Infusing Finetuning with Semantic Dependencies

Zhaofeng Wu[♠] **Hao Peng**[♠] **Noah A. Smith**[♠◇]

[♠]Paul G. Allen School of Computer Science & Engineering, University of Washington
[◇]Allen Institute for Artificial Intelligence
{zfw7,hapeng,nasmith}@cs.washington.edu

Abstract

For natural language processing systems, two kinds of evidence support the use of text representations from neural language models ''pretrained'' on large unannotated corpora: performance on application-inspired benchmarks (Peters et al., 2018, *inter alia*), and the emergence of syntactic abstractions in those representations (Tenney et al., 2019, *inter alia*). On the other hand, the lack of grounded supervision calls into question how well these representations can ever capture meaning (Bender and Koller, 2020). We apply novel probes to recent language models—specifically focusing on predicate-argument structure as operationalized by semantic dependencies (Ivanova et al., 2012)—and find that, unlike syntax, semantics is not brought to the surface by today's pretrained models. We then use convolutional graph encoders to explicitly incorporate semantic parses into task-specific finetuning, yielding benefits to natural language understanding (NLU) tasks in the GLUE benchmark. This approach demonstrates the potential for general-purpose (rather than task-specific) linguistic supervision, above and beyond conventional pretraining and finetuning. Several diagnostics help to localize the benefits of our approach.[1]

1 Introduction

The past decade has seen a paradigm shift in how NLP systems are built, summarized as follows:

- Before, general-purpose linguistic modules (e.g., part-of-speech taggers, word-sense disambiguators, and many kinds of parsers) were constructed using supervised learning from linguistic datasets. These were often applied as preprocessing to text as part of larger systems for information extraction, question answering, and other applications.
- Today, general-purpose representation learning is carried out on large, unannotated corpora—effectively a kind of unsupervised learning known as ''pretraining''—and then the representations are ''finetuned'' on application-specific datasets using conventional end-to-end neural network methods.

The newer paradigm encourages an emphasis on corpus curation, scaling up pretraining, and translation of end-user applications into trainable ''tasks,'' purporting to automate most of the labor requiring experts (linguistic theory construction, annotation of data, and computational model design). Apart from performance improvements on virtually every task explored in the NLP literature, a body of evidence from probing studies has shown that pretraining brings linguistic abstractions to the surface, without explicit supervision (Liu et al., 2019a; Tenney et al., 2019; Hewitt and Manning, 2019; Goldberg, 2019, *inter alia*).

There are, however, reasons to pause. First, some have argued from first principles that learning mappings from form to meaning is hard from forms alone (Bender and Koller, 2020).[2] Second, probing studies have focused more heavily on *syntax* than on *semantics* (i.e., mapping of forms to abstractions of meaning intended by people speaking in the world). Tenney et al. (2019) noted that the BERT model (Devlin et al., 2019) offered more to syntactic tasks like constituent and dependency relation labeling than semantic ones

[1]https://github.com/ZhaofengWu/SIFT.

[2]In fact, Bender and Koller (2020) argued that this is impossible for grounded semantics. Our probing analysis, along with recent efforts (Kovaleva et al., 2019; Liu et al., 2019a), suggests that modern pretrained models are poor at surfacing predicate-argument semantics.

Transactions of the Association for Computational Linguistics, vol. 9, pp. 226–242, 2021. https://doi.org/10.1162/tacl_a_00363
Action Editor: Benjamin Van Durme. Submission batch: 9/2020; Revision batch: 11/2020; Published 3/2021.

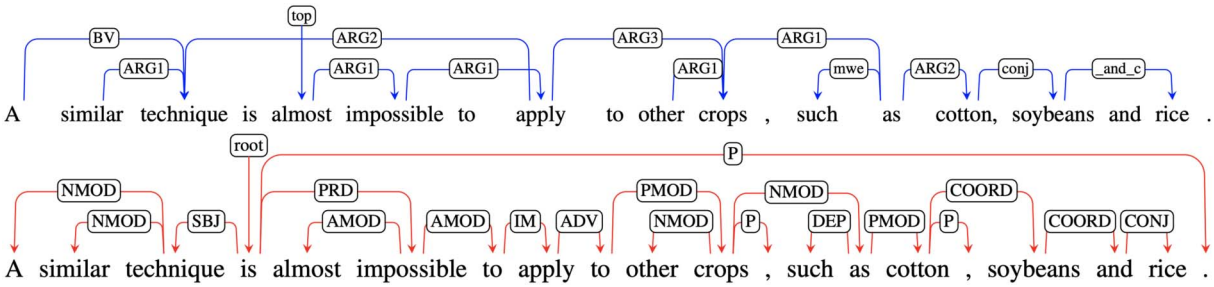

Figure 1: An example sentence in the DM (top, blue) and Stanford Dependencies (bottom, red) format, taken from Oepen et al. (2015) and Ivanova et al. (2012).

like Winograd coreference and semantic proto-role labeling. Liu et al. (2019a) showed that pretraining did not provide much useful information for entity labeling or coreference resolution. Kovaleva et al. (2019) found minimal evidence that the BERT attention heads capture FrameNet (Baker et al., 1998) relations. We extend these findings in §3, showing that representations from the RoBERTa model (Liu et al., 2019b) are relatively poor at surfacing information for a predicate-argument semantic parsing probe, compared to what can be learned with finetuning, or what RoBERTa offers for *syntactic* parsing. The same pattern holds for BERT.

Based on that finding, we hypothesize that semantic supervision may still be useful to tasks targeting natural language ''understanding.'' In §4, we introduce **semantics-infused finetuning** (SIFT), inspired by pre-neural pipelines. Input sentences are first passed through a semantic dependency parser. Though the method can accommodate any graph over tokens, our implementation uses the DELPH-IN MRS-derived dependencies, known as ''DM'' (Ivanova et al., 2012), illustrated in Figure 1. The task architecture learned during finetuning combines the pretrained model (here, RoBERTa) with a relational graph convolutional network (RGCN; Schlichtkrull et al., 2018) that reads the graph parse. Though the same graph parser can be applied at inference time (achieving our best experimental results), benefits to task performance are in evidence in a ''light'' model variant without inference time parsing and with the same inference cost as a RoBERTa-only baseline.

We experiment with the GLUE benchmarks (§5), which target many aspects of natural language understanding (Wang et al., 2018). Our model consistently improves over both base and large-sized RoBERTa baselines.[3] Our focus is not on achieving a new state of the art, but we note that SIFT can be applied orthogonally alongside other methods that have improved over similar baselines, such as Raffel et al. (2020) and Clark et al. (2020), which used alternative pretraining objectives, and Jiang et al. (2020), which proposed an alternative finetuning optimization framework. In §6, we use the HANS (McCoy et al., 2019) and GLUE (Wang et al., 2018) diagnostics to better understand where our method helps on natural language inference tasks. We find that our model's gains strengthen when finetuning data is reduced, and that our approach is more effective than alternatives that do not use the full labeled semantic dependency graph.

2 Predicate-Argument Semantics as Dependencies

Though many formalisms and annotated datasets have been proposed to capture various facets of natural language semantics, here our focus is on predicates and arguments evoked by words in sentences. Our experiments focus on the DELPH-IN dependencies formalism (Ivanova et al., 2012), commonly referred to as ''DM'' and derived from minimal recursion semantics (Copestake et al., 2005) and head-driven phrase structure grammar (Pollard and Sag, 1994). This formalism, illustrated in Figure 1 (top, blue) has the appealing property that a sentence's meaning is represented as a labeled, directed graph. Vertices are words (though not every word is a vertex), and 59 labels are used to characterize argument and adjunct relationships, as well as conjunction.

[3]RoBERTa-base and RoBERTa-large use the same pretraining data and only differ in the number of parameters.

Other semantic formalisms such as PSD (Hajic et al., 2012), EDS (Oepen and Lønning, 2006), and UCCA (Abend and Rappoport, 2013) also capture semantics as graphs. Preliminary experiments showed similar findings using these. Frame-based predicate-argument representations such as those found in PropBank (Palmer et al., 2005) and FrameNet (Baker et al., 1998) are not typically cast as graphs (rather as ''semantic role labeling''), but see Surdeanu et al. (2008) for data transformations and Peng et al. (2018b) for methods that help bridge the gap.

Graph-based formalizations of predicate-argument semantics, along with organized shared tasks on semantic dependency parsing (Oepen et al., 2014, 2015), enabled the development of data-driven parsing methods following extensive algorithm development for dependency *syntax* (Eisner, 1996; McDonald et al., 2005). Even before the advent of the pretraining-finetuning paradigm, labeled F_1 scores above 0.9 were achieved (Peng et al., 2017).

Some similarities between DM and dependency syntax (e.g., the Stanford dependencies, illustrated in Figure 1, bottom, red; de Marneffe et al., 2006) are apparent: both highlight *bilexical* relationships. However, semantically empty words (like infinitival *to*) are excluded from the semantic graph, allowing direct connections between semantically related pairs (e.g., *technique* $\leftarrow$ *apply*, *impossible* $\rightarrow$ *apply*, and *apply* $\rightarrow$ *crops*, all of which are mediated by other words in the syntactic graph). DM analyses need not be trees as in most syntactic dependency representations,[4] so they may more directly capture the meaning of many constructions, such as control.

3 Probing RoBERTa for Predicate-Argument Semantics

The methodology known as ''linguistic probing'' seeks to determine the level to which a pretrained model has rediscovered a particular linguistic abstraction from raw data (Shi et al., 2016; Adi et al., 2017; Hupkes et al., 2018; Belinkov and Glass, 2019, *inter alia*). The procedure is:

1. Select an annotated dataset that encodes the theoretical abstraction of interest into a predictive task, usually mapping sentences to linguistic structures. Here we will consider the Penn Treebank (Marcus et al., 1993) converted to Stanford dependencies and the DM corpus from CoNLL 2015's shared task 18 (Oepen et al., 2015).[5]
2. Pretrain. We consider RoBERTa and BERT.
3. Train a full-fledged ''ceiling'' model with finetuned representations. It can be seen as proxy to the best performance one can get with the pretrained representations.
4. Train a supervised ''probe'' model for the task with the pretrained representations. Importantly, the pretrained representations should be *frozen*, and the probe model should be lightweight with limited capacity, so that its performance is attributable to pretraining. We use a linear probe classifier.
5. Compare, on held-out data, the probe model against the ceiling model. Through such a comparison, we can estimate the extent to which the pretrained model ''already knows'' how to do the task, or, more precisely, brings relevant features to the surface for use by the probing model.

Liu et al. (2019a) included isolated DM arc prediction and labeling tasks and Tenney et al. (2019) conducted ''edge probing.'' To our knowledge, full-graph semantic dependency parsing has not been formulated as a probe.

For both syntactic and semantic parsing, our full ceiling model and our probing model are based on the Dozat and Manning (2017, 2018) parser that underlies many state-of-the-art systems (Clark et al., 2018; Li et al., 2019, *inter alia*). Our ceiling model contains nonlinear multilayer perceptron (MLP) layers between RoBERTa/BERT and the arc/label classifiers, as in the original parser, and finetunes the pretrained representations. The probing model, trained on the same data, freezes the representations and removes the MLP layers, yielding a linear model with limited capacity. We measure the conventionally reported metrics: labeled attachment score for dependency parsing and labeled F_1 for semantic parsing, as well as labeled and unlabeled exact match scores. We follow the standard practice and use

[4] The enhanced universal dependencies of Schuster and Manning (2016) are a counterexample.

[5] These are both derived from the same *Wall Street Journal* corpus and have similar size: the syntactic dependency dataset has 39,832/2,416 training/test examples, while the DM dataset has 33,964/1,410.

Metrics	PTB SD				CoNLL 2015 DM			
	Abs Δ	Rel Δ	Ceiling	Probe	Abs Δ	Rel Δ	Ceiling	Probe
LAS/F_1	$-13.5_{\pm 0.2}$	$-14.2\%_{\pm 0.2}$	$95.2_{\pm 0.1}$	$81.7_{\pm 0.1}$	$-23.5_{\pm 0.1}$	$-24.9\%_{\pm 0.2}$	$94.2_{\pm 0.0}$	$70.7_{\pm 0.2}$
LEM	$-36.4_{\pm 0.8}$	$-72.4\%_{\pm 1.1}$	$50.3_{\pm 0.5}$	$13.9_{\pm 0.5}$	$-45.4_{\pm 1.1}$	$-93.5\%_{\pm 0.5}$	$48.5_{\pm 1.2}$	$3.1_{\pm 0.2}$
UEM	$-46.3_{\pm 0.7}$	$-73.2\%_{\pm 0.5}$	$63.3_{\pm 0.8}$	$17.0_{\pm 0.3}$	$-48.8_{\pm 1.0}$	$-92.8\%_{\pm 0.5}$	$52.6_{\pm 1.0}$	$3.8_{\pm 0.2}$

(a) Base.

Metrics	PTB SD				CoNLL 2015 DM			
	Abs Δ	Rel Δ	Ceiling	Probe	Abs Δ	Rel Δ	Ceiling	Probe
LAS/F_1	$-17.6_{\pm 0.1}$	$-18.5\%_{\pm 0.1}$	$95.3_{\pm 0.0}$	$77.7_{\pm 0.1}$	$-26.7_{\pm 0.3}$	$-28.3\%_{\pm 0.3}$	$94.4_{\pm 0.1}$	$67.7_{\pm 0.2}$
LEM	$-40.0_{\pm 0.6}$	$-77.2\%_{\pm 0.4}$	$51.9_{\pm 0.6}$	$11.8_{\pm 0.2}$	$-46.6_{\pm 1.1}$	$-94.4\%_{\pm 0.1}$	$49.3_{\pm 1.1}$	$2.7_{\pm 0.0}$
UEM	$-50.2_{\pm 0.6}$	$-77.4\%_{\pm 0.2}$	$64.8_{\pm 0.7}$	$14.6_{\pm 0.2}$	$-50.0_{\pm 1.1}$	$-93.9\%_{\pm 0.2}$	$53.2_{\pm 1.0}$	$3.3_{\pm 0.1}$

(b) Large.

Table 1: The RoBERTa-base (top) and RoBERTa-large (bottom) parsing results for the full ceiling model and the probe on the PTB Stanford Dependencies (SD) test set and CoNLL 2015 in-domain test set. We also report their absolute and relative differences (probe – full). The smaller the magnitude of the difference, the more relevant content the pretrained model already encodes. We report the canonical parsing metric (LAS for PTB dependency and labeled F_1 for DM) and labeled/unlabeled exact match scores (LEM/UEM). All numbers are mean ± standard deviation across three seeds.

the Chu-Liu-Edmonds algorithm (Chu and Liu, 1965; Edmonds, 1967) to decode the syntactic dependency trees and greedily decode the semantic graphs with local edge/label classification decisions. See Appendix B for training details.

Comparisons between absolute scores on the two tasks are less meaningful. Instead, we are interested in the *difference* between the probe (largely determined by pretrained representations) and the ceiling (which benefits also from finetuning). Prior work leads us to expect that the semantic probe will exhibit a larger difference than the syntactic one, signalling that pretraining surfaces syntactic abstractions more readily than semantic ones. This is exactly what we see in Tables 1 across all metrics, for both RoBERTa-base and RoBERTa-large, where all relative differences (probe – full) are greater in magnitude for parsing semantics than syntax. Surprisingly, RoBERTa-large achieves worse semantic and syntactic probing performance than its base-sized counterpart across all metrics. This suggests that larger pretrained representations do not necessarily come with better structural information for downstream models to exploit. In Appendix C, we also show that BERT-base shows the same qualitative pattern.

4 Finetuning with Semantic Graphs

Given pretrained RoBERTa's relative incapability of surfacing semantic structures (§3) and the importance of modeling predicate-argument semantics (§2), we hypothesize that incorporating such information into the RoBERTa finetuning process should benefit downstream NLU tasks.

SIFT, briefly outlined in §4.1, is based on the relational graph convolutional network (RGCN; Schlichtkrull et al., 2018). §4.2 introduces a lightweight variant of SIFT aiming to reduce test time memory and runtime.

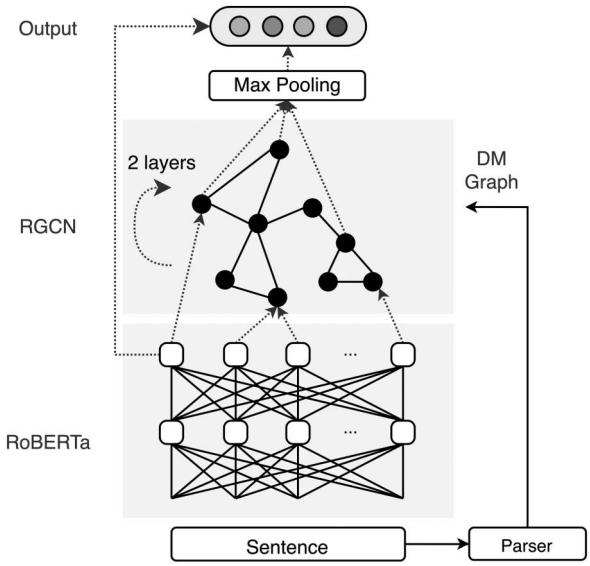

Figure 2: SIFT architecture. The sentence is first contextualized using RoBERTa, and then parsed. RGCN encodes the graph structures on top of RoBERTa. We max-pool over the RGCN's outputs for onward computation.

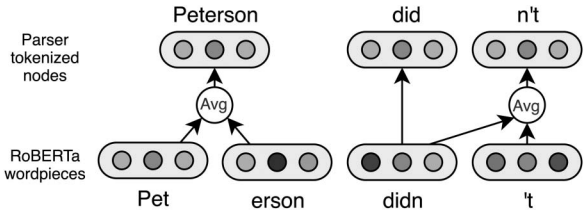

Figure 3: To get the representation of a node, we average the vectors of the wordpieces it is aligned to.

4.1 SIFT

SIFT first uses an external parser to get the semantic analysis for the input sentence. Then it contextualizes the input with a pretrained RoBERTa model, the output of which is fed into a graph encoder building on the semantic parse. We use RGCN to encode the DM structures, which are labeled graphs. The model is trained end-to-end. Figure 2 diagrams this procedure.

RGCN. RGCN can be understood as passing vector ''messages'' among vertices in the graph. The nodes are initially represented with RoBERTa token embeddings. At each RGCN layer, each node representation is updated with a learned composition function, taking as input the vector representations of the node's neighbors as well itself. Each DM relation type is associated with a separately parameterized composition function. For tasks such as text classification or regression, we max-pool over the final RGCN layer's output to obtain a sequence-level representation for onward computation. Readers are referred to Appendix A and Schlichtkrull et al. (2018) for further details.

Note on Tokenization. RoBERTa uses byte-pair encodings (BPE; Sennrich et al., 2016), differing from the CoNLL 2019 tokenizer (Oepen et al., 2019) used by the parser. To get each token's initial representation for RGCN, we average RoBERTa's output vectors for the BPE wordpieces that the token is aligned to (illustrated in Figure 3).

4.2 SIFT-Light

Inspired by the scaffold model of Swayamdipta et al. (2018), we introduce SIFT-Light, a light-weight variant of SIFT that aims to reduce time and memory overhead at test time. During inference it does *not* rely on explicit semantic structures and therefore has the same computational cost as the RoBERTa baseline.

SIFT-Light learns two classifiers (or regressors): (1) a main linear classifier on top of RoBERTa f_{RoBERTa}; (2) an auxiliary classifier f_{RGCN} based on SIFT. They are separately parameterized at the classifier level, but share the same underlying RoBERTa. They are trained on the same downstream task and jointly update the RoBERTa model. At test time, we only use f_{RoBERTa}. The assumption behind SIFT-Light is similar to the scaffold framework of Swayamdipta et al. (2018): by sharing the RoBERTa parameters between the two classifiers, the contextualized representations steer towards downstream classification with semantic encoding. One key difference is that SIFT-Light learns with two different architectures for the same task, instead of using the multitask learning framework of Swayamdipta et al. (2018). In §6.3, we find that SIFT-Light outperforms a scaffold.

4.3 Discussion

Previous works have used GCN (Kipf and Welling, 2016), a similar architecture, to encode *unlabeled* syntactic structures (Marcheggiani and Titov, 2017; Bastings et al., 2017; Zhang et al., 2020c,a, *inter alia*). We use RGCN to explicitly encode *labeled* semantic graphs. Our analysis shows that it outperforms GCN, as well as alternatives such as multitask learning with parameter-sharing (§6.3). However, this comes with a cost. In RGCN, the number of parameters linearly increases with the number of relation types.[6] In our experiments, on top of the 125M RoBERTa-base parameters, this adds approximately 3–118M parameters to the model, depending on the hyperparameter settings (see Appendix B). On top of RoBERTa-large, which itself has 355M parameters, this adds 6–121M additional parameters. The inference runtime of SIFT is 1.41–1.79× RoBERTa's with the base size and 1.30–1.53× with the large size.

SIFT incorporates semantic information only during finetuning. Recent evidence suggests that structural information can be learned with specially-designed pretraining procedures. For example, Swayamdipta et al. (2019) pretrain with syntactic chunking, requiring the entire pretraining corpus to be parsed which is computationally

[6] In experiments we upper-bound the number of the parameters by imposing a low-rank constraint on the parameter matrices by construction. See Appendix A.

prohibitive at the scale of RoBERTa's pretraining dataset. With a distillation technique, Kuncoro et al. (2020) bake syntactic supervision into the pretraining objective. Despite better accuracy on tasks that benefit from syntax, they show that the obtained syntactically-informed model *hurts* the performance on other tasks, which could restrict its general applicability. Departing from these alternatives, SIFT augments general-purpose pretraining with task-specific structural finetuning, an attractively modular and flexible solution.

5 Experiments

We next present experiments with SIFT to test our hypothesis that pretrained models for natural language understanding tasks benefit from explicit predicate-argument semantics.

5.1 Settings

We use the GLUE datasets, a suite of tests targeting natural language understanding detailed in Table 2 (Wang et al., 2018).[7] Most are classification datasets, while STS-B considers regression. Among the classifications datasets, MNLI has three classes while others have two; CoLA and SST-2 classify single sentences while the rest classify sentence pairs. We follow Dodge et al. (2020) and Vu et al. (2020) and only report development set results due to restricted GLUE test set access.

We compare the following models:

- **RoBERTa**, both the base and large variants, following Liu et al. (2019b).
- **SIFT** builds on pretrained RoBERTa, with 2 RGCN layers. To generate semantic graphs, we use the semantic dependency parser by Che et al. (2019) which held the first place in the CoNLL 2019 shared task (Oepen et al., 2019) with 92.5 labeled F_1 for DM.[8]
- **SIFT-Light** (§4.2) is trained similarly to SIFT, but does not rely on inference-time parsing.

[7]Following Devlin et al. (2019), we do not report WNLI results because it is hard to outperform the majority class baseline using the standard classification finetuning routine.

[8]About half of the CoNLL 2019 evaluation set is out-of-domain. Without gold semantic graph annotations for our target datasets, this can be seen as a reasonable estimation of the parser's performance for our use case.

Data	Task	|Train|	|Dev.|
CoLA	Acceptability	8.5K	1K
MRPC	Paraphrase	2.7K	409
QNLI	Entailment	105K	5.5K
RTE	Entailment	2.5K	278
SST-2	Sentiment	67K	873
STS-B	Similarity	5.8K	1.5K
QQP	Paraphrase	363K	40K
MNLI	Entailment	392K	9.8K

Table 2: GLUE datasets and statistics. CoLA: Warstadt et al. (2019); MRPC: Dolan and Brockett (2005); SST-2: Socher et al. (2013); STS-B: Cer et al. (2017); QQP: Csernai (2017); MNLI: Williams et al. (2018); QNLI is compiled by GLUE's authors using Rajpurkar et al. (2016). RTE is the concatenation of Dagan et al. (2005); Bar-Haim et al. (2006); Giampiccolo et al. (2007); Bentivogli et al. (2009).

- **Syntax-infused finetuning** is similar to SIFT but uses the *syntactic* Universal Dependencies parser (Straka, 2018; Straka and Straková, 2019) from the CoNLL 2019 shared task (Oepen et al., 2019). We include this model to confirm that any benefits to task performance are due specifically to the semantic structures.

Hyperparameters are summarized in Appendix B.

Implementation Details. We run all models across 3 seeds for the large datasets QNLI, MNLI, and QQP (due to limited computational resources), and 4 seeds for all others. As we do not aim for state of the art, we do *not* use intermediate task training, ensemble models, or re-formulate QNLI as a ranking task as done by Liu et al. (2019b). For sentence-pair classification tasks such as MNLI, we use structured decomposable attention (Parikh et al., 2016) and 2 additional RGCN layers to further propagate the attended information (Chen et al., 2017). The two graphs are separately max-pooled to obtain the final representation. See Appendix A for more details.

5.2 Main Findings

Tables 3 summarizes the GLUE development set performance of the four aforementioned models when they are implemented with RoBERTa-base and RoBERTa-large. With RoBERTa-base

Models	CoLA	MRPC	RTE	SST-2	STS-B	QNLI	QQP	MNLI ID.	MNLI OOD.	Avg.
RoBERTa	$63.1_{\pm 0.9}$	$90.1_{\pm 0.8}$	$79.0_{\pm 1.6}$	$94.6_{\pm 0.3}$	$91.0_{\pm 0.0}$	$93.0_{\pm 0.3}$	$91.8_{\pm 0.1}$	$87.7_{\pm 0.2}$	$87.3_{\pm 0.3}$	86.4
SIFT	$\mathbf{64.8}_{\pm 0.4}$	$90.5_{\pm 0.7}$	$81.0_{\pm 1.4}$	$95.1_{\pm 0.4}$	$\mathbf{91.3}_{\pm 0.1}$	$93.2_{\pm 0.2}$	$91.9_{\pm 0.1}$	$87.9_{\pm 0.2}$	$\mathbf{87.7}_{\pm 0.1}$	87.0
SIFT-Light	$64.1_{\pm 1.3}$	$90.3_{\pm 0.5}$	$80.6_{\pm 1.4}$	$94.7_{\pm 0.1}$	$\mathbf{91.2}_{\pm 0.1}$	$92.8_{\pm 0.3}$	$91.7_{\pm 0.0}$	$87.7_{\pm 0.1}$	$87.6_{\pm 0.1}$	86.7
Syntax	$63.5_{\pm 0.6}$	$90.4_{\pm 0.5}$	$80.9_{\pm 1.0}$	$94.7_{\pm 0.5}$	$91.1_{\pm 0.2}$	$92.8_{\pm 0.2}$	$91.8_{\pm 0.0}$	$87.9_{\pm 0.1}$	$\mathbf{87.7}_{\pm 0.1}$	86.7

(a) Base.

Models	CoLA	MRPC	RTE	SST-2	STS-B	QNLI	QQP	MNLI ID.	MNLI OOD.	Avg.
RoBERTa	$68.0_{\pm 0.6}$	$90.1_{\pm 0.8}$	$85.1_{\pm 1.0}$	$96.1_{\pm 0.3}$	$92.3_{\pm 0.2}$	$94.5_{\pm 0.2}$	$91.9_{\pm 0.1}$	$90.3_{\pm 0.1}$	$89.8_{\pm 0.3}$	88.7
SIFT	$\mathbf{69.7}_{\pm 0.5}$	$\mathbf{91.3}_{\pm 0.4}$	$\mathbf{87.0}_{\pm 1.1}$	$96.3_{\pm 0.3}$	$\mathbf{92.6}_{\pm 0.0}$	$94.7_{\pm 0.1}$	$\mathbf{92.1}_{\pm 0.1}$	$90.4_{\pm 0.1}$	$90.1_{\pm 0.1}$	89.3
Syntax	$69.6_{\pm 1.2}$	$91.0_{\pm 0.5}$	$86.0_{\pm 1.6}$	$95.9_{\pm 0.3}$	$92.4_{\pm 0.1}$	$94.6_{\pm 0.1}$	$\mathbf{92.0}_{\pm 0.0}$	$90.4_{\pm 0.3}$	$90.0_{\pm 0.2}$	89.1

(b) Large.

Table 3: GLUE development set results with RoBERTa-base (top) and RoBERTa-large (bottom). We report Matthews correlation for CoLA, Pearson's correlation for STS-B, and accuracy for others. We report mean ± standard deviation; for each bold entry, the mean minus standard deviation is no worse than RoBERTa's corresponding mean plus standard deviation.

(Table 3a), SIFT achieves a consistent improvement over the baseline across the board, suggesting that despite heavy pretraining, RoBERTa still benefits from explicit semantic structural information. Among the datasets, smaller ones tend to obtain larger improvements from SIFT, e.g., 1.7 Matthews correlation for CoLA and 2.0 accuracy for RTE, while the gap is smaller on the larger ones (e.g., only 0.1 accuracy for QQP). Moreover, SIFT-Light often improves over RoBERTa, with a smaller gap, making it a compelling model choice when latency is prioritized. This shows that encoding semantics using RGCN is not only capable of producing better standalone output representations, but can also benefit the finetuning of the RoBERTa-internal weights through parameter sharing. Finally, the syntax-infused model underperforms SIFT across all tasks. It only achieves minor improvements over RoBERTa, if not hurting performance. These results provide evidence supporting our hypothesis that incorporating semantic structures is more beneficial to RoBERTa than syntactic ones.

We observe a similar trend with RoBERTa-large in Table 3b, where SIFT's absolute improvements are very similar to those in Table 3a. Specifically, both achieve an 0.6 accuracy improvement over RoBERTa, averaged across all datasets. This indicates that the increase from RoBERTa-base to RoBERTa-large added little to surfacing semantic information.

6 Analysis and Discussion

In this section, we first analyze in which scenarios incorporating semantic structures helps RoBERTa. We then highlight SIFT's data efficiency and compare it to alternative architectures. We show ablation results for architectural decisions in Appendix D. All analyses are conducted on RoBERTa-base.

6.1 When Do Semantic Structures Help?

Using two diagnostic datasets designed for evaluating and analyzing natural language inference models, we find that SIFT (1) helps guard the model against frequent but *invalid* heuristics in the data, and (2) better captures nuanced sentence-level linguistic phenomena than RoBERTa.

Results on the HANS Diagnostic Data. We first diagnose the model using the HANS dataset (McCoy et al., 2019). It aims to study whether a natural language inference (NLI) system adopts three heuristics, summarized and exemplified in Table 4. The premise and the hypothesis have high surface form overlap, but the heuristics are *not* valid for reasoning. Each heuristic has both positive and negative (i.e., entailment and non-entailment) instances constructed. Due to the high surface similarity, many models tend to predict ''entailment'' for the vast majority of instances. As a result, they often reach decent accuracy on the entailment examples, but struggle on the

Heuristic	Premise	Hypothesis	Label	RoBERTa	SIFT
Lexical Overlap	The banker near the judge saw the actor.	The banker saw the actor.	E	98.3	**98.9**
	The judge by the actor stopped the banker.	The banker stopped the actor.	N	68.1	**71.0**
Sub-sequence	The artist and the student called the judge.	The student called the judge.	E	99.7	**99.8**
	The judges heard the actors resigned.	The judges heard the actors.	N	25.8	**29.5**
Constituent	Before the actor slept, the senator ran.	The actor slept.	E	**99.3**	98.8
	If the actor slept, the judge saw the artist.	The actor slept.	N	**37.9**	37.6

Table 4: HANS heuristics and RoBERTa-base and SIFT's accuracy. Examples are due to McCoy et al. (2019). ''E'': entailment. ''N'': non-entailment. Bold font indicates better result in each category.

''non-entailment'' ones (McCoy et al., 2019), on which we focus our analysis. The 30,000 test examples are evenly spread among the 6 classes (3 heuristics, 2 labels).

Table 4 compares SIFT against the RoBERTa baseline on HANS. Both struggle with non-entailment examples. SIFT yields improvements on the lexical overlap and subsequence heuristics, which we find unsurprising, given that semantic analysis directly addresses the underlying differences in meaning between the (surface-similar) premise and hypothesis in these cases. SIFT performs similarly to RoBERTa on the constituent heuristic with a 0.3% accuracy difference for the non-entailment examples. Here the hypothesis corresponds to a constituent in the premise, and therefore we expect its semantic parse to often be a subgraph of the premise's; accuracy hinges on the meanings of the connectives (e.g., *before* and *if* in the examples), not on the structure of the graphs.

Results on the GLUE Diagnostic Data. GLUE's diagnostic set (Wang et al., 2018) contains 1,104 artificially curated NLI examples to test a model's performance on various linguistic phenomena including **predicate-argument structure** (e.g., ''I opened the door.'' entails ''The door opened.'' but not ''I opened.''), **logic** (e.g., ''I have no pet puppy.'' entails ''I have no corgi pet puppy.'' but not ''I have no pets.''), **lexical semantics** (e.g., ''I have a dog.'' entails ''I have an animal.'' but not ''I have a cat.''), and **knowledge & common sense** (e.g., ''I went to the Grand Canyon.'' entails ''I went to the U.S..'' but not ''I went to Antarctica.''). Table 5 presents the results in R_3 correlation coefficient (Gorodkin, 2004). Explicit semantic dependencies help SIFT perform better on predicate-argument structure and sentence logic. On the other hand, SIFT underperforms the baseline on lexical semantics and world knowledge. We would not expect a benefit here, since semantic graphs do not add lexical semantics or world knowledge; the drop in performance suggests that some of what RoBERTa learns is lost when it is finetuned through sparse graphs. Future work might seek graph encoding architectures that mitigate this loss.

Phenomenon	RoBERTa	SIFT
Predicate Argument Structure	43.5	**44.6**
Logic	36.2	**38.3**
Lexical Semantics	**45.6**	44.8
Knowledge	**28.0**	26.3

Table 5: R_3 correlation coefficient of RoBERTa-base and SIFT on the GLUE diagnostic set.

6.2 Sample Efficiency

In §5.2, we observe greater improvements from SIFT on smaller finetuning sets. We hypothesize that the structured inductive bias helps SIFT more when the amount of finetuning data is limited. We test this hypothesis on MNLI by training different models varying the amount of finetuning data. We train all configurations with the same three random seeds. As seen in Table 6, SIFT offers larger improvements when less finetuning data is used. Given the success of the pretraining paradigm, we expect many new tasks to emerge with tiny finetuning sets, and these will benefit the most from methods like SIFT.

6.3 Comparisons to Other Graph Encoders

In this section we compare RGCN to some commonly used graph encoders. We aim to study whether or not (1) encoding graph labels helps, and (2) explicitly modeling discrete structures is necessary. Using the same experiment setting as in §5.1, we compare SIFT and SIFT-Light to

		ID.				OOD.			
Fraction	**|Train|**	**RoBERTa**	**SIFT**	**Abs Δ**	**Rel Δ**	**RoBERTa**	**SIFT**	**Abs Δ**	**Rel Δ**
100%	392k	87.7	87.9	0.2	0.2%	87.3	87.7	0.4	0.4%
0.5%	1,963	76.1	77.6	1.5	1.9%	77.1	78.2	1.1	1.4%
0.2%	785	68.6	71.0	2.5	3.5%	70.0	71.8	1.8	2.5%
0.1%	392	58.7	61.2	2.6	4.2%	60.5	63.7	3.3	5.1%

Table 6: RoBERTa-base and SIFT's performance on the entire MNLI development sets and their absolute and relative differences, with different numbers of finetuning instances randomly subsampled from the training data.

								MNLI		
Models	**CoLA**	**MRPC**	**RTE**	**SST-2**	**STS-B**	**QNLI**	**QQP**	**ID.**	**OOD.**	**Avg.**
RoBERTa	63.1	90.1	79.0	94.6	91.0	93.0	91.8	87.7	87.3	86.4
GCN	**65.2**	90.2	80.2	94.8	91.1	92.9	91.8	87.8	**87.7**	86.8
GAT	63.4	90.0	79.4	94.7	91.2	92.9	91.8	87.7	87.6	86.5
Hidden	64.2	90.2	79.7	94.5	91.0	92.8	91.8	87.1	86.7	86.4
Scaffold	62.5	**90.5**	71.1	94.3	91.0	92.6	91.7	87.7	87.6	85.5
SIFT	64.8	**90.5**	**81.0**	**95.1**	**91.3**	**93.2**	**91.9**	**87.9**	**87.7**	**87.0**
SIFT-Light	64.1	90.3	80.6	94.7	91.2	92.8	91.7	87.7	87.6	86.7

Table 7: GLUE development set results for different architectures for incorporating semantic information. The settings and metrics are identical to Table 3a. All models use the base size variant.

- Graph convolutional network (**GCN**; Kipf and Welling, 2016). GCN does *not* encode relations, but is otherwise the same as RGCN.
- Graph attention network (**GAT**; Veličković et al., 2018). Similarly to GCN, it encodes *unlabeled* graphs. Each node aggregates representations of its neighbors using an attention function (instead of convolutions).
- **Hidden** (Pang et al., 2019; Zhang et al., 2020a). It does *not* explicitly encode structures, but uses the hidden representations from a pretrained parser as additional features to the classifier.
- **Scaffold** (Swayamdipta et al., 2018) is based on multitask learning. It aims to improve the downstream task performance by additionally training the model on the DM data with a full parsing objective.

To ensure fair comparisons, we use comparable implementations for these models. We refer the readers to the works cited for further details.

Table 7 summarizes the results, with SIFT having the highest average score across all datasets. Notably, the 0.2 average absolute benefit of SIFT over GCN and 0.5 over GAT demonstrates the benefit of including the semantic relation types (labels). Interestingly, on the linguistic acceptability task—which focuses on well-formedness and therefore we expect relies more on syntax —GCN outperforms RGCN-based SIFT. GAT underperforms GCN by 0.3 on average, likely because the sparse semantic structures (i.e., small degrees of each node) make attended message passing less useful. Hidden does not on average outperform the baseline, highlighting the benefit of discrete graph structures (which it lacks). Finally, the scaffold underperforms across most tasks.

7 Related Work

Using Explicit Linguistic Information. Before pretrained contextualized representations emerged, linguistic information was commonly incorporated into deep learning models to improve their performance including part of speech (Sennrich and Haddow, 2016; Xu et al., 2016, *inter alia*) and syntax (Eriguchi et al., 2017;

Chen et al., 2017; Miwa and Bansal, 2016, *inter alia*). Nevertheless, recent attempts in incorporating syntax into pretrained models have little success on NLU: Strubell et al. (2018) found syntax to only marginally help semantic role labeling with ELMo, and Kuncoro et al. (2020) observed that incorporating syntax into BERT conversely hurts the performance on some GLUE NLU tasks. On the other hand, fewer attempts have been devoted to incorporating sentential predicate-argument semantics into NLP models. Zhang et al. (2020b) embedded semantic role labels from a pretrained parser to improve BERT. However, these features do not constitute full sentential semantics. Peng et al. (2018a) enhanced a sentiment classification model with DM but only used one-hop information and no relation modeling.

Probing Syntax and Semantics in Models. Many prior works have probed the syntactic and semantic content of pretrained transformers, typically BERT. Wallace et al. (2019) observed that BERT displays suboptimal numeracy knowledge. Clark et al. (2019) discovered that BERT's attention heads tend to surface syntactic relationships. Hewitt and Manning (2019) and Tenney et al. (2019) both observed that BERT embeds a significant amount of syntactic knowledge. Besides pretrained transformers, Belinkov et al. (2020) used syntactic and semantic dependency relations to analyze machine translation models.

8 Conclusion

We presented strong evidence that RoBERTa and BERT do not bring predicate-argument semantics to the surface as effectively as they do for syntactic dependencies. This observation motivates SIFT, which aims to incorporate explicit semantic structures into the pretraining-finetuning paradigm. It encodes automatically parsed semantic graphs using RGCN. In controlled experiments, we find consistent benefits across eight tasks targeting natural language understanding, relative to RoBERTa and a syntax-infused RoBERTa. These findings motivate continued work on task-independent semantic analysis, including training methods that integrate it into architectures serving downstream applications.

Acknowledgments

The authors thank the anonymous reviewers for feedback that improved the paper. We also thank Stephan Oepen for help in producing the CoNLL 2019 shared task companion data, Yutong Li for contributing to early experiments, and Elizabeth Clark and Lucy Lin for their suggestions and feedback. This research was supported in part by a Google Fellowship to HP and NSF grant 1562364.

References

Omri Abend and Ari Rappoport. 2013. Universal conceptual cognitive annotation (UCCA). In *Proceedings of ACL*.

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. In *Proceedings of ICLR*.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *Proceedings of ACL*. **DOI:** https://doi.org/10.3115/980845.980860

Roy Bar-Haim, Ido Dagan, Bill Dolan, Lisa Ferro, and Danilo Giampiccolo. 2006. The second PASCAL recognising textual entailment challenge. *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*.

Jasmijn Bastings, Ivan Titov, Wilker Aziz, Diego Marcheggiani, and Khalil Sima'an. 2017. Graph convolutional encoders for syntax-aware neural machine translation. In *Proceedings of EMNLP*. **DOI:** https://doi.org/10.18653/v1/D17-1209

Yonatan Belinkov, Nadir Durrani, Hassan Sajjad, Fahim Dalvi, and James Glass. 2020. On the linguistic representational power of neural machine translation models. *Computational Linguistics*, 46(1):1–52. **DOI:** https://doi.org/10.1162/coli_a_00367

Yonatan Belinkov 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

Emily M. Bender and Alexander Koller. 2020. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of ACL*. **DOI:** https://doi.org/10.18653/v1/2020.acl-main.463

Luisa Bentivogli, Ido Dagan, Hoa Trang Dang, Danilo Giampiccolo, and Bernardo Magnini. 2009. The fifth PASCAL recognizing textual entailment challenge. In *Proceedings of TAC*.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of SemEval*.

Wanxiang Che, Longxu Dou, Yang Xu, Yuxuan Wang, Yijia Liu, and Ting Liu. 2019. HIT-SCIR at MRP 2019: A unified pipeline for meaning representation parsing via efficient training and effective encoding. In *Proceedings of MRP*.

Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced LSTM for natural language inference. In *Proceedings of ACL*. **DOI:** https://doi.org/10.18653/v1/P17-1152

Yoeng-Jin Chu and Tseng-Hong Liu. 1965. On the shortest arborescence of a directed graph. *Science Sinica*, 14:1396–1400.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. **DOI:** https://doi.org/10.18653/v1/W19-4828, **PMID:** 31709923

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *Proceedings of ICLR*.

Kevin Clark, Minh-Thang Luong, Christopher D. Manning, and Quoc Le. 2018. Semi-supervised sequence modeling with cross-view training. In *Proceedings of EMNLP*. **DOI:** https://doi.org/10.18653/v1/D18-1217

Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan A Sag. 2005. Minimal recursion semantics: An introduction. *Research on Language and Computation*, 3(2–3):281–332. **DOI:** https://doi.org/10.1007/s11168-006-6327-9

Kornél Csernai. 2017. (accessed September 1, 2020). *First Quora Dataset Release: Question Pairs*.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The PASCAL recognising textual entailment challenge. In *Proceedings of the First International Conference on Machine Learning Challenges: Evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment*. **DOI:** https://doi.org/10.1007/11736790_9

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL*.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing*.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *Proceedings of ICLR*.

Timothy Dozat and Christopher D. Manning. 2018. Simpler but more accurate semantic dependency parsing. In *Proceedings of ACL*. **DOI:** https://doi.org/10.18653/v1/P18-2077

Jack Edmonds. 1967. Optimum branchings. *Journal of Research of the National Bureau of Standards*, 71B:233–240. **DOI:** https://doi.org/10.6028/jres.071B.032

Jason M. Eisner. 1996. Three new probabilistic models for dependency parsing: An exploration. In *Proceedings of COLING*. **DOI:**

https://doi.org/10.3115/992628.992688

Akiko Eriguchi, Yoshimasa Tsuruoka, and Kyunghyun Cho. 2017. Learning to parse and translate improves neural machine translation. In *Proceedings of ACL*. **DOI:** https://doi.org/10.18653/v1/P17-2012

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third PASCAL recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*. **DOI:** https://doi.org/10.3115/1654536.1654538

Yoav Goldberg. 2019. Assessing BERT's syntactic abilities. *arXiv preprint arXiv:1901.05287*.

Jan Gorodkin. 2004. Comparing two k-category assignments by a k-category correlation coefficient. *Computational Biology and Chemistry*, 28(5–6):367–374. **DOI:** https://doi.org/10.1016/j.compbiolchem.2004.09.006, **PMID:** 15556477

Jan Hajic, Eva Hajicová, Jarmila Panevová, Petr Sgall, Ondrej Bojar, Silvie Cinková, Eva Fucíková, Marie Mikulová, Petr Pajas, Jan Popelka, et al. 2012. Announcing Prague Czech-English dependency treebank 2.0. In *Proceedings of LREC*.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of NAACL*.

Dieuwke Hupkes, Sara Veldhoen, and Willem Zuidema. 2018. Visualisation and 'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure. In *Proceedings of IJCAI*.

Angelina Ivanova, Stephan Oepen, Lilja Øvrelid, and Dan Flickinger. 2012. Who did what to whom?: A contrastive study of syntacto-semantic dependencies. In *Proceedings LAW*.

Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Tuo Zhao. 2020. SMART: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. In *Proceedings of ACL*. **DOI:** https://doi.org/10.18653/v1/2020.acl-main.197, **PMCID:** PMC7218724

Thomas N. Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. In *Proceedings of ICLR*.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the dark secrets of BERT. In *Proceedings of EMNLP*. **DOI:** https://doi.org/10.18653/v1/D19-1445

Adhiguna Kuncoro, Lingpeng Kong, Daniel Fried, Dani Yogatama, Laura Rimell, Chris Dyer, and Phil Blunsom. 2020. Syntactic structure distillation pretraining for bidirectional encoders. *arXiv preprint arXiv:2005.13482*. **DOI:** https://doi.org/10.1162/tacl_a_00345

Zuchao Li, Hai Zhao, Zhuosheng Zhang, Rui Wang, Masao Utiyama, and Eiichiro Sumita. 2019. SJTU-NICT at MRP 2019: Multi-task learning for end-to-end uniform semantic graph parsing. In *Proceedings of MRP*.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew Peters, and Noah A. Smith. 2019a. Linguistic knowledge and transferability of contextual representations. In *Proceedings of NAACL*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *Proceedings of ICLR*.

Diego Marcheggiani and Ivan Titov. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. In *Proceedings of EMNLP*. **DOI:** https://doi.org/10.18653/v1/D17-1159

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2): 313–330. **DOI:** https://doi.org.10.21236/ADA273556

Thomas McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of ACL*. **DOI:** https://doi.org/10.18653/v1/P19-1334

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of NAACL*. **DOI:** https://doi.org/10.3115/1220575.1220641

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using LSTMs on sequences and tree structures. In *Proceedings of ACL*. **DOI:** https://doi.org/10.18653/v1/P16-1105

Stephan Oepen, 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 MRP*.

Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Silvie Cinkova, Dan Flickinger, Jan Hajic, and Zdenka Uresova. 2015. Semeval 2015 task 18: Broad-coverage semantic dependency parsing. In *Proceedings of SemEval*.

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 SemEval*.

Stephan Oepen and Jan Tore Lønning. 2006. Discriminant-based MRS banking. In *Proceedings of LREC*.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106. **DOI:** https://doi.org/10.1162/0891201053630264

Deric Pang, Lucy H. Lin, and Noah A. Smith. 2019. Improving natural language inference with a pretrained parser. *arXiv preprint arXiv:1909.08217*.

Ankur P. Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. In *Proceedings of EMNLP*. **DOI:** https://doi.org/10.18653/v1/D16-1244

Hao Peng, Sam Thomson, and Noah A. Smith. 2017. Deep multitask learning for semantic dependency parsing. In *Proceedings of ACL*. **DOI:** https://doi.org/10.18653/v1/P17-1186

Hao Peng, Sam Thomson, and Noah A. Smith. 2018a. Backpropagating through structured argmax using a SPIGOT. In *Proceedings of ACL*. **DOI:** https://doi.org/10.18653/v1/P18-1173, **PMID:** 30080257

Hao Peng, Sam Thomson, Swabha Swayamdipta, and Noah A. Smith. 2018b. Learning joint semantic parsers from disjoint data. In *Proceedings of NAACL*. **DOI:** https://doi.org/10.18653/v1/N18-1135, **PMCID:** PMC6327562

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL*. **DOI:** https://doi.org/10.18653/v1/N18-1202

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of EMNLP*. **DOI:** https://doi.org/10.18653/v1/D16-1264

Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*. **DOI:** https://doi.org/10.1007/978-3-319-93417-4_38

Sebastian Schuster and Christopher D. Manning. 2016. Enhanced english universal dependencies: An improved representation for natural language understanding tasks. In *Proceedings of LREC*.

Rico Sennrich and Barry Haddow. 2016. Linguistic input features improve neural machine translation. In *Proceedings of the First Conference on Machine Translation*. **DOI:** https://doi.org/10.18653/v1/W16-2209

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings ACL*. **DOI:** https://doi.org/10.18653/v1/P16-1162

Xing Shi, Inkit Padhi, and Kevin Knight. 2016. Does string-based neural MT learn source syntax? In *Proceedings of EMNLP*. **DOI:** https://doi.org/10.18653/v1/D16-1159

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of EMNLP*.

Milan Straka. 2018. UDPipe 2.0 prototype at CoNLL 2018 UD shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. **DOI:** https://doi.org/10.18653/v1/K19-2012

Milan Straka and Jana Straková. 2019. ÚFAL MRPipe at MRP 2019: UDPipe goes semantic in the meaning representation parsing shared task. In *Proceedings of MRP*.

Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. 2018. Linguistically-informed self-attention for semantic role labeling. In *Proceedings of EMNLP*. **DOI:** https://doi.org/10.18653/v1/D18-1548

Mihai Surdeanu, Richard Johansson, Adam Meyers, Lluís Màrquez, and Joakim Nivre. 2008. The CoNLL 2008 shared task on joint parsing of syntactic and semantic dependencies. In *Proceedings of CoNLL*. **DOI:** https://doi.org/10.3115/1596324.1596352

Swabha Swayamdipta, Matthew Peters, Brendan Roof, Chris Dyer, and Noah A. Smith. 2019. Shallow syntax in deep water. *arXiv preprint arXiv:1908.11047*.

Swabha Swayamdipta, Sam Thomson, Kenton Lee, Luke Zettlemoyer, Chris Dyer, and Noah A. Smith. 2018. Syntactic scaffolds for semantic structures. In *Proceedings of EMNLP*. **DOI:** https://doi.org/10.18653/v1/D18-1412

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick. 2019. What do you learn from context? Probing for sentence structure in contextualized word representations. In *Proceedings of ICLR*.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. In *Proceedings of ICLR*.

Tu Vu, Tong Wang, Tsendsuren Munkhdalai, Alessandro Sordoni, Adam Trischler, Andrew Mattarella-Micke, Subhransu Maji, and Mohit Iyyer. 2020. Exploring and predicting transferability across NLP tasks. In *Proceedings of EMNLP*. **DOI:** https://doi.org/10.18653/v1/2020.emnlp-main.635

Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner. 2019. Do NLP models know numbers? Probing numeracy in embeddings. In *Proceedings of EMNLP*. **DOI:** https://doi.org/10.18653/v1/D19-1534

Alex Wang, Amapreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of EMNLP*. **DOI:** https://doi.org/10.18653/v1/W18-5446

Alex Warstadt, 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

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of NAACL*. **DOI:** https://doi.org/10.18653/v1/N18-1101

Kun Xu, Siva Reddy, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2016. Question answering on freebase via relation extraction and textual evidence. In *Proceedings of ACL*. **DOI:** https://doi.org/10.18653/v1/P16-1220

Bo Zhang, Yue Zhang, Rui Wang, Zhenghua Li, and Min Zhang. 2020a. Syntax-aware opinion role labeling with dependency graph convolutional networks. In *Proceedings of ACL*. **DOI:** https://doi.org/10.18653/v1/2020.acl-main.297

Zhuosheng Zhang, Yuwei Wu, Hai Zhao, Zuchao Li, Shuailiang Zhang, Xi Zhou, and Xiang Zhou. 2020b. Semantics-aware BERT for language understanding. In *Proceedings of AAAI*. **DOI:** https://doi.org/10.1609/aaai.v34i05.6510

Zhuosheng Zhang, Yuwei Wu, Junru Zhou, Sufeng Duan, Hai Zhao, and Rui Wang. 2020c. Sg-net: Syntax-guided machine reading comprehension. In *Proceedings of AAAI*. **DOI:** https://doi.org/10.1609/aaai.v34i05.6511

A Detailed Model Architecture

In this section we provide a detailed illustration of our architecture.

Graph Initialization Because RoBERTa's BPE tokenization differs from the Che et al. (2019) semantic parser's CoNLL 2019 tokenization, we align the two tokenization schemes using character level offsets, as illustrated in Figure 3. For each node i, we find wordpieces $[t_j, \cdots, t_k]$ that it aligns to. We initialize its node embedding by averaging the vectors of these wordpiece followed by an learned affine transformation and a ReLU nonlinearity:

$$\mathbf{h}_i^{(0)} = \mathrm{ReLU}\left(\mathbf{W}_e \frac{1}{k-j+1}\sum_{s=j}^{k}\mathbf{e}_s\right)$$

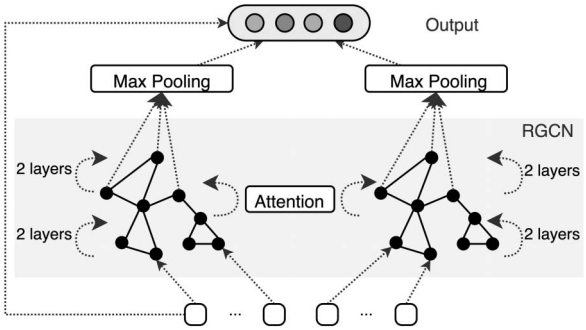

Figure 4: SIFT architecture for sentence pair tasks. Two graphs are first separately encoded using RGCN, then structured decomposable attention is used to capture the inter-graph interaction. Additional RGCN layers are used to further propagate the structured information. Finally two vectors max-pooled from both graphs are concatenated and used for onward computation. RoBERTa and the external parser are suppressed for clarity.

Here $\mathbf{W}_e$ is a learned matrix, and the $\mathbf{e}$ vectors are the wordpiece representations. The superscript on $\mathbf{h}$ denotes the layer number, with 0 being the input embedding vector fed into the RGCN layers.

Graph Update In each RGCN layer ℓ, every node's hidden representation is propagated to its direct neighbors:

$$\mathbf{h}_i^{(\ell+1)} = \mathrm{ReLU}\left(\sum_{r\in\mathcal{R}}\sum_{j\in\mathcal{N}_i^r}\frac{1}{|\mathcal{N}_i^r|}\mathbf{W}_r^{(\ell)}\mathbf{h}_j^{(\ell)} + \mathbf{W}_0^{(\ell)}\mathbf{h}_i^{(\ell)}\right)$$

where $\mathcal{R}$ is the set of all possible relations (i.e., edge labels; including inverse relations for inverse edges that we manually add corresponding to the original edges) and $\mathcal{N}_i^r$ denotes v_i's neighbors with relation r. $\mathbf{W}_r$ and $\mathbf{W}_0$ are learned parameters representing a relation-specific transformation and a self-loop transformation, respectively. We also use the basis-decomposition trick described in Schlichtkrull et al. (2018) to reduce the number of parameters and hence the memory requirement. Specifically, we construct B basis matrices; where $|\mathcal{R}| > B$, the transformation of each relation is constructed by a learned linear combination of the basis matrices. Each RGCN layer captures the neighbors information that is one hop away. We use $\ell = 2$ RGCN layers for our experiments.

Sentence Pair Tasks For sentence pair tasks, it is crucial to model sentence interaction (Parikh et al.,

Metrics	PTB SD				CoNLL 2015 DM			
	Abs Δ	Rel Δ	Full	Probe	Abs Δ	Rel Δ	Full	Probe
LAS/F_1	–13.6	–14.4%	94.6	81.0	–23.2	–24.8%	93.6	70.4
LEM	–35.8	–73.7%	48.6	12.8	–39.4	–91.6%	43.0	3.6
UEM	–44.7	–74.1%	60.3	15.7	–42.0	–91.5%	45.9	3.9

Table 8: The BERT-base parsing results for the full ceiling model and the probing model on the PTB Stanford Dependencies (SD) test set and CoNLL 2015 in-domain test set. The metrics and settings are identical to Table 1 except only one seed is used.

			MNLI	
	MRPC	STS-B	ID.	OOD.
Full	**90.5**	**91.3**	**87.9**	**87.7**
– attention	90.1	91.2	**87.9**	**87.7**
– concat	90.2	91.0	87.8	87.6

Table 9: Ablation results on the development sets of 3 GLUE datasets with a RoBERTa-base backbone.

2016). We therefore use a similar structured decomposable attention component to model the interaction between the two semantic graphs. Each node attends to the other graph's nodes using biaffine attention; its output is then concatenated to its node representation calculated in its own graph. Specifically, for two sentences a and b, we obtain an updated representation $\mathbf{h}'^{(\ell),a}$ for a as follows:

$$\alpha_{i,j} = \text{biaffine}\left(\mathbf{h}_i^{(\ell),a}, \mathbf{h}_j^{(\ell),b}\right)$$
$$\tilde{\mathbf{h}}_i^{(\ell),a} = \sum_j \alpha_{i,j}\mathbf{h}_j^{(\ell),b}$$
$$\mathbf{h}'^{(l),a} = \text{ReLU}\left(\mathbf{W}_\alpha [\mathbf{h}_i^{(\ell),a}; \tilde{\mathbf{h}}_i^{(\ell),a}; \mathbf{h}_i^{(\ell),a} - \tilde{\mathbf{h}}_i^{(\ell),a}; \mathbf{h}_i^{(\ell),a} \odot \tilde{\mathbf{h}}_i^{(\ell),a}]\right)$$

where $\mathbf{W}_\alpha$ is a learned matrix, and $\odot$ denotes the elementwise product. We do the same operation to obtain the updated $\mathbf{h}'^{(\ell),b}$. Inspired by Chen et al. (2017), we add another ℓ RGCN composition layers to further propagate the attended representation. They result in additional parameters and runtime cost compared to what was presented in §4.3.

Graph Pooling The NLU tasks we experiment with require one vector representation for each instance. We max-pool over the sentence graph (for sentence pair tasks, separately for the two graphs whose pooled output are then concatenated), concatenate it with RoBERTa's *[CLS]* embedding, and feed the result into a layer normalization layer (LN) to get the final output.

B Hyperparameters

Probing Hyperparameters. No hyperparameter tuning is conducted for the probing experiments. For the full models, we use intermediate MLP layers with dimension 512 for arc projection and 128 for label projection. The probing models do not have such layers. We minimize the sum of the arc and label cross entropy losses for both dependency and DM parsing. All models are optimized with AdamW (Loshchilov and Hutter, 2019) for 10 epochs with batch size 8 and learning rate 2×10^{-5}.

Main Experiment Hyperparameters. For SIFT, we use 2 RGCN layers for single-sentence tasks and 2 additional composition RGCN layers after the structured decomposable attention component for sentence-pair tasks. The RGCN hidden dimension is searched in $\{256, 512, 768\}$, the number of bases in $\{20, 60, 80, 100\}$, dropout between RGCN layers in $\{0, 0.2, 0.3\}$, and the final dropout after all RGCN layers in $\{0, 0.1\}$. For SIFT-Light, the training loss is obtained with $0.2\text{loss}_{\text{RGCN}} + 0.8\text{loss}_{\text{RoBERTa}}$. For all models, the number of training epochs is searched in $\{3, 10, 20\}$ and the learning rate in $\{1 \times 10^{-4}, 2 \times 10^{-5}\}$. We use 0.1 weight decay and 0.06 warmup ratio. All models are optimized with AdamW with an effective batch size of 32.

C BERT Probing Results

We replicate the RoBERTa probing experiments described in §3 for BERT. We observe similar trends where the probing model degrades more from the full model for DM than dependency syntax. This demonstrates that, like RoBERTa,

BERT also less readily surfaces semantic content than syntax.

D Ablations

In this section we ablate two major architectural choices: the sentence pair structured decomposable attention component and the use of a concatenated RoBERTa and RGCN representation rather than only using the latter. We select 3 sentence-pair datasets covering different dataset sizes and tasks with identical experimental setup as §5.1. The ablation results in Table 9 show that the full SIFT architecture performs the best.