Neural Maximum Subgraph Parsing for Cross-Domain Semantic Dependency Analysis

Yufei Chen[♠], **Sheng Huang**[♠], **Fang Wang**[♠], **Weiwei Sun**[♠♡] and **Xiaojun Wan**[♠]
[♠] Institute of Computer Science and Technology, Peking University
[♠] The MOE Key Laboratory of Computational Linguistics, Peking University
[♡] Center for Chinese Linguistics, Peking University
{yufei.chen,huangsheng,foundwang,ws,wanxiaojun}@pku.edu.cn

Abstract

We present experiments for cross-domain semantic dependency analysis with a neural Maximum Subgraph parser. Our parser targets 1-endpoint-crossing, pagenumber-2 graphs which are a good fit to semantic dependency graphs, and utilizes an efficient dynamic programming algorithm for decoding. For disambiguation, the parser associates words with BiLSTM vectors and utilizes these vectors to assign scores to candidate dependencies. We conduct experiments on the data sets from SemEval 2015 as well as Chinese CCGBank. Our parser achieves very competitive results for both English and Chinese. To improve the parsing performance on cross-domain texts, we propose a data-oriented method to explore the linguistic generality encoded in English Resource Grammar, which is a precision-oriented, hand-crafted HPSG grammar, in an implicit way. Experiments demonstrate the effectiveness of our data-oriented method across a wide range of conditions.

1 Introduction

Semantic Dependency Parsing (SDP) is defined as the task of recovering sentence-internal bilexical semantic dependency structures, which encode predicate–argument relationships for all content words. Such sentence-level semantic analysis of text is concerned with the characterization of events and is therefore important to understand the essential meaning of a natural language sentence. With the advent of many supporting resources, SDP has become a well-defined task with a substantial body of work and comparative evaluation. (Almeida and Martins, 2015; Du et al., 2015a; Zhang et al., 2016; Peng et al., 2017; Wang et al., 2018). Two SDP shared tasks have been run as part of the 2014 and 2015 International Workshops on Semantic Evaluation (SemEval) (Oepen et al., 2014, 2015).

There are two key dimensions of the data-driven dependency parsing approach: decoding and disambiguation. Existing decoding approaches to syntactic or semantic analysis into bilexical dependencies can be categorized into two dominant types: transition-based (Zhang et al., 2016; Wang et al., 2018) and graph-based, i.e., Maximum Subgraph (Kuhlmann and Jonsson, 2015; Cao et al., 2017a) approaches. For disambiguation, while early work on dependency parsing focused on global linear models, e.g., structured perceptron (Collins, 2002), recent work shows that deep learning techniques, e.g., LSTM (Hochreiter and Schmidhuber, 1997), is able to significantly advance the state-of-the-art of the parsing accuracy. From the above two perspectives, i.e., the decoding and disambiguation frameworks, we find that what is still underexploited is neural Maximum Subgraph parsing for highly constrained graph classes, e.g., noncrossing graphs. In this paper, we fill this gap in the literature by developing a neural Maximum Subgraph parser.

Previous work showed that the 1-endpoint-crossing, pagenumber-2 (1EC/P2) graphs are an appropriate graph class for modeling semantic dependency structures (Cao et al., 2017a). In this paper, we build a parser that targets 1EC/P2 graphs. Based on an efficient first-order Maximum Subgraph decoder, we implement a data-driven parser that scores arcs based on stacked bidirectional-LSTM (BiLSTM) together with a multi-layer perceptron. Using the benchmark data sets from the SemEval 2015 Task 18 (Oepen et al., 2015), our parser gives very competitive results for English semantic parsing. To test the ability for cross-lingual parsing, we also conduct experiments on the Chinese CCGBank (Tse and Curran, 2010) and Enju HPSGBank (Yu et al., 2010) data. Our parser plays equally well for Chinese, resulting in an error reduction of 23.5% and 9.4% over the best

Proceedings of the 22nd Conference on Computational Natural Language Learning (CoNLL 2018), pages 562–572
Brussels, Belgium, October 31 - November 1, 2018. ©2018 Association for Computational Linguistics

published result reported in Zhang et al. (2016) and Du et al. (2015b).

Most studies on semantic parsing focused on the in-domain setting, meaning that both training and testing data are drawn from the same domain. Even a data-driven parsing system achieves a high in-domain accuracy, it usually performs rather poorly on the out-of-domain data (Oepen et al., 2015). How to build robust semantic dependency parsers that can learn across domains remains an under-addressed problem. To improve the cross-domain parsing performance, we propose a data-oriented model to explore the linguistic generality encoded in a hand-crafted, domain-independent, linguistically-precise English grammar, namely English Resource Grammar (ERG; Flickinger, 2000). In particular, we introduce a cost-sensitive training model to learn cross-domain semantic information implicitly encoded in WikiWoods (Flickinger et al., 2010), i.e., a corpus that collects the wikipedia[1] texts as well as their automatic syntactico-semantic annotations produced by ERG. Evaluation demonstrates the usefulness of the imperfect annotations automatically created by ERG.

Our parser is available at `https://github.com/draplater/msg-parser`.

2 Semantic Dependency Parsing

2.1 Semantic Dependency Analysis

SDP is the task of mapping a natural language sentence into a formal meaning representation in the form of a dependency graph. Figure 1 shows an Minimal Recursion Semantics (MRS; Copestake et al., 2005) reduced semantic dependency analysis (Ivanova et al., 2012). In this example, the semantic analysis is represented as a labeled directed graph in which the vertices are tokens in the sentence. The graph abstracts away from syntactic analysis (e.g., the complementizer—*that*—and passive construction are excluded) and includes most semantically relevant non-anaphoric local (e.g., from "wants" to "Mark") and long-distance (e.g., from "buy" to "company") dependencies. The arc labels encode linguistically-motivated, broadly-applicable semantic relations that are grounded under the type-driven semantics. It is worth noting that semantic dependency graphs are not necessarily trees: (1) a token may be multiply headed because a word can be the arguments of more than one predicate; (2) cycles are allowed if the direction of arcs are not taken into account.

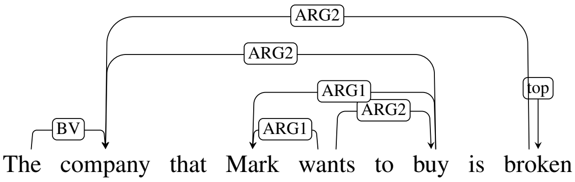

Figure 1: A fragment of a semantic dependency graph.

2.2 Previous Work

Some recent work on parsing targets the graph-structured semantic representations that are more general than the tree representation. Existing approaches can be categorized into two dominant types: the transition-based (Zhang et al., 2016; Wang et al., 2018) and graph-based, i.e., Maximum Subgraph (Kuhlmann and Jonsson, 2015; Cao et al., 2017a), approaches. Previous investigations on transition-based string-to-semantic-graph parsing adopt many ideas from syntactic string-to-tree parsing, such as how to handle crossing arcs and how to perform neural disambiguation. Zhang et al. (2016) introduced two transition systems that can generate arbitrary graphs and augmented them into practical semantic dependency parsers with a structured perceptron model. Wang et al. (2018) evaluated the effectiveness of deep learning techniques for transition-based SDP.

Kuhlmann and Jonsson (2015) proposed to formulate SDP as the search for the maximum subgraphs for some particular graph classes. This proposal is called Maximum Subgraph parsing, which is a generalization of the graph-based parsing framework for syntactic parsing. For arbitrary graphs, Du et al. (2015a) proved that the second-order Maximum Subgraph problem is an NP-hard problem. Nevertheless, Almeida and Martins (2015) and Du et al. (2015a) showed that dual decomposition is a practical technique to solve the problem. Considering more restricted graph classes, Kuhlmann and Jonsson (2015) introduced a dynamic programming algorithm for parsing to noncrossing graphs. Cao et al. (2017a; 2017b) showed that 1EC/P2 graphs are more suitable for describing semantic graphs than the noncrossing graphs, and they also allow low-degree dynamic programming algorithms for decoding.

[1] `https://www.wikipedia.org`

3 A Neural Maximum Subgraph Parser

3.1 Maximum Subgraph Parsing

Usually, syntactic dependency analysis employs the *tree*-shaped representation. Dependency parsing, thus, can be formulated as the search for a maximum spanning tree (MST) from an arc-weighted (complete) graph. For SDP where the target representation are no longer trees, Kuhlmann and Jonsson (2015) proposed to generalize the MST model to other types of subgraphs. In general, dependency parsing is formulated as the search for Maximum Subgraph regarding to a particular graph class, viz. $\mathcal{G}$: Given a graph $G = (V, A)$, find a subset $A' \subseteq A$ with maximum total weight such that the induced subgraph $G' = (V, A')$ belongs to $\mathcal{G}$. Formally, we have the following optimization problem:

$$\begin{aligned} G'(s) &= \arg \max_{H \in \mathcal{G}(s,G)} \textsc{Score}(H) \\ &= \arg \max_{H \in \mathcal{G}(s,G)} \sum_{p \text{ in } H} \textsc{ScorePart}(s, p) \end{aligned} \tag{1}$$

Here, $\mathcal{G}(s, G)$ is the set of all graphs that belong to $\mathcal{G}$ and are compatible with s and G. For parsing, G is usually a complete graph. $\textsc{ScorePart}(s, p)$ evaluates whether a small subgraph p of a candidate graph H is a good partial analysis for sentence s.

For some graph classes and some types of score functions, there exists efficient algorithms for solving (1). For example, when $\mathcal{G}$ is the set of noncrossing graphs and SCOREPART is limited to handle individual dependencies, (1) can be solved in cubic-time (Kuhlmann and Jonsson, 2015).

3.2 Parsing to 1EC/P2 Graphs

Previous work showed that the Maximum Subgraph framework is not only elegant in theory but also effective in practice (Kuhlmann and Jonsson, 2015; Cao et al., 2017a,b). In particular, 1EC/P2 graphs are an appropriate graph class for modeling semantic dependency structures (Cao et al., 2017a). Figure 2 presents an example to illustrate the 1-endpoint-crossing property, while Figure 3 shows a case for pagenumber-2. Below we present the formal description of the two properties that are adopted from Pitler et al. (2013) and Kuhlmann and Jonsson (2015) respectively.

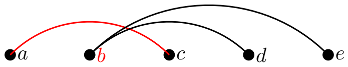

Figure 2: (a, c)'s crossing edges (b, d) and (b, e) share an endpoint b.

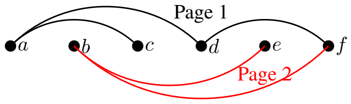

Figure 3: A pagenumber-2 graph. The upper and the lower figures represent two half-planes respectively.

Definition 1 *A dependency graph is 1-Endpoint-Crossing if for any edge e, all edges that cross e share an endpoint p named pencil point.*

Definition 2 *A pagenumber-k graph means it consists at most k half-planes, and arcs on each half-plane are noncrossing.*

If $\mathcal{G}$ is the set of 1-endpoint-crossing graphs or more restricted 1EC/P2 graphs, the optimization problem (1) in the first-order case can be solved in quintic-time (Cao et al., 2017a) by using dynamic programming. Furthermore, ignoring one linguistically-rare structure in 1EC/P2 graphs descreases the complexity to $O(n^4)$ (Cao et al., 2017a). In this paper, we implement Cao et al. Cao et al. (2017a)'s algorithm as the basis of our parser.

3.3 Disambiguation with an LSTM

3.3.1 The Architecture

A semantic graph mainly consists of two parts: the structural part and the label part. The former describes the predicate–argument relation in the sentence, and the latter describes the type of this relation. In our model, the structural part and the label part are regarded as independent of each other. We use a coarse-to-fine strategy: finding the maximum unlabeled subgraph first and assigning a label for every edge in this subgraph then. The motivation is to avoid the calculation of a number of unnecessary label scores in order to improve the processing efficiency.

Following Kiperwasser and Goldberg (2016)'s successful experience on syntactic tree parsing and Peng et al. (2017)'s experience on semantic graph parsing, we employ a stacked bidirectional-LSTM (BiLSTM) based model to assign scores. In our system, the BiLSTM vectors associated with the input words are utilized to calculate scores for the

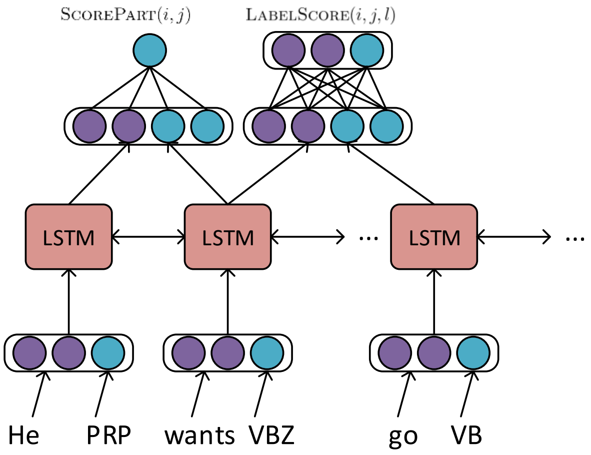

Figure 4: The architecture of the network when processing *He wants to go*. The upper-left nonlinear transform is used for edge scoring while the upper right one is used for label scoring.

candidate dependencies as well as their relation types. Figure 4 shows the architecture of our system.

3.3.2 Dense Representations

We use words as well as POS tags as clues for scoring an individual arc. In particular, we transform all of them into continuous and dense vectors. Inspired by Costa-jussà and Fonollosa (2016)'s work, we utilize character-based embedding for low-frequency words, i.e., words that appear more than k times in the training data, and word-based embeddings for other words. The word-based embedding module applies the common lookup-table mechanism, while the character-based word embedding $\boldsymbol{w}_i$ is implemented by extracting the features (denoted as $\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_n$) within a character-based BiLSTM:

$$\boldsymbol{x}_1 : \boldsymbol{x}_n = \text{BiLSTM}(\boldsymbol{c}_1 : \boldsymbol{c}_n)$$

$$\boldsymbol{w_i} = \boldsymbol{x}_1 + \boldsymbol{x}_n$$

3.3.3 Lexical Feature Extractor

The concatenation of word embedding $\boldsymbol{w}_i$ and POS-tag embedding $\boldsymbol{p}_i$ of each word in specific sentence is used as the input of BiLSTMs to extract context-related feature vectors r_i for each position i.

$$\boldsymbol{a}_i = \boldsymbol{w}_i \oplus \boldsymbol{p}_i$$

$$\boldsymbol{r}_1 : \boldsymbol{r}_n = \text{BiLSTM}(\boldsymbol{a}_1 : \boldsymbol{a}_n)$$

3.3.4 Factorized Scoring

In our first order model, the SCORE function evaluates the preference of a semantic dependency graph by considering every bilexical relation in this graph one by one. In particular, the corresponding SCOREPART function assigns a score to a candidate arc between word i and word j using a non-linear transform from the two feature vectors, viz. r_i and r_j, associated to the two words:

$$\text{SCOREPART}(i, j) = \\ \boldsymbol{W}_2 \cdot \text{ReLU}(\boldsymbol{W}_{1,1} \cdot \boldsymbol{r}_i + \boldsymbol{W}_{1,2} \cdot \boldsymbol{r}_j + \boldsymbol{b})$$

The assignment task for dependency labels can be regarded as a classification task. Our label scoring process is similar to the prediction of dependencies:

$$\text{LABEL}(i, j) = \arg\max \\ \boldsymbol{W}_2 \cdot \text{ReLU}(\boldsymbol{W}_{1,1} \cdot \boldsymbol{r}_i + \boldsymbol{W}_{1,2} \cdot \boldsymbol{r}_j + \boldsymbol{b}) + \boldsymbol{b}_2$$

We can see here the two *local* score functions explicitly utilize the positions of a semantic head and a semantic dependent. It is similar to the first-order factorization as defined in a number of linear parsing models, e.g., the models defined by Martins and Almeida (2014) and Cao et al. (2017a).

3.3.5 Training

In order to update graphs which achieve high model scores but are actually wrong, we use a margin-based approach to compute loss from the gold graph G^* and the best prediction $\hat{G}$ under current model. We define the *loss* term as:

$$\max(0, \Delta(G^*, \hat{G}) - \text{SCORE}(G^*) + \text{SCORE}(\hat{G}))$$

The margin objective Δ measures the similarity between the gold graph G^* and the prediction $\hat{G}$. Follow Peng et al. (2017)'s approach, we define Δ as weighted Hamming to trade off between precision and recall.

4 Cross-Domain Parsing with a Precision Grammar and a Data-Oriented Model

4.1 Precision Grammar-Guided Parsing

Semantic dependency graphs like Minimal Recursion Semantics (MRS) reduced analysis (dubbed DM) and Head-driven Phrase Structure Grammar (HPSG) grounded predicate–argument analysis (dubbed PAS) are derived from the linguistic analysis licensed by a deep linguistic grammar.

They are parallel with the deep syntactic analysis, and the semantic construction process of them is strictly compositional. Another type of domain-independent, sentence-level semantic annotations are based on annotators' reflection of the meanings of particular natural language sentences. No syntactic constraints on linguistic signals are introduced explicitly introduced. A representative example is Abstract Meaning Representation (AMR; Banarescu et al., 2013).

Different from data-driven syntactic parsing, semantic parsing for the first type of annotation can leverage a precision grammar-guided model. Such a model applies a rich set of precise linguistic rules to constrain their search for a preferable syntactic or semantic analysis. In recent years, several of these linguistically motivated parsing systems achieved high performances that are comparable or even superior to the treebank-based purely data-driven parsers. For example, using ERG (Flickinger, 2000), which provides precise linguistic analyses for a broad range of phenomena, as the the core engine, PET[2] (Callmeier, 2000) and ACE[3] produce better results than all existing data-driven semantic parsers for sentences that can be parsed by ERG.

The main weakness of the precision grammar-guided parsers is their robustness with respect to both coverage and efficiency. Even for treebanking on the newswire data, i.e., the Wall Street Journal data from Penn TreeBank, ERG lacks analyses for c.a. 11% sentences (Oepen et al., 2015). For the texts from the web, e.g., tweets, this problem is much more serious. Moreover, checking all linguistic constraints makes a grammar-guided parser too slow for many realistic NLP applications. On the contrary, light-weight, data-driven parsers usually have complementary strengthes in terms of both coverage and efficiency.

4.2 The Parser-Oriented Model

Intuitively, a hand-crafted precision grammar, e.g., ERG, reflects highly generalized properties of a particular language and is thus highly resilient to domain shifts. Accordingly, one should expect that a precision grammar-guided parser which guarantees the a rich set of domain-independent linguistic constraints to be met can be more robust to domain shifts than a purely data-driven parser. In related work for syntactic parsing, Ivanova et al. (2013) showed that the ERG-based parser was more robust to domain variation than several representative data-driven parsers.

Zhang and Wang (2009) proposed to derive features from syntactic parses generated by PET to assist a data-driven dependency tree parser and observed some encouraging results for cross-domain evaluation. However, there are at least two drawbacks of their ERG-guided parser based method:

1. A considerable number of sentences cannot benefit from ERG since PET may produce no analysis.

2. This method fails to take parsing efficiency into account.

4.3 Our Data-Oriented Model

In this paper, we introduce a new data-oriented strategy to consume a precision grammar. The key idea is to take a grammar as an imperfect annotator: We let a precision grammar-guided parser parse large-scale raw texts in an *offline* way, and then utilize the automatically generated analysis as imperfect training data. Because we only need raw texts to be parsed once, even if this process takes much time, it is still reasonable. A grammar-guided parser cannot parse a considerable portion of data, but this will not cause serious problems because we can take an enormous amount of sentences as annotation candidates. Just considering the wikipedia, we can collect at least dozens of millions of comparatively high-quality sentences.

An essential problem of this method is that such imperfect annotations bring in annotation errors which may hurt parser training. To deal with this problem, we adopted a cost-sensitive training method to train our model on the extended training data. In each epoch, we trained on imperfect corpus first and then on gold-standard corpus. When processing an imperfect sentence, we do not take a loss into consideration if the loss of this sentence is too small. In particular, if a loss of a bilexical relation between two tokens is less than 0.05, we would exclude the loss. As for label assigning, we exclude losses less than 0.5. These threshold numbers are tuned on the development data.

[2] http://pet.opendfki.de/

[3] http://sweaglesw.org/linguistics/ace/

	System		**DM**			**PAS**			**PSD**		
			LP	LR	LF	LP	LR	LF	LP	LR	LF
IN-DOMAIN	Du et al.	ensemble	90.93	87.32	89.09	92.90	89.67	91.26	78.60	72.93	75.66
	Almeida and Martins	single	89.84	86.64	88.21	91.87	89.92	90.88	78.62	74.23	76.36
	Peng et al.	single	- -	- -	89.4	- -	- -	92.2	- -	- -	**77.6**
	Peng et al.	multitask	- -	- -	90.4	- -	- -	92.7	- -	- -	78.5
	Wang et al.	single	- -	- -	89.3	- -	- -	91.4	- -	- -	76.1
	Wang et al.	ensemble	- -	- -	90.3	- -	- -	91.7	- -	- -	78.6
	Ours	single	90.74	90.40	**90.57**	92.26	92.43	**92.35**	76.42	76.33	76.38
	Ours (E[3])	ensemble	92.17	91.35	91.76	93.50	92.98	93.24	78.83	77.07	77.95
	Ours ([E10])	ensemble	92.81	91.65	**92.23**	93.91	93.22	**93.56**	79.33	78.00	**78.66**
OUT-OF-DOMAIN	Du et al.	ensemble	84.29	79.53	81.84	89.47	85.10	87.23	77.36	69.61	73.28
	Almeida and Martins	single	84.81	78.90	81.75	88.52	85.30	86.88	78.68	71.31	74.82
	Peng et al.	single	- -	- -	84.5	- -	- -	88.3	- -	- -	75.3
	Peng et al.	multitask	- -	- -	85.3	- -	- -	89.0	- -	- -	**76.4**
	Wang et al.	single	- -	- -	83.2	- -	- -	87.2	- -	- -	73.2
	Wang et al.	ensemble	- -	- -	84.9	- -	- -	87.6	- -	- -	75.9
	Ours	single	85.70	85.02	**85.37**	89.11	88.85	**88.98**	73.54	73.19	73.36
	Ours (E[3])	ensemble	87.65	86.24	86.94	90.72	89.31	90.01	76.10	73.83	74.95
	Ours (E[10])	ensemble	88.13	86.37	**87.24**	91.19	89.50	**90.34**	76.75	74.48	75.60

Table 1: Labeled F_1 on the test data from SemEval 2015.

Hyper-parameter	Val
Randomly-initialized word embedding dimension	100
Pre-trained word embedding dimension	100
Randomly-initialized character embedding dimension	100
Character LSTM layers for each direction	2
Randomly-initialized POS-Tag embedding dimension	50
POS-Tag dropout	0.5
Batch size	32
BiLSTM dimension for each direction	150
BiLSTM layers	5
MLP hidden layers	1
MLP hidden layer dimension	100

Table 2: Hyper-parameter setting of our model.

5 Experiments

5.1 Set-up for the Baseline System

To evaluate neural Maximum Subgraph parsing in practice, we first conduct experiments on the three English data sets, namely DM, PAS and PSD[4], which are from the SemEval 2015 Task18 (Oepen et al., 2015). We use the "standard" training, validation, and test splits to facilitate comparisons. In other words, the data splitting policy follows the shared task. In addition to English parsing, we consider Chinese SDP and use two data sets: (1) Chinese PAS data provided by SemEval 2015, and (2) Chinese CCGBank (Tse and Curran, 2010) to evaluate the cross-lingual ability of our model. All the SemEval data sets are publicly available from LDC (Oepen et al., 2016).

We use DyNet[5] to implement our neural models. We use the automatic batch technique (Neubig et al., 2017) in DyNet to perform mini-batch gradient descent training. The batch size is 32. The detailed network hyper-parameters are summarized in Table 2. We use the same pre-trained word embedding as Kiperwasser and Goldberg (2016).

5.2 Main Results of English Parsing

Table 1 lists the parsing accuracy of our system as well as the best published results in the literature for comparison. Results from other papers are of different yet representative decoding or disambiguation frameworks. Du et al. (2015a)'s and Almeida and Martins (2015)'s parsers use global linear models to perform disambiguation. These systems obtained the best parsing accuracy for the SemEval 2015 shared task. Peng et al. (2017)'s and Wang et al. (2018)'s parsers utilize neural models, LSTMs in particular, to score either arcs or transitions. Our single models get the highest scores on not only in-domain but also out-of-domain test sets for the DM and PAS data sets, and they obtain comparable results with the state-of-art parser on the PSD data set. Comparing our results to the results obtained by parsers based on linear models, we can see the effectiveness of the BiLSTM based disambiguation model. The preci-

[4] DM, PAS and PSD are short for DeepBank, Enju HPSGBank and Prague Dependency Treebank.

[5] https://github.com/clab/dynet

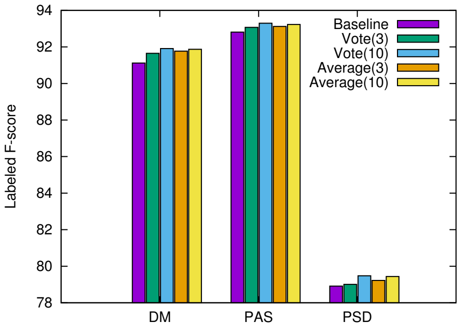

Figure 5: Labeled F_1 relative to different ensemble methods. Results are obtained on the development data.

sion of the two linear model-based parsers is comparable or even superior to our neural parser, but the recall is far behind.

5.3 Model Ensemble

Ensemble methods have been shown very helpful to boost the accuracy of neural network based parsing. We evaluate two ensemble methods, voting and score averaging. In the voting method, each model parses the sentence to graph respectively. An edge will exist on the combined graph only if more than half output graphs of these models contain this edge. The label of this edge will be the most common label. In the score averaging method, we use averaged score parts to get a maximum graph and classify labels.

We choose 3/10 kind of different initial parameters to train models for ensemble. Figure 5 shows the result of the two ensemble methods. The averaging method has slightly better performance on the 3 datasets. The performance of this method on test data is shown on Table 1.

5.4 Data for Cross-Domain Experiments

Since around 2001, the ERG has been accompanied by syntactico-semantic annotations, where for each sentence an annotator has selected the intended analysis among all alternatives licensed by the grammar. This derived resource, namly Redwoods[6] (Oepen et al., 2002; Flickinger et al., 2017), is a collection of hand-annotated corpora and consists of data sets from several distinct domains. Redwoods also includes (re)treebanking results of the first 22 sections of the venerable Wall Street Journal (WSJ) text and the section of Brown Corpus in the Penn Treebank (Marcus et al., 1993). The WSJ part is also known as DeepBank (Flickinger et al., 2012). The Brown corpus part is used as the out-of-domain test data by SemEval 2015. The DM data sets for both SemEval 2014 and 2015 SDP shared tasks are based on the RedWoods corpus.

Besides gold standard annoations, Flickinger et al. (2010) built the WikiWoods corpus[7], which provides automatically created annotations for the texts from wikipedia. The annotations are disambiguated using the MaxEnt model trained using redwoods without DeepBank. We use a small portion of Wikiwoods, which contains 857,329 sentences in total.

To evaluate the (positive) impact of ERG on out-of-domain parsing, we conduct experiments on the DM data. The first group of experiments are designed to be comparable with the results obtained by various participant systems of SemEval 2015. The detailed data set-up is as follows:

- **Test Data**. We use the Brown corpus section which is provided by SemEval 2015.

- **Training Data**. We use three data sets for training: (1) DeepBank, (2) RedWoods and (3) a small portion of WikiWoods reparsed using the MaxEnt model trained on DeepBank. We denote this reparsed WikiWoods as WikiWoods-ACE, since the HPSG analysis is provided by the ACE parser. To extract the semantic dependency graph, we use the pydelphin tool[8].

For the second group of experiments, we use the section *wsj21* from the DeepBank as test data, which is the official in-domain test of the SemEval 2015. The training data includes the “RedWoods minus DeepBank” annotations (RedwoodsWOD for short) as well as the official WikiWoods annotations. Note that the MaxEnt model used to obtain the official WikiWoods annotations are compatible with RedwoodswWOD. Due to the diversity of the RedwoodsWOD and DeepBank sentences, this set-up can also be viewed as an out-of-domain evaluation.

5.5 Results of Cross-Domain Parsing

Table 3 summarizes experimental results for different cross-domain evaluation set-ups. For the

[6] http://moin.delph-in.net/RedwoodsTop

[7] http://moin.delph-in.net/WikiWoods

[8] https://github.com/delph-in/pydelphin

Training Data		LP	LR	LF
IN-DOMAIN (SEMEVAL)				
DeepBank	S	90.74	90.40	90.57
Redwoods	S	91.50	90.57	91.03
DeepBank+WikiWoods-ACE	S	91.93	90.72	91.32
DeepBank+WikiWoods-ACE	E[3]	92.73	91.48	92.11
OUT-OF-DOMAIN (SEMEVAL)				
DeepBank	S	85.70	85.02	85.37
Redwoods	S	86.28	84.85	85.56
DeepBank+WikiWoods-ACE	S	88.30	86.42	87.35
DeepBank+WikiWoods-ACE	E[3]	89.53	87.57	88.54
OUT-OF-DOMAIN (REDWOODSWOD)				
DeepBank	S	90.74	90.40	90.57
RedwoodsWOD	S	81.40	78.99	80.18
RedwoodsWOD+WikiWoods	S	84.05	79.86	81.90
RedwoodsWOD+WikiWoods	E[3]	84.84	81.02	82.88

Table 3: Labeled F_1 on the DM test sets. "S" denotes single model, while "E[3]" denotes ensemble model with 3 sub-models.

first group of experiments, we test the parser using different training data sets. The baseline utilizes the WSJ portion only. While more reliable training data is added, the performances increase consistently. We notice that the improvement extending the training data from DeepBank to Redwoods is quite limited for the out-of-domain evaluation. One reason is that the amount of enlarged gold standard annotations is still limited: The DeepBank training data contains 35,656 sentences (838,374 tokens, i.e., roughly words), while the additional training data contains 35,950 sentences (538,659 tokens). For comparison, we select 480,564 sentences (5,346,703 tokens) from WikiWoods to train another model, and leave out other parts of Redwoods. The performance improvement is more remarkable when providing more data, even though such data contains annotation errors. For the second group of experiments, we use the RedwoodsWOD sentences for training and the DeepBank WSJ sentences for evaluation. For this set-up, consistent improvements of the parser quality are observed.

5.6 Results of Chinese Parsing

To test the ability for cross-lingual parsing, we conduct experiments on HPSG and CCG grounded semantic analyses respectively. The HPSG grounded analysis is provided by SemEval 2015 and the underlying framework is the same to the English PAS data. The CCG grounded analysis is from Chinese CCGBank. We use the same set-up as Zhang et al. (2016). Both data sets are transformed from Chinese TreeBank with two rich sets of heuristic rules (Yu et al., 2010; Tse and Curran, 2010). Table 4 and 5 presents all results. Our parser significantly outperforms Zhang et al. (2016)'s Zhang et al. (2016) system on Chinese CCGBank, which achieved best reported performance.

Chinese POS tagging has a great impact on parsing. In this paper, we consider two POS taggers: a symbol-refined generative HMM tagger (SR-HMM) (Huang et al., 2009) and a BiLSTM-CRF model when assisting Chinese SDG. For the neural tagging model, in addition to a BiLSTM layer for encoding words, we set a BiLSTM layer for encoding characters, which supports us to derive character-level representations for all words. In particular, vectors from the character-level LSTM is concatenated with the pre-trained word embedding before feeding into the other word-level BiLSTM network to capture contextual information. The final module of our CRF tagger is a linear chain CRF which scores the output sequence by factoring it in local tag bi-grams. From Table 5, we can see that POS information is very important to Chinese SDP. This phenomenon is consist with Chinese syntactic parsing, including both constituency and dependency parsing. Mandarin Chinese is recognized as a morphology-poor language: POS tags are defined mainly according to words' distributional rather than morphological properties. The LSTM-based tagger can leverage

Model	LP	LR	LF
Peking	84.75	82.15	83.43
Ours	85.49	84.11	84.79

Table 4: Labeled F_1 on the test set of SemEval 2015 for Chinese. "Peking" is the participant system that obtained the best parsing accuracy for Chinese in SemEval 2015.

Model	POS	LP	LR	LF
ZDSW	Gold	82.09	81.81	81.95
Ours	Gold	86.37	86.00	86.19
	SR-HMM	80.19	80.53	80.37
	BiLSTM-CRF	81.13	81.74	81.43

Table 5: Labeled F_1 on the test set of Chinese CCGBank. "ZDSW" is the system that obtained the best parsing accuracy on the Chinese CCGBank data in the literature.

the power of the RNN architecture to learn non-local dependencies and thus benefit our semantic dependency parser a lot.

6 Conclusion

Parsing sentences to linguistically-rich semantic representations is a key goal of Natural Language Understanding. We introduce a new parser for semantic dependency analysis, which combines two promising parsing techniques, i.e., decoding based on Maximum Subgraph algorithms and disambiguation based on BiLSTMs. To our knowledge, this is the first neural Maximum Subgraph parser. Our parser significantly improves state-of-the-art accuracy on three out of total four data sets from SemEval 2015 for English/Chinese parsing and the CCGBank data for Chinese parsing. We also propose a new data-oriented method to leverage ERG, a linguistically-motivated, hand-crafted grammar, to improve cross-domain performance. Experiments demonstrate the effectiveness of taking ERG as an imperfect annotator. We think this method can be re-used for other types of data-driven semantic parsing models.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (61772036, 61331011) and the Key Laboratory of Science, Technology and Standard in Press Industry (Key Laboratory of Intelligent Press Media Technology). We thank the anonymous reviewers for their helpful comments. Weiwei Sun is the corresponding author.

References

C. Mariana S. Almeida and T. André F. Martins. 2015. Lisbon: Evaluating TurboSemanticParser on Multiple Languages and Out-of-Domain Data. *Proceedings of SemEval 2015.*

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for Sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Ulrich Callmeier. 2000. Pet. a platform for experimentation with efficient hpsg processing techniques. *Journal of Natural Language Engineering*, 6(1):99–108.

Junjie Cao, Sheng Huang, Weiwei Sun, and Xiaojun Wan. 2017a. Parsing to 1-endpoint-crossing, pagenumber-2 graphs. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2110–2120, Vancouver, Canada. Association for Computational Linguistics.

Junjie Cao, Sheng Huang, Weiwei Sun, and Xiaojun Wan. 2017b. Quasi-second-order parsing for 1-endpoint-crossing, pagenumber-2 graphs. In *Proceedings of EMNLP 2017*. Association for Computational Linguistics.

Michael Collins. 2002. 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*, pages 1–8. Association for Computational Linguistics.

Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan A. Sag. 2005. Minimal Recursion Semantics: An introduction. *Research on Language and Computation*, pages 281–332.

Marta R. Costa-jussà and José A. R. Fonollosa. 2016. Character-based neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 357–361, Berlin, Germany.

Yantao Du, Weiwei Sun, and Xiaojun Wan. 2015a. A data-driven, factorization parser for CCG dependency structures. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1545–1555, Beijing, China. Association for Computational Linguistics.

Yantao Du, Fan Zhang, Xun Zhang, Weiwei Sun, and Xiaojun Wan. 2015b. Peking: Building semantic dependency graphs with a hybrid parser. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 927–931, Denver, Colorado. Association for Computational Linguistics.

Dan Flickinger. 2000. On building a more efficient grammar by exploiting types. *Nat. Lang. Eng.*, 6(1):15–28.

Dan Flickinger, Stephan Oepen, and Emily M. Bender. 2017. *Sustainable Development and Refinement of Complex Linguistic Annotations at Scale*. Springer Netherlands, Dordrecht.

Dan Flickinger, Stephan Oepen, and Gisle Ytrestøl. 2010. Wikiwoods: Syntacto-semantic annotation for English wikipedia. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Daniel Flickinger, Yi Zhang, and Valia Kordoni. 2012. Deepbank: A dynamically annotated treebank of the wall street journal. In *Proceedings of the Eleventh International Workshop on Treebanks and Linguistic Theories*, pages 85–96.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Comput.*, 9(8):1735–1780.

Zhongqiang Huang, Vladimir Eidelman, and Mary Harper. 2009. Improving a simple bigram hmm part-of-speech tagger by latent annotation and self-training. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 213–216, Boulder, Colorado. Association for Computational Linguistics.

Angelina Ivanova, Stephan Oepen, Rebecca Dridan, Dan Flickinger, and Lilja Øvrelid. 2013. On different approaches to syntactic analysis into bi-lexical dependencies. an empirical comparison of direct, PCFG-based, and HPSG-based parsers. In *Proceedings of The 13th International Conference on Parsing Technologies (IWPT-2013)*, pages 63–72, Nara, Japan.

Angelina Ivanova, Stephan Oepen, Lilja Øvrelid, and Dan Flickinger. 2012. Who did what to whom? A contrastive study of syntacto-semantic dependencies. In *Proceedings of the Sixth Linguistic Annotation Workshop*, pages 2–11, Jeju, Republic of Korea.

Eliyahu Kiperwasser and Yoav Goldberg. 2016. Simple and accurate dependency parsing using bidirectional LSTM feature representations. *Transactions of the Association for Computational Linguistics*, 4:313–327.

Marco Kuhlmann and Peter Jonsson. 2015. Parsing to noncrossing dependency graphs. *Transactions of the Association for Computational Linguistics*, 3:559–570.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: the penn treebank. *Computational Linguistics*, 19(2):313–330.

André F. T. Martins and Mariana S. C. Almeida. 2014. Priberam: A turbo semantic parser with second order features. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 471–476, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

Graham Neubig, Yoav Goldberg, and Chris Dyer. 2017. On-the-fly operation batching in dynamic computation graphs. In *Advances in Neural Information Processing Systems*.

Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Silvie Cinková, Dan Flickinger, Jan Hajič, Angelina Ivanova, and Zdeňka Urešová. 2016. Semantic Dependency Parsing (SDP) graph banks release 1.0 LDC2016T10. Web Download.

Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Silvie Cinková, Dan Flickinger, Jan Hajic, and Zdenka Uresová. 2015. Semeval 2015 task 18: Broad-coverage semantic dependency parsing. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*.

Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Dan Flickinger, Jan Hajic, Angelina Ivanova, and Yi Zhang. 2014. Semeval 2014 task 8: Broad-coverage semantic dependency parsing. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 63–72, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

Stephan Oepen, Kristina Toutanova, Stuart Shieber, Christopher Manning, Dan Flickinger, and Thorsten Brants. 2002. The lingo redwoods treebank motivation and preliminary applications. In *Proceedings of the 19th International Conference on Computational Linguistics - Volume 2*, COLING '02, pages 1–5, Stroudsburg, PA, USA. Association for Computational Linguistics.

Hao Peng, Sam Thomson, and Noah A. Smith. 2017. Deep multitask learning for semantic dependency parsing. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2037–2048, Vancouver, Canada. Association for Computational Linguistics.

Emily Pitler, Sampath Kannan, and Mitchell Marcus. 2013. Finding optimal 1-endpoint-crossing trees. *TACL*, 1:13–24.

Daniel Tse and James R. Curran. 2010. Chinese CCGbank: extracting CCG derivations from the penn Chinese treebank. In *Proceedings of the 23rd International Conference on Computational Linguistics*

(Coling 2010), pages 1083–1091, Beijing, China. Coling 2010 Organizing Committee.

Yuxuan Wang, Wanxiang Che, Jiang Guo, and Ting Liu. 2018. A neural transition-based approach for semantic dependency graph parsing. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*.

Kun Yu, Miyao Yusuke, Xiangli Wang, Takuya Matsuzaki, and Junichi Tsujii. 2010. Semi-automatically developing Chinese hpsg grammar from the penn Chinese treebank for deep parsing. In *Coling 2010: Posters*, pages 1417–1425, Beijing, China. Coling 2010 Organizing Committee.

Xun Zhang, Yantao Du, Weiwei Sun, and Xiaojun Wan. 2016. Transition-based parsing for deep dependency structures. *Computational Linguistics*, 42(3):353–389.

Yi Zhang and Rui Wang. 2009. Cross-domain dependency parsing using a deep linguistic grammar. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 378–386, Suntec, Singapore. Association for Computational Linguistics.