Comparing Knowledge-Intensive and Data-Intensive Models for English Resource Semantic Parsing

Junjie Cao*
Peking University
Wangxuan Institute of
Computer Technology
junjie.junjiecao@alibaba-inc.com

Zi Lin*
Peking University
Department of Chinese Language
and Literature
lzi@google.com

Weiwei Sun**
Peking University
Wangxuan Institute of Computer
Technology and
Center for Chinese Linguistics
ws390@cam.ac.uk

Xiaojun Wan
Peking University
Wangxuan Institute of
Computer Technology
wanxiaojun@pku.edu.cn

In this work, we present a phenomenon-oriented comparative analysis of the two dominant approaches in English Resource Semantic (ERS) parsing: classic, knowledge-intensive and neural, data-intensive models. To reflect state-of-the-art neural NLP technologies, a factorization-based parser is introduced that can produce Elementary Dependency Structures much more

* Equal contribution. The work was done while the first three authors were at Peking University. Junjie Cao is now at Alibaba, and Zi Lin is now at Google.
** Corresponding author. Now at the Department of Computer Science and Technology of University of Cambridge.

Submission received: 20 April 2019; revised version received: 3 October 2020; accepted for publication: 18 November 2020.

https://doi.org/10.1162/COLI_a_00395

accurately than previous data-driven parsers. We conduct a suite of tests for different linguistic phenomena to analyze the grammatical competence of different parsers, where we show that, despite comparable performance overall, knowledge- and data-intensive models produce different types of errors, in a way that can be explained by their theoretical properties. This analysis is beneficial to in-depth evaluation of several representative parsing techniques and leads to new directions for parser development.

1. Introduction

Recent work in task-independent semantic parsing shifts from the knowledge-intensive approach to the data-intensive approach. Early attempts in semantic parsing leverage explicitly expressed symbolic rules in a deep grammar formalism, for example, Combinatory Categorial Grammar (CCG; Steedman 1996, 2000) and Head-driven Phrase Structure Grammar (HPSG; Pollard and Sag 1994), to model the syntactico-semantic composition process (Bos et al. 2004; Callmeier 2000). Then, statistical machine learning technologies, especially structured prediction models, are utilized to enhance deep grammar-driven parsers (Clark and Curran 2007; Zhang, Oepen, and Carroll 2007; Miyao and Tsujii 2008). Recently, various deep learning models together with vector-based embeddings induced from large-scale raw texts have been making considerable advances (Chen, Sun, and Wan 2018; Dozat and Manning 2018).

This article is concerned with comparing knowledge-intensive and data-intensive parsing models for English Resource Semantics (ERS; Flickinger, Bender, and Oepen 2014b,2014a), a comprehensive framework for in-depth linguistic analysis. Figures 1 and 2 are two examples illustrating the ERS representations. Our comparison is based not only on the general evaluation metrics for semantic parsing, but also a fine-grained construction-focused evaluation that sheds light on the kinds of strengths each type of parser exhibits. Characterizing such models may benefit parser development for not only ERS but also other frameworks, for example, Groningen Meaning Bank (GMB; Basile et al. 2012; Bos et al. 2017) and Abstract Meaning Representation (AMR; Banarescu et al. 2013).

To reflect the state-of-the-art deep learning technologies that are already available for data-intensive parsing, we design and implement a new factorization-based system for string-to-conceptual graph parsing (Kuhlmann and Oepen 2016). This parser learns to produce conceptual graphs for sentences from an annotated corpus and does not assume the existence of a grammar that explicitly defines syntactico-semantic patterns. The core engine is scoring functions that use contextualized word and concept embeddings to discriminate good parses from bad for a given sentence, regardless of its semantic composition process.

To evaluate the effectiveness of the new parser, we conduct experiments on DeepBank (Flickinger, Zhang, and Kordoni 2012). Our parser achieves an accuracy of 95.05[1] for Elementary Dependency Structure (EDS; Oepen and Lønning 2006) in terms of SMATCH, which shows 8.05-point improvement over the best transition-based model (Buys and Blunsom 2017) and 4.19-point improvement over the composition-based parser (Chen, Sun, and Wan 2018). We take it as a reflection that the models induced from large-scale data by neural networks have a strong coherence with linguistic knowledge. Our parser has been re-implemented or extended by two best-performing

1 The results are obtained based on gold-standard tokenization.

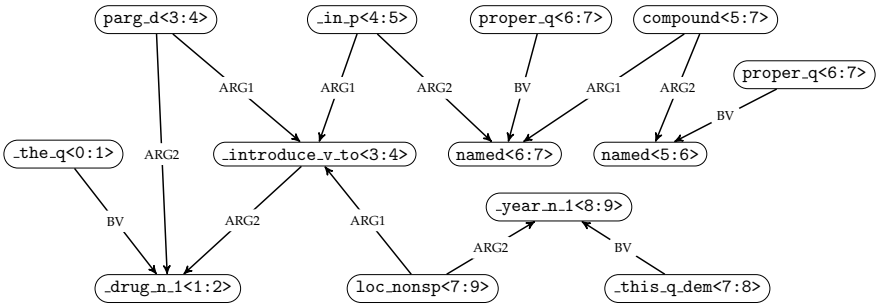


Figure 1
An example of EDS graph. Some concepts are surface relations, meaning that they are related to a single lexical unit, e.g., `_the_q` or `_introduce_v_to`, while others are abstract relations representing grammatical meanings, e.g., `compound` (multiword expression), `parg_d` (passive), and `loc_nonsp` (temporal). ERS corpus provides alignment between concept nodes and surface strings, e.g., `<0:1>` that is associated to `_the_q` indicates that this concept is signaled by the first word.

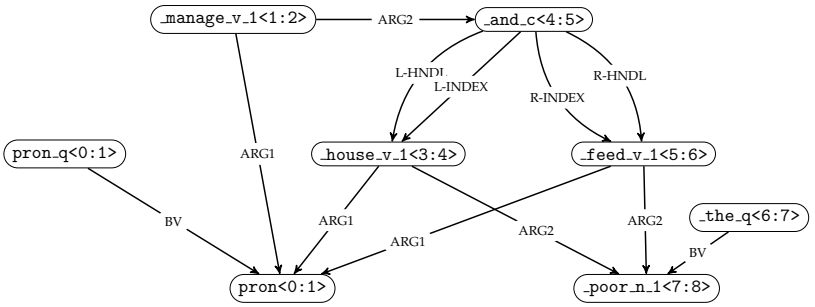


Figure 2
An example of EDS graph to represent complicated phenomena like *right node raising* and *raising/control* constructions.

systems (Zhang et al. 2019; Chen, Ye, and Sun 2019) in the CoNLL 2019 Shared Task on Cross-Framework Meaning Representation Parsing (Oepen et al. 2019).

Despite the numerical improvements brought by neural networks, they have typically come at the cost of our understanding of the systems, that is, it is still unclear to what extent we can expect supervised training or pretrained embeddings to induce the implicit linguistic knowledge and thus help semantic parsing. To answer this question, we utilize linguistically informed data sets based on previous work (Bender et al. 2011) and create an extensive suite of other widely discussed linguistic phenomena, covering a rich set of linguistic phenomena related to various lexical, phrasal, and non-local dependency constructions. Based on the probing study, we find several non-obvious facts:

1. The data-intensive parser is good at capturing local information at the lexical level even when the training data set is rather small.

2. The data-intensive parser performs better on some peripheral phenomena but may suffer from data sparsity.
3. The knowledge-intensive parser produces more coherent semantic structures, which may have a great impact on advanced natural language understanding tasks, such as textual inference.
4. It is difficult for both parsers to find long-distance dependencies, and their performance varies across phenomena.

There is no a priori restriction that a data-intensive approach must remove all explicitly defined grammatical rules, or a knowledge-intensive approach cannot be augmented by data-based technologies. Our comparative analysis appears highly relevant, in that these insights may be explored further to design new computational models with improved performance.[2]

2. Background

2.1 Graph-Based Meaning Representations

Considerable NLP research has been devoted to the transformation of natural language utterances into a desired linguistically motivated semantic representation. Such a representation can be understood as a class of discrete structures that describe lexical, syntactic, semantic, pragmatic, as well as many other aspects of the phenomenon of human language. In this domain, graph-based representations provide a light-weight yet effective way to encode rich semantic information of natural language sentences and have been receiving heightened attention in recent years (Kuhlmann and Oepen 2016). Popular frameworks under this umbrella includes Bi-lexical Semantic Dependency Graphs (SDG; Clark, Hockenmaier, and Steedman 2002; Ivanova et al. 2012; Oepen et al. 2014, 2015), AMR (Banarescu et al. 2013), Graph-based Representations for ERS (Oepen and Lønning 2006; Copestake 2009), and Universal Conceptual Cognitive Annotation (UCCA; Abend and Rappoport 2013).

2.2 Parsing to Semantic Graphs: Knowledge-intensive vs. Data-intensive

Parsing to the graph-based representations has been extensively studied recently (Flanigan et al. 2014; Artzi, Lee, and Zettlemoyer 2015; Du, Sun, and Wan 2015; Peng, Song, and Gildea 2015; Zhang et al. 2016; Buys and Blunsom 2017; Cao et al. 2017; Hershcovich, Abend, and Rappoport 2017; Konstas et al. 2017; Peng, Thomson, and Smith 2017; Chen, Sun, and Wan 2018). Work in this area can be divided into different types, according to how information about the mapping between natural language utterances and target graphs is formalized, acquired, and utilized. In this article, we are concerned with two dominant types of approaches in ERS parsing, which won the DM and EDS sections of the CoNLL 2019 shared task (Oepen et al. 2019).

In the first type of approach, a semantic graph is derived according to a set of lexical and syntactico-semantic rules, which extensively encode explicit linguistic knowledge. Usually, such rules are governed by a well-defined grammar formalism, for

2 The code of our semantic parser, the test sets of linguistic phenomena, as well as the evaluation tool can be found at https://github.com/zi-lin/feds-parser for research purposes.

example, CCG, HPSG, and Hyperedge Replacement Grammar, and exploit compositionality (Callmeier 2000; Bos et al. 2004; Artzi, Lee, and Zettlemoyer 2015; Peng, Song, and Gildea 2015; Chen, Sun, and Wan 2018; Groschwitz et al. 2018). In this article, we call this the **knowledge-intensive** approach.

The second type of approach explicitly models the target semantic structures. It may associate each basic part with a target graph score, and casts parsing as the search for the graphs with the highest sum of partial scores (Flanigan et al. 2014; Kuhlmann and Jonsson 2015; Peng, Thomson, and Smith 2017). The essential computational module in this architecture is the score function, which is usually induced based on moderate-sized annotated sentences. Various deep learning models, together with vector-based encodings induced from large-scale raw texts, have been making considerable advances in shaping a score function (Dozat and Manning 2018). This type of approach is also referred to as graph-based or factorization-based in the context of bi-lexical dependency parsing. In this article, we call this the **data-intensive** approach.

2.3 DELPH-IN English Resource Semantics

In this article, we take the representations from ERS (Flickinger, Bender, and Oepen 2014b)[3] as our case study. Compared to other meaning representations, ERS exhibits at least the following features: (1) ERS has a relatively high coverage for English text (Adolphs et al. 2008; Flickinger, Oepen, and Ytrestøl 2010; Flickinger, Zhang, and Kordoni 2012); (2) ERS has a strong transferability across difference domains (Copestake and Flickinger 2000; Ivanova et al. 2013); and (3) ERS has comparable and relatively high performance in terms of knowledge-intensive and data-intensive parsing technologies (Callmeier 2000; Zhang, Oepen, and Carroll 2007; Chen, Sun, and Wan 2018).

ERS is the semantic annotation associated with ERG (Flickinger 2000), an open-source, domain-independent, linguistically precise and broad-coverage grammar of English, which encapsulates the linguistic knowledge required to produce many of the types of compositional meaning annotations. The ERG is an implementation of the grammatical theory of Head-driven Phrase Structure Grammar (HPSG; Pollard and Sag 1994). ERG is a resource grammar that can be used for both parsing and generation. Development of the ERG began in 1993, and after continuously evolving through a series of projects, it allows the grammatical analysis of most running text across domains and genres.

In the most recent stable release (version 1214), the ERG contains 225 syntactic rules and 70 lexical rules for derivation and inflection. The hand-built lexicon of the ERG contains 39,000 lemmata, instantiating 975 leaf lexical types providing part-of-speech and valence constraints, which aims at providing complete coverage of function words and open-class words with "non-standard" syntactic properties (e.g., argument structure). The ERG also supports light-weight named entity recognition and an unknown word mechanism, allowing the grammar to derive full syntactico-semantic analyses for 85–95% of all utterances in real corpora such as newspaper text, the English Wikipedia, or bio-medical academic papers (Adolphs et al. 2008; Flickinger, Oepen, and Ytrestøl 2010; Flickinger, Zhang, and Kordoni 2012). For more than 20 years of development, ERS has shown its advantages of explicit formalization and large scalability (Copestake and Flickinger 2000).

3 http://moin.delph-in.net/ErgSemantics.

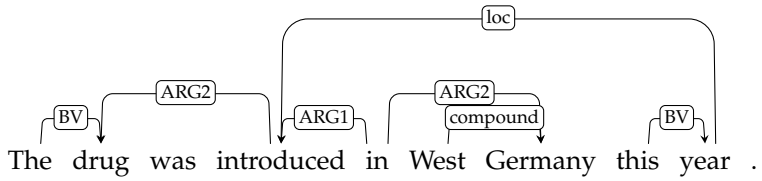


Figure 3
The standard SDG that is converted from the EDS in Figure 1.

The Minimal Recursion Semantics (MRS; Copestake et al. 2005) is the associated semantic representation used by the ERG. MRS is based on the first-order language family with generalized quantifiers. ERS can also be expressed as other semantic graphs, including SDG (Ivanova et al. 2012), EDS (Oepen and Lønning 2006), and Dependency-based Minimal Recursion Semantics (DMRS; Copestake 2009). In this article, we illustrate our models using the EDS format.[4]

The graphs in Figures 1 and 2 are examples of an EDS graph. Figure 2 further shows that EDS does have the ability to represent more complicated linguistic phenomena such as the *right node raising* and *raising/control* constructions.[5] The semantic structure is a directed graph where nodes labeled with semantic predicates/relations are related to a constituent of the sentence, and arcs are labeled with semantic arguments roles. By linking concepts with lexical units, this EDS graph can be reduced to an SDG, as shown in Figure 3. In these forms, **relation** is the predicate name of an Elementary Predication from the MRS, and **role** is an argument label such as *ARG1*.

2.4 Analyzing Neural Networks for NLP

What are the representations that the neural network learns and how can we explore that? Concerns of this question have led to the interpretability of the system being an active area of research. Related work tries to answer these questions by: (1) investigating specific components of the architectures (Karpathy, Johnson, and Fei-Fei 2015; Qian, Qiu, and Huang 2016; Bau et al. 2019), (2) testing models on specific tasks, including part-of-speech tagging (Shi, Padhi, and Knight 2016; Belinkov et al. 2017; Blevins, Levy, and Zettlemoyer 2018), semantic role labeling (Tenney et al. 2019), word sense disambiguation (Peters et al. 2018a), coreference (Peters et al. 2018b), and so forth, and (3) building a linguistically informed data set for evaluation (Linzen, Dupoux, and Goldberg 2016; Burchardt et al. 2017; Isabelle, Cherry, and Foster 2017; Sennrich 2017; Wang et al. 2018; Warstadt, Singh, and Bowman 2019).

In this article, we try to probe this question by applying the models built on the state-of-the-art technologies to the string-to-conceptual graph parsing task, and utilizing linguistically informed data sets based on previous work (Bender et al. 2011) and our own creation.

4 `http://moin.delph-in.net/EdsTop`.

5 *Right node raising* often involves coordination where they share the same component (e.g., the subject *they* here for the predicates *house* and *feed*); the *raising/control* construction refers to raising and control verbs that select for an infinitival VP complement and stipulate that another of their arguments (subject or direct object in the example) is identified with the unrealized subject position of the infinitival VP. For further details, see Bender et al. (2011).

3. A Knowledge-Intensive Parser

There are two main existing knowledge-intensive parsers with unification-based grammars for the ERG: the PET system[6] (Callmeier 2000; Zhang, Oepen, and Carroll 2007) and the ACE system.[7] PET is an efficient open-source parser for unification grammars. Coupled with ERG, it can produce HPSG-style syntactico-semantic derivations and MRS-style semantic representations in logic forms. Similar to PET, ACE is another industrial strength implementation of the typed feature structure formalism. Note that the key algorithms implemented by PET and ACE are the same. We choose to use ACE in this work given the fact that, compared with PET's parsing performance, ACE is faster in certain common configurations. Coupled with ERG, it serves as a valid companion to study our problem: comparing knowledge- and data-intensive approaches.

4. A Data-Intensive Parser

To empirically analyze data-intensive parsing technologies, we design, implement, and evaluate a new target structure–centric parser for ERS graphs, trained on *(string, graph)* pairs without explicitly incorporating linguistic knowledge. The string-to-graph parsing is formulated as a problem of finding the Maximum Subgraph for a graph class $\mathcal{G}$ of a sentence $s = l_1, \ldots, l_m$: Given a graph $G = (V, A)$ related to s, the goal is to search for a subset $A' \subseteq A$ with maximum total score such that the induced subgraph $G' = (V, A')$ belongs to $\mathcal{G}$. Formally, we have the following optimization problem:

$$\arg \max_{G^* \in \mathcal{G}(s,G)} \sum_{p \in \text{FACTORIZE}(G^*)} \text{SCORE}_{\text{part}}(s, p)$$

where $\mathcal{G}(s, G)$ denotes the set of all graphs belonging to $\mathcal{G}$ and compatible with s and G. This view matches a classic solution to the structured prediction that captures elemental and structural information through part-wise factorization. To evaluate the goodness of a semantic graph is to calculate the sum of *local* scores assigned to those parts.

In the literature of bi-lexical dependency parsing, the above architecture is also widely referred to as factorization-based, as such a parser factors all valid structures for a given sentence into smaller units, which can be scored somewhat independently.

For string-to-graph parsing, we consider two basic factors, namely, single concepts and single dependencies. Formally, we use the following objective function:

$$\sum_{n \in \text{NODE}(G)} \text{SC}_{\text{n}}(s, n) + \sum_{(p,a) \in \text{ARC}(G)} \text{SC}_{\text{a}}(s, p, a)$$

Our parser adopts a two-step architecture to produce EDS graphs: (1) it identifies the concept nodes based on contextualized word embeddings by solving a simplified optimization problem, viz. $\max \sum_{n \in \text{NODE}(G)} \text{SC}_{\text{n}}(s, n)$; and (2) it identifies the dependencies between concepts based on concept embeddings by solving another optimization problem, namely, $\max \sum_{(p,a) \in \text{ARC}(G)} \text{SC}_{\text{a}}(s, p, a)$. Particularly, our architecture is a pipeline: Single-best prediction of the first step is utilized as the input for the second step.

6 http://moin.delph-in.net/PetTop.
7 http://sweaglesw.org/linguistics/ace/.

4.1 Concept Identification

Usually, the nodes in a conceptual graph have a strong correspondence to surface lexical units, namely, tokens, in a sentence. Take the graph in Figure 1 for example; the generalized quantifier `_the_q` corresponds to *the* and the property concept `_drug_n_1` corresponds to *drug*. Because the concepts are highly *lexicalized*, it is reasonable to use a sequence labeler to predict concepts triggered by tokens.

Nodes may be aligned with arbitrary parts of the sentence, including sub-token or multi-token sequences, which affords more flexibility in the representation of meaning contributed by derivational morphemes (e.g., `parg_d` that indicates a passive construction) or phrasal constructions (e.g., `compound_name` that indicates a multiword expression). To handle these types of concepts by a word-based sequence labeler, we align them to words based on their span information and a small set of heuristic rules. Take Figure 4 for example—we align `parg_d` to the word where *-ed* is attached to, and `compound_name` to the first word of the compound.

The concept predicate may contain the lexical part aligning to the surface predicate, which leads to a serious data sparseness problem for training. To deal with this problem, we *delexicalize* lexical predicates as described in Buys and Blunsom (2017): Replacing the lemma part by a placeholder "`*`". Figure 4 shows a complete example. In summary, the concept identification problem is formulated as a word tagging problem:

$$\sum_{n \in \text{NODE}(G)} \text{SC}_\text{n}(s, n) \approx \sum_{1 \le i \le m} \max_{st_i \in \mathcal{ST}} \text{SC}_\text{st}(s, i, st_i)$$

Our parser applies a neural sequence labeling model to predict concepts. In particular, a BiLSTM model is utilized to capture words' contextual information and another softmax layer for classification. Usually, words and POS tags are needed to be transformed into continuous and dense representation in neural models. Inspired by Costa-jussà and Fonollosa (2016), we use word embedding of two granularities in our model: character-based and word-based, for low frequency and high-frequency words (the words appear more than k times in the training data), respectively. A character-based model can capture rich affix information of low-frequency words for better word representations. The word-based embedding uses a common lookup-table mechanism. The character-based word embedding w_i is implemented by extracting features with bidirectional LSTM from character embeddings $c_1, \ldots, c_n$:

Contextualized representation models such as CoVe (McCann et al. 2017), ELMo (Peters et al. 2018a), OpenAI GPT (Radford et al. 2018), and BERT (Devlin et al. 2019)

	The	drug	was	introduced	in	West	Germany	this	year
N	`_the_q`	`_drug_n_1`	∅	`_introduce_v_to` `parg_d`	`_in_p`	`named` `compound_name` `proper_q`	`named` `proper_q`	`_this_q_dem`	`_year_n_1` `loc_nonsp`
S	`*_q`	`*_n_1`	∅	`*_v_to` `parg_d`	`*_p`	`named` `compound_name` `proper_q`	`named` `proper_q`	`*_q_dem`	`*_n_1` `loc_nonsp`

Figure 4
An example for illustrating concept identification. The "N" row presents the results of lexicalization. The "S" row presents the gold tags assigned to tokens that are utilized to train a sequence labeling based concept identifier.

have recently achieved the state-of-the-art results on downstream NLP models across many domains. In this article, we use pretrained ELMo model and learn a weighted average of all ELMo layers for our embedding e_i to capture richer contextual information. The concatenation of word embedding w_i, ELMo embedding, and POS-tag embedding t_i of each word in a specific sentence is used as the input of BiLSTMs to extract context-related feature vectors r_i for each position i. Finally we use r_i as input of a softmax layer to get the probability $\text{SC}_{\text{st}}(s, i, st_i)$.

$$\begin{aligned} a_i &= w_i \oplus e_i \oplus t_i \\ r_1 : r_m &= \text{BiLSTM}(a_1 : a_m) \\ \text{SC}_{\text{st}}(s, i, st_i) &= \text{softmax}(r_i) \end{aligned}$$

4.2 Dependency Identification

Given a set of concept nodes N that are predicted by our concept identifier, the semantic dependency identification problem is formalized as the following optimization problem:

$$\hat{G} = \arg\max_{G \in \mathcal{G}(N)} \sum_{(p,a) \in \text{ARC}(G)} \text{SC}_{\text{a}}(s, N, p, a)$$

where $\mathcal{G}(N)$ denotes the set of all possible graphs that take N as their vertex set. Following the factorization principle, we measure a graph using a sum of local scores.

In order to effectively learn a local score function, that is, SC_a, we represent concepts with the concatenation of two embeddings: textual and conceptual embeddings.

$$c_i = r_i \oplus n_i$$

To represent two concept nodes' textual information, we use stacked BiLSTMs that are similar to the proposed structure of our concept identifier to get r_i.

Besides contextual information, we also need to transform a concept into a dense vector n_i. Similar to word embedding and POS-tag embedding, we also use a common lookup-table mechanism and let our parser automatically induce conceptual embeddings from semantic annotations.

We calculate scores for all directional arcs between two concepts in the graph, which can be scored with a non-linear transformation from the two feature vectors of each concept pair:

$$\text{SC}_a(s, N, p, a) = W_2 \cdot \delta(W_1 \cdot (c_p \oplus c_a) + b)$$

Similar to unlabeled arcs, we also use MLP to get each arc's scores for all labels, and select the max one as its label.

For training, we use a margin-based approach to compute loss from the gold graph G^* and the best prediction $\hat{G}$ under the current model and decoder. 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 $\hat{G}$. Following Peng, Thomson, and Smith (2017)'s approach, we define Δ as weighted Hamming to trade off between precision and recall.

Inspired by the **maximum spanning connected subgraph** algorithm proposed by Flanigan et al. (2014), we also consider using an additional constraint to restrict the generated graph to be connected. The algorithm is simple and effective: generating a maximum spanning tree (MST) firstly, and then adding all arcs with positive local scores. During the training, our dependency identifier ignores this constraint.

4.3 Evaluation

We conduct experiments on DeepBank v1.1 that correspond to ERG version 1214, and adopt the *standard* data split. The PyDelphin[8] library and the jigsaw tool[9] are leveraged to extract EDS graphs and to separate punctuations from their attached words respectively. The TensorFlow *ELMo* model[10] is trained on the 1B Word Benchmark for pretrained feature, and we use the same pretrained word embedding introduced in Kiperwasser and Goldberg (2016). DyNet2.0[11] is utilized to implement the neural models. The automatic batch technique (Neubig, Goldberg, and Dyer 2017) in DyNet is applied to perform mini-batch gradient descent training, where the batch size equals 32.

Different models are tested to explore the contribution of BiLSTM and ELMo, including (1) *ELMo** using BiLSTM and ELMo features, (2) *ELMo* using only ELMo features and softmax layer for classification, (3) *W2V* using BiLSTM and word2vec features (Mikolov et al. 2013), and (4) *Random* using BiLSTM and random embedding initialization. In all these experiments, we only learn a weighted average of all biLM layers but froze other parameters in ELMo.

4.3.1 Results on Concept Identification. Because we predict concepts by composing them together as a word tag, there are two strategies for evaluating concept identification: the accuracy of tag (viz., concept set) prediction and concept (decomposed from tags) prediction. For the former, we take "$\emptyset$" as a unique tag and compare each word's predicted tag as a whole part; for the latter, we ignore the empty concepts, such as *was* in Figure 1. We can see that the *ELMo** model performs better than the others. Empirically speaking, the numeric performance of concept prediction is better than the tag prediction. The results are illustrated in Table 1.

4.3.2 Results on Dependency Identification. In the dependency identification step, we train the parsing model on sentences with golden concepts and alignment. Both unlabeled and labeled results are reported. Because golden concepts are used, the accuracy will obviously be much higher than the total system with predicted concepts. Nevertheless, the numbers here serve as a good reflection of the goodness of our models. We can see that the *ELMo** model still performs the best, but the *ELMo* model is much lower than the others, indicating that BiLSTM layers are much more important for dependency identification. Table 2 shows the results. The measure for comparing two dependency

8 www.github.com/delph-in/pydelphin.
9 www.coli.uni-saarland.de/∼yzhang/files/jigsaw.jar.
10 www.github.com/allenai/bilm-tf.
11 www.github.com/clab/dynet.

Table 1
Accuracy of concept identification on development data.

	Tag	Concept		
	Accuracy	Precision	Recall	F-Score
Random	92.74	95.13	94.60	94.87
W2V	94.51	96.68	96.11	96.39
ELMo	92.34	95.73	95.16	95.45
ELMo*	95.38	97.31	96.77	97.04

Table 2
Accuracy of dependency identification on development data.

Model	UP	UR	UF	LP	LR	LF
Random	94.47	95.95	95.21	94.25	95.57	94.98
W2V	94.91	96.30	95.60	94.72	96.12	95.42
ELMo	88.97	92.95	90.92	88.44	92.40	90.38
ELMo*	96.00	96.99	96.49	95.80	96.79	96.29

graphs is the precision/recall of concept tokens that are defined as $\langle c_h, c_d, l \rangle$ tuples, where c_h is the functor concept, c_d is the dependent concept, and l is their dependency relation. Labeled precision/recall (LP/LR) is the ratio of tuples correctly identified by the automatic generator, whereas unlabeled precision/recall (UP/UR) is the ratio regardless of l. F-score (LF/UF) is a harmonic mean of precision and recall.

4.3.3 Results for Graph Identification. As for the overall evaluation, we report parsing accuracy in terms of SMATCH (Cai and Knight 2013), which considers both nodes and relations, and was initially used for evaluating AMR parsers. The Smatch metric (Cai and Knight 2013), proposed for evaluating AMR graphs, also measures graph overlap, but does not rely on sentence alignments to determine the correspondences between graph nodes. SMATCH is computed by performing inference over graph alignments to estimate the maximum F-score obtainable from a one-to-one matching between the predicted and gold graph nodes. Different from EDM (Dridan and Oepen 2011), we only use each node's predicate but ignore the span information while aligning the two nodes. But the results of these two evaluations are positively related.

Considering the difference between the AMR graph and the EDS graph, we implement our own tool for the disconnected graph, and calculate the scores in Table 3. The *ELMo**'s concept and arc score are obviously higher than the others, while *ELMo*'s arc prediction yields the lowest SMATCH score.

We compare our system with the ERG-guided ACE parser, the data-driven parser introduced in Buys and Blunsom (2017), and the composition-based parser in Chen, Sun, and Wan (2018) on the test data (Table 4). As the ACE parser fails to parse some sentences (more than 1%),[12] the outputs of the whole data and the successfully parsed part are evaluated separately. For the other parsers, the results on the whole data and

12 Note that the DeepBank data already removes a considerable portion (ca. 11%) of sentences.

Table 3
Accuracy of the whole graphs on the development data. *Concept* and *Arc* in the table header are the F-Score of concept and arc mapping for the highest SMATCH score. *Smatch* is the Smatch score of each model.

Model	Concept	Arc	SMATCH
Random	94.69	91.25	92.95
W2V	96.07	92.41	94.22
ELMo	95.06	86.75	90.82
ELMo*	96.71	93.86	95.27

Table 4
Accuracy (SMATCH) on the test data. ACE_1 is evaluated on the whole data set: sentences that do not receive results are taken as empty graphs. ACE_2 is evaluated on the successfully parsed data only.

Model	Node	Arc	SMATCH
ACE_1	95.51	91.90	93.69
ACE_2	96.42	92.84	94.61
Buys and Blunsom (2017)	89.06	84.96	87.00
Chen, Sun, and Wan (2018)	94.47	88.44	91.42
W2V	95.65	91.97	93.78
ELMo	94.74	86.64	90.60
ELMo*	96.42	93.73	95.05

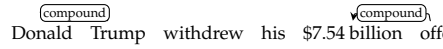
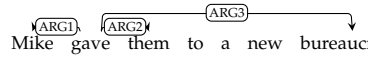
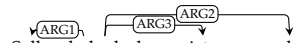
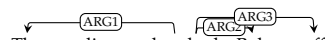
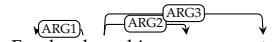
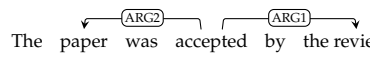
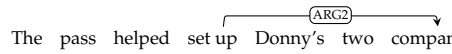
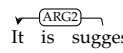
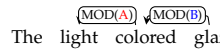
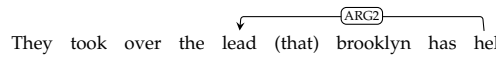
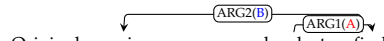
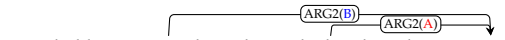
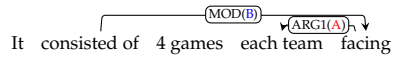
those ACE parsed data are very similar (less than 0.05% lower), so we show the results on the whole data for brevity. The numbers of ACE and Buys and Blunsom's (2017) are different from the results they reported due to the different SMATCH evaluation tools. Our *ELMo** model achieves an accuracy of 95.05, which is significantly better than existing parsers, demonstrating the effectiveness of this parsing architecture.

5. Linguistic Phenomena in Question

Most benchmark data sets in NLP are drawn from text corpora, reflecting a natural frequency distribution of language phenomena. Such data sets are usually insufficient for evaluating and analyzing neural models in many advanced NLP tasks, because they may fail to capture a wide range of complex and low-frequency phenomena (Kuhnle and Copestake 2018; Belinkov and Glass 2019). Therefore, an extensive suite of unit-tests should be considered to evaluate models on their ability to handle specific linguistic phenomena (Lin and Xue 2019).

In this section, we discuss several important linguistic phenomena for evaluating semantic parsers, including lexical constructions, predicate–argument structures, phrase constructions, and non-local dependencies (Fillmore, Kay, and O'connor 1988; Kay and Fillmore 1999; Michaelis and Lambrecht 1996; Hilpert 2014), which diverge from the common average-case evaluation but are critical for understanding natural language (Goldberg 2003). The phenomena and the corresponding examples are summarized in Table 5.

Table 5
Definitions and examples of the linguistic phenomena for evaluation.

Definition	Head	Examples
`comp`: compound/ named entity	head word in compound or last name	
`as`: basic predicate-argument structures	core predicate	
`ditr`: ditransitive construction	core predicate	
`causemo`: cause motion construction	core predicate	
`way`: way construction	core predicate	
`passive`: passive verb construction	Passive verb	
`vpart`: verb-particle constructions	(B) verb+particle	
`itexpl`: expletive *it*	*it*-subject taking verb	
`ned`: adj/Noun2+ Noun1-*ed*	(A) head noun (B) Noun1-*ed*	
`argadj`: interleaved arg/adj	(A) selecting verb (B) selecting verb	
`barerel`: bare relatives (*that*-less)	(B) grapped predicate in relative	
`tough`: tough adjectives	(A) tough adjective (B) grapped predicate in *to*-VP	
`rnr`: right node raising	(A) verb/prep2 (B) verb/prep1	
`absol`: absolutives	(A) absolutive predicate (B) main clause predicate	
`vger`: verbal gerunds	(A) selecting head (B) gerund	
`control`: raising/ control constructions	(A) "upstairs" verb (B) "downstairs" verb	

5.1 Lexical Constructions: Multiword Expression

Multiword Expressions (MWEs) are lexical items made up of multiple lexemes that undergo idiosyncratic constraints and therefore offer a certain degree of idiomaticity. MWEs cover a wide range of linguistic phenomena, including fixed and semi-fixed expressions, phrasal verbs, and named entities.

Although MWEs can lead to various categorization schemes and its definitions observed in the literature tend to stress different aspects, in this article we mainly focus on **compound** and **multiword named entity**. Roughly speaking, a compound is a lexeme formed by the juxtaposition of adjacent lexemes. Compounds can be subdivided according to their syntactic function. Thus, nominal compounds are headed by a noun (e.g., *bank robbery*) or a nominalized verb (e.g., *cigarette smoking*) and verbal compounds

are headed by a verb (e.g., *London-based*). Multiword named entity is a multiword linguistic expression that rigidly designates an entity in the world, typically including persons, organizations, and locations (e.g., *International Business Machines*).

5.2 Basic Argument Structure

The term "argument structure" refers to a relationship that holds between a predicate denoting an activity, state, or event and the respective participants, which are called arguments. Argument structure is often referred to as valency (Tesnière 2018). A verb can attract a certain number of arguments, just as an atom's valency determines the number of bonds it can engage in (Ágel and Fischer 2009).

Valency is first and foremost a characteristic of verbs, but the concept can also be applied to adjectives and nouns. For instance, the adjective *certain* can form a bond with a *that*-clause in the sentence *I am certain that he left* or an infinitival clause (*John is certain to win the election*). Nouns such as *fact* can bond to a *that*-clause as well: *the fact that he left*. We view all these valency relations as basic argument structures.

5.3 Phrasal Constructions

In the past two decades, the constructivist perspective to syntax is more and more popular. For example, Goldberg (1995) argued that argument structure could not be wholly explained in terms of lexical entries alone, and syntactic constructions also lead hearers to understand some meanings. Though this perspective is very controversial, we think the related phenomena are relevant to computational semantics. To test a parser's *adaptation* ability to handle *peripheral phenomena*, we evaluate the performance on several valency-increasing and decreasing constructions, including the ditransitive construction, cause motion construction, way construction, and passive.

Ditransitive Construction. The ditransitive construction links a verb with three arguments — a subject and two objects. Whereas English verbs like *give*, *send*, *offer* conventionally include two objects in their argument structure, the same cannot be said of other verbs that occur with the ditransitive construction.

(1) Sally baked her sister a cake.

The above sentence means that Sally produced a cake so that her sister could willingly receive it. We can posit the ditransitive construction as a symbolic unit that carries meaning and that is responsible for the observed increase in the valency of *bake*. In general, the ditransitive construction conveys, as its basic sense, the meaning of a transfer between an intentional agent and a willing recipient (indirect object).

Caused-Motion Construction. The caused-motion construction can also change the number of arguments with which a verb combines and yield an additional meaning. For example, in the sentence (2), the event structure of *laugh* specifies someone who is laughing and the unusually added argument leads to a new motion event in which both the agent and the goal are specified.

(2) The audience laughed Bob off the stage.

Way Construction. This construction specifies the lexical element *way* and a possessive determiner in its form. For example, in (3), the construction evokes a scenario in which an agent moves along a path that is difficult to navigate, thus adds up two arguments in the process — the *way* argument and a path/goal argument that is different from the caused-motion construction.

(3) Frank dug his way out of prison.

Passive. The passive construction with *be* is most often discussed as the marked counterpart of active sentences with transitive verbs. For example, in (4) the subject of the active (*the reviewer*) appears in the corresponding passive sentences as an oblique object marked with the preposition *by*, and it is possible to omit this argument in the passive. It is this type of omission that justifies categorizing the passive as a valency-decreasing construction.

(4) The paper was accepted (by the reviewer).

5.4 BFOZ's Ten Constructions

Bender et al. (2011) proposed a selection of ten challenging linguistic phenomena, each of which consists of 100 examples from English Wikipedia and occurs with reasonably high frequency in running text. Their selection (hereafter BFOZ) considers lexical (e.g., `vpart`), phrasal (e.g., `ned`), as well as non-local (e.g., `rnr`) constructions. The definitions and examples of the linguistic phenomena are outlined in Table 5, which considers representative local and non-local dependencies. Refer to their paper for more information.

In some phenomena, there are subtypes A and B, corresponding to different arcs in the structure. It is noted that the number of "A" and the number of "B" are not necessarily equal, as illustrated in the example of `control` in the table. Some sentences contain more than one instance of the phenomenon they illustrate and multiple sets of dependencies are recorded. In total, the evaluation data consists of 2,127 dependency triples for the 1,000 sentences.

6. Evaluation and Analysis

To investigate the type of representations manipulated by different parsers, in this section, we evaluate the ACE parser and our parser regarding the linguistic phenomena discussed in Section 5. In order to get the bilexical relationship, we use SMATCH to obtain all alignments between the output and ground truth.

6.1 Lexical Construction

`MRS` uses the abstract predicate `compound` to denote compounds as well as light-weight named entities. The edge labeled with *ARG1* denotes the root of the compound structure and thus can help to distinguish the type of the compound (nominal or verbal compounds), and the nodes in named entities are labeled as *named-relation*. The head words of the compound in the test set can be other types such as adjectives, and due to their data sparsity in the test data, we just omit this part. The results are illustrated in Table 6.

Table 6
Accuracy of lexical constructions.

Type	Example	#	ACE	W2V	ELMo	ELMo*
Compound	-	2,266	80.58	87.56	84.33	**89.67**
Nominal *w/ nominalization*	flag burning	22	85.71	**90.91**	81.82	**90.91**
Nominal *w/ noun*	pilot union	1,044	85.28	89.27	88.51	**90.04**
Verbal	state-owned	23	40.91	**82.61**	47.83	65.22
Named Entity	Donald Trump	1,153	82.92	86.93	82.74	**90.55**

Table 7
Accuracies on basic argument structure over the 1,474 test data. The accuracies are based on complete match, i.e., the predicates, arguments (nodes, edges, and the edge labels in the graph) should all be correctly parsed to their gold standard graphs.

Type	#	ACE	W2V	ELMo	ELMo*
Overall	7,108	**86.98**	81.44	74.56	84.70
Total verb	4,176	**85.34**	77.59	69.08	81.81
Basic verb	2,354	**85.79**	80.76	73.70	83.90
ARG1	1,683	**90.25**	87.17	80.45	89.07
ARG2	1,995	**90.48**	84.96	81.95	87.85
ARG3	195	**82.63**	58.46	55.90	72.31
Verb-particle	1,761	**84.69**	73.31	62.86	78.99
ARG1	1,545	**89.57**	80.45	75.15	84.72
ARG2	923	**86.27**	78.80	68.42	82.73
ARG3	122	**81.88**	58.44	47.40	73.38
Total noun	394	**92.41**	87.56	72.34	90.61
Total adjective	2,538	**89.27**	87.31	84.48	89.05

From the table, we find that ELMo* performs much better than ACE, especially for the named entity recognition (the total number of verbal compounds in the test set is rather small and does not affect the overall performance too much). It is noted that even the pure ELMo alone can achieve fairly good results, indicating that those pretrained embedding-based models are good at capturing local semantic information such as compound constructions and named entities.

6.2 Basic Argument Structure

The detailed performances on the 1,474 test data in terms of basic argument structure are shown in Table 7. In MRS, different senses of a predicate are distinguished by optional sense labels. Usually, the verb with its basic sense will be assigned the sense label as _v_1 (e.g., _look_v_1), while verb particle construction is handled semantically by having the verb contribute a relation particular to the combination (e.g., _look_v_up). In the evaluation, we also categorize the verb into basic verbs and verb particle constructions and show the detailed performance.

As can be observed from the table, the overall performance of ELMo* is relatively worse than the one of ACE, and this is mainly due to the relatively low accuracy of verb particle constructions and ARG3. As for the pure ELMo model, this issue will be exacerbated. The verb particle construction emphasizes combinations, and ARG3

Table 8
Accuracies on phrasal construction, including the valency increasing evaluation over 300 selected sentences and the valency decreasing evaluation over the 1,474 test data.

Type	#	ACE	W2V	ELMo	ELMo*
`ditr`	100	87.36	90.00	88.00	**93.00**
ARG1	98	**97.65**	95.92	94.90	96.94
ARG2	100	**100.00**	99.00	98.00	99.00
ARG3	100	87.36	94.00	93.00	**95.00**
`causemo`	100	41.11	27.00	32.00	**55.00**
ARG1	94	91.86	90.43	75.53	**93.62**
ARG2	100	**100.00**	99.00	97.00	99.00
ARG3	100	43.33	30.00	45.00	**60.00**
`way`	100	**7.14**	0.00	3.00	4.00
ARG1	94	81.25	86.46	79.17	**88.54**
ARG2	100	61.22	96.00	59.00	**99.00**
ARG3	100	**9.18**	1.00	4.00	7.00
`passive`	522	**85.12**	82.57	76.05	84.87

often denotes long-distance cross words within the sentence, while pure ELMo (without LSTM) is weak in capturing such information.

6.3 Phrasal Construction

For each valency increasing construction (ditransitive construction, caused-motion construction, and way construction) introduced in Section 5.3, based on some predefined patterns, we first automatically obtained a large set of candidate sentences from Linguee,[13] a Web service that provides access to large amounts of appropriate bilingual sentence pairs found online. The paired sentences identified undergo automatic quality evaluation by a human-trained machine learning algorithm that estimates the quality of those sentences. We manually select 100 sentences for each type of linguistic phenomena to guarantee the quality of pattern-based selection in terms of correctness and diversity. In order to form the gold standard for the subsequent evaluation, we then ask a senior linguistic student who is familiar with `ERG` to annotate the argument structure of those sentences. The annotation format is based on dependency triples, identifying the head words and dependents by the surface form of the head words in the sentence suffixed with a number indicating the word position.

As for the valency decreasing construction, namely, the passive construction, `MRS` gives special treatment to passive verbs, identified by the abstract node `parg_d`. Similar to the previous evaluation, we test the parsing accuracy on `parg_d` over the 1,474 test data. The results of phrasal constructions are shown in Table 8.

The results are shown in Table 8, from which we find that all the parsers perform worse on the way construction, while on the other valency increasing constructions, ELMo* yields the best results. The performances on the three constructions are mainly affected by the performances on *ARG3*, where ELMo* performs relatively better on ditransitive and caused-motion constructions. Interestingly, it is a contrast to the results on *ARG3* in basic argument constructions.

13 `https://www.linguee.com/`.

The parsers may run across a variety of cases where a predicate appeared to be in an atypical context: None of the senses listed in the lexicon provided the appropriate role label choices for novel arguments encountered. The issue can be addressed by either adding many individual senses for every predicate compatible with a particular construction, as what the rule-based parser has done, or learning the construction pattern from the training data, as what the data-driven parser has done.

The latter option clearly had a practical advantage of requiring far less time-consuming manual expansion of the lexicon although it may also suffer from data sparsity. The annotated MRS data provides considerably wide coverage for the most frequent and predictably patterned linguistic phenomena, although it sometimes fails to include some of the rarer structures found in the long tail of language. According to our statistics, the cause motion and way constructions are very sparse in the training data — appearing 12 and 0 times, respectively, in the 35,314 sentences, severely limiting the prediction on these constructions.

6.4 BFOZ's Ten Constructions

Although the annotated style for those 10 linguistic phenomena introduced in Bender et al. (2011) is not the same as the one for MRS, we were able to associate our parser-specific results with the manually annotated target non-local dependencies, and Table 9 shows the results.

All the parsers perform markedly worse on the dependencies of `rnr(B)`, `absol(B)`, and `argadj(B)`, which have very long average distances of dependencies. Each of the parsers attempt with some success to analyze each of these phenomena, but they vary across phenomena:

1. Comparing pure ELMO and ELMO*, we can observe that in most cases, ELMO* outperforms pure ELMO especially for long-distance dependencies

Table 9
Recall of individual dependencies on Bender et al.'s ten constructions. Arc refers to the average distance between the two nodes; Δ_{+x} refers to the improvement of performance when add feature x, compared with Random.

Phenomena	Arc	ACE_1	AEC_2	Rand	W2V	ELMo	ELMo*	Δ_{+W2V}	Δ_{+ELMo}	Δ_{+LSTM}
`vpart`	3.8	79	81	71	69	46	**85**	−2	+14	+39
`itexpl`	-	**91**	**91**	52	48	63	74	−4	+23	+11
`ned(A)`	2.7	63	72	78	83	79	**88**	+5	+10	+9
`ned(B)`	1.0	81	**93**	75	79	47	83	+4	+7	+35
`argadj(A)`	1.6	78	**84**	74	75	69	76	+1	+3	+8
`argadj(B)`	6.3	50	53	39	47	43	**56**	+8	+17	+13
`barerel`	3.4	60	67	70	72	73	**75**	+2	+6	+3
`tough(A)`	2.2	88	90	**91**	90	90	86	−1	−5	−4
`tough(B)`	6.4	83	**85**	68	70	47	83	+2	+16	+37
`rnr(A)`	2.8	69	76	75	72	**77**	73	−3	−2	−4
`rnr(B)`	6.2	43	**47**	17	17	10	32	+1	+16	+22
`absol(A)`	1.8	61	68	81	83	73	**92**	+3	+11	+18
`absol(B)`	9.5	6	**7**	3	3	3	3	0	0	0
`vger(A)`	1.9	56	62	**69**	62	61	**69**	−7	0	+8
`vger(B)`	2.4	80	88	88	**89**	79	84	+2	−4	+5
`control(A)`	3.0	90	91	83	87	82	**92**	+4	+8	+9
`control(B)`	4.8	87	**89**	**89**	88	63	91	−1	+2	+28

such as `tough(B)`, `vpart`, and `control(B)`, indicating that LSTM features help to capture distant information to some extent. For example, `tough(B)` is a relatively long-distance relation, and the significant improvement could be observed when we add ELMo and LSTM.

2. Similar to the conclusion drawn in Section 6.1, in general, compared with ACE, ELMO is good at capturing relatively local dependencies that have short distances, for example, `absol(A)`, `vger(A)`, and `ned(A)`.

3. For some of the phenomena, such as `itexpl`, ACE works pretty well while all the neural models fail to achieve competitive results, and this is because `itexpl` could be completely covered by a predefined grammar whereas it is very hard to learn the relation implicitly. On the other hand, the knowledge-intensive parser will be confused when handling phenomena that are not covered by a predefined grammar (e.g., `barerel`, `absol(A)`).

6.5 Down-Sampling Data Size

Our next experiment examines this effect in a more controlled environment by down-sampling the training data and observing the performances on the development set. The results are shown in Figures 5 and 6, where we test the overall performance for lexical (compound and named entities), basic argument, and phrasal (valency-increasing and passive) constructions.

As can be seen from Figure 5, adding training data cannot help the parser predict valency-increasing constructions that much. When it comes to local constructions (lexical construction), even rather small training data can lead to relatively high performance, especially for the light-weight named entity recognition. The learning curve of the basic argument structures also serves as another complementary reflection. From Figure 6, we also find that due to the low frequency of the valency-increasing constructions in the data, the performance will stay low as the training data grows.

7. Conclusion and Discussions

In this work, we proposed a new target structure-centric parser that can produce semantic graphs (EDS here) much more accurately than previous data-driven parsers. Specifically, we achieve an accuracy of 95.05 for EDS in terms of SMATCH, which yields a significant improvement over previous data-driven parsing models. Comparing this data-intensive parser with the knowledge-intensive ACE parser sheds light on complementary strengths the different types of parser exhibit.

To this end, we have presented a thorough study of the difference in errors made between systems that leverage different methods to express the mapping between string and meaning representation. To achieve that, we used a construction-focused parser evaluation methodology as an alternative to the exclusive focus on incremental improvements in overall accuracy measures such as SMATCH. We have shown that knowledge- and data-intensive models make different types of errors and such differences can be quantified concerning linguistic constructions. Our analysis provides

insights that may lead to better semantic parsing models in the future. Below we sketch some possible new directions:

- neural tagging for the knowledge-intensive parser,
- structural inference for the data-intensive parser,
- syntactic knowledge for the data-intensive parser, and
- parser ensemble.

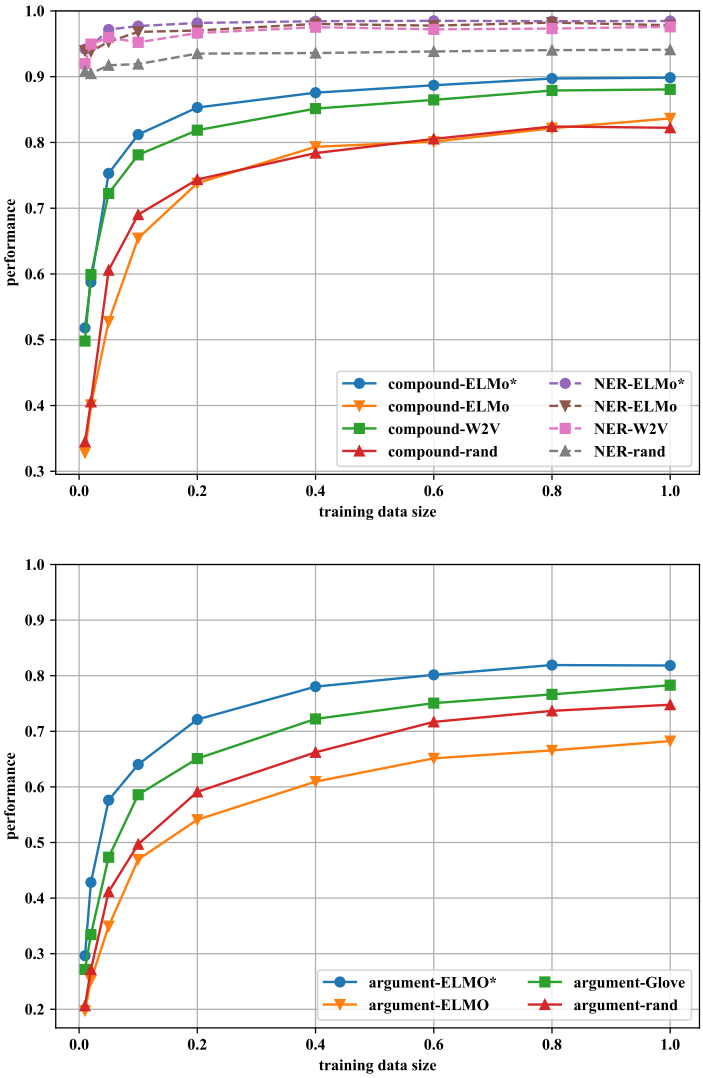


Figure 5
Performance of compound (compound), named entity (NER), and basic argument (ARG) on development set when down-sampling the training data size.

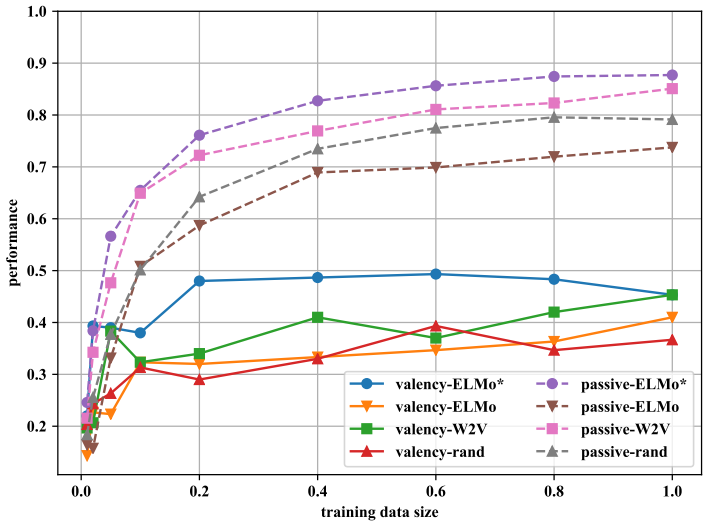


Figure 6
Performance of valency-increasing constructions (valency) and passive (passive) on the development set when down-sampling the training data size.

References

Abend, Omri and Ari Rappoport. 2013. Universal conceptual cognitive annotation (UCCA). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 228–238, Association for Computational Linguistics, pages 228–238, Sofia.

Adolphs, Peter, Stephan Oepen, Ulrich Callmeier, Berthold Crysmann, Dan Flickinger, and Bernd Kiefer. 2008. Some fine points of hybrid natural language parsing. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, pages 1380–1387, Marrakech.

Ágel, Vilmos and Klaus Fischer. 2009. In Dependency grammar and valency theory, *The Oxford Handbook of Linguistic Analysis*, Oxford University Press, pages 225–258. DOI: https://doi.org/10.1093/oxfordhb/9780199544004.013.0010

Artzi, Yoav, Kenton Lee, and Luke Zettlemoyer. 2015. Broad-coverage CCG semantic parsing with AMR. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1699–1710, Lisbon. DOI: https://doi.org/10.18653/v1/D15-1198

Banarescu, Laura, 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, Association for Computational Linguistics, Sofia.

Basile, Valerio, Johan Bos, Kilian Evang, and Noortje Venhuizen. 2012. Developing a large semantically annotated corpus. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012)*, pages 3196–3200, Istanbul.

Bau, Anthony, Yonatan Belinkov, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. 2019. Identifying and controlling important neurons in neural machine translation. In *International Conference on Learning Representations*.

Belinkov, Yonatan and James Glass. 2019. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72. DOI: https://doi.org/10.1162/tacl_a_00254

Belinkov, Yonatan, Lluís Màrquez, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. 2017. Evaluating layers of representation in neural machine translation on part-of-speech and semantic

tagging tasks. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 1, pages 1–10, Taipei.

Bender, Emily M., Dan Flickinger, Stephan Oepen, and Yi Zhang. 2011. Parser evaluation over local and non-local deep dependencies in a large corpus. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 397–408, Edinburgh.

Blevins, Terra, Omer Levy, and Luke Zettlemoyer. 2018. Deep RNNs encode soft hierarchical syntax. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 14–19, Melbourne. DOI: https://doi.org/10.18653/v1/P18-2003

Bos, Johan, Valerio Basile, Kilian Evang, Noortje Venhuizen, and Johannes Bjerva. 2017. The Groningen Meaning Bank. In Ide, Nancy and James Pustejovsky, editors, *Handbook of Linguistic Annotation*, volume 2. Springer, pages 463–496. DOI: https://doi.org/10.1007/978-94-024-0881-2_18, PMID: 29036840

Bos, Johan, Stephen Clark, Mark Steedman, James R. Curran, and Julia Hockenmaier. 2004. Wide-coverage semantic representations from a CCG parser. In *Proceedings of Coling 2004*, pages 1240–1246, Geneva. DOI: https://doi.org/10.3115/1220355.1220535

Burchardt, Aljoscha, Vivien Macketanz, Jon Dehdari, Georg Heigold, Jan-Thorsten Peter, and Philip Williams. 2017. A linguistic evaluation of rule-based, phrase-based, and neural MT engines. *Prague Bulletin of Mathematical Linguistics*, 108(1):159–170. DOI: https://doi.org/10.1515/pralin-2017-0017

Buys, Jan and Phil Blunsom. 2017. Robust incremental neural semantic graph parsing. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1215–1226, Association for Computational Linguistics, Vancouver. DOI: https://doi.org/10.18653/v1/P17-1112

Cai, Shu and Kevin Knight. 2013. Smatch: an evaluation metric for semantic feature structures. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 748–752, Sofia.

Callmeier, Ulrich. 2000. PET: A platform for experimentation with efficient HPSG processing techniques. *Journal of Natural Language Engineering*, 6(1):99–108. DOI: https://doi.org/10.1017/S1351324900002369

Cao, Junjie, Sheng Huang, Weiwei Sun, and Xiaojun Wan. 2017. 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, Association for Computational Linguistics, Vancouver. DOI: https://doi.org/10.18653/v1/P17-1193, PMCID: PMC5666792

Chen, Yufei, Weiwei Sun, and Xiaojun Wan. 2018. Accurate SHRG-based semantic parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 408–418, Melbourne. DOI: https://doi.org/10.18653/v1/P18-1038

Chen, Yufei, Yajie Ye, and Weiwei Sun. 2019. Peking at MRP 2019: Factorization- and composition-based parsing for elementary dependency structures. In *Proceedings of the Shared Task on Cross-Framework Meaning Representation Parsing at the 2019 Conference on Natural Language Learning*, pages 166–176, Hong Kong. DOI: https://doi.org/10.18653/v1/K19-2016

Clark, Stephen and James R. Curran. 2007. Wide-coverage efficient statistical parsing with CCG and log-linear models. *Computational Linguistics*, 33(4):493–552. DOI: https://doi.org/10.1162/coli.2007.33.4.493

Clark, Stephen, Julia Hockenmaier, and Mark Steedman. 2002. Building deep dependency structures using a wide-coverage CCG parser. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 327–334, Philadelphia, PA. DOI: https://doi.org/10.3115/1073083.1073138

Copestake, Ann. 2009. ***Invited Talk:*** slacker semantics: Why superficiality, dependency and avoidance of commitment can be the right way to go. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 1–9, Athens. DOI: https://doi.org/10.1007/s11168-006-6327-9

Copestake, Ann, Dan Flickinger, Carl Pollard, and Ivan A. Sag. 2005. Minimal Recursion Semantics: An introduction. *Research on Language and Computation*, 3:281–332. DOI: https://doi.org/10.1007/s11168-006-6327-9

Copestake, Ann A. and Dan Flickinger. 2000. An open source grammar development

environment and broad-coverage English grammar using HPSG. In *LREC*, pages 591–600, Athens.

Costa-jussà, Marta R. 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. DOI: https://doi.org/10.18653/v1/P16-2058

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, MN.

Dozat, Timothy and Christopher D. Manning. 2018. Simpler but more accurate semantic dependency parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 484–490, Association for Computational Linguistics, Melbourne, Australia. DOI: https://doi.org/10.18653/v1/P18-2077

Dridan, Rebecca and Stephan Oepen. 2011. Parser evaluation using elementary dependency matching. In *Proceedings of the 12th International Conference on Parsing Technologies*, pages 225–230, Dublin.

Du, Yantao, Weiwei Sun, and Xiaojun Wan. 2015. 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, Association for Computational Linguistics, Beijing. DOI: https://doi.org/10.3115/v1/P15-1149

Fillmore, Charles J., Paul Kay, and Mary Catherine O'connor. 1988. Regularity and idiomaticity in grammatical constructions: The case of let alone. *Language*, 64:501–538. DOI: https://doi.org/10.2307/414531

Flanigan, Jeffrey, Sam Thomson, Jaime Carbonell, Chris Dyer, and Noah A. Smith. 2014. A discriminative graph-based parser for the Abstract Meaning Representation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1426–1436, Association for Computational Linguistics, Baltimore. DOI: https://doi.org/10.3115/v1/P14-1134

Flickinger, Dan. 2000. On building a more efficient grammar by exploiting types. *Natural Language Engineering*, 6(1):15–28. DOI: https://doi.org/10.1017/S1351324900002370

Flickinger, Dan, Emily M. Bender, and Stephan Oepen. 2014a. ERG semantic documentation. http://www.delph-in.net/esd. Accessed on 2019-01-24.

Flickinger, Dan, Emily M. Bender, and Stephan Oepen. 2014b. Towards an encyclopedia of compositional semantics: Documenting the interface of the English Resource Grammar. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*, pages 875–881, Reykjavik.

Flickinger, Dan, 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)*, pages 1665–1671, Valletta.

Flickinger, Daniel, 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, Lisbon.

Goldberg, Adele E. 1995. *Constructions: A Construction Grammar Approach to Argument Structure*. University of Chicago Press.

Goldberg, Adele E. 2003. Constructions: A new theoretical approach to language. *Trends in Cognitive Sciences*, 7(5):219–224. DOI: https://doi.org/10.1016/S1364-6613(03)00080-9

Groschwitz, Jonas, Matthias Lindemann, Meaghan Fowlie, Mark Johnson, and Alexander Koller. 2018. AMR dependency parsing with a typed semantic algebra. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1831–1841, Melbourne. DOI: https://doi.org/10.18653/v1/P18-1170

Hershcovich, Daniel, Omri Abend, and Ari Rappoport. 2017. A transition-based directed acyclic graph parser for UCCA. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1127–1138, Association for Computational Linguistics, Vancouver. DOI: https://doi.org/10.18653/v1/P17-1104

Hilpert, Martin. 2014. *Construction Grammar and its Application to English*. Edinburgh University Press.

Isabelle, Pierre, Colin Cherry, and George Foster. 2017. A challenge set approach to evaluating machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2486–2496, Copenhagen. DOI: https://doi.org/10.18653/v1/D17-1263

Ivanova, Angelina, 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.

Ivanova, Angelina, 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 Island.

Karpathy, Andrej, Justin Johnson, and Li Fei-Fei. 2015. Visualizing and understanding recurrent networks. *arXiv preprint arXiv:1506.02078*.

Kay, Paul and Charles J Fillmore. 1999. Grammatical constructions and linguistic generalizations: The what's x doing y? construction. *Language*, 75:1–33. DOI: https://doi.org/10.2307/417472

Kiperwasser, Eliyahu and Yoav Goldberg. 2016. Simple and accurate dependency parsing using bidirectional LSTM feature representations. *Transactions of the Association for Computational Linguistics*, 4313–327. DOI: https://doi.org/10.1162/tacl_a_00101

Konstas, Ioannis, Srinivasan Iyer, Mark Yatskar, Yejin Choi, and Luke Zettlemoyer. 2017. Neural AMR: Sequence-to-sequence models for parsing and generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 146–157, Association for Computational Linguistics, Vancouver. DOI: https://doi.org/10.18653/v1/P17-1014

Kuhlmann, Marco and Peter Jonsson. 2015. Parsing to noncrossing dependency graphs. *Transactions of the Association for Computational Linguistics*, 3:559–570. DOI: https://doi.org/10.1162/tacl_a_00158

Kuhlmann, Marco and Stephan Oepen. 2016. Towards a catalogue of linguistic graph banks. *Computational Linguistics*, 42(4):819–827. DOI: https://doi.org/10.1162/COLI_a_00268

Kuhnle, Alexander and Ann Copestake. 2018. Deep learning evaluation using deep linguistic processing. In *Proceedings of the Workshop on Generalization in the Age of Deep Learning*, pages 17–23, New Orleans, LA. DOI: https://doi.org/10.18653/v1/W18-1003

Lin, Zi and Nianwen Xue. 2019. Parsing meaning representations: Is easier always better? In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages 34–43, Florence. DOI: https://doi.org/10.18653/v1/W19-3304

Linzen, Tal, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4521–535. DOI: https://doi.org/10.1162/tacl_a_00115

McCann, Bryan, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*, pages 6294–6305.

Michaelis, Laura A. and Knud Lambrecht. 1996. Toward a construction-based theory of language function: The case of nominal extraposition. *Language*, 72:215–247. DOI: https://doi.org/10.2307/416650

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

Miyao, Yusuke and Jun'ichi Tsujii. 2008. Feature forest models for probabilistic HPSG parsing. *Computational Linguistics*, 34(1):35–80. DOI: https://doi.org/10.1162/coli.2008.34.1.35

Neubig, Graham, Yoav Goldberg, and Chris Dyer. 2017. On-the-fly operation batching in dynamic computation graphs. In *Advances in Neural Information Processing Systems*, pages 3974–3984.

Oepen, Stephan, Omri Abend, Jan Hajic, Daniel Hershcovich, Marco Kuhlmann, Tim O'Gorman, Nianwen Xue, Jayeol Chun, Milan Straka, and Zdenka Uresova. 2019. 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*, pages 1–27, Hong Kong. DOI: https://doi.org/10.18653/v1/K19-2001

Oepen, Stephan, 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)*, pages 915–926, Denver, CO. DOI: https://doi.org/10.18653/v1/S15-2153

Oepen, Stephan, 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. DOI: https://doi.org/10.3115/v1/S14-2008

Oepen, Stephan and Jan Tore Lønning. 2006. Discriminant-based MRS banking. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation*, pages 1250–1255, Genoa.

Peng, Hao, 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, Association for Computational Linguistics, Vancouver. DOI: https://doi.org/10.18653/v1/P17-1186

Peng, Xiaochang, Linfeng Song, and Daniel Gildea. 2015. A Synchronous Hyperedge Replacement Grammar based approach for AMR parsing. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 32–41, Beijing. DOI: https://doi.org/10.18653/v1/K15-1004

Peters, Matthew, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018a. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 1, pages 2227–2237, New Orleans, LA. DOI: https://doi.org/10.18653/v1/N18-1202

Peters, Matthew, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. 2018b. Dissecting contextual word embeddings: Architecture and representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509, Brussels. DOI: https://doi.org/10.18653/v1/D18-1179

Pollard, Carl and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. The University of Chicago Press, Chicago.

Qian, Peng, Xipeng Qiu, and Xuanjing Huang. 2016. Analyzing linguistic knowledge in sequential model of sentence. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 826–835, Austin, TX. DOI: https://doi.org/10.18653/v1/D16-1079

Radford, Alec, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. https://s3-us-west-2.amazonaws.com/openai-assets/researchcovers/languageunsupervised/language_understanding_paper.pdf

Sennrich, Rico. 2017. How grammatical is character-level neural machine translation? Assessing MT quality with contrastive translation pairs. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, 2, pages 376–382, Valencia. DOI: https://doi.org/10.18653/v1/E17-2060

Shi, Xing, Inkit Padhi, and Kevin Knight. 2016. Does string-based neural MT learn source syntax? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1526–1534, Austin, TX. DOI: https://doi.org/10.18653/v1/D16-1159

Steedman, M. 1996. *Surface Structure and Interpretation*, Linguistic Inquiry Monographs. MIT Press.

Steedman, Mark. 2000. *The syntactic process*. MIT Press, Cambridge, MA, USA.

Tenney, Ian, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R. Thomas McCoy, Najoung Kim, Benjamin Van Durme, Sam Bowman, Dipanjan Das, and Ellie Pavlick. 2019. What do you learn from context? Probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*, pages 1–17.

Tesnière, Lucien. 2018. *Elements of Structural Syntax*. John Benjamins Publishing Company.

Wang, Alex, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*,

pages 353–355, Association for Computational Linguistics, Brussels. DOI: https://doi.org/10.18653/v1/W18-5446

Warstadt, Alex, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641. DOI: https://doi.org/10.1162/tacl_a_00290

Zhang, Xun, Yantao Du, Weiwei Sun, and Xiaojun Wan. 2016. Transition-based parsing for deep dependency structures. *Computational Linguistics*, 42(3):353–389. DOI: https://doi.org/10.1162/COLI_a_00252

Zhang, Yi, Stephan Oepen, and John Carroll. 2007. Efficiency in unification-based n-best parsing. In *Proceedings of the Tenth International Conference on Parsing Technologies*, pages 48–59, Prague. DOI: https://doi.org/10.3115/1621410.1621417, PMCID: PMC1942019

Zhang, Yue, Wei Jiang, Qingrong Xia, Junjie Cao, Rui Wang, Zhenghua Li, and Min Zhang. 2019. SUDA-Alibaba at MRP 2019: Graph-based models with BERT. In *Proceedings of the Shared Task on Cross-Framework Meaning Representation Parsing at the 2019 Conference on Natural Language Learning*, pages 149–157, Hong Kong. DOI: https://doi.org/10.18653/v1/K19-2014x