A survey of syntactic-semantic parsing based on constituent and dependency structures

ZHANG MeiShan*

School of New Media and Communication, Tianjin University, Tianjin 300072, China

Syntactic and semantic parsing has been investigated for decades, which is one primary topic in the natural language processing community. This article aims for a brief survey on this topic. The parsing community includes many tasks, which are difficult to be covered fully. Here we focus on two of the most popular formalizations of parsing: constituent parsing and dependency parsing. Constituent parsing is majorly targeted to syntactic analysis, and dependency parsing can handle both syntactic and semantic analysis. This article briefly reviews the representative models of constituent parsing and dependency parsing, and also dependency graph parsing with rich semantics. Besides, we also review the closely-related topics such as cross-domain, cross-lingual and joint parsing models, parser application as well as corpus development of parsing in the article.

syntax parsing, semantic parsing, constituent parsing, dependency parsing, semantic graph parsing

1 Introduction

Sentence-level syntactic and semantic parsing is one major topic in the natural language processing (NLP) community, which aims to uncover the internal structural relations in sentences [1–4]. On the one hand, from the view of linguistics, the goal of parsing is to disclose how words are combined to form sentences and the rules that govern the formation of sentences. On the other hand, from the view of NLP applications, parsing can be beneficial for a number of tasks, such as machine translation, question answering, information extraction, sentiment analysis and generation [5–7], and the performance of parsing matters greatly.

Parsing has been extensively studied for decades. The goal of syntactic parsing is to derive the syntax information in sentences, such as the subjects, objects, modifiers and topics. There have been a number of achievements for the task, and large-scale corpora for a range of languages have been already available. Compared with syntactic parsing, semantic parsing is much more difficult due to the complex structure of various semantics such as predicate-argument, and it is also a long-range goal of NLP. With the recent advance in data-driven machine learning models, semantic parsing has received increasing interests, especially under the neural setting. Several datasets based on certain formalizations have been developed to facilitate research.

Parsing often relies on specific grammars, which are used to refine the output structures of syntax and semantics. There are many sophisticated grammars for accurately expressing the syntactic and semantic information at the sentence-level. In this paper, we focus on two popular grammars which are concerned mostly. Context-free grammar (CFG), well known as constituent parsing (or phrase-structure parsing [4] thus, also as constituent grammar or phrase-structure grammar), adopts hierarchal phrase-structural trees to organize sentence-level syntactic information, which has been re-

---

*Corresponding author (email: mason.zms@gmail.com)

searched intensively since very early. Dependency grammar is another widely-adopted grammar for syntactic and semantic parsing, where words are directly connected by dependency links, with labels indicating their syntactic or semantic relations [2]. Because of the conciseness and easy annotation of dependency structures, dependency parsing has received more attention than constituent parsing.

Besides, there are many other great grammars. The representative topics include combinatory categorial grammar (CCG), head-driven phrase structure grammar (HPSG), lexical functional grammar (LFG), abstract meaning representation (AMR), minimal recursion semantics (MRS), universal conceptual cognitive annotation (UCCA) and also several logic-targeted formalizations. All these categories have been researched for a long time and in particular several of which are now quickly developed because of the powerfulness of neural networks as well as pretrained contextualized word representations. However, this article leaves these studies for future more comprehensive surveys.

Here we make a brief survey for syntactic and semantic parsing based on constituent grammar and bi-lexicalized dependency grammar. In Sects. 2 and 3, we review the studies of constituent parsing and dependency parsing, respectively, where the dependency parsing is based tree structure and specifically targeted to syntax. We further investigate semantic-oriented dependency graph parsing in Sect. 4. Sects. 5 and 6 review cross-domain and cross-lingual parsing, which is one hot direction. Sect. 7 reviews the joint models which are targeted to parsing as the final goal, while Sect. 8 reviews the parser application strategies, where parsers are evaluated on downstream applications. Sect. 9 introduces the related treebank work, which serves the major training corpus for various parsers as well as for parser model evaluations. Finally, in Sect. 10, the conclusion and future work are summarized.

2   Constituent parsing

Constituent parsing is one fundamental task for syntax parsing, which has received great interest for decades [1, 3, 4]. Figure 1 shows an example constituent tree, where nodes in the constituent tree are constituent spans, also known as phrases. The goal of constituent parsing is to uncover these phrases as well as their relations. The standard evaluation method of constituent parsers is based on recognition of the phrases, where precision, recall and the F1-measure scores are adopted as the major metrics.

The mainstream approaches of constituent parsing include the chart-based and the transition-based models. Current neural models have achieved state-of-the-art performances under

both two kinds of methods. In fact, neural constituent parsing starts very early before the prosperity of deep learning [8]. In this section, first we introduce the chart-based and transition-based constituent models, and then show several other models out of the two categories. Here before the detailed introduction, we show an overall picture of the performances of various representative constituent parsers in Table 1, where ensemble models are excluded for fair comparisons.

2.1   Chart-based parsing

2.1.1   Statistical models

Early successful constituent parsing models exploit the productive CFG rules to guide the generation of constituent trees. The chart parsing algorithms are exploited universally for decoding, and most of the effort is focused on the refinement of CFG rules, which serve as the major sources of parameter estimation. Refs. [9, 10] extended probabilistic context-free grammar (PCFG) with head lexicalization, associating PCFG rules with head words, which can effectively boost the PCFG parsing performance. Unlexicalized models have also received great attention, by using fine-grained structural annotation [32] or automatic latent variables [12] to enrich PCFG rules, leading to comparable or even better performance than lexicalized models.

The above models suffer the difficulty of integrating nonlocal features since future decisions are invisible during decoding which is critical for global inference. Condition random field (CRF) is one way for global modeling. Ref. [13] proposed a strong constituent parsing model by adapting the standard n-gram CRF models for CFG, and meanwhile presenting rich sophisticated features. The dependencies among adjacent CFG rules can be modeled, which are used for global inference.

2.1.2   Neural models

Ref. [33] is the first work to define scores over phrases by
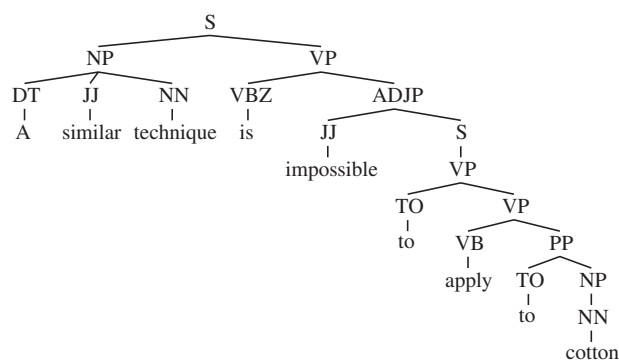
Figure 1   An example of constituent tree.

Table 1 A comparison of representative constituent parsing models, where phrase-level F1 scores are reported, PTB and CTB are two benchmark datasets for the English and Chinese parsing, respectively

Model	Main features	PTB	CTB
Chart-based, statistical models			
Ref. [9]	Head-lexicalization	88.2	N/A
Ref. [10]	Max-entropy	89.5	80.8
Ref. [11]	Self-training	**92.3**	N/A
Ref. [12]	PCFG	90.1	**83.3**
Ref. [13]	CRF	89.9	N/A
Transition-based, statistical models			
Ref. [14]	Greedy	86.0	N/A
Ref. [15]	Global learning, beam	**91.3**	**85.6**
Chart-based, neural models			
Ref. [16]	Recursive NN	90.4	N/A
Ref. [17]	CNN	91.1	N/A
Ref. [18]	LSTM, span	91.8	N/A
Ref. [19]	Self-attentive	93.5	N/A
Ref. [19]	+ELMo	**95.1**	N/A
Transition-based, neural models			
Ref. [20]	Neural+discrete	90.7	**86.6**
Ref. [21]	Global learning, beam	90.7	N/A
Ref. [22]	Language modelling	92.4	82.7
Ref. [23]	Dynamic oracle	91.3	N/A
Ref. [24]	In-order	91.8	86.1
Ref. [25]	Policy gradient	92.6	86.0
Ref. [26]	Policy gradient	**95.4**	86.0
Other methods (report neural models only)			
Ref. [27]	Distance to tree	91.8	86.5
Ref. [28]	Local classification	92.7	87.3
Ref. [29]	Sequence labeling	91.1	85.6
Ref. [30]	HPSG grammar	**96.3**	**92.2**
Ref. [31]	HPSG, improved attention	**96.3**	**N/A**

recursive neural networks. The CFG-based constituent trees can be naturally modeled in this way. Neural CRF parsing is accordingly proposed by ref. [17], which can be regarded as a neural enhancing of ref. [13]. The work simply uses feed-forward neural networks to encode atomic features instead of human composition. Notice that it is different from ref. [33] as no recursive composition is used here.

Ref. [18] proposed state-of-the-art chart-based neural models. On the one hand, they use deep bidirectional long-short term memory (LSTM) neural networks to enhance sentence representations, designing sophisticated strategies for span representation. On the other hand, they also adopt top-down incremental parsing for decoding, which dilutes the differences between chart-based and transition-based approaches. Their results are very strong on par with the state-of-the-art transition-based methods at the same time. The work is further followed by ref. [34] with extensive analysis

and ref. [19] with a self-attentive encoder. In particular, ref. [19] exploited contextualized word representation including ELMo [35] and BERT [36], leading to almost the best parsing performance in the literature.

2.2 Transition-based parsing

2.2.1 Statistical models

The transition-based models demonstrate highly promising for constituent parsing [14, 37]. The key idea is to define a transition system with transition states and actions, where states denote partial parsing outputs, and actions specify next-step state-transition operations. Transition actions indicate the incremental tree construction process. For constituent parsing, typical actions include the *shift* to building terminal nodes, the *unary* to building unary nodes, and the *binary* to building binary nodes. The details can be referred to as ref. [14]. The model is also commonly referred to as the shift-reduce model, where *unary* and *binary* are actions of reduction. By converting constituent parsing into predicting a sequence of transition actions, discriminant classifiers such as max-entropy and support vector machine (SVM) can be applied for the prediction, with rich manually-crafted features.

The initial shift-reduce model classifies the sequence of actions for a single constituent tree separately, greedily searching for the best output constituent tree, which may suffer the error propagation problem since the early step errors can affect later predictions. To this end, globally modeling with beam search is proposed to alleviate the problem, which decodes the total sequence of actions for a full constituent tree as a whole [15, 38]. The discriminative perceptron-style online learning greatly promotes this line of work [39], which enables legal parameter optimizations towards inexact search. For feature engineering, the contextual lexicalized words, POS tags, distances and their compositions are all extensively investigated, which can be referred to ref. [15] for details.

2.2.2 Neural models

Refs. [20, 21] could be the direct extensions of ref. [15] by using neural networks. The composition of atomic features is achieved by feed-forward neural networks. Ref. [40] found that the greedy style decoding can also achieve highly competitive performance when a deep LSTM encoder is exploited. Then, several studies suggest dynamic oracles to optimize greedy constituent parsers [23, 41]. The main idea is to let models make optimum decisions when facing erroneous transition states [42]. A proportion of training instances with erroneous transition states and their oracle actions are added into the original training corpus.

There have been several studies exploiting different transition strategies. Ref. [22] suggested the recurrent neural work grammar (RNNG), which is a top-down transition-based system. Ref. [24] designed an in-order transition system to make a compromise between top-down and bottom-up transitions. Ref. [43] presented a novel system with an additional GAP action for discontinuous constituency parsing, and they also find that unlexicalized models give better performance. Ref. [44] optimized the transition actions to facilitate the construction of non-binarized constituent nodes, avoiding the preprocessing of binarization for constituent trees. Ref. [26] suggested the tetra-tagging system, which combines sequence tagging and transition action classification. The system achieves state-of-the-art performance on the benchmark PTB dataset with BERT representations.

2.3   Other frameworks

Neural networks such as deep LSTM and multi-head self-attention are capable of encoding global features implicitly into their final representations, which weakens the role of decoding as a source of feature induction. Based on the observation, several studies attempt to use simple frameworks, aiming for a wide community for parsing.

One representative attempt is to exploit neural sequence-to-sequence models for structural constituent parsing [45,46]. The key idea is to first linearize a phrase-structural constituent tree into a sequence of symbols by certain traversing strategies, and then directly feed the pair of input words and output symbols into a standard sequence-to-sequence model. These models require large-scale corpora for training, which could be obtained by auto-parsed high-confidence constituent trees from other state-of-the-art parsers.

Neural sequence labeling models have also been investigated for constituent parsing [47]. Ref. [47] proposed the first work of this line, which exploits the lowest common ancestor between adjacent words as clues to encode the word roles. Ref. [48] extended the work by language modeling and enhance parsing with pretraining. Further, more direct schemes have been proposed with local modeling for constituent parsing. Ref. [27] directly predicted the distance of constituent phrases and then decode greedily in a top-down way for a full constituent tree. Similarly, ref. [28] proposed two models based on local span prediction, achieving highly competitive performance on par with transition-based models. Recently, ref. [30] presented to exploit the HPSG-based grammar for constituent parsing, and further power the model with XLNet word representations [49], achieving the top performances for both CTB and PTB datasets. Ref. [31] revised the multi-head self-attention mechanism in ref. [30], leading to a similar performance with a smaller number of layers.

2.4   Semi-supervised models

The semi-supervised architecture aims to enhance a supervised model by statistical information extracted from raw text. Ref. [11] presented the first work which achieves improved performance for constituent parsing by self-training, and ref. [50] studied self-training empirically to show the conditions of usefulness. Ref. [51] exploited unsupervised word clusters learned from raw text to enhance constituent parsing. While recent studies shift to the neural network setting, the borderline between semi-supervised and supervised is becoming vague, as pretraining from raw text is one critical for the successfulness of neural models.

2.5   Model ensemble

The model ensemble is one effective way to boost the performance of constituent parsing. Initial work focuses on the output reranking [52, 53]. We can take either the k-best outputs of a constituent parser or one-best outputs from heterogeneous parsers as the inputs, and then build a new constituent tree by using a feature-rich reranking model. Benefiting from sophisticated manually-crafted non-local features, the framework can improve the parser performance significantly. However, related studies under the neural setting have received much less concern, which can be potentially due to that the majority of state-of-the-art neural models exploit the same sentence encoders, indicating that features are resemble in different kinds of models, and meanwhile homogeneous ensemble (e.g., different random seeds) by simply voting can achieve unsurpassable performances.

3   Dependency parsing

Dependency parsing is developed for syntax and semantic analysis by using bilexicalized dependency grammar, where all syntactic and semantic phenomena are represented by bilexicalized dependencies [2]. Figure 2 shows an example tree of dependency parsing. For the evaluation of various dependency parsers, dependency accuracy is used as the major metric, in terms of the unlabeled attachment score (UAS) and the labeled attachment score (LAS). In the early stage, dependency parsing is constrained to trees, projective or non-projective [54]. Recently, several studies have devoted to dependency parsing over graphs [55]. On the one hand, initial
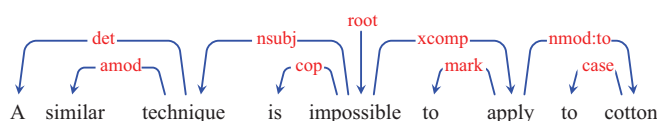
Figure 2   (Color online) An example of dependency tree.

dependency trees are mostly syntactic oriented, while recently there are growing interests focusing on semantic relations between words [54, 55]. This section mainly focuses on dependency tree parsing, while dependency graph parsing will be discussed in the next section.

The majority of dependency parsing models can be divided into two types, graph-based and transition-based [56], both of which have been extensively investigated under the traditional statistical setting [57–60] and the neural setting [56]. There also exist other interesting approaches for dependency parsing outside the two categories [61]. Table 2 shows an overall picture of the performances of several representative dependency parsers, and all ensemble models are excluded in this table. The graph-based and transition-based models are almost comparable (graph-based models are slightly higher) across both traditional statistical and neural settings, and other types of models achieve good performance with the support of sophisticated neural networks. Currently, neural models achieve state-of-the-art performances for dependency parsing [86].

3.1 Graph-based parsing

3.1.1 Statistical models

Graph-based methods exploit the maximum spanning tree (MST) algorithm for decoding, which decomposes a full dependency tree into small factors, such as dependency edges, and scores the full tree by summing the scores of all the included factors. The score of each factor can be calculated independently by the features extracted from it. The models by using dependency edges as the basic scoring factor are referred to as first-order models, where the order indicates the maximum number of edges in a factor. Ref. [57] proposed a feature-rich first-order MST parser based on discriminative max-margin training.

Later, higher-order MST parsers have been studied. With larger factors, the parsing models can exploit more sophisticated features, and thus can potentially bring improved performance. Second order MST parsing models have studied extensively [58, 62, 87, 88], where the newly added features include the relations from parent-sibling and parent-child-grandchild factors. Notice that higher-order MST decoding can have higher time complexity (i.e., from $O(n^3)$ to $O(n^4)$), which may lead to intolerable parsing speed. The problem could be largely alleviated by ref. [88] with feature hashing. Ref. [66] proposed an efficient third-order dependency parsing model, which adds grand-sibling and tri-sibling features into the model. Ref. [89] exploited low-rank tensor to alleviate the burden of feature engineering. Fourth-order dependency parsing has been investigated by ref. [67]. As a whole,

second-order and third-order parsers could be good choices considering both performance and efficiency.

3.1.2 Neural models

Ref. [71] presented a graph-based neural model by embedding all discrete atomic features in the traditional statistical MST models and then composing these embeddings with a similar feed-forward network of ref. [78]. Convolution neural network is then applied for neural feature composition in

Table 2 A comparison of representative dependency parsing models, where UAS are reported, PTB and CTB5.1 (CTB in the table for short) are two benchmark datasets for the English and Chinese parsing, respectively

Model	Main features	PTB	CTB
Graph-based, statistical models			
Ref. [57]	1-order	90.9	83.0
Ref. [62]	2-order	91.5	85.2
Ref. [63]	Word clusters	93.2	N/A
Ref. [64]	Auto subtrees	93.2	86.7
Ref. [65]	Feature hashing	92.9	N/A
Ref. [66]	3-order	93.0	86.0
Ref. [67]	4-order	**93.4**	**87.4**
Transition-based, statistical models			
Ref. [68]	Arc-standard	89.7	82.7
Ref. [68]	Arc-eager	89.9	80.3
Ref. [69]	Global learning, beam	91.4	84.3
Ref. [70]	Rich non-local features	**92.9**	**86.0**
Ref. [42]	Dynamic oracle	91.0	84.7
Graph-based, neural models			
Ref. [71]	Feed-forward	93.3	N/A
Ref. [72]	CNN	93.4	87.7
Ref. [73]	2-layer LSTM	94.1	87.6
Ref. [74]	2-layer LSTM	93.1	86.6
Ref. [75]	3-layer LSTM, biaffine	95.7	88.9
Ref. [76]	Self-attentive	95.9	92.2
Ref. [76]	+ELMO	96.6	90.3
Ref. [76]	+BERT	**96.7**	**92.2**
Ref. [77]	GNN	96.0	N/A
Transition-based, neural models			
Ref. [78]	Feed-forward	91.8	83.9
Ref. [79]	Stack-LSTM	93.1	87.2
Ref. [80]	Global learning, beam	93.3	N/A
Ref. [81]	Global learning, beam	94.6	N/A
Ref. [74]	2-layer LSTM	93.9	87.6
Ref. [82]	Char, stack-LSTM	93.6	87.6
Ref. [83]	3-layer LSTM	**95.9**	**90.6**
Other Methods (report neural models only)			
Ref. [84]	Easy-first	93.0	87.1
Ref. [61]	Sequence-to-sequence	92.1	86.2
Ref. [85]	Sequence labeling	93.7	N/A
Ref. [30]	HPSG grammar	97.2	**91.2**
Ref. [31]	HPSG, improved attention	**97.3**	**N/A**

ref. [72]. Following, deep bidirectional LSTMs are exploited to substitute the simple neural feed-forward network [73,74]. As sentence-level global information can be encoded through these neural structures, the performance gap between first- and higher-order decoding is largely reduced.

Ref. [75] proposed a deep biaffine parser which achieves the impressive performances, boosting the UAS and LAS numbers into a new degree. The parser exploits a three-layer bidirectional LSTM as the encoder, and a biaffine operation as the decoder to score all possible dependency edges. This work adopts several tricks to reach their final performance, e.g., the node-level dropouts, and the same dropout mask at every recurrent timestep. Ref. [76] further enhanced the bi-affine parser with self-attentive encoder and contextualized word representations such as ELMo and BERT [35, 36]. Ref. [77] exploited graph neural networks to better the input sentence encoder.

3.2   Transition-based parsing

Transition-based models have achieved great success on dependency parsing. To some extent, the transition-based framework is then received great attention to other NLP tasks involving structural learning because of the successfulness of dependency parsing. For example, the transition-based constituent parsing is initially inspired by transition-based dependency parsing. On the one hand, the transition-based models can obtain nearly equivalent performance compared with graph-based methods. On the other hand, these models are highly efficient, which can achieve linear time complexity. Transition-based models convert dependency parsing into an incremental state-transition process, where states denote partial outputs and they are advanced step by step by predefined transition actions.

3.2.1   Statistical models

The initial work for feature-rich transition-based dependency parsing is suggested by refs. [59,60], and then the framework is extensively investigated [68, 90]. There are two typical transition configurations, arc-standard and arc-eager, respectively, which are comparable in parsing performances. Typically, the transition actions include *shift* operation (aiming for starting next word processing), *arc-left* (aiming for building a left directional dependency), and *arc-right* (aiming for right directional dependencies). Besides, several researchers propose other transition configurations [90–93], which can handle various complex cases, such as non-projective dependencies.

Early transition-based methods usually exploit discriminative classifiers for action prediction when a certain transition state is given, which processes the parsing in a local manner.

The scheme may suffer the error propagation problem, where early errors can influence future predictions. To alleviate the problem, global learning with beam-search is one effective way. Ref. [69] firstly applies the strategy. Rich global features that have been exploited in high-order graph-based dependency parsers can be also integrated into the model [70]. The strategy can be also enhanced with dynamic programming further [94, 95].

Another alternative strategy is the dynamic oracle, which is firstly proposed by ref. [42] for transition-based models by using arc-eager. The method defines dynamic gold-standard oracle based on a sample of erroneous states, and then add these instances to enhance model training. Thus, we can minimize global performance losses when errors occur. Although the strategy gives slightly worse performance than ref. [70], it enables dependency parsing in a greedy way, greatly increasing the parsing efficiency. The strategy has been investigated by several studies with different configurations, such as arc-standard and non-projective parsing [96, 97].

3.2.2   Neural models

Ref. [78] is one millstone work for neural dependency parsing, which substitutes traditional manually-crafted discrete features with neural features. The work uses simple feed-forward neural networks to compose the embeddings of all atomic features automatically, and thus is free of feature engineering. Finally, the proposed model obtained much better performance than the corresponding statistical baseline. Pretrained word embeddings and the neural composition function are the keys to success.

There exist several directions to improve the performance of neural transition-based dependency parsing. First, we can exploit better neural network structures. Stack-LSTM is presented by ref. [79] and then followed by several studies [82, 98, 99], which can represent transition states by utilizing partial structural information. In parallel, deep bidirectional LSTM is also investigated [74,83]. Ref. [83] exploited a similar encoder as ref. [75], achieving slightly better performances than ref. [75]. In fact, with powerful neural encoders, especially pretrained contextualized word representations, the performance difference between graph-based and transition-based is very marginal [86].

Several researchers suggest global learning with beam-search strategy in ref. [70] under the neural setting. Ref. [80] made the pioneer attempts for this goal, which is further perfected with a theoretical guaranty by ref. [81]. These models have achieved state-of-the-art performance before the biaffine parser [75]. One major drawback is that the strategy suffers from the efficiency problem due to the beam search. The dynamic oracle strategy is applied as well making the

greedy transition-based neural dependency parsers [100]. Recently, both global learning and dynamic oracle are difficult to give much-improved capacity when pretrained contextualized word representations are exploited.

3.3 Other frameworks

Several interesting models outside the graph-based and transition-based framework are also concerned. For example, the grammar-based framework can be applied to dependency parsing as well. First, a dependency tree is converted to an equivalent phrase-structural constituent tree, and then a grammar-based constituent parsing model can be applied for dependency parsing. The method is proposed firstly by ref. [87], and also highly emphasized in ref. [2]. Several studies have exploited this method as one component for model ensembling [101]. Recently, refs. [30, 31] adopted the HPSG grammar for the same goal, achieving very competitive performances.

Ref. [102] presented an easy-first dependency parsing model, which processes the input sentences in a nondirectional way. The output dependency tree is constructed recursively, where the highest-confidence dependency arc is selected at each time. The neural version of the work is exploited by ref. [84] by using hierarchical LSTMs. Sequence-to-sequence learning can be also applied to neural dependency parsing, where the transition-based linearization can be served as one natural solution. Ref. [61] presented a strong sequence-to-sequence model by head prediction for each word. Ref. [85] suggested a sequence labeling model for dependency parsing.

3.4 Semi-supervised models

Here we briefly offer a survey for semi-supervised dependency parsing under the traditional statistical setting, which utilizes statistical information extracted from a raw text to enhance a baseline model. This scheme has received little attention under the neural setting because of pretraining. As a whole, the semi-supervised dependency parsing models can be categorized into three types according to the extracted information from the raw text, namely word-level, partial-tree level, and sentence-level methods.

For word-level information, one representative work is ref. [63], which augments the atomic features of a baseline model with word clusters. Ref. [103] exploited selectional preference information from web texts to improve dependency parsing. Actually, word embeddings can be also regarded as a kind of semi-supervised word-level information, which has been suggested by ref. [104] for NLP, but not experimented on dependency parsing. Ref. [105] further extended

the idea into feature embeddings, embedding all features including words.

For the partial-tree level integration, ref. [106] exploited high-frequency auto-parsed bilexical dependencies to enhance the baseline supervised model. Further, ref. [64] extended the work by using higher-order subtrees. Ref. [107] could be regarded as a general framework for the partial tree level integration, by utilizing dependency language models learned from auto-parsed dependency trees.

Self-training, co-training as well as tri-training are straightforward methods for sentence-level semi-supervised learning [108], where high-confidence auto-parsed dependency trees from several baseline models, are used to augment the training dataset. Ref. [109] proposed an ambiguity-aware learning method to effectively model the confidence of auto-parsed dependency trees, leading to significant performance improvements.

3.5 Model ensemble

By effectively combining heterogeneous models, the dependency parsing performance can be further boosted. Ref. [56] first analyzed the differences between graph-based and transition-based models, and then combine the two kinds of models to utilize their complementary information, resulting in better performances. Ref. [101] performed parsing ensemble by including grammar-based models further, which are highly diverse with the graph-based and transition-based models. Under the neural setting, simple voting can achieve very strong performances.

The above studies are all targeted at different parsing models based on the same treebank. There are several studies aimed at the parser ensemble based on heterogeneous treebanks, whose annotation guidelines are highly different. Ref. [110] exploited stacked learning combine with quasi-synchronous grammars for effective integration. Ref. [111] studied a similar ensemble by using deep multitask learning, where treebanks of different languages are also concerned. Ref. [112] presented and study the task of supervised treebank conversion, which can be served as one method for integration.

4 Semantic dependency graph

The dependency parsing models mentioned in the previous section are all aimed for dependency tree parsing, which majorly reflects syntactic and shallow-semantic information in sentences. As there are growing demands of deep semantic parsing, which is difficult to be expressed by dependency tree only, dependency graph parsing has received increasing interests [55, 113, 114], which allows multiple (including zero)

heads for one word in sentences. Note that the semantic graph is still formalized by a set of bilexicalized dependencies, with nodes corresponding to surface lexical words, and edges indicating the semantic relations between nodes.

There are different formalizations of the semantic dependency graph. We can combine syntactic tree-based dependency parsing and semantic role labeling (SRL) to result in a dependency graph, which is referred to as joint dependency syntax and SRL [54, 115]. Recently, the conception of semantic dependency parsing (SDP) has been introduced [55, 113, 114], which provides different views of semantic relations, such as DELPH-IN MRS (DM), predicate-argument structures (PDS) and Prague semantic dependencies (PSD). Following, we will review the studies of the two types of semantic dependency graph parsing.

4.1   Joint dependency syntax and SRL

Figure 3 shows an example dependency graph of joint syntactic and semantic dependencies. Here we do not intend to introduce the pipeline models, which train syntactic and semantic models separately, and then output the dependency graph by either two steps or jointly [116, 117]. Although these models can perform dependency graph parsing, they receive less attention as this topic. We focus on the models of joint learning and decoding for full dependency graph parsing. Table 3 shows the performance of several studies on this line.

Ref. [118] extended the transition-based dependency parsing with a particular *swap* operation, enable the model to process non-planarity multiple graphs jointly, and thus dependency graph parsing can be performed jointly. Ref. [119] also exploited the transition-based framework to derive syntactic and semantic dependencies concurrently based on a similar transition system as ref. [118], but adopt a different model estimation by using an incremental sigmoid belief network with latent variables. Ref. [121] presented a graph-based model with a dual decomposition algorithm for decoding, assigning syntactic and semantic dependencies concurrently.

All the aforementioned studies are based on the traditional statistical setting. Under the neural setting, there is little work focus on the task, with one exception. Ref. [120] presented a transition-based stack-LSTM model for joint syntactic and semantic dependencies, where their transition system is largely followed [119]. Since then, neural dependency graph dependency parsing models are centered on other datasets.
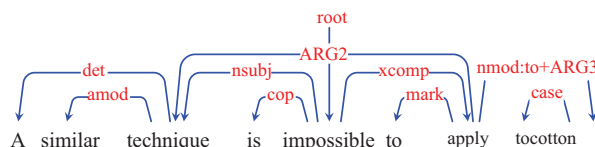
4.2   Semantic dependency parsing

SDP could be regarded as an extension from syntactic dependency parsing by characterizing more semantic relations over

the bilexical dependencies [114, 122], which can be greatly benefited from the advances of dependency parsing. While recently, refs. [55, 113] presented SDP from a different view, which converts the already available linguistic-informed semantic annotations into dependencies, including three different formalisms: DM, PAS and PAD, and currently it has been widely accepted for deep semantic parsing. Figure 4 shows an example of SDP. For SDP, graph- and transition-based models are also the mainstream methods, and most of these models are adapted from dependency tree parsing. Table 4 shows the performance of several representative SDP models.

4.2.1   Graph-based

There are a range of graph-based SDP models for the shared tasks of SDP in SemEval [124, 129]. Generally, it is hard to develop a graph-based decoding algorithm targeted to

Figure 3   (Color online) An example of joint syntactic and semantic dependencies.

Table 3   A comparison of typical joint dependency syntax and SRL models on the CONLL08 English dataset

Model	Main features	Syn	Sem	All
Ref. [117]	Joint inference	86.6	77.1	81.8
Ref. [118]	Transition-based	87.5	76.1	81.8
Ref. [119]	Sigmoid belief network	87.5	76.1	81.8
Ref. [120]	Neural, stack-LSTM	**89.1**	**80.5**	**84.5**
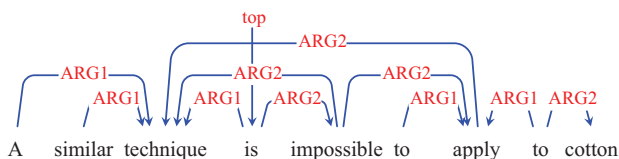
Figure 4   (Color online) An example of semantic dependency graph.

Table 4   A comparison of typical dependency parsing models on the SemEval-2015 shared dataset, where WSJ and Brown indicate the in-domain and out-of-domain test sections

Model	Main features	WSJ	Brown
Ref. [123]	Tree approximations	85.4	80.8
Ref. [124]	2-order graph	85.2	81.2
Ref. [125]	Multi-task learning	87.2	83.6
Ref. [126]	Transition, LSTM	86.9	82.8
Ref. [127]	LSTM, biaffine	89.5	86.3
Ref. [128]	2-order graph, LSTM	**89.8**	**86.9**

arbitrary dependency graphs. Thus, most models have imposed particular constraints. Ref. [130] presented a cubic-time exact inference algorithm for non-crossing dependency graphs. Refs. [131, 132] investigated the maximum subgraph algorithm for 1-endpoint-crossing, pagenumber-2 graphs. Ref. [133] attempted to solve the dependency graph parsing by subgraph decomposition and merging. Ref. [134] proposed an interesting book embedding strategy for SDP.

All the above models exploit manually-crafted discrete features. Under the neural setting, ref. [125] presented a multi-task learning framework to different views of SDP. Ref. [127] extended the biaffine dependency parsing for SDP. Recently, ref. [128] proposed a second-order SDP model based on ref. [127]. As a whole, neural models can obtain better performances for SDP.

4.2.2 Transition-based

The transition-based SDP models can also achieve competitive performance, and meanwhile, these models are more efficient and free of constraints, thus they have received great attention [135, 136]. Actually, transition-based dependency graph parsing can be dated back to ref. [137], and the model is enhanced with dynamic oracle by ref. [138]. Ref. [122] defined a K-permutation transition system to handle dependency graph generation. Ref. [139] presented two novel transition systems for deep semantic dependency parsing. Ref. [140] presented a transition-based system by including a cache to capture dependency graphs.

Recently, ref. [126] proposed a strong transition-based SDP model by using neural networks. They exploit deep bidirectional LSTM as sentential encoder together with stack-LSTM for better representation of transition states. Ref. [141] presented a transition-based model for general semantic graph parsing, which is also suitable for SDP.

4.2.3 Other methods

Dependency graph parsing by using tree approximations and post-processing is also able to obtain competitive performance. These kinds of models first convert dependency graphs into trees, and then tree-based parsing can be applied [142, 143]. Ref. [123] ensemble several tree approximation strategies and achieved the top performance in SemEval 2015 [55]. Ref. [144] conducted a comprehensive investigation of semantic dependency graph parsing using tree approximations.
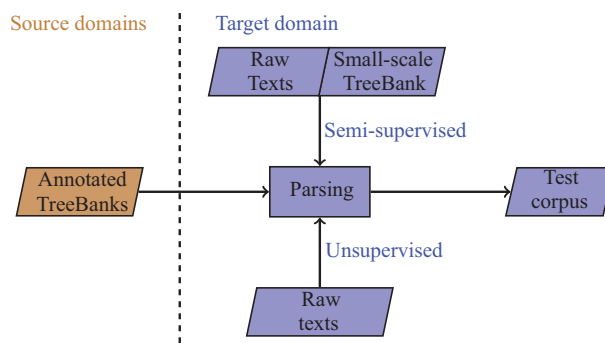
5 Cross-domain parsing

Cross-domain adaption is one hot topic in the NLP community, especially for the syntactic and semantic parsing tasks,

where the data annotation is extremely laborious and expensive. Currently supervised parsing has achieved incredibly high performances thanks to the recent advances of neural networks. However, the performance could drop significantly when the well-trained parsers are applied to texts in different domains as the training corpus. It is impractical to annotate training datasets for all domains. Thus, cross-domain adaption is very important to make parser applicable. The studies of cross-domain parsing are focused on two settings majorly: unsupervised domain adaption, where no target domain training dataset is available, and semi-supervised domain adaption, where a small-scale of training instances are available for a target domain. Figure 5 shows the architecture of cross-domain parsing, where the differences between the two settings are illustrated.

5.1 Unsupervised domain adaption

Self-training is one useful strategy for cross-domain parser adaption, although it has achieved very limited gains under the in-domain semi-supervised setting. Initial work mostly focuses on constituent parsing. Ref. [145] exploited a reranking strategy to obtain a set of high-confidence auto-parsed outputs, and then add them to the training corpus. Ref. [146] showed that without reranking self-training alone can also give significant improvements. Ref. [147] firstly applied self-training successfully on dependency parsing, which exploits an extra classifier to determine whether a parsed tree is reliable. Ref. [148] exploited only high-confidence partial dependencies for next-round training. Ref. [149] proposed a novel confidence estimation method, leading to improved performance on the out-of-domain dataset.

Besides self-training, there are several other methods for unsupervised domain adaption. Ref. [150] applied co-training to constituent parsing, which is similar to self-training but difference in that the example selection is performed by two parsers. Ref. [151] used a similar co-training method for dependency parsing. Further, ref. [108] exploited

Figure 5 (Color online) The architecture of cross-domain parsing.

tri-training for domain adaption of dependency parsing, extending two parsers into parsers. Interestingly, ref. [152] selected training instances from the source-domain dataset instead, where the instances most relevant to the target domain are chosen. Ref. [153] exploited deep belief neural networks to enhance the dependency parsing performance on out-of-domain test data, which can effectively extract useful information from target-domain raw texts.

Multi-source domain adaption is also a promising direction, which assumes that training corpora of several source domains are available. The setting is highly matched with the real practical scenario. Ref. [154] presented the first work of this setting for dependency parsing. They linearly combine the parsing models of different domains with the weights learned from a regression model, considering the performance of each individual parser on the target domain.

5.2 Semi-supervised domain adaption

With a small number of target domain training dataset, ref. [155] showed that self-training can effectively improve the performance of constituent parsing. Recently, most work focuses on effectively training on the mixed source and target training instances by separating the domain-dependent and domain-invariant features [156]. By treating these features differently, the final model can accurately transfer the useful knowledge from the source domain into the target. Ref. [157] extended the idea with a hierarchical Bayesian model and evaluate it on dependency parsing, achieving better performance on the target domain than training with only the target-domain data. Under the neural setting, adversarial learning is one effective method for the same purpose [158]. Ref. [159] firstly applied the method on dependency parsing.

Active learning can be one promising approach for semi-supervised domain adaption. Considering that full-sentence syntax/semantic annotation is extremely expensive, partial annotation might be preferable. For constituent parsing, ref. [160] suggested partial annotation of constituent brackets to enhance domain adaption. For dependency parsing, ref. [161] exploited partial annotation combined with active learning for cross-domain dependency parsing in Japanese. Recently, ref. [162] investigated the strategy comprehensively for Chinese dependency parsing under the neural setting.

6 Cross-lingual parsing

Cross-lingual parsing, which aims to parse the sentence structures of low-resource languages with the help of resource-rich languages such as English. There have been a number of studies for this task, and the majority of work focuses on dependency parsing due to the relatively structural conciseness

as well as well-developed universal dependencies. In particular, with the recent development of cross-lingual or universal word representations based on neural pretraining techniques, the task has been concerned with increasing interests. The task includes two main settings, the unsupervised setting assuming that no training corpus is available for target languages, and the semi-supervised/supervised setting where there exists a certain scale of corpora for the target languages. The architecture of cross-lingual parsing is shown in Figure 6, where the detailed difference between unsupervised and semi-supervised/supervised settings are illustrated as well.

6.1 Unsupervised setting

For unsupervised cross-lingual parsing, the mainstream methods can be classified into two categories, model transferring and annotation projection, where the first category trains a model on the source-language training corpus, and then directly uses it to parse the target-language texts, and the second category projects the source-language parse annotations into the target-language by using a parallel corpus, resulting in a pseudo training corpus for the target language, and then trains a target-language parsing model on the pseudo corpus.

6.1.1 Model transferring

The model transferring approach is straightforward for cross-lingual parsing. The most effort is concerned with language-independent features, which play consistent functions across languages. This line of work is initially presented by ref. [163] which suggests delexiciallized models for cross-lingual dependency parsing, and is further developed by ref. [164] for multi-source transferring, where multiple source languages are used to enhance a target language. Several researchers resort to various non-lexical features to enhance the delexicalized cross-lingual models [165, 166].

Recently, ref. [167] exploited cross-lingual word clusters, which is one king of cross-lingual word representations. Under the neural setting, the exploration of cross-lingual word
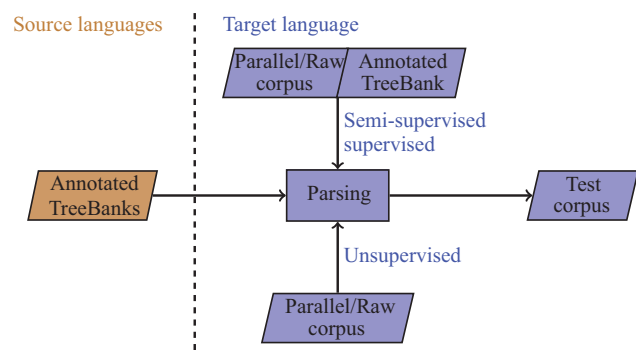
Figure 6 (Color online) The architecture of cross-lingual parsing.

representations is greatly facilitated. Ref. [168] proposed to use cross-lingual word embeddings for lexicalized cross-lingual dependency parsing. This method is then received much attention and can be further enhanced by various ways such as better neural structures [169] and multi-source adaption [170, 171].

Cross-lingual pretrained contextualized word representations give the state-of-the-art performances of this category. Ref. [172] provided a method to learn contextual ELMO representations effectively and then apply the representations on the task, achieving much better performances than cross-lingual word embeddings. Refs. [173, 174] applied cross-lingual mBERT to zero-shot cross-lingual dependency parsing. Ref. [175] introduced the XLM concurrently to mBERT, which is also a kind of strong multilingual contextualized word representations for cross-lingual parsing [176]. All these recent studies lead to state-of-the-art performances in the literature of this category.

6.1.2 Annotation projection

The annotation projection approach requires slightly more effort compared with model transferring, which aims to build a pseudo training corpus through bitext projection. With the pseudo training corpus, the final model can capture rich target-language characteristics. The method relies on a set of parallel sentences between the source and target languages. A source parser trained on the source treebank is used to parse the source-side sentences of the parallel corpus, and then the automatic source annotations are projected onto the target language sentences according to word alignments, resulting in the final pseudo training corpus. There are a range of strategies to achieve the goal. For example, we can use different kinds of parallel corpora, such as EuroParl and the book Bible, and can also exploit various sophisticated methods to improve the projection quality.

For constituent parsing, ref. [177] exploited the method for unsupervised constituent parsing, and find that it can significantly outperform the purely-unsupervised models. Ref. [178] suggested an EM algorithm to incremental boost the quality of the projected constituent trees with relaxing constraints. The number of studies on constituent parsing is relatively small, which may be possible due to that the projection of constituent structures is very complex.

For dependency parsing, ref. [179] presented the first work of this category, and then the approach has been extensively studied under different settings, such as confidence-aware learning [180], neural network enhancing [181, 182], and multi-source adaption [183, 184]. Interestingly, ref. [185] proposed a joint model for cross-lingual constituent and dependency parsing with annotation projection. The approach

achieves great success for cross-lingual dependency parsing.

6.1.3 Other methods

There are also several other methods for unsupervised cross-lingual parsing. Treebank translation is one representative strategy, which is essentially highly similar to annotation projection. The approach also aims to construct a pseudo training corpus. Different from annotation projection, it directly translates the source training corpus into the target language. Besides bitext projection, it requires translation to produce target language sentences. Ref. [186] firstly proposed this method and their method is further perfected by their later studies [187]. Ref. [188] studied the approach under the neural setting with partial translation, and combine their model with model transferring.

The methods exploited in cross-domain parsing may be also suitable (e.g., self-training) for this setting because of the cross-lingual word representations. However, these kinds of methods have been seldom studied. Ref. [189] combined the advantages of model transferring, annotation projection, treebank translation as well as self-training to obtain a very strong model for cross-lingual dependency parsing.

Sentence reordering is one interesting method presented recently, which aims to reorder the input source language syntactic trees to make it highly similar to the target language. The idea is first studied by ref. [190]. Ref. [191] exploited the method with two strong reordering strategies, obtaining very competitive performance compared with even supervised parsing models.

6.2 Semi-supervised/supervised setting

As the availability of treebanks for a range of languages, how to effectively exploit both source and target language treebanks is one interesting problem. Since very early, several studies show that two languages are better than one language alone for parsing. Ref. [192] showed that joint training the English and the Korean parser can bring better performance. Ref. [193] also demonstrated the same observation.

Under the neural setting, this line of work can be conducted more conveniently due to the cross-lingual word representations. Ref. [194] proposed to use one single universal model to parse all languages. However, their final performance is still below the corresponding individual baselines. Ref. [195] trained 34 models for 46 different languages. By aggregating multiple treebanks from one language or closely related languages, we can achieve competitive performances and meanwhile reduce the number of required models greatly. Most recently, ref. [196] proposed a sophisticated strategy to train one universal model for 75 languages by leveraging a

multilingual BERT self-attention, which achieves better performances than the corresponding individual models.
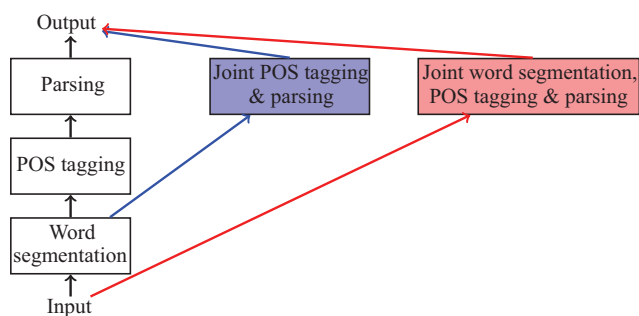
7   Joint models

In this section we discuss joint models of parsing, focusing only on the final goal being the parsing task. The studies of jointly modeling syntax-semantic parsing as well as a targeted downstream task will be introduced in the next section. The development of joint models is mainly motivated by the error prorogation problem of the preconditioned tasks. POS tagging is one of the major preconditioned tasks, as POS tags are one kind of valuable feature source for parsing. Before POS tagging, several languages such as Chinese require word segmentation as a prerequisite step. Parsing is generally performed based on words, while sentences of these languages do not have explicit word boundaries. In summary, here we briefly investigate two kinds of joint models: joint POS tagging and parsing, joint segmentation & tagging and parsing, and we show their relationship in Figure 7.

Noticeably, there are several studies for joint syntactic and semantic parsing. The dependency-based joint models have been already described in Sect. 4.1. Thus, one can refer to there for details. For joint constituent parsing and semantic role labeling, there are very few studies. The representative work is ref. [197], which is the first work of this kind by using sophisticated manually-crafted features. The work shows that their joint model is able to give better performances for both Chinese constituent parsing and SRL.

7.1   Joint POS tagging and parsing

For joint POS tagging and constituent parsing, the chart-based PCFG parsing naturally performs the two tasks concurrently [9, 10, 12], where POS tags can be directly induced from the bottom lexical rules. Based on the transition-based framework, joint POS tagging and constituent parsing can be easily achieved by the shift operation with one additional

Figure 7   (Color online) The architecture of joint models targeted for parsing, where word segmentation is only available to the Chinese language.

parameter to indicate the POS tag of the processing word. Ref. [198] investigated the joint task and present a number of non-local features.

Ref. [199] proposed the first joint model of POS tagging and dependency parsing based on graph factoring, where the basic scoring units are augmented with POS tags. Ref. [200] enhanced the model with better learning strategies. Ref. [201] is the first transition-based model for joint POS tagging and dependency parsing. Ref. [202] extended the transition-based model for non-projective dependency parsing. The two kinds of models achieve comparable performances for both tasks.

Under the neural setting, ref. [203] investigated the model of ref. [202] with neural features. Ref. [204] suggested a joint POS tagging and dependency parsing model by stack propagation. Ref. [205] further investigated the neural joint task with LSTMs by using graph-based and transition-based frameworks, respectively. In fact, the importance of joint modeling has been largely weakened as parsing without POS tags can also obtain strong performance which is close to the same model with POS tags [75].

7.2   Joint segmentation, tagging and parsing

The task of joint segmentation, tagging and parsing is majorly targeted to Chinese parsing. The series of this work starts very early [206] by character-level parsing. Later, ref. [207] demonstrated that Chinese dependency parsing based on characters is better, which can naturally perform the three tasks. Recently, ref. [208] proposed a transition-based joint model for word segmentation, POS tagging and dependency parsing. Ref. [209] suggested a similar transition-based joint model by using indivisible subwords as well as their internal structures. Refs. [210, 211] conducted character-level constituent and dependency parsing by extending word-level annotations into characters, achieving state-of-the-art performances for both tasks under the discrete setting. All the four models exploit transition-based framework. Ref. [212] proposed the first work by using graph-based inference, with efficient hill-climb decoding.

Ref. [213] is the first work of adopting neural networks for character-level constituent parsing, achieving comparable with the state-of-the-art discrete model by a simple convolutional neural network. Ref. [214] presented a neural model for character-level dependency parsing. Ref. [215] proposed a strong joint model for word segmentation and dependency parsing only, state-of-the-art biaffine parser and pretrained BERT are exploited in this work. Under the neural network, the joint framework might be highly challenging, as the baselines are strong and meanwhile neural networks can learn global high-level features implicitly.

8 Parser application

When a well-trained syntactic/semantic parser is available, how to use it effectively to benefit for downstream applications is one important topic in the parsing community, which is also related to the verification of the usefulness of syntactic and semantic parsing. In fact, the topic has been extensively studied, and the parsing outputs have been demonstrated effective for a number of tasks such as semantic role labeling [216, 217], relation extraction [218, 219], sentiment analysis [7, 220] and machine translation [5, 221]. The exploration methods have major changes from the statistical discrete models to the neural models. Here we briefly summarize the mainstream approaches of parser exploration in terms of the two settings.

8.1 Feature-based statistical methods

Under the traditional statistical setting, the exploration of parser resorts to manually-crafted discrete features, which are mostly designed sophisticatedly according to the targeted tasks. We briefly summarize the widely-adopted features here. For constituent trees, such features include nonterminal categories, CFG rules, phrase-level word ngrams, syntax paths to the root or some other word, the matching with a completed phrase. For dependency trees, dependency-based ngrams, dependency labels, dependency paths are widely-used features. All these kinds of features are further adapted to various tasks aiming to get most of the parsing information effectively [6, 7, 216, 222, 223]. Besides, the tree-kernel based approach can also be good alternatives [218, 224–227]. Several approaches suggest using multiple heterogeneous parsers for better performances, including the integration of constituent and dependency parsers as well as parsers trained on heterogeneous treebanks [228].

8.2 Representation learning with neural networks

One simple method to use parsing features based on neural networks is to embed all the atomic features, and then exploit sophisticated neural networks to compose them automatically. The most representative method of this kind is the path-based LSTMs, which exploit LSTM over sequential-level constituent or dependency paths [229, 230]. The recent tendency of using the end-to-end framework for the majority of NLP tasks leads to universal representations based on parser outputs. We build a universal encoder with structural outputs of a parser, and then adapt them to different tasks by decoders, as shown by Figure 8. There are several ways to build the encoder. Here we divide the methods into four types: recursive neural network; linearization-based; impli-
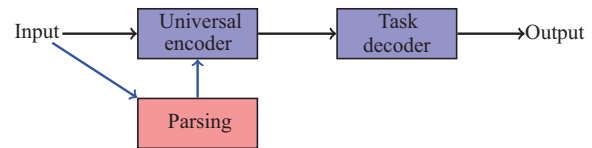
Figure 8 (Color online) Parser enhanced universal encoder for downstream tasks.

cated structural-aware word representations and graph neural networks (GNN).

The recursive neural network is one natural method to model tree-structural outputs, which composes a tree input from bottom-to-up or top-to-down incrementally. We can use various composition operations leading to more sophisticated tree-level neural networks such as tree convolutions suggested by ref. [231] and Tree-LSTM proposed by ref. [220]. All these related studies give improved performances for a range of tasks [232, 233].

The key idea of the linearization-based methods is to convert structural inputs into a sequence of symbols, and then adopt standard sequential encoders to model the new sequence directly [234, 235]. Usually, the conversion can be referred to as the linearization process of transition-based parsers, or we can incrementally traverse a tree or graph in different ways. The method has received fewer concerns which might be due to its extreme simplicity, although it is effective and meanwhile much efficient [221].

The implicit structural-aware word representations, firstly presented by ref. [236] for relation extraction, are similar to the idea of contextualized word representations, which exploit the hidden outputs of a well-pretrained parser as inputs for the downstream tasks [221, 237]. This method can also efficiently represent structural information such as syntax and semantics. Besides, the method can be easily adapted to the multi-task-learning strategy for parser application [217], while parser requires to be jointly trained in multi-task-learning.

Recently, there are grown interests on the topic of graph neural networks, which can be naturally applied to encode structural syntactic and semantic graphs. Indeed, there have been several studies already by using either graph convolutional networks or graph attention networks [238–240], and all these works demonstrate the effectiveness of GNN for structure encoding.

9 Corpus and shared tasks

Finally, we review the work of corpus development in syntactic and semantic parsing, which is critical to the performance of supervised parsing. There are several classical treebanks such as the Penn Treebanks of English and Chinese

languages, which greatly promote the development of the parsing community. In fact, there are treebanks for a range of languages, and here we focus majorly on the Chinese and English treebanks. In addition, there are a number of shared tasks, which also offer valuable corpora for syntactic and semantic parsing.

9.1   Penn treebank

The English Penn Treebank (PTB) by ref. [241] could be the most famous resource for syntactic parsing, which annotates bracketed syntactic phrase structures for over 40000 sentences covering about 4.5 million words. In addition, ref. [242] annotated the Penn Treebank for the Chinese language, for short as CTB, and now there are over 130000 sentences with phrase-structure annotations covering over 2 million words. Both the two datasets have annotated POS tags as well, which are important to automatic syntactic parsing. For Chinese, gold-standard word segmentation has been annotated in CTB as well.

The two datasets are also converted into dependency treebanks for dependency parsing, which could be achieved by rule-based head lexicalization over the constituent trees [60, 69, 243–245]. Recently, Stanford dependencies have been exploited the most popularly especially for the English language, where the conversion rules are relatively more fine-grained [246] and meanwhile can reflect more syntactic and semantic phenomena.

There are several small-scale treebanks with the same annotation guideline as PTB, which can be useful resources for domain adaption studies of constituent and dependency parsing, regarding that PTB are focused on the news genre data. For example, the Brown Treebank is exploited most frequently for cross-domain parsing as the literature genre. Ref. [247] offered a treebank of the biomedical domain. The two treebanks are targeted to researches on constituent parsing. Recently, ref. [248] annotated a treebank for twitter texts based on dependency grammar.

9.2   Universal dependencies

The present of Universal Dependencies (UD) has received great attention for facilitating multilingual researches, which aims to develop cross-linguistically consistent treebank annotation for multiple languages. UD can capture similarities as well as idiosyncrasies among typologically different languages such as English-alike languages, morphologically-rich languages and pro-drop languages. The development of UD is initially based on Stanford typed dependencies [249] and the universal Google dependency scheme [250, 251]. Now it goes through several versions [252, 253], with sig-

nificant changes on the guidelines, also supporting language-specific extensions when necessary. Currently the UD treebank version 2.5 includes 157 treebanks over 90 languages. Besides multilingual dependency parsing, there is an increasing tendency to exploit them for evaluating monolingual dependency parsing based on the datasets as well [77, 86].

9.3   Chinese treebank

For the Chinese languages, treebank development has been concerned by several studies besides the CTB. The Sinica Treebank has offered phrase-structural syntactic trees over about 360000 words in traditional Chinese [254]. Ref. [255] released a constituent treebank covering about one million words for simplified Chinese. Ref. [256] also annotated constituent trees over a scale of 0.9 million words for Chinese. The guidelines of all these phrase-structural treebanks are quite different.

There are several treebank resources directly based on the dependency structure, as it is believed that dependency grammar is simpler and easier to be developed. Refs. [257, 258] constructed a Chinese dependency treebank consuming over 1.1 million words. Ref. [259] created a multi-view Chinese dependency treebank containing 14463 sentences, which is further augmented with predicate-argument information by ref. [260] for a semantic-oriented dependency treebank. Most recently, ref. [162] released a large scale Chinse dependency treebank covering about 3 million words as well as different domains, including news, web blogs, literature texts.

9.4   Shared tasks

Nearly all the shared tasks are focused on dependency parsing, and most of which devote to multilingual parsing with the support of several treebanks in different languages. These shared tasks, on the one hand, can evaluate the current state-of-the-art parsing models, and on the other hand offer valuable datasets for parsing, facilitating the future research work.

The ConLL06 organizes the first shared task for dependency parsing involving 13 languages [261], and domain adaption is considered later in ConLL07 [262]. At ConLL08 and ConLL09, semantic dependencies extracted from SRL are integrated, leading to joint syntactic-semantic parsing [54, 115]. Recently, the shared task on ConLL 2017 starts to adopt universal dependencies for dependency parsing [263], and at ConLL 2018, 82 UD treebanks in 57 languages are included for evaluation [264]. Besides ConLL, SANCL 2012 organizes a shared task on parsing English web text [265], which offers a benchmark dataset for cross-domain dependency parsing in English. In addition, the NLPCC 2019 shared task on cross-domain dependency parsing also offers

a valuable dataset in Chinese [266].

The above shared tasks focus on syntactic dependency parsing. For semantic dependency parsing, ref. [267] presented the first shared task to Chinese texts in SemEval, where dependency trees are used in the evaluation. Ref. [114] started to use dependency graphs for formal semantic representation. For the English language, ref. [113] organized a shared task for broad coverage semantic parsing by using three distinct dependency-based semantic formalizations. Dependency graphs are exploited to represent various semantics. Ref. [55] extended the shared task of ref. [113] with more languages including Chinese and Czech. Ref. [268] covered more topics of semantic graph parsing for deep semantics, including not only dependency-based graphs, but also several other formalizations such as UCCA and AMR.

10 Conclusion and future directions

In this article, we made a thorough review of the past work of syntactic and semantic parsing focusing on constituent parsing and dependency parsing. Traditional statistical models, as well as currently-dominant neural network methods, were both summarized. First, for the parsing models, neural network methods with pretrained contextualized word representations have achieved the top performances for almost all datasets. There is a grown tendency to use simple encoder-decoder frameworks for parsing, so that well-investigated training strategies can be applied. Second, broad-coverage semantic parsing is receiving increasing attention, which might be the next stage hop topic. The task performances are now gradually acceptable thanks to the neural network models as well as the development of linguistic resources.

The cross-domain and cross-lingual settings are important scenarios for parsing, which are difficult to be resolved yet play the key role to the real applications. For the cross-domain setting, there is still a large demand for resources. While for cross-lingual parsing, there exist a number of methods. A comprehensive and fair comparison of these methods as well as their integrations might be valuable. In addition, the difference between cross-domain and cross-language is becoming smaller because of the universal word representations. One can regard cross-lingual parsing as a special case of cross-domain technically.

The importance of joint models is decreasing. By using neural networks, global features across different tasks can be directly captured by sophisticated neural structures such as deep LSTM and self-attention, and on the other hand, we can build one share encoder across tasks to reduce the influence of error propagation. For parser application, which might be regarded as the reverse direction of joint models, neural network encoders can lead to highly effective and elegant universal representations with syntactic and semantic information. Also, all current state-of-the-art methods still require a comprehensive and fair comparison.

Finally, treebank development is the major source of the advances of syntactic and semantic parsing, which might be the most difficult and highly valuable job. In particular, the semantic knowledge of one sentence can have several different views. Comprehensive annotations require extremely-high costs. How to effectively perform treebank annotation is one task deserved investigation.

For future directions, there is still a lot of work left to be followed. Most importantly, parsing with more complex grammars would receive increasing attention, although this survey is no covered. For syntactic parsing, the performances of CCG, HPSG and LFG parsing are still unsatisfactory, especially for non-English languages. For semantic parsing, the dependency-based grammar is not enough for rich semantics, even being relaxed with graph constraints. Non-lexicalized nodes are necessary to express several complicated semantics. Thus, AMR, UCCA and MRS could be promising for practical deep semantic parsing. Based on the CFG and dependency grammars, the cross-domain and cross-lingual settings are deserved to be concerned, which can be further unified. Lightly-supervised or zero-shot models might be practical solutions. For the joint models as well as parser applications, multi-task-learning and pretraining might become more popular architectures for adaption.

1 Manning C D, Schütze H. Foundations of Statistical Natural Language Processing. Massachusetts: The MIT Press, 1999

2 Kübler S, McDonald R, Nivre J. Dependency parsing. Synt Lectures Human Language Tech, 2009, 1: 1–127

3 Chengqing Z. Statistical Natural Language Processing. Beijing: Tsinghua University Press, 2013

4 Jurafsky D, H. Martin J. Speech and Language Processing. 3rd ed. New Jersey: Prentice Hall PTR, 2019

5 Yamada K, Knight K. A syntax-based statistical translation model. In: Proceedings of Association for Computational Linguistic, 39th Annual Meeting and 10th Conference of the European Chapter. Toulouse, 2001. 523–530

6 Chan Y S, Roth D. Exploiting syntactico-semantic structures for relation extraction. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. Portland, 2011. 551–560

7 Zou H, Tang X, Xie B, et al. Sentiment classification using machine learning techniques with syntax features. In: Proceedings of 2015 International Conference on Computational Science and Computational Intelligence. IEEE, 2015. 175–179

8 Henderson J. Discriminative training of a neural network statistical parser. In: Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics. Barcelona, 2004. 95–102

9 Collins M. Three generative, lexicalised models for statistical pars-

ing. In: Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference. Madrid, 1997. 16–23

10 Charniak E. A maximum-entropy-inspired parser. In: Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics. Seattle, 2000

11 McClosky D, Charniak E, Johnson M. Effective self-training for parsing. In: Proceedings of Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics. New York, 2006. 152–159

12 Petrov S, Klein D. Improved inference for unlexicalized parsing. In: Proceedings of Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics. Rochester, 2007. 404–411

13 Hall D, Durrett G, Klein D. Less grammar, more features. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. Baltimore, 2014. 228–237

14 Sagae K, Lavie A. A classifier-based parser with linear run-time complexity. In: Proceedings of the Ninth International Workshop on Parsing Technology. Vancouver, 2005. 125–132

15 Zhu M, Zhang Y, Chen W, et al. Fast and accurate shift-reduce constituent parsing. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics. Sofia, 2013. 434–443

16 Socher R, Bauer J, Manning C D, et al. Parsing with compositional vector grammars. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics. Sofia, 2013. 455–465

17 Durrett G, Klein D. Neural CRF parsing. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing. Beijing, 2015. 302–312

18 Stern M, Andreas J, Klein D. A minimal span-based neural constituency parser. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. Vancouver, 2017. 818–827

19 Kitaev N, Klein D. Constituency parsing with a self-attentive encoder. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics. Melbourne, 2018. 2676–2686

20 Wang Z, Mi H, Xue N. Feature optimization for constituent parsing via neural networks. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing. Beijing, 2015. 1138–1147

21 Watanabe T, Sumita E. Transition-based neural constituent parsing. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing. Beijing, 2015. 1169–1179

22 Dyer C, Kuncoro A, Ballesteros M, et al. Recurrent neural network grammars. In: Proceedings of The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. San Diego, 2016. 199–209

23 Cross J, Huang L. Span-based constituency parsing with a structure-label system and provably optimal dynamic oracles. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. Austin, 2016. 1–11

24 Liu J, Zhang Y. In-order transition-based constituent parsing. Trans Assoc Comput Linguist, 2017, 5: 413–424

25 Fried D, Klein D. Policy gradient as a proxy for dynamic oracles in constituency parsing. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics. Melbourne, 2018. 469–476

26 Kitaev N, Klein D. Tetra-tagging: Word-synchronous parsing with

linear-time inference. ArXiv: 1904.09745

27 Shen Y, Lin Z, Jacob A P, et al. Straight to the tree: Constituency parsing with neural syntactic distance. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics. Melbourne, 2018. 1171–1180

28 Teng Z, Zhang Y. Two local models for neural constituent parsing. In: Proceedings of the 27th International Conference on Computational Linguistics. Santa Fe, 2018. 119–132

29 Vilares D, Abdou M, Søgaard A. Better, faster, stronger sequence tagging constituent parsers. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Minneapolis, 2019. 3372–3383

30 Zhou J, Zhao H. Head-driven phrase structure grammar parsing on Penn treebank. In: Proceedings of the 57th Conference of the Association for Computational Linguistics. Florence, 2019. 2396–2408

31 Mrini K, Dernoncourt F, Bui T, et al. Rethinking self-attention: An interpretable self-attentive encoder-decoder parser. ArXiv: 1911.03875

32 Klein D, Manning C D. Accurate unlexicalized parsing. In: Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics. Sapporo, 2003. 423–430

33 Socher R, Manning C D, Ng A Y. Learning continuous phrase representations and syntactic parsing with recursive neural networks. In: Proceedings of the NIPS-2010 Deep Learning and Unsupervised Feature Learning Workshop. Vancouver, 2010. 1–9

34 Gaddy D, Stern M, Klein D. What's going on in neural constituency parsers? An analysis. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Minneapolis, 2018. 999–1010

35 Peters M, Neumann M, Iyyer M, et al. Deep contextualized word representations. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Minneapolis, 2018. 2227–2237

36 Devlin J, Chang M W, Lee K, et al. BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Minneapolis, 2019. 4171–4186

37 Ratnaparkhi A. A linear observed time statistical parser based on maximum entropy models. In: Proceedings of Second Conference on Empirical Methods in Natural Language Processing. Providence, 1997

38 Zhang Y, Clark S. Transition-based parsing of the chinese treebank using a global discriminative model. In: Proceedings of the 11th International Workshop on Parsing Technologies. Paris, 2009. 162–171

39 Collins M. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In: Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing. Philadelphia, 2002. 1–8

40 Cross J, Huang L. Incremental parsing with minimal features using bi-directional LSTM. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. Berlin, 2016. 32–37

41 Coavoux M, Crabbé B. Neural greedy constituent parsing with dynamic oracles. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. Berlin, 2016. 172–182

42 Goldberg Y, Nivre J. A dynamic oracle for arc-eager dependency parsing. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. Jeju Island, 2012. 959–976

43 Coavoux M, Crabbé B, Cohen S B. Unlexicalized Transition-based Discontinuous Constituency Parsing. Trans Assoc Comput Linguist, 2019, 7: 73–89

44 Fernández-González D, Gómez-Rodríguez C. Faster shift-reduce constituent parsing with a non-binary, bottom-up strategy. Artif In-

tell, 2019, 275: 559–574

45 Vinyals O, Kaiser Ł, Koo T, et al. Grammar as a foreign language. In: Proceedings of Advances in Neural Information Processing Systems. Montreal, 2015. 2773–2781

46 Choe D K, Charniak E. Parsing as language modeling. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. Austin, 2016. 2331–2336

47 Gómez-Rodríguez C, Vilares D. Constituent parsing as sequence labeling. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Brussels, 2018. 1314–1324

48 Vilares D, Strzyz M, Søgaard A, et al. Parsing as pretraining. ArXiv: 2002.01685

49 Yang Z, Dai Z, Yang Y, et al. Xlnet: Generalized autoregressive pretraining for language understanding. In: Proceedings of Thirty-third Conference on Neural Information Processing Systems. Vancouver, 2019. 5754–5764

50 McClosky D, Charniak E, Johnson M. When is self-training effective for parsing? In: Proceedings of the 22nd International Conference on Computational Linguistics. Manchster, 2008. 561–568

51 Candito M, Crabbé B. Improving generative statistical parsing with semi-supervised word clustering. In: Proceedings of the 11th International Workshop on Parsing Technologies. Paris, 2009. 138–141

52 Collins M, Koo T. Discriminative reranking for natural language parsing. Comput Linguist, 2005, 31: 25–70

53 Huang L. Forest reranking: Discriminative parsing with non-local features. In: Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics. Columbus, 2008. 586–594

54 Hajič J, Ciaramita M, Johansson R, et al. The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In: Proceedings of the Thirteenth Conference on Computational Natural Language Learning. Boulder, 2009. 1–18

55 Oepen S, Kuhlmann M, Miyao Y, et al. Semeval 2015 task 18: Broad-coverage semantic dependency parsing. In: Proceedings of the 9th International Workshop on Semantic Evaluation. Denver, 2015. 915–926

56 Nivre J, McDonald R. Integrating graph-based and transition-based dependency parsers. In: Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics. Columbus, 2008. 950–958

57 McDonald R, Crammer K, Pereira F. Online large-margin training of dependency parsers. In: Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics. University of Michigan, 2005. 91–98

58 Carreras X. Experiments with a higher-order projective dependency parser. In: Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL. Prague, 2007

59 Nivre J. An efficient algorithm for projective dependency parsing. In: Proceedings of the Eighth International Conference on Parsing Technologies. Nancy, 2003

60 Yamada H, Matsumoto Y. Statistical dependency analysis with support vector machines. In: Proceedings of the Eighth International Conference on Parsing Technologies. Nancy, 2003

61 Li Z, Cai J, He S, et al. Seq2seq dependency parsing. In: Proceedings of the 27th International Conference on Computational Linguistics. Santa Fe, 2018. 3203–3214

62 McDonald R, Pereira F. Online learning of approximate dependency parsing algorithms. In: Proceedings of the 11st Conference of the European Chapter of the Association for Computational Linguistics. Trento, 2006

63 Koo T, Carreras X, Collins M. Simple semi-supervised dependency parsing. In: Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics. Columbus, 2008. 595–603

64 Chen W, Kazama J, Uchimoto K, et al. Improving dependency parsing with subtrees from auto-parsed data. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing.

Singapore, 2009. 570–579

65 Bohnet B. Top accuracy and fast dependency parsing is not a contradiction. In: Proceedings of the 23rd International Conference on Computational Linguistics. Beijing, 2010. 89–97

66 Koo T, Collins M. Efficient third-order dependency parsers. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. Uppsala, 2010. 1–11

67 Ma X, Zhao H. Fourth-order dependency parsing. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. Jeju Island, 2012. 785–796

68 Nivre J. Algorithms for deterministic incremental dependency parsing. Comput Linguist, 2008, 34: 513–553

69 Zhang Y, Clark S. A tale of two parsers: Investigating and combining graph-based and transition-based dependency parsing. In: Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing. Honolulu, 2008. 562–571

70 Zhang Y, Nivre J. Transition-based dependency parsing with rich non-local features. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. Portland, 2011. 188–193

71 Pei W, Ge T, Chang B. An effective neural network model for graph-based dependency parsing. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing. Beijing, 2015. 313–322

72 Zhang Z, Zhao H, Qin L. Probabilistic graph-based dependency parsing with convolutional neural network. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. Berlin, 2016. 1382–1392

73 Wang W, Chang B. Graph-based dependency parsing with bidirectional LSTM. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. Berlin, 2016. 2306–2315

74 Kiperwasser E, Goldberg Y. Simple and accurate dependency parsing using bidirectional LSTM feature representations. Trans Assoc Comput Linguist, 2016, 4: 313–327

75 Dozat T, Manning C D. Deep biaffine attention for neural dependency parsing. ArXiv: 1611.01734

76 Li Y, Li Z, Zhang M, et al. Self-attentive biaffine dependency parsing. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence. Macao, 2019. 5067–5073

77 Ji T, Wu Y, Lan M. Graph-based dependency parsing with graph neural networks. In: Proceedings of the 57th Conference of the Association for Computational Linguistics. Florence, 2019. 2475–2485

78 Chen D, Manning C. A fast and accurate dependency parser using neural networks. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. Doha, 2014. 740–750

79 Dyer C, Ballesteros M, Ling W, et al. Transition-based dependency parsing with stack long short-term memory. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing. Beijing, 2015. 334–343

80 Zhou H, Zhang Y, Huang S, et al. A neural probabilistic structured-prediction model for transition-based dependency parsing. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing. Beijing, 2015. 1213–1222

81 Andor D, Alberti C, Weiss D, et al. Globally normalized transition-based neural networks. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. Berlin, 2016. 2442–2452

82 Ballesteros M, Dyer C, Goldberg Y, et al. Greedy transition-based

dependency parsing with stack LSTMs. Comput Linguist, 2017, 43: 311–347

83 Ma X, Hu Z, Liu J, et al. Stack-pointer networks for dependency parsing. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics. Melbourne, 2018. 1403–1414

84 Kiperwasser E, Goldberg Y. Easy-first dependency parsing with hierarchical tree LSTMs. Trans Assoc Comput Linguist, 2016, 4: 445–461

85 Strzyz M, Vilares D, Gómez-Rodríguez C. Viable dependency parsing as sequence labeling. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Minneapolis, 2019. 717–723

86 Kulmizev A, de Lhoneux M, Gontrum J, et al. Deep contextualized word embeddings in transition-based and graph-based dependency parsing - a tale of two parsers revisited. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing. Hong Kong, 2019. 2755–2768

87 McDonald R. Discriminative learning and spanning tree algorithms for dependency parsing. Dissertation for Dcotoral Degree. Philadelphia: University of Pennsylvania, 2006

88 Bohnet B. Very high accuracy and fast dependency parsing is not a contradiction. In: Proceedings of the 23rd International Conference on Computational Linguistics, Proceedings of the Conference. Beijing, 2010. 89–97

89 Lei T, Xin Y, Zhang Y, et al. Low-rank tensors for scoring dependency structures. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. Baltimore, 2014. 1381–1391

90 Gómez-Rodríguez C, Nivre J. Divisible transition systems and multiplanar dependency parsing. Comput Linguist, 2013, 39: 799–845

91 Nivre J. Non-projective dependency parsing in expected linear time. In: Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP. Singapore, 2009. 351–359

92 Sartorio F, Satta G, Nivre J. A transition-based dependency parser using a dynamic parsing strategy. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics. Sofia, 2013. 135–144

93 Noji H, Miyao Y. Left-corner transitions on dependency parsing. In: Proceedings of the 25th International Conference on Computational Linguistics. Dublin, 2014. 2140–2150

94 Huang L, Sagae K. Dynamic programming for linear-time incremental parsing. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. Uppsala, 2010. 1077–1086

95 Kuhlmann M, Gómez-Rodríguez C, Satta G. Dynamic programming algorithms for transition-based dependency parsers. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. Portland, 2011. 673–682

96 Goldberg Y, Sartorio F, Satta G. A tabular method for dynamic oracles in transition-based parsing. Trans Assoc Comput Linguist, 2014, 2: 119–130

97 Gómez-Rodríguez C, Sartorio F, Satta G. A polynomial-time dynamic oracle for non-projective dependency parsing. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. Doha, 2014. 917–927

98 Ballesteros M, Dyer C, Smith N A. Improved transition-based parsing by modeling characters instead of words with LSTMs. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Lisbon, 2015. 349–359

99 de Lhoneux M, Ballesteros M, Nivre J. Recursive subtree composition in LSTM-based dependency parsing. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Minneapolis, 2019. 1566–1576

100 Fernández-González D, Gómez-Rodríguez C. A dynamic oracle for linear-time 2-planar dependency parsing. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Minneapolis, 2018. 386–392

101 Sun W, Wan X. Data-driven, PCFG-based and pseudo-PCFG-based models for Chinese dependency parsing. Trans Assoc Comput Linguist, 2013, 1: 301–314

102 Goldberg Y, Elhadad M. An efficient algorithm for easy-first non-directional dependency parsing. In: Proceedings of Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics. Los Angeles, 2010. 742–750

103 Zhou G, Zhao J, Liu K, et al. Exploiting web-derived selectional preference to improve statistical dependency parsing. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. Portland, 2011. 1556–1565

104 Turian J, Ratinov L A, Bengio Y. Word representations: A simple and general method for semi-supervised learning. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. Uppsala, 2010. 384–394

105 Chen W, Zhang Y, Zhang M. Feature embedding for dependency parsing. In: Proceedings of the 25th International Conference on Computational Linguistics. Dublin, 2014. 816–826

106 Chen W, Kawahara D, Uchimoto K, et al. Dependency parsing with short dependency relations in unlabeled data. In: Proceedings of Third International Joint Conference on Natural Language Processing. Hyderabad, 2008

107 Chen W, Zhang M, Li H. Utilizing dependency language models for graph-based dependency parsing models. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics. Jeju Island, 2012. 213–222

108 Søgaard A, Rishøj C. Semi-supervised dependency parsing using generalized tri-training. In: Proceedings of the 23rd International Conference on Computational Linguistics. Beijing, 2010. 1065–1073

109 Li Z, Zhang M, Chen W. Ambiguity-aware ensemble training for semi-supervised dependency parsing. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. Baltimore, 2014. 457–467

110 Li Z, Liu T, Che W. Exploiting multiple treebanks for parsing with quasi-synchronous grammars. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics. Jeju Island, 2012. 675–684

111 Guo J, Che W, Wang H, et al. A universal framework for inductive transfer parsing across multi-typed treebanks. In: Proceedings of the 26th International Conference on Computational Linguistics. Osaka, 2016. 12–22

112 Jiang X, Li Z, Zhang B, et al. Supervised treebank conversion: Data and approaches. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics. Melbourne, 2018. 2706–2716

113 Oepen S, Kuhlmann M, Miyao Y, et al. SemEval 2014 task 8: Broad-coverage semantic dependency parsing. In: Proceedings of the 8th International Workshop on Semantic Evaluation. Dublin, 2014. 63–72

114 Che W, Shao Y, Liu T, et al. SemEval-2016 task 9: Chinese semantic dependency parsing. In: Proceedings of the 10th International Workshop on Semantic Evaluation. San Diego, 2016. 1074–1080

115 Surdeanu M, Johansson R, Meyers A, et al. The conll 2008 shared task on joint parsing of syntactic and semantic dependencies. In: Proceedings of the Twelfth Conference on Computational Natural Language Learning. Manchester, 2008. 159–177

116 Che W, Li Z, Li Y, et al. Multilingual dependency-based syntactic and semantic parsing. In: Proceedings of the Thirteenth Conference on Computational Natural Language Learning. Boulder, 2009. 49–54

117 Johansson R. Statistical bistratal dependency parsing. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing. Singapore, 2009. 561–569

118 Titov I, Henderson J, Merlo P, et al. Online graph planarisation for synchronous parsing of semantic and syntactic dependencies. In: Proceedings of the 21st International Joint Conference on Artificial Intelligence. Pasadena, 2009

119 Henderson J, Merlo P, Titov I, et al. Multilingual joint parsing of syntactic and semantic dependencies with a latent variable model. Comput Linguist, 2013, 39: 949–998

120 Swayamdipta S, Ballesteros M, Dyer C, et al. Greedy, joint syntactic-semantic parsing with stack LSTMs. In: Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning. Berlin, 2016. 187–197

121 Lluís X, Carreras X, Màrquez L. Joint Arc-factored parsing of syntactic and semantic dependencies. Trans Assoc Comput Linguist, 2013, 1: 219–230

122 Sun W, Du Y, Kou X, et al. Grammatical relations in Chinese: GB-ground extraction and data-driven parsing. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. Baltimore, 2014. 446–456

123 Du Y, Zhang F, Zhang X, et al. Peking: Building semantic dependency graphs with a hybrid parser. In: Proceedings of the 9th International Workshop on Semantic Evaluation. Denver, 2015. 927–931

124 Almeida M S, Martins A F. Lisbon: Evaluating turbosemanticparser on multiple languages and out-of-domain data. In: Proceedings of the 9th International Workshop on Semantic Evaluation. Denver, 2015. 970–973

125 Peng H, Thomson S, Smith N A. Deep multitask learning for semantic dependency parsing. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. Vancouver, 2017. 2037–2048

126 Wang Y, Che W, Guo J, et al. A neural transition-based approach for semantic dependency graph parsing. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence. New Orleans, 2018

127 Dozat T, Manning C D. Simpler but more accurate semantic dependency parsing. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics. Melbourne, 2018. 484–490

128 Wang X, Huang J, Tu K. Second-order semantic dependency parsing with end-to-end neural networks. In: Proceedings of the 57th Conference of the Association for Computational Linguistics. Florence, 2019. 4609–4618

129 Thomson S, O'Connor B, Flanigan J, et al. CMU: Arc-factored, discriminative semantic dependency parsing. In: Proceedings of the 8th International Workshop on Semantic Evaluation. Dublin, 2014. 176–180

130 Kuhlmann M, Jonsson P. Parsing to noncrossing dependency graphs. Trans Assoc Comput Linguist, 2015, 3: 559–570

131 Cao J, Huang S, Sun W, et al. Parsing to 1-endpoint-crossing, pagenumber-2 graphs. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. Vancouver, 2017. 2110–2120

132 Cao J, Huang S, Sun W, et al. Quasi-second-order parsing for 1-endpoint-crossing, pagenumber-2 graphs. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. Copenhagen, 2017. 24–34

133 Sun W, Du Y, Wan X. Parsing for grammatical relations via graph merging. In: Proceedings of the 21st Conference on Computational Natural Language Learning. Vancouver, 2017. 26–35

134 Sun W, Cao J, Wan X. Semantic dependency parsing via book embed-

ding. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. Vancouver, 2017. 828–838

135 Ribeyre C, Villemonte de la Clergerie E, Seddah D. Alpage: Transition-based semantic graph parsing with syntactic features. In: Proceedings of the 8th International Workshop on Semantic Evaluation. Dublin, 2014. 97–103

136 Kanerva J, Luotolahti J, Ginter F. Turku: Semantic dependency parsing as a sequence classification. In: Proceedings of Proceedings of the 9th International Workshop on Semantic Evaluation. Denver, 2015. 965–969

137 Sagae K, Tsujii J. Shift-reduce dependency DAG parsing. In: Proceedings of the 22nd International Conference on Computational Linguistics. Manchster, 2008. 753–760

138 Tokgöz A, Eryiğit G. Transition-based dependency dag parsing using dynamic oracles. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing: Student Research Workshop. Beijing, 2015. 22–27

139 Zhang X, Du Y, Sun W, et al. Transition-based parsing for deep dependency structures. Comput Linguist, 2016, 42: 353–389

140 Gildea D, Satta G, Peng X. Cache transition systems for graph parsing. Comput Linguist, 2018, 44: 85–118

141 Buys J, Blunsom P. Robust incremental neural semantic graph parsing. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. Vancouver, 2017. 1215–1226

142 Agić Ž, Koller A. Potsdam: Semantic dependency parsing by bidirectional graph-tree transformations and syntactic parsing. In: Proceedings of the 8th International Workshop on Semantic Evaluation. Dublin, 2014. 465–470

143 Schluter N, Søgaard A, Elming J, et al. Copenhagen-malmö: Tree approximations of semantic parsing problems. In: Proceedings of the 8th International Workshop on Semantic Evaluation. Dublin, 2014. 213–217

144 Agić Ž, Koller A, Oepen S. Semantic dependency graph parsing using tree approximations. In: Proceedings of the 11th International Conference on Computational Semantics. London, 2015

145 McClosky D, Charniak E, Johnson M. Reranking and self-training for parser adaptation. In: Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics. Sydney, 2006. 337–344

146 Sagae K. Self-training without reranking for parser domain adaptation and its impact on semantic role labeling. In: Proceedings of Workshop on Domain Adaptation for Natural Language Processing. Uppsala, 2010. 37–44

147 Kawahara D, Uchimoto K. Learning reliability of parses for domain adaptation of dependency parsing. In: Proceedings of Third International Joint Conference on Natural Language Processing. Hyderabad, 2008

148 Chen W, Wu Y, Isahara H. Learning reliable information for dependency parsing adaptation. In: Proceedings of the 22nd International Conference on Computational Linguistics. Manchster, 2008. 113–120

149 Yu J, Elkaref M, Bohnet B. Domain adaptation for dependency parsing via self-training. In: Proceedings of the 14th International Conference on Parsing Technologies. Bilbao, 2015. 1–10

150 Steedman M, Hwa R, Clark S, et al. Example selection for bootstrapping statistical parsers. In: Proceedings of Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics. Edmonton, 2003. 236–243

151 Sagae K, Tsujii J. Dependency parsing and domain adaptation with LR models and parser ensembles. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. Prague, 2007. 1044–1050

152 Plank B and van Noord G. Effective measures of domain similarity for parsing. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. Portland, 2011. 1566–1576

153 Yang H, Zhuang T, Zong C. Domain adaptation for syntactic and semantic dependency parsing using deep belief networks. Trans Assoc Comput Linguist, 2015, 3: 271–282

154 McClosky D, Charniak E, Johnson M. Automatic domain adaptation for parsing. In: Proceedings of Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics. Los Angeles, 2010. 28–36

155 Reichart R, Rappoport A. Self-training for enhancement and domain adaptation of statistical parsers trained on small datasets. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics. Prague, 2007. 616–623

156 Daumé III H. Frustratingly easy domain adaptation. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics. Prague, 2007. 256–263

157 Finkel J R, Manning C D. Hierarchical Bayesian domain adaptation. In: Proceedings of Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics. Boulder, 2009. 602–610

158 Ganin Y, Lempitsky V. Unsupervised domain adaptation by backpropagation. In: Proceedings of the 32nd International Conference on Machine Learning. Lille, 2015. 1180–1189

159 Sano M, Manabe H, Noji H, et al. Adversarial training for crossdomain universal dependency parsing. In: Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. Vancouver, 2017. 71–79

160 Joshi V, Peters M, Hopkins M. Extending a parser to distant domains using a few dozen partially annotated examples. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics. Melbourne, 2018. 1190–1199

161 Flannery D, Mori S. Combining active learning and partial annotation for domain adaptation of a japanese dependency parser. In: Proceedings of the 14th International Conference on Parsing Technologies. Bilbao, 2015. 11–19

162 Li Z, Peng X, Zhang M, et al. Semi-supervised domain adaptation for dependency parsing. In: Proceedings of the 57th Conference of the Association for Computational Linguistics. Florence, 2019. 2386–2395

163 Zeman D, Resnik P. Cross-language parser adaptation between related languages. In: Proceedings of the Third International Joint Conference on Natural Language Processing-08 Workshop on NLP for Less Privileged Languages. Hyderabad, 2008

164 McDonald R, Petrov S, Hall K. Multi-source transfer of delexicalized dependency parsers. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing. Edinburgh, 2011. 62–72

165 Cohen S B, Das D, Smith N A. Unsupervised structure prediction with non-parallel multilingual guidance. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing. Edinburgh, 2011. 50–61

166 Naseem T, Barzilay R, Globerson A. Selective sharing for multilingual dependency parsing. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics. Jeju Island, 2012. 629–637

167 Täckström O, McDonald R, Uszkoreit J. Cross-lingual word clusters for direct transfer of linguistic structure. In: Proceedings of the Conference of the North American Chapter of the Association of Computational Linguistics: Human Language Technologies. Montreal, 2012. 477–487

168 Guo J, Che W, Yarowsky D, et al. Cross-lingual dependency parsing based on distributed representations. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and

the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing. Beijing, 2015. 1234–1244

169 Zhang Y, Barzilay R. Hierarchical low-rank tensors for multilingual transfer parsing. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Lisbon, 2015. 1857–1867

170 Guo J, Che W, Yarowsky D, et al. A representation learning framework for multi-source transfer parsing. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence. Phoenix, 2016

171 Wick M, Kanani P, Pocock A. Minimally-constrained multilingual embeddings via artificial code-switching. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence. Phoenix, 2016

172 Schuster T, Ram O, Barzilay R, et al. Cross-lingual alignment of contextual word embeddings, with applications to zero-shot dependency parsing. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Minneapolis, 2019. 1599–1613

173 Wang Y, Che W, Guo J, et al. Cross-lingual BERT transformation for zero-shot dependency parsing. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing. Hong Kong, 2019. 5721–5727

174 Wu S, Dredze M. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing. Hong Kong, 2019. 833–844

175 Lample G, Conneau A. Cross-lingual language model pretraining. ArXiv: 1901.07291, 2019

176 Wu S, Conneau A, Li H, et al. Emerging cross-lingual structure in pretrained language models. ArXiv: 1911.01464

177 Snyder B, Naseem T, Barzilay R. Unsupervised multilingual grammar induction. In: Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP. Singapore, 2009. 73–81

178 Jiang W, Liu Q, Lv Y. Relaxed cross-lingual projection of constituent syntax. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing. Edinburgh, 2011. 1192–1201

179 Hwa R, Resnik P, Weinberg A, et al. Bootstrapping parsers via syntactic projection across parallel texts. Nat Lang Eng, 2005, 11: 311–325

180 Li Z, Zhang M, Chen W. Soft cross-lingual syntax projection for dependency parsing. In: Proceedings of the 25th International Conference on Computational Linguistics. Dublin, 2014. 783–793

181 Ma X, Xia F. Unsupervised dependency parsing with transferring distribution via parallel guidance and entropy regularization. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. Baltimore, 2014. 1337–1348

182 Schlichtkrull M, Søgaard A. Cross-lingual dependency parsing with late decoding for truly low-resource languages. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics. Valencia, 2017. 220–229

183 Rasooli M S, Collins M. Density-driven cross-lingual transfer of dependency parsers. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Lisbon, 2015. 328–338

184 Agić Ž, Johannsen A, Plank B, et al. Multilingual projection for parsing truly low-resource languages. Trans Assoc Comput Linguist, 2016, 4: 301–312

185 Jiang W, Liu Q, Supnithi T. Joint learning of constituency and dependency grammars by decomposed cross-lingual induction. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence. Austin, 2015

186 Tiedemann J, Agić Ž, and Nivre J. Treebank translation for crosslingual parser induction. In: Proceedings of the 2014 Conference

on Empirical Methods in Natural Language Processing. Doha, 2014. 130–140

187 Tiedemann J, Agić Ž. Synthetic treebanking for cross-lingual dependency parsing. jair, 2016, 55: 209–248

188 Zhang M, Zhang Y, Fu G. Cross-lingual dependency parsing using code-mixed TreeBank. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing. Hong Kong, 2019. 997–1006

189 Rasooli M S, Collins M. Cross-lingual syntactic transfer with limited resources. Trans Assoc Comput Linguist, 2017, 5: 279–293

190 Wang D, Eisner J. Synthetic data made to order: The case of parsing. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Brussels, 2018. 1325–1337

191 Rasooli M S, Collins M. Low-resource syntactic transfer with unsupervised source reordering. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Minneapolis, 2019. 3845–3856

192 Smith D A, Smith N A. Bilingual parsing with factored estimation: Using english to parse korean. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing. Barcelona, 2004. 49–56

193 Burkett D, Klein D. Two languages are better than one (for syntactic parsing). In: Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing. Honolulu, 2008. 877–886

194 Ammar W, Mulcaire G, Ballesteros M, et al. Many languages, one parser. Trans Assoc Comput Linguist, 2016, 4: 431–444

195 Smith A, Bohnet B, de Lhoneux M, et al. 82 treebanks, 34 models: Universal dependency parsing with multi-treebank models. In: Proceedings of the 22nd Conference on Computational Natural Language Learning. Brussels, 2018. 113–123

196 Kondratyuk D, Straka M. 75 languages, 1 model: Parsing universal dependencies universally. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing. Hong Kong, 2019. 2779–2795

197 Li J, Zhou G, Ng H T. Joint syntactic and semantic parsing of Chinese. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. Uppsala, 2010. 1108–1117

198 Wang Z, Xue N. Joint POS tagging and transition-based constituent parsing in Chinese with non-local features. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. Baltimore, 2014. 733–742

199 Li Z, Zhang M, Che W, et al. Joint models for Chinese POS tagging and dependency parsing. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing. Edinburgh, 2011. 1180–1191

200 Li Z, Zhang M, Che W, et al. A separately passive-aggressive training algorithm for joint POS tagging and dependency parsing. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. Jeju Island, 2012. 1681–1698

201 Hatori J, Matsuzaki T, Miyao Y, et al. Incremental joint POS tagging and dependency parsing in Chinese. In: Proceedings of Fifth International Joint Conference on Natural Language Processing. Chiang Mai, 2011. 1216–1224

202 Bohnet B, Nivre J. A transition-based system for joint part-of-speech tagging and labeled non-projective dependency parsing. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. Jeju Island, 2012. 1455–1465

203 Alberti C, Weiss D, Coppola G, et al. Improved transition-based parsing and tagging with neural networks. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing.

Lisbon, 2015. 1354–1359

204 Zhang Y, Weiss D. Stack-propagation: Improved representation learning for syntax. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. Berlin, 2016. 1557–1566

205 Yang L, Zhang M, Liu Y, et al. Joint pos tagging and dependence parsing with transition-based neural networks. IEEE/ACM Trans Audio Speech Language Process, 2017, 26: 1352–1358

206 Luo X. A maximum entropy Chinese character based parser. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. Sapporo, 2003. 192–199

207 Zhao H. Character-level dependencies in Chinese: Usefulness and learning. In: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics. Athens, 2009. 879–887

208 Hatori J, Matsuzaki T, Miyao Y, et al. Incremental joint approach to word segmentation, POS tagging, and dependency parsing in Chinese. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics. Jeju Island, 2012. 1045–1053

209 Li Z, Zhou G. Unified dependency parsing of Chinese morphological and syntactic structures. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. Jeju Island, 2012. 1445–1454

210 Zhang M, Zhang Y, Che W, et al. Chinese parsing exploiting characters. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics. Sofia, 2013. 125–134

211 Zhang M, Zhang Y, Che W, et al. Character-level Chinese dependency parsing. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. Baltimore, 2014. 1326–1336

212 Zhang Y, Li C, Barzilay R, et al. Randomized greedy inference for joint segmentation, POS tagging and dependency parsing. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Denver, 2015. 42–52

213 Zheng X, Peng H, Chen Y, et al. Character-based parsing with convolutional neural network. In: Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence. Buenos Aires, 2015

214 Li H, Zhang Z, Ju Y, et al. Neural character-level dependency parsing for chinese. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence. New Orleans, 2018

215 Yan H, Qiu X, Huang X. A unified model for joint chinese word segmentation and dependency parsing. ArXiv: 1904.04697

216 Johansson R, Nugues P. Dependency-based semantic role labeling of PropBank. In: Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing. Honolulu, 2008. 69–78

217 Strubell E, Verga P, Andor D, et al. Linguistically-informed self-attention for semantic role labeling. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Brussels, 2018. 5027–5038

218 Zhang M, Zhang J, Su J. Exploring syntactic features for relation extraction using a convolution tree kernel. In: Proceedings of Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics. New York, 2006. 288–295

219 Miwa M, Bansal M. End-to-end relation extraction using LSTMs on sequences and tree structures. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. Berlin, 2016. 1105–1116

220 Tai K S, Socher R, Manning C D. Improved semantic representations from tree-structured long short-term memory networks. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natu-

ral Language Processing of the Asian Federation of Natural Language Processing. Beijing, 2015. 1556–1566

221 Zhang M, Li Z, Fu G, et al. Syntax-enhanced neural machine translation with syntax-aware word representations. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Minneapolis, 2019. 1151–1161

222 Liu Y, Liu Q, Lin S. Tree-to-string alignment template for statistical machine translation. In: 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics. Sydney, 2006. 609–616

223 Qiu L, Zhang Y. ZORE: A syntax-based system for Chinese open relation extraction. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. Doha, 2014. 1870–1880

224 Che W, Zhang M, Liu T, et al. A hybrid convolution tree kernel for semantic role labeling. In: 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics. Sydney, 2006. 73–80

225 Yang X, Su J, Tan C L. Kernel-based pronoun resolution with structured syntactic knowledge. In: 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics. Sydney, 2006. 41–48

226 Zhou G, Zhang M, Ji D H, et al. Tree kernel-based relation extraction with context-sensitive structured parse tree information. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. Prague, 2007. 728–736

227 Zhang M, Li H. Tree kernel-based SVM with structured syntactic knowledge for BTG-based phrase reordering. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing. Singapore, 2009. 698–707

228 Johansson R, Nugues P. The effect of syntactic representation on semantic role labeling. In: Proceedings of the 22nd International Conference on Computational Linguistics. Manchster, 2008. 393–400

229 Xu Y, Mou L, Li G, et al. Classifying relations via long short term memory networks along shortest dependency paths. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Lisbon, 2015. 1785–1794

230 Roth M, Lapata M. Neural semantic role labeling with dependency path embeddings. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. Berlin, 2016. 1192–1202

231 Mou L, Peng H, Li G, et al. Discriminative neural sentence modeling by tree-based convolution. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Lisbon, 2015. 2315–2325

232 Zhang X, Lu L, Lapata M. Top-down tree long short-term memory networks. In: Proceedings of The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. San Diego, 2016. 310–320

233 Teng Z, Zhang Y. Head-lexicalized bidirectional tree LSTMs. Trans Assoc Comput Linguist, 2017, 5: 163–177

234 Li J, Xiong D, Tu Z, et al. Modeling source syntax for neural machine translation. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. Vancouver, 2017. 688–697

235 Wu S, Zhou M, Zhang D. Improved neural machine translation with source syntax. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence. Melbourne, 2017. 4179–4185

236 Zhang M, Zhang Y, Fu G. End-to-end neural relation extraction with global optimization. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. Copenhagen, 2017. 1730–1740

237 Yu N, Zhang M, Fu G. Transition-based neural RST parsing with implicit syntax features. In: Proceedings of the 27th International Conference on Computational Linguistics. Santa Fe, 2018. 559–570

238 Bastings J, Titov I, Aziz W, et al. Graph convolutional encoders for syntax-aware neural machine translation. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. Copenhagen, 2017. 1957–1967

239 Zhang Y, Qi P, Manning C D. Graph convolution over pruned dependency trees improves relation extraction. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Brussels, 2018. 2205–2215

240 Marcheggiani D, Bastings J, Titov I. Exploiting semantics in neural machine translation with graph convolutional networks. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Minneapolis, 2018. 486–492

241 Marcus M P, Santorini B, Marcinkiewicz M A. Building a large annotated corpus of English: The Penn Treebank. Comput Linguist, 1993, 19: 313–330

242 Xue N, Xia F, Chiou F D, et al. The penn chinese treebank: Phrase structure annotation of a large corpus. NLE, 2005, 11: 207–238

243 Johansson R, Nugues P. Extended constituent-to-dependency conversion for english. In: Proceedings of the 16th Nordic Conference of Computational Linguistics. Tartu, 2007. 105–112

244 Johansson R, Nugues P. Lth: semantic structure extraction using non-projective dependency trees. In: Proceedings of the 4th International Workshop on Semantic Evaluations. Prague, 2007. 227–230

245 Zhenghua L, Wanxiang C, Ting L. A study on the conversion of phrase-structural trees into dependencies. J Chin Inform Process, 2008, 22: 14–19

246 De Marneffe M C, MacCartney B, Manning C D, et al. Generating typed dependency parses from phrase structure parses. In: Proceedings of the Fifth International Conference on Language Resources and Evaluation. Genoa, 2006. 449–454

247 Tateisi Y, Yakushiji A, Ohta T, et al. Syntax annotation for the genia corpus. In: Proceedings of Natural Language Processing - IJCNLP 2005, Second International Joint Conference. Jeju Island, 2005. 222–227

248 Kong L, Schneider N, Swayamdipta S, et al. A dependency parser for tweets. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. Doha, 2014. 1001–1012

249 De Marneffe M C, Dozat T, Silveira N, et al. Universal stanford dependencies: A cross-linguistic typology. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation. Reykjavik, 2014. 4585–4592

250 Petrov S, Das D, McDonald R. A universal part-of-speech tagset. In: Proceedings of the Eighth International Conference on Language Resources and Evaluation. Istanbul, 2012

251 McDonald R, Nivre J, Quirmbach-Brundage Y, et al. Universal dependency annotation for multilingual parsing. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics. Sofia, 2013. 92–97

252 Nivre J, Agić Ž, Aranzabe M J, et al. Universal dependencies 1.2. 2015

253 Nivre J, Agić Ž, Ahrenberg L, et al. Universal dependencies 2.1. 2017

254 Chen K J, Luo C C, Chang M C, et al. Sinica treebank: Design criteria, representational issues and implementation, chapter 13, 2003

255 Qiang Z. Annotation scheme for Chinese treebank. J Chin Infor Proc, 2004, 18: 1–8

256 Zhan W. The application of treebank to assist chinese grammar instruction: A preliminary investigation. J Tech Chin Language Teach, 2012, 3: 16–29

257 Liu T, Ma J, Li S. Building a dependency treebank for improving chinese parser. J Chin Language Comput, 2006, 16: 207–224

258 Che W, Li Z, Liu T. Chinese Dependency Treebank 1.0 ldc2012t05. Philadelphia: Linguistic Data Consortium, 2012

259 Qiu L, Zhang Y, Jin P, et al. Multi-view Chinese treebanking. In: Proceedings of the 25th International Conference on Computational Linguistics. Dublin, 2014. 257–268

260 Qiu L, Zhang Y, Zhang M. Dependency tree representations of predicate-argument structures. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence. Phoenix, 2016

261 Buchholz S, Marsi E. Conll-x shared task on multilingual dependency parsing. In: Proceedings of the Tenth Conference on Computational Natural Language Learning. New York, 2006. 149–164

262 Nivre J, Hall J, Kübler S, et al. The CoNLL 2007 shared task on dependency parsing. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. Prague, 2007. 915–932

263 Zeman D, Popel M, Straka M, et al. CoNLL 2017 shared task: Multilingual parsing from raw text to universal dependencies. In: Proceedings of the 21st Conference on Computational Natural Language Learning. Vancouver, 2017. 1–19

264 Zeman D, Hajič J, Popel M, et al. CoNLL 2018 shared task: Multilingual parsing from raw text to universal dependencies. In: Proceedings of the 22nd Conference on Computational Natural Language Learning. Brussels, 2018. 1–21

265 Petrov S, McDonald R. Overview of the 2012 shared task on parsing the web. In: Proceedings of First Workshop on Syntactic Analysis of Non-Canonical Language (SANCL). Montreal, 2012

266 Peng X, Li Z, Zhang M, et al. Overview of the nlpcc 2019 shared task: cross-domain dependency parsing. In: Proceedings of Natural Language Processing and Chinese Computing - 8th CCF International Conference. Dunhuang, 2019. 760–771

267 Che W, Zhang M, Shao Y, et al. Semeval-2012 task 5: Chinese semantic dependency parsing. In: Proceedings of the 6th International Workshop on Semantic Evaluation. Montreal, 2012. 378–384

268 Oepen S, Abend O, Hajic J, et al. MRP 2019: Cross-framework meaning representation parsing. In: Proceedings of the Shared Task on Cross-Framework Meaning Representation Parsing at the 2019 Conference on Natural Language Learning. Hong Kong, 2019. 1–27