolingual heterogeneous treebanks. Figure 1(b) shows an English sentence annotated with respectively the universal dependencies which are *content-head* and the CONLL dependencies which instead take the functional heads. Despite the structural divergences, these treebanks express the syntax of the same language, thereby sharing a large amount of common knowledge that can be effectively transferred.

† Corresponding author: Wanxiang Che

This work is licensed under a Creative Commons Attribution 4.0 International License. License details: <http://creativecommons.org/licenses/by/4.0/>

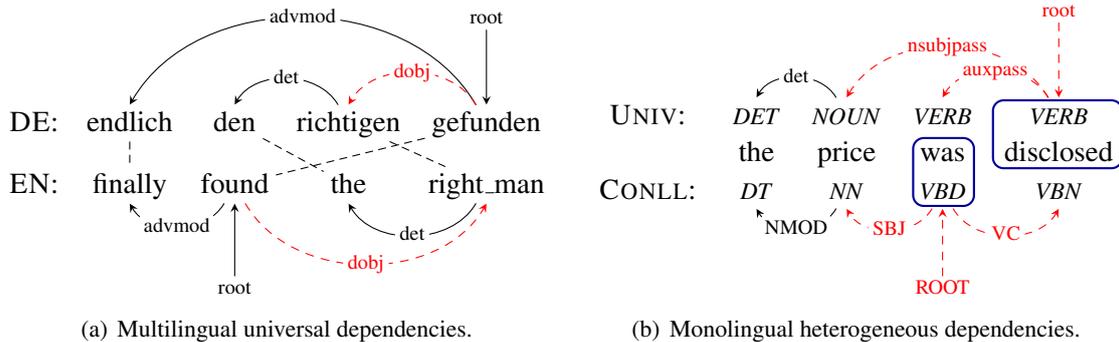


Figure 1: Comparisons between multilingual universal dependencies (a) and monolingual heterogeneous dependencies (b).

The present paper proposes a simple yet effective framework that aims at making full use of the consistencies while avoiding suffering from the inconsistencies across treebanks. Our framework effectively ties together the deep neural parsing models with multi-task learning, using multi-level parameter sharing to control the information flow across tasks. More specifically, learning with each treebank is maintained as an individual task, and their interactions are achieved through parameter sharing in different abstraction levels on the deep neural network, thus referred to as *deep multi-task learning*. We find that different parameter sharing strategies should be applied for different typed source treebanks adaptively, due to the different types of consistencies and inconsistencies (Figure 1).

We investigate the effect of multilingual transfer parsing using the Universal Dependency Treebanks (UDT) (McDonald et al., 2013). We show that our approach improves significantly over strong supervised baseline systems in six languages. We further study the effect of monolingual heterogeneous transfer parsing using UDT and the CONLL-X shared task dataset (Buchholz and Marsi, 2006). We consider using UDT and CoNLL-X as source treebanks respectively, to investigate their mutual benefits. Experiment results show significant improvements under both settings. Moreover, indirect comparisons on the Chinese Penn Treebank 5.1 (CTB5) using the Chinese Dependency Treebank (CDT)¹ as source treebank show the merits of our approach over previous work.²

2 Related Work

The present work is related to several strands of previous studies.

Monolingual resources for parsing Exploiting heterogeneous treebanks for parsing has been explored in various ways. Niu et al. (2009) automatically convert the dependency-structure CDT into the phrase-structure style of CTB5 using a trained constituency parser on CTB5, and then combine the converted treebanks for constituency parsing. Li et al. (2012) capture the annotation inconsistencies among different treebanks by designing several types of *transformation patterns*, based on which they introduce *quasi-synchronous grammar* features (Smith and Eisner, 2009) to augment the baseline parsing models. Johansson (2013) also adopts the idea of parameter sharing to incorporate multiple treebanks. They focus on parameter sharing at feature-level with discrete representations, which limits its scalability to multilingual treebanks where feature surfaces might be totally different. On the contrary, our approach is capable of utilizing representation-level parameter sharing, making full use of the multi-level abstractive representations generated by deep neural network. This is the key that makes our framework scalable to multi-typed treebanks and thus more practically useful.

Aside from resource utilization, attempts have also been made to integrate different parsing models through stacking (Torres Martins et al., 2008; Nivre and McDonald, 2008) or joint inference (Zhang and Clark, 2008; Zhang et al., 2014).

¹<https://catalog.ldc.upenn.edu/LDC2012T05>

²Our code is available at: <https://github.com/jiangfeng1124/mtl-nndep>.

Multilingual resources for parsing Cross-lingual transfer has proven to be a promising way of inducing parsers for low-resource languages, either through *data transfer* (Hwa et al., 2005; Tiedemann, 2014; Rasooli and Collins, 2015) or *model transfer* (McDonald et al., 2011; Täckström et al., 2012; Guo et al., 2015; Zhang and Barzilay, 2015; Guo et al., 2016).

Duong et al. (2015b) and Ammar et al. (2016) both adopt parameter sharing to exploit multilingual treebanks in parsing, but with a few important differences to our work. In both of their models, most of the neural network parameters are shared in two (or multiple) parsers except the feature embeddings,³ which ignores the important *syntactical inconsistencies* of different languages and is also inapplicable for heterogeneous treebanks that have different transition actions. Besides, Duong et al. (2015b) focus on low resource parsing where the target language has a small treebank of ~3K tokens. Their models may sacrifice accuracy on target languages with a large treebank. Ammar et al. (2016) and Vilares et al. (2016) instead train a single parser on a multilingual set of rich-resource treebanks, which is a more similar setting to ours. We refer to their approach as *shallow multi-task learning* (SMTL) and will include as one of our baseline systems (Section 4.2). Note that SMTL is a special case of our approach in which all tasks use the same set of parameters.

Bilingual parallel data has also proven beneficial in various ways (Chen et al., 2010; Huang et al., 2009; Burkett and Klein, 2008), demonstrating the potential of cross-lingual transfer learning.

Multi-task learning for NLP There has been a line of research on joint modeling pipelined NLP tasks, such as word segmentation, POS tagging and parsing (Hatori et al., 2012; Li et al., 2011; Bohnet and Nivre, 2012). Most multi-task learning or joint training frameworks can be summarized as parameter sharing approaches proposed by Ando and Zhang (2005). In the context of neural models for NLP, the most notable work was proposed by Collobert and Weston (2008), which aims at solving multiple NLP tasks within one framework by sharing common word embeddings. Henderson et al. (2013) present a joint dependency parsing and semantic role labeling model with the Incremental Sigmoid Belief Networks (ISBN) (Henderson and Titov, 2010). More recently, the idea of neural multi-task learning was applied to sequence-to-sequence problems with recurrent neural networks. Dong et al. (2015) use multiple decoders in neural machine translation systems that allows translating one source language to many target languages. Luong et al. (2015) study the ensemble of a wide range of tasks (e.g., syntactic parsing, machine translation, image caption, etc.) with multi-task sequence-to-sequence models.

To the best of our knowledge, we present the first work that successfully integrate both monolingual and multilingual treebanks for parsing, with or without consistent annotation schemes.

3 Approach

This section describes the deep multi-task learning architecture, using a formalism that extends on the transition-based dependency parsing model with LSTM networks (Dyer et al., 2015) which is further enhanced by modeling characters (Ballesteros et al., 2015). We first revisit the parsing approach of Ballesteros et al. (2015), then present our framework for learning with multi-typed source treebanks.

3.1 Transition-based Neural Parsing

Neural models for parsing have gained a lot of interests in recent years, particularly boosted by Chen and Manning (2014). The heart of transition-based parsing is the challenge of representing the *state* (configuration) of a transition system, based on which the most likely transition action is determined. Typically, a state includes three primary components, a *stack*, a *buffer* and a set of *dependency arcs*. Traditional parsing models deal with features extracted from manually defined feature templates in a discrete feature space, which suffers from the problems of *Sparsity*, *Incompleteness* and *Expensive feature computation*. The neural network model proposed by Chen and Manning (2014) instead represents features as continuous, low-dimensional vectors and use a *cube* activation function for implicit feature composition. More recently, this architecture has been improved in several different ways (Dyer et al., 2015; Weiss et al., 2015; Zhou et al., 2015; Andor et al., 2016). Here, we employ the LSTM-based architecture enhanced with character bidirectional LSTMs (Ballesteros et al., 2015) for the following major reasons:

³Duong et al. (2015b) used *L2* regularizers to tie the lexical embeddings with a bilingual dictionary.

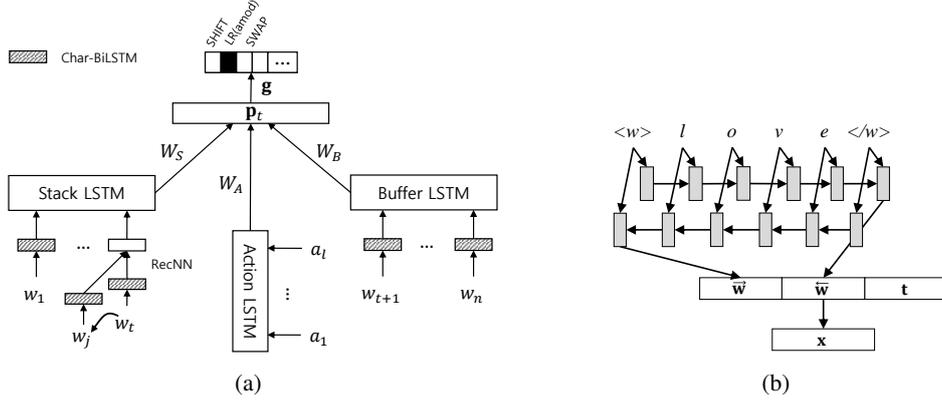


Figure 2: The LSTM-based neural parser (a) and the Char-BiLSTM for modeling words (b).

- Compared with Chen & Manning’s architecture, it makes full use of the non-local features by modeling the full history information of a *state* with stack LSTMs.
- By modeling words, stack, buffer and action sequence separately which indicate hierarchical abstractions of representations, we can control the information flow across tasks via parameter sharing with more flexibility (Section 3.2).

Besides, we did not use the earlier ISBN parsing model (Titov and Henderson, 2007) due to its lack of scalability to large vocabulary. Figure 2(a) illustrates the transition-based parsing architecture using LSTMs. Bidirectional LSTMs are used for modeling the word representations (Figure 2(b)), which we refer to as Char-BiLSTMs henceforth. Char-BiLSTMs learn features for each word, and then the representation of each token can be calculated as:

$$\mathbf{x} = \text{ReLU}(\mathbf{V}[\vec{\mathbf{w}}; \overleftarrow{\mathbf{w}}; \mathbf{t}] + \mathbf{b}) \quad (1)$$

where \mathbf{t} is the POS tag embedding. The token embeddings are then fed into subsequent LSTM layers to obtain representations of the *stack*, *buffer* and *action sequence* respectively referred to as \mathbf{s}_t , \mathbf{b}_t and \mathbf{a}_t (The subscript t represents the time step). Note that the subtrees within the stack and buffer are modeled with a *recursive neural network* (RecNN) as described in Dyer et al. (2015). Next, a linear mapping (\mathbf{W}) is applied to the concatenation of \mathbf{s}_t , \mathbf{b}_t and \mathbf{a}_t , and passed through a component-wise ReLU:

$$\mathbf{p}_t = \text{ReLU}(\mathbf{W}[\mathbf{s}_t; \mathbf{b}_t; \mathbf{a}_t] + \mathbf{d}) \quad (2)$$

Finally, the probability of next action $z \in \mathcal{A}(S, B)$ is estimated using a softmax function:

$$p(z|\mathbf{p}_t) = \frac{\exp(\mathbf{g}_z^\top \mathbf{p}_t + \mathbf{q}_z)}{\sum_{z' \in \mathcal{A}(S, B)} \exp(\mathbf{g}_{z'}^\top \mathbf{p}_t + \mathbf{q}_{z'})} \quad (3)$$

where $\mathcal{A}(S, B)$ represents the set of valid actions given the current content in the *stack* and *buffer*.

We apply the non-projective transition system originally introduced by Nivre (2009) since most of the treebanks we consider in this study has a noticeable proportion of non-projective trees. In the SWAP-based system, both the *stack* and *buffer* may contain tree fragments, so RecNN is applied both in S and B to obtain representations of each position.

3.2 Deep Multi-task Learning

Multi-task learning (MTL) is the procedure of inductive transfer that improves learning for one task by using the information contained in the training signals of other related tasks. It does this by learning tasks in parallel while using a shared representation. A good overview, especially focusing on neural networks, can be found in Caruana (1997).

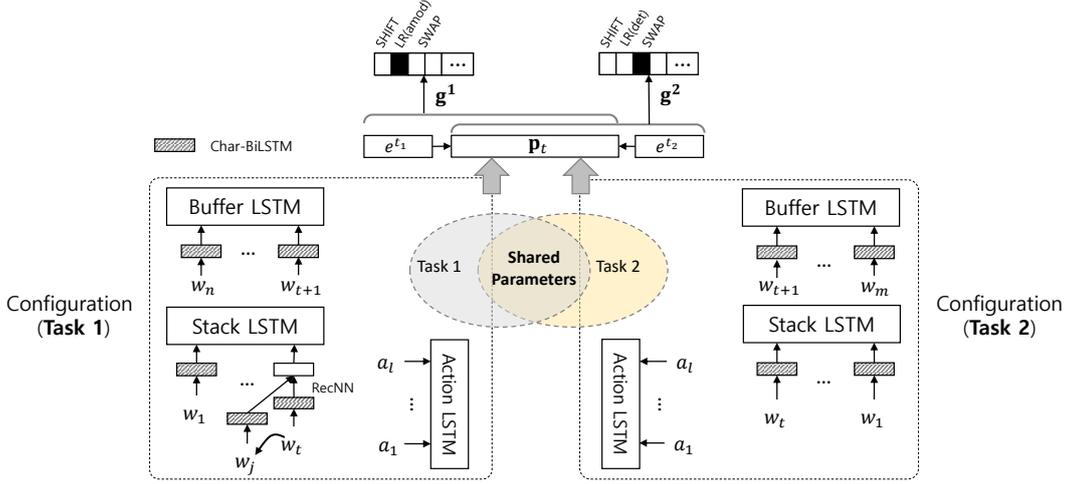


Figure 3: Illustration of the MTL framework.

We illustrate our multi-task learning architecture in Figure 3. As discussed in previous sections, multi-ple treebanks, either multilingual or monolingual heterogeneous, contain knowledge that can be mutually beneficial. We consider the target treebank processing as the *primary task*, and the source treebank as a *related task*. The two tasks are interacted through multi-level parameter sharing (Section 3.2.1). Inspired by Ammar et al. (2016), we introduce a task-specific vector e^t (*task embedding*) which is first combined with $s_t, \mathbf{b}_t, \mathbf{a}_t$ to compute \mathbf{p}_t , and then further concatenated with \mathbf{p}_t to compute the probability distribution of transition actions. Therefore, Eqn 2, 3 become:

$$\mathbf{p}_t = \text{ReLU}(\mathbf{W}[s_t; \mathbf{b}_t; \mathbf{a}_t; e^t] + \mathbf{d}) \quad (4)$$

$$p(z|\mathbf{p}_t) = \text{softmax}(\mathbf{g}_z^\top [\mathbf{p}_t; e^t] + \mathbf{q}_z) \quad (5)$$

The joint cross-entropy is used as the objective function. The key of *multi-task* learning is *parameter sharing*, without which the correlation between tasks will not be exploited. In this work, we design sophisticated parameter sharing strategies according to the linguistic similarities and differences between the tasks.

3.2.1 Parameter Sharing

Deep neural networks automatically learn features for a specific task with hierarchical abstractions, which gives us the flexibility to control parameter sharing in different levels accordingly.

In this study, different parameter sharing strategies are applied according to the source and target treebanks being used. We consider two different scenarios: MTL with multilingual universal treebanks as source (**MULTI-UNIV**) and MTL with monolingual heterogeneous treebanks as source (**MONO-HETERO**). Table 1 presents our parameter sharing strategies for each setting.

MULTI-UNIV Multilingual universal treebanks are annotated with the same set of POS tags (Petrov et al., 2012), dependency relations, and share the same set of transition actions. However, the vocabularies (word, characters) are language-specific. Therefore, it makes sense to share the lookup tables (embeddings) of POS tags (E_{pos}), relations (E_{rel}) and actions (E_{act}), but separate the character embeddings (E_{char}) as well as the Char-BiLSTMs (BiLSTM(chars)). Additionally, linguistic typologies such as the order of *subject-verb-object* and *adjective-noun* (Figure 1(a)) varies across languages, which result in the divergence of inherent grammars of transition action sequences. So we set the action LSTM (LSTM(A)) as *task-specific*.

MONO-HETERO Monolingual heterogeneous treebanks instead share the same lexical representations, but have different POS tags, structures and relations due to the different annotation schemes. Hence the transition actions set varies across treebanks. For simplicity reasons, we convert the language-specific

	MULTI-UNIV	MONO-HETERO
Shared	$E_{pos}, E_{rel}, E_{act}$ LSTM(S), LSTM(B), RecNN W_A, W_S, W_B	E_{pos}, E_{char} LSTM(S), LSTM(B), BiLSTM(chars), RecNN W_A, W_S, W_B
Task-specific	E_{char}, e^t LSTM(A), BiLSTM(chars) \mathbf{g}	E_{rel}, E_{act}, e^t LSTM(A) \mathbf{g}

Table 1: Parameter sharing strategies for **MULTI-UNIV** and **MONO-HETERO**. LSTM(S) – *stack* LSTM; LSTM(B) – *buffer* LSTM; LSTM(A) – *action* LSTM; BiLSTM(chars) – Char-BiLSTM; RecNN – recursive NN modeling the subtrees; W_A, W_S, W_B – weights from A, S, B to the state (\mathbf{p}_t); \mathbf{g} – weights from the state to output layer; E – embeddings.

POS tags of the heterogeneous treebanks into universal POS tags (Petrov et al., 2012). Consequently, E_{char} and BiLSTM(chars), E_{pos} are shared across tasks, but $E_{rel}, E_{act}, \text{LSTM(A)}$ are *task-specific*.

Besides, the LSTM parameters for modeling the *stack* and *buffer* (LSTM(S), LSTM(B)), the RecNN for modeling tree compositions, and the weights from S, B, A to the state \mathbf{p}_t (W_A, W_B, W_S) are shared for both MULTI-UNIV and MONO-HETERO. As standard in multi-task learning, the weights at the output layer (\mathbf{g}) are *task-specific* in both settings.

3.2.2 Learning

Training is achieved in a stochastic manner by looping over the tasks:

1. Randomly select a task.
2. Select a sentence from the task, and generate instances for classification.
3. Update the corresponding parameters by back-propagation w.r.t. the instances.
4. Go to 1.

We adopt the development data of the target treebank (primary task) for early-stopping.

4 Experiments

We first describe the data and settings in our experiments, then the results and analysis.

4.1 Data and Settings

We conduct experiments on UDT v2.0⁴ and the CoNLL-X shared task data. For monolingual heterogeneous source, we also experiment on CTB5 using CDT as the source treebank, to compare with the previous work of Li et al. (2012). Statistics of the datasets are summarized in Table 2. We investigate the following experiment settings:

- **MULTILINGUAL (UNIV→UNIV)**. In this setting, we study the integration of multilingual universal treebanks. Specifically, we consider the DE, ES, FR, PT, IT and SV universal treebanks as target treebanks, and the EN treebank as the common source treebank.
- **MONOLINGUAL (CONLL↔UNIV)**. Here we study the integration of monolingual heterogeneous treebanks. The CoNLL-X corporas (DE, ES, PT, SV) and the UDT treebank of corresponding languages are used as source and target treebanks mutually.
- **MONOLINGUAL (CDT→CTB5)**. We follow the same settings of Li et al. (2012), and consider two scenarios using automatic POS tags and gold-standard POS tags respectively.

⁴<https://github.com/ryanmcd/uni-dep-tb>

	Train	Dev	Test	Train	Dev	Test
	UDT			CoNLL-X		
EN	39,832	1,700	2,416	–	–	–
DE	14,118	800	1,000	35,295	3,921	357
ES	14,138	1,569	300	2,976	330	206
FR	14,511	1,611	300	–	–	–
PT	9,600	1,200	1,198	8,164	907	288
IT	6,389	400	400	–	–	–
SV	4,447	493	1,219	9,938	1,104	389
	CDT			CTB5		
ZH	55,500	1,500	3,000	16,091	803	1,910

Table 2: Statics of UDT v2.0 and CoNLL-X treebanks (with languages presented in UDT v2.0).

	MULTILINGUAL (UNIV → UNIV)							
	SUP		CAS _{EN}		SMTL _{EN}		MTL _{EN}	
	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS
DE	84.24	78.40	84.24	78.65	84.37	79.07	84.93	79.34
ES	85.31	81.23	85.42	81.42	85.78	81.54	86.78	82.92
FR	85.55	81.13	84.57	80.14	86.13	81.77	86.44	82.01
PT	88.40	86.54	88.88	87.07	89.08	87.24	89.24	87.50
IT	86.53	83.72	86.58	83.67	86.53	83.64	87.26	84.27
SV	84.91	79.88	86.43	81.92	86.79	82.31	85.98	81.35
AVG	85.82	81.82	86.02	82.15	86.45	82.60	86.77	82.90

Table 3: Parsing accuracies of MULTILINGUAL (UNIV→UNIV). Significance tests with MaltEval yield p-values < 0.01 for (MTL vs. SUP) on all languages.

4.2 Baseline Systems

We compare our approach with the following baseline systems.

- Monolingual supervised training (SUP). Models are trained only on the target treebank, with the LSTM-based parser.
- Cascaded training (CAS). This system has two stages. First, models are trained using the source treebank. Then the parameters are used to initialize the neural network for training target parsers. Similar approach was studied in Duong et al. (2015a) and Guo et al. (2016) for low-resource parsing.

For MULTILINGUAL (UNIV→UNIV), we also compare with the *shallow multi-task learning* (SMTL) system, as described in Section 2, which is representative of the approach of Duong et al. (2015b) and Ammar et al. (2016). In SMTL all the parameters are shared except the character embeddings (E_{char}), and task embeddings (e^t) are not used. Unlike Duong et al. (2015b) and Ammar et al. (2016), we don’t use external resources such as cross-lingual word clusters, embeddings and dictionaries which is beyond the scope of this work.

4.3 Results

In this section, we present empirical evaluations under different settings.

4.3.1 Multilingual Universal Source Treebanks

Table 3 shows the results under the MULTILINGUAL (UNIV→UNIV) setting. CAS yields slightly better performance than SUP, especially for SV (+1.52% UAS and +2.04% LAS), indicating that *pre-training* with EN training data indeed provides a better initialization of the parameters for cascaded training. SMTL in turn outperforms CAS overall (comparable for IT), which implies that training two treebanks jointly helps even with a unique model.

	DE	ES	FR
SUP	58.93	61.99	60.45
CAS	64.08	70.45	68.72
SMTL	63.57	69.01	65.04
+ <i>weighted sampling</i>	<i>63.50</i>	<i>70.17</i>	<i>68.52</i>
MTL	62.43	66.67	64.23
+ <i>weighted sampling</i>	64.22	<i>68.42</i>	<i>66.67</i>
Duong et al.	61.2	69.1	65.3
Duong et al. + Dict	61.8	70.5	67.2

Table 4: Low resource setup (3K tokens), evaluated with LAS.

Furthermore, with appropriate parameter sharing, our deep multi-task learning approach (MTL) outperforms SUP overall and achieves the best performances in five out of six languages. An exception is Swedish. As we can see, both CAS and SMTL outperforms MTL by a significant margin for SV. The underlying reasons we suggest are two-fold.

1. SV morphology is similar to EN with less inflections, encouraging the morphology-related parameters like BiLSTM(chars) to be shared.
2. SV has a much smaller treebank compared with EN (1:9). We suggest that SMTL and CAS work better than MTL in low resource setting. To our intuition, since knowledge contained in the low-resource target treebank is very limited, it is reasonable for us to put more emphasis on the source treebank through SMTL or CAS.

To verify the first issue, we conduct tests on SMTL without sharing Char-BiLSTMs, and observe significant degradation in performance (-0.73 in UAS and -0.81 in LAS). This observation also suggests that MTL has the potential to reach higher performances through *language-specific tuning* of parameter sharing strategies.

To verify the second issue, we consider a low resource setup following Duong et al. (2015b), where the target language has a small treebank (3K tokens). We train our models on identical sampled dataset shared by the authors on DE, ES and FR. As we can find in Table 4, while all the models outperform SUP, both CAS and SMTL work better than MTL, which confirms our assumption. Although not the primary focus of this work, we find that SMTL and MTL can be significantly improved in low resource setting through weighted sampling of tasks during training. Specifically, in the training procedure (Section 3.2.2), we sample from the source language (EN) which has a much richer treebank with larger probability of 0.9, while sample from the target language with probability of 0.1. In this way, the two tasks are encouraged to converge at a similar rate. As shown in Table 4, both SMTL and MTL benefit from weighted task sampling.

4.3.2 Monolingual Heterogeneous Source Treebanks

Among the four languages here, the SV universal treebank is mainly converted from the Talbanken part of the Swedish bank (Nivre and Megyesi, 2007), thus has a large overlap with the CONLL-X Swedish treebank. Therefore, we exclude the sentences in SV test set that appear in the source treebank for evaluation. Table 5 shows the results of MONOLINGUAL (CONLL \leftrightarrow UNIV). Overall MTL systems outperforms the supervised baselines by significant margins in both conditions, showing the mutual benefits of UDT and CONLL-X treebanks.⁵

To show the merit of our approach against previous approaches, we further conduct experiments on CTB5 using CDT as heterogeneous source treebank (Table 2). For CTB5, we follow (Li et al., 2012) and consider two scenarios which use automatic POS tags and gold-standard POS tags respectively. To compare with their results, we run SUP, CAS and MTL on CTB5. Table 6 presents the results.

⁵An exception is PT in MONOLINGUAL (UNIV \rightarrow CONLL). This may be due to the low quality of the PT universal treebank caused by the automatic construction process. We discussed and verified this with the author of UDT v2.0.

	MONOLINGUAL (CONLL→UNIV)						MONOLINGUAL (UNIV→CONLL)					
	SUP		CAS		MTL		SUP		CAS		MTL	
	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS
DE	84.24	78.40	85.02	80.05	85.73	80.64	89.06	86.48	89.64	86.66	89.98	87.50
ES	85.31	81.23	85.90	81.73	85.80	81.45	85.41	80.50	86.46	81.37	86.07	81.41
PT	88.40	86.54	89.12	87.32	89.40	87.60	90.16	85.53	89.50	85.03	89.98	85.23
SV	82.61	77.42	85.39	80.60	85.29	81.22	79.61	72.71	82.91	74.96	84.86	77.36
AVG	<u>85.14</u>	<u>80.90</u>	<u>86.35</u>	<u>82.43</u>	<u>86.56</u>	<u>82.73</u>	<u>86.06</u>	<u>81.31</u>	<u>87.13</u>	<u>82.01</u>	<u>87.72</u>	<u>82.88</u>

Table 5: MONOLINGUAL (CONLL↔UNIV) performance.

		Auto-POS			Gold-POS		
		SUP	CAS	MTL	SUP	CAS	MTL
		OURS	UAS	79.34	80.25 (+0.91)	81.13 (+1.79)	85.25
	LAS	76.23	77.26 (+1.03)	78.24 (+2.01)	83.59	84.72 (+1.13)	85.18 (+1.59)
		SUP	with QG		SUP	with QG	
LI12-O2	UAS	79.67	81.04 (+1.37)		86.13	86.44 (+0.31)	
LI12-O2SIB	UAS	79.25	80.45 (+1.20)		85.63	86.17 (+0.54)	

Table 6: Parsing accuracy comparisons of MONOLINGUAL (CDT→CTB5). LI12-O2 use the O2 graph-based parser with both sibling and grandparent structures, while LI12-O2SIB only use the sibling parts.

The indirect comparison indicates that our approach can achieve larger improvement than their method in both scenarios. Beside the empirical comparison, our method has the additional advantages in its scalability to multi-typed source treebanks without the painful human efforts of feature design.

5 Conclusion

This paper propose a universal framework based on deep multi-task learning that can integrate arbitrary-typed source treebanks to enhance the parsing models on target treebanks. We study two scenarios, respectively using multilingual universal source treebanks and monolingual heterogeneous source treebanks, and design effective parameter sharing strategies for each scenario.

We conduct extensive experiments on benchmark treebanks in various languages. Results demonstrate that our approach significantly improves over baseline systems under various experiment settings.

Acknowledgments

We thank Ryan McDonald for helpful discussions, and thank Dr. Zhenghua Li for sharing the processed CTB and CDT dataset. We also thank the anonymous reviewers for their insightful comments and suggestions. This work was supported by the National Key Basic Research Program of China via grant 2014CB340503 and the National Natural Science Foundation of China (NSFC) via grant 61300113 and 61370164.

References

- Waleed Ammar, George Mulcaire, Miguel Ballesteros, Chris Dyer, and Noah A Smith. 2016. One parser, many languages. *TACL*, 4:431–444.
- Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *JMLR*, 6:1817–1853, December.
- Daniel Andor, Chris Alberti, David Weiss, Aliaksei Severyn, Alessandro Presta, Kuzman Ganchev, Slav Petrov, and Michael Collins. 2016. Globally normalized transition-based neural networks. In *Proc. of the 54th ACL*, pages 2442–2452, Berlin, Germany, August.

- Miguel Ballesteros, Chris Dyer, and Noah A. Smith. 2015. Improved transition-based parsing by modeling characters instead of words with lstms. In *Proc. of EMNLP*, pages 349–359, September.
- Bernd Bohnet and Joakim Nivre. 2012. A transition-based system for joint part-of-speech tagging and labeled non-projective dependency parsing. In *Proc. of EMNLP-CoNLL*, pages 1455–1465, July.
- Sabine Buchholz and Erwin Marsi. 2006. Conll-x shared task on multilingual dependency parsing. In *Proc. of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*, pages 149–164, June.
- David Burkett and Dan Klein. 2008. Two languages are better than one (for syntactic parsing). In *Proc. of the 2008 Conference on EMNLP*, pages 877–886, October.
- Rich Caruana. 1997. Multitask learning. *Machine learning*, 28(1):41–75.
- Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proc. of EMNLP*, pages 740–750, October.
- Wenliang Chen, Jun’ichi Kazama, and Kentaro Torisawa. 2010. Bitext dependency parsing with bilingual subtree constraints. In *Proc. of the 48th ACL*, pages 21–29, July.
- Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proc. of the 25th ICML*, pages 160–167.
- Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proc. of the 53rd ACL and the 7th IJCNLP (Volume 1: Long Papers)*, pages 1723–1732, July.
- Long Duong, Trevor Cohn, Steven Bird, and Paul Cook. 2015a. Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser. In *Proc. of ACL-IJCNLP*, pages 845–850, July.
- Long Duong, Trevor Cohn, Steven Bird, and Paul Cook. 2015b. A neural network model for low-resource universal dependency parsing. In *Proc. of the 2015 Conference on EMNLP*, pages 339–348, September.
- Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. 2015. Transition-based dependency parsing with stack long short-term memory. In *Proc. of ACL-IJCNLP*, pages 334–343, July.
- Jiang Guo, Wanxiang Che, David Yarowsky, Haifeng Wang, and Ting Liu. 2015. Cross-lingual dependency parsing based on distributed representations. In *Proc. of ACL-IJCNLP*, pages 1234–1244, July.
- Jiang Guo, Wanxiang Che, David Yarowsky, Haifeng Wang, and Ting Liu. 2016. A representation learning framework for multi-source transfer parsing. In *Proc. of AAAI*, February.
- Jun Hatori, Takuya Matsuzaki, Yusuke Miyao, and Jun’ichi Tsujii. 2012. Incremental joint approach to word segmentation, pos tagging, and dependency parsing in chinese. In *Proc. of ACL*, pages 1045–1053, July.
- James Henderson and Ivan Titov. 2010. Incremental sigmoid belief networks for grammar learning. *J. Mach. Learn. Res.*, 11:3541–3570, December.
- James Henderson, Paola Merlo, Ivan Titov, and Gabriele Musillo. 2013. Multilingual joint parsing of syntactic and semantic dependencies with a latent variable model. *Computational Linguistics*, 39(4):949–998.
- Liang Huang, Wenbin Jiang, and Qun Liu. 2009. Bilingually-constrained (monolingual) shift-reduce parsing. In *Proc. of the 2009 Conference on EMNLP*, pages 1222–1231, August.
- Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. 2005. Bootstrapping parsers via syntactic projection across parallel texts. *Natural language engineering*, 11(03):311–325.
- Richard Johansson. 2013. Training parsers on incompatible treebanks. In *Proc. of NAACL*, pages 127–137, June.
- Zhenghua Li, Min Zhang, Wanxiang Che, Ting Liu, Wenliang Chen, and Haizhou Li. 2011. Joint models for chinese pos tagging and dependency parsing. In *Proc. of EMNLP*, pages 1180–1191, July.
- Zhenghua Li, Ting Liu, and Wanxiang Che. 2012. Exploiting multiple treebanks for parsing with quasi-synchronous grammars. In *Proc. of the 50th ACL (Volume 1: Long Papers)*, pages 675–684, July.
- Minh-Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2015. Multi-task sequence to sequence learning. *CoRR*, abs/1511.06114.
- Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330.

- Ryan McDonald, Slav Petrov, and Keith Hall. 2011. Multi-source transfer of delexicalized dependency parsers. In *Proc. of the 2011 Conference on EMNLP*, pages 62–72, July.
- Ryan McDonald, Joakim Nivre, Yvonne Quirnbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, and Jungmee Lee. 2013. Universal dependency annotation for multilingual parsing. In *Proc. of ACL*, pages 92–97, August.
- Zheng-Yu Niu, Haifeng Wang, and Hua Wu. 2009. Exploiting heterogeneous treebanks for parsing. In *Proc. of the Joint Conference of the 47th ACL and the 4th IJCNLP of the AFNLP*, pages 46–54, August.
- Joakim Nivre and Ryan McDonald. 2008. Integrating graph-based and transition-based dependency parsers. In *Proc. of ACL-08: HLT*, pages 950–958, June.
- Joakim Nivre and Beata Megyesi. 2007. Bootstrapping a swedish treebank using cross-corpus harmonization and annotation projection. In *Proc. of the 6th International Workshop on Treebanks and Linguistic Theories*, pages 97–102.
- Joakim Nivre. 2009. Non-projective dependency parsing in expected linear time. In *Proc. of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP*, pages 351–359, August.
- Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proc. of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 2089–2096, May.
- Mohammad Sadegh Rasooli and Michael Collins. 2015. Density-driven cross-lingual transfer of dependency parsers. In *Proc. of EMNLP*, pages 328–338, September.
- David A. Smith and Jason Eisner. 2009. Parser adaptation and projection with quasi-synchronous grammar features. In *Proc. of the 2009 Conference on EMNLP*, pages 822–831, August.
- Oscar Täckström, Ryan McDonald, and Jakob Uszkoreit. 2012. Cross-lingual word clusters for direct transfer of linguistic structure. In *Proc. of NAACL: HLT*, pages 477–487, June.
- Jörg Tiedemann. 2014. Rediscovering annotation projection for cross-lingual parser induction. In *Proc. of COLING 2014*, pages 1854–1864, August.
- Ivan Titov and James Henderson. 2007. Fast and robust multilingual dependency parsing with a generative latent variable model. In *Proc. of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 947–951, June.
- André Filipe Torres Martins, Dipanjan Das, Noah A. Smith, and Eric P. Xing. 2008. Stacking dependency parsers. In *Proc. of the 2008 Conference on EMNLP*, pages 157–166, October.
- David Vilares, Carlos Gómez-Rodríguez, and Miguel A. Alonso. 2016. One model, two languages: training bilingual parsers with harmonized treebanks. In *Proc. of the 54th ACL*, pages 425–431, August.
- David Weiss, Chris Alberti, Michael Collins, and Slav Petrov. 2015. Structured training for neural network transition-based parsing. In *Proc. of ACL-IJCNLP*, pages 323–333, July.
- Yuan Zhang and Regina Barzilay. 2015. Hierarchical low-rank tensors for multilingual transfer parsing. In *Proc. of the 2015 Conference on EMNLP*, pages 1857–1867, September.
- Yue Zhang and Stephen Clark. 2008. A tale of two parsers: Investigating and combining graph-based and transition-based dependency parsing. In *Proc. of the 2008 Conference on EMNLP*, pages 562–571, October.
- Meishan Zhang, Wanxiang Che, Yanqiu Shao, and Ting Liu. 2014. Jointly or separately: Which is better for parsing heterogeneous dependencies? In *Proc. of COLING 2014*, pages 530–540, August.
- Hao Zhou, Yue Zhang, Shujian Huang, and Jiajun Chen. 2015. A neural probabilistic structured-prediction model for transition-based dependency parsing. In *Proc. of ACL-IJCNLP*, pages 1213–1222, July.