An Empirical Revisiting of Linguistic Knowledge Fusion in Language Understanding Tasks

Changlong Yu[1]   **Tianyi Xiao**[1]   **Lingpeng Kong**[2]   **Yangqiu Song**[1]   **Wilfred Ng**[1]

[1]HKUST, Hong Kong   [2]The University of Hong Kong, Hong Kong

{cyuaq, yqsong, wilfred}@cse.ust.hk, txiao@connect.ust.hk, lpk@cs.hku.hk

Abstract

Though linguistic knowledge emerges during large-scale language model pretraining, recent work attempt to explicitly incorporate human-defined linguistic priors into task-specific fine-tuning. Infusing language models with syntactic or semantic knowledge from parsers has shown improvements on many language understanding tasks. To further investigate the effectiveness of structural linguistic priors, we conduct empirical study of replacing parsed graphs or trees with trivial ones (rarely carrying linguistic knowledge e.g., balanced tree) for tasks in the GLUE benchmark. Encoding with trivial graphs achieves competitive or even better performance in fully-supervised and few-shot settings. It reveals that the gains might not be significantly attributed to explicit linguistic priors but rather to more feature interactions brought by fusion layers. Hence we call for attention to using trivial graphs as necessary baselines to design advanced knowledge fusion methods in the future.

1 Introduction

Recently large-scale pretrained language models (Devlin et al., 2019; Liu et al., 2019; Raffel et al., 2020) have shown to gain linguistic knowledge from unlabeled corpus and achieve strong performance on many downstream natural language processing (NLP) tasks. Though probing analysis indicate that, to some extent, they can implicitly capture syntactic or semantic structures (Hewitt and Manning, 2019; Goldberg, 2019; Tenney et al., 2018; Hou and Sachan, 2021), whether they can further benefit from more explicit linguistic knowledge remains an open problem. Attempts have been made to inject syntactic biases into language model pretraining (Kuncoro et al., 2020; Wang et al., 2021; Xu et al., 2021b) or infuse finetuning with semantic information (Zhang et al., 2020a; Wu et al., 2021), and positive results are reported on downstream tasks.

However, the concerns about the effect or viability of linguistic knowledge have been raised. On the one hand, the performance gains highly rely on the availability of human-annotated dependency parsers (Sachan et al., 2021) or oracle semantic graphs (Prange et al., 2022), which limits the real-world applications. Developing accurate semantic graph parsers is yet challenging (Oepen et al., 2019; Bai et al., 2022). On the other hand, incorporating trees induced from pretrained language models (Wu et al., 2020) can outperform the ones fused with dependency-parsed trees for aspect-level sentiment analysis (Dai et al., 2021). This discovery is in line with the similar findings of trivial trees for tree-LSTM encoders in sequence modeling tasks (Shi et al., 2018). In this work, we push the envelop and answer the following two questions. Do knowledge fusion methods in Wu et al. (2021) benefit from trivial graphs that contain no linguistic information? If that's the case, where might the performance gains come from?

With the above questions, we empirically revisit the effectiveness of linguistic knowledge fusion in language understanding tasks. Motivated by Shi et al. (2018), we compare the performance between original dependency-parsed trees and balanced trees for syntax fusion, and compare the results between parsed semantic graphs and sequential graphs for semantic fusion. To our surprise, trivial graphs outperform syntactic trees or semantic graphs in full-supervised setting and achieve competitive results in few-shot setting. All the evidence again shows that the linguistic inductive bias might not be the major contributor of consistent improvements over baselines. Additional analysis gives some clues that the possible reasons are extra model parameters and feature interactions from fusion modules. This work encourages future research to add trivial graphs as necessary baselines when designing more advanced knowledge fusion methods for

downstream tasks. Our experimental code is available at `https://github.com/HKUST-KnowComp/revisit-nlu-linguistic-knowledge`.

2 Study Design

In this section, we briefly introduce two linguistic graphs, i.e., syntactic dependency trees and semantic graphs. As a comparison, we manually construct two trivial graphs to infuse with task-specific finetuning.

2.1 Linguistic Graph

Graphs have intuitively represented various linguistic phenomena in natural languages including sentence structures (Chomsky, 1957) and meanings (Koller et al., 2019).

Syntactic Dependency Tree. Syntactic trees are one of the most commonly-used linguistic structures and have long been shown useful for many NLP tasks. Syntactic dependency mainly models *head-dependent* relations between words. Dependency parsers parse the sentence into tree structures, which are further incorporated into LMs via syntax-aware attention (Nguyen et al., 2019) or graph neural networks (GNN Sachan et al. 2021).

Semantic Graphs. Different from syntactic dependency, semantic graphs aim to map sentences to high-order meaning representations with more complex structures. Normally semantics concern about *predicate-argument* relations, where predicates evoke relations of various arity and arguments filled with semantic roles that are related to each specific predicate.[1] One example is shown in Figure 1, and the characteristics of semantic graphs are the following: 1) Argument sharing leads to nodes whose in-degrees are more than one. 2) Some tokens do not contribute to meaning and not appear in the graphs. 3) There exist multiple roots. Complex semantic structures enable them to capture information that is not explicit in the single-rooted syntactic trees. Semantics could be formalized by different frameworks with respect to special linguistic assumptions. Some representative semantic formalisms are AMR (Abstract Meaning Representation, Banarescu et al., 2013) and UCCA (Abend and Rappoport, 2013). Recently Wu et al. (2021) proposed semantics-infused finetuning (SIFT) to infuse DM (DELPH-IN Minimal Recursion Semantics, Ivanova et al., 2012) graphs and achieved

---

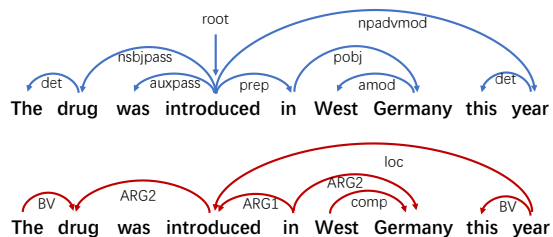[1]We refer the readers to the ACL tutorial Koller et al. (2019) for detailed explanations.

Figure 1: An example of dependency tree (blue) and DM semantic graph (red).

consistent improvements over RoBERTa (Liu et al., 2019) baselines on the GLUE (Wang et al., 2019).

DM graphs (Ivanova et al., 2012) define 59 types to characterize *predicate-argument* relationships. In order to investigate the effect of different semantic relations, we consider to only keep six common relation types, which appear in most parsed graphs, named **skeleton graphs**. These relations include ARG1, ARG2, ARG3, ARG4, compound and BV. We are interested in whether downstream tasks would still benefit from the core semantics instead of the entire linguistic graphs.

2.2 Trivial Graph

Though linguistic graphs convey useful structures, high-quality parsers are not easily available due to limited annotated graph banks (Oepen et al., 2019). If structure priors are unavailable, Shi et al. (2018) demonstrated that trivial trees, such as *gumbel tree* outperform syntactic trees when they are incorporated into tree LSTM encoders (Tai et al., 2015).

However, infusing trivial linguistic graphs with pretrained transformer models has not been explored. Similarly, we also create two types of trivial trees or graphs, which rarely contain linguistic inductive bias, to reproduce knowledge fusion experiments in Wu et al. (2021).

Binary Balanced Tree. Compared with syntactic trees, binary balanced trees are shallower and possibly easier to propagate information from leaves to the root. We assume GNN layers might benefit from the shallowness of balanced trees.

Sequential Bidirectional Graph. As the most natural and straight-forward way, tokens in the sentence are connected in the sequential order, which combines left-to-right and right-to-left chains. By doing so, GNN layers only aggregate local information rather than potentially long dependency from linguistic graphs.

Models	CoLA	MRPC	STS-B	SST-2	RTE	QNLI	QQP	MNLI	Avg.
RoBERTa	$63.1_{\pm0.9}$	$90.1_{\pm0.8}$	$91.0_{\pm0.0}$	$94.6_{\pm0.3}$	$79.0_{\pm1.6}$	$93.0_{\pm0.3}$	$91.8_{\pm0.1}$	$87.7_{\pm0.2}$	86.3
SIFT	$64.8_{\pm0.4}$	$90.5_{\pm0.7}$	$91.3_{\pm0.1}$	$95.1_{\pm0.4}$	$81.0_{\pm1.4}$	$93.2_{\pm0.2}$	$91.9_{\pm0.1}$	$87.9_{\pm0.2}$	87.0
SIFT (Ske)	$\mathbf{65.6}_{\pm0.8}$	$90.6_{\pm0.8}$	$91.4_{\pm0.2}$	$95.2_{\pm0.2}$	$\mathbf{81.8}_{\pm0.3}$	$93.2_{\pm0.2}$	$91.9_{\pm0.0}$	$87.9_{\pm0.0}$	87.2
SIFT (Seq)♠	$65.5_{\pm0.3}$	$90.9_{\pm0.3}$	$91.4_{\pm0.0}$	$95.2_{\pm0.2}$	$81.3_{\pm1.8}$	$93.2_{\pm0.1}$	$91.9_{\pm0.1}$	$87.9_{\pm0.0}$	87.2
Syntax	$63.5_{\pm0.6}$	$90.4_{\pm0.5}$	$91.1_{\pm0.2}$	$94.7_{\pm0.5}$	$80.9_{\pm1.0}$	$92.8_{\pm0.2}$	$91.8_{\pm0.0}$	$87.9_{\pm0.1}$	86.6
Balanced♡	$65.3_{\pm0.9}$	$90.5_{\pm0.2}$	$\mathbf{91.5}_{\pm0.1}$	$\mathbf{95.4}_{\pm0.3}$	$81.1_{\pm0.7}$	$93.2_{\pm0.1}$	$91.9_{\pm0.1}$	$87.9_{\pm0.1}$	87.1

(a) Base

Models	CoLA	MRPC	STS-B	SST-2	RTE	QNLI	QQP	MNLI	Avg.
RoBERTa	$68.0_{\pm0.6}$	$90.1_{\pm0.8}$	$92.3_{\pm0.2}$	$96.1_{\pm0.3}$	$85.1_{\pm1.0}$	$94.5_{\pm0.2}$	$91.9_{\pm0.1}$	$90.3_{\pm0.1}$	88.5
SIFT	$69.7_{\pm0.5}$	$91.3_{\pm0.4}$	$92.6_{\pm0.0}$	$96.3_{\pm0.3}$	$87.0_{\pm1.1}$	$94.7_{\pm0.1}$	$92.1_{\pm0.1}$	$90.4_{\pm0.1}$	89.3
SIFT (Ske)	$\mathbf{71.6}_{\pm0.3}$	$91.8_{\pm0.3}$	$92.6_{\pm0.1}$	$96.4_{\pm0.2}$	$88.5_{\pm1.2}$	$\mathbf{94.8}_{\pm0.1}$	$92.1_{\pm0.1}$	$90.4_{\pm0.2}$	89.8
SIFT (Seq)♠	$70.7_{\pm0.7}$	$\mathbf{91.9}_{\pm0.5}$	$92.6_{\pm0.1}$	$96.5_{\pm0.3}$	$\mathbf{88.7}_{\pm0.3}$	$94.7_{\pm0.1}$	$92.1_{\pm0.1}$	$90.4_{\pm0.1}$	89.7
Syntax	$69.6_{\pm1.2}$	$91.0_{\pm0.5}$	$92.4_{\pm0.1}$	$95.9_{\pm0.3}$	$86.0_{\pm1.6}$	$94.6_{\pm0.1}$	$92.0_{\pm0.0}$	$90.4_{\pm0.3}$	89.0
Balanced♡	$70.5_{\pm1.0}$	$91.5_{\pm0.4}$	$92.6_{\pm0.0}$	$96.5_{\pm0.2}$	$88.0_{\pm0.3}$	$94.7_{\pm0.1}$	$92.1_{\pm0.1}$	$90.4_{\pm0.1}$	89.5

(a) Large

Table 1: GLUE benchmark development results using RoBERTa base (top) and large (bottom). ♠: we replace parsed semantic graphs (SIFT) with trivial sequential graphs. ♡: we replace parsed dependency trees (Syntax) with balanced trees. SIFT (SKE) means encoding with skeleton graphs. Trivial graphs could outperform parsed linguistic graphs in both base and large models.

2.3 Encoding Graph Structures

Structural information can be incorporated into pre-trained transformer models by two typical strategies: adopt GNN on top of the output of transformers (Wu et al., 2021; Peng et al., 2021) and fuse structures with transformer attention layers (Nguyen et al., 2019; Zhang et al., 2020b). Followed Wu et al. (2021), in this work we use the former one, where the linguistic effects are easily disentangled for analysis.

Formally, given an input sentence $x_i = \{w_1, w_2, ..., w_L\}$ with the length L and the corresponding graph $\mathcal{G}_i$ (either linguistic graph or trivial graph), we obtain the last hidden representation, $\mathbf{H}$ after pretrained transformer layers (Vaswani et al., 2017), which also serves as the node embedding initialization of relational graph neural networks (RGCN Schlichtkrull et al. 2018) to encode $\mathcal{G}_i$. At RGCN layers, each node's representation would be updated by aggregating its neighbors' features with relational bias. We max-pool over the final RGCN layer's output as the graph representation $\mathbf{O}_g$:

$$\mathbf{O}_g = \text{MaxPooling}(\text{RGCN}([\mathcal{G}_i, \mathbf{H}])). \quad (1)$$

And the final classification feature is the concatenation of [CLS] token embedding $\mathbf{H}_0$ and pooled graph representation $\mathbf{O}_g$. Note that vanilla transformer-based models only take the [CLS] embedding as the classification feature. For sentence pair tasks, two graphs are separately encoded by RGCN with inner-attention and then aggregated to one representation as Wu et al. (2021).

3 Experiments

3.1 Implementation Details

We use the GLUE benchmark (Wang et al., 2019), a general natural language understanding test suite which contains eight datasets for text classification tasks (Details listed in Appendix A). Following the common practice, we report the averaged results of development sets over multiple seeds.

We directly adopt parsed semantic graphs from Wu et al. (2021)[2] for each dataset, using the ranked-first parser (Che et al., 2019) in the CoNLL2019 shared task (Oepen et al., 2019). For the construction of trivial graphs, we randomly sample edge labels from the relation list due to their unavailability. To be comparable with Wu et al. (2021), we apply the same model architecture and tuned the same set of parameters such as learning rate, RGCN graph layer, RGCN hidden dimension, when infusing graph structures into LMs.
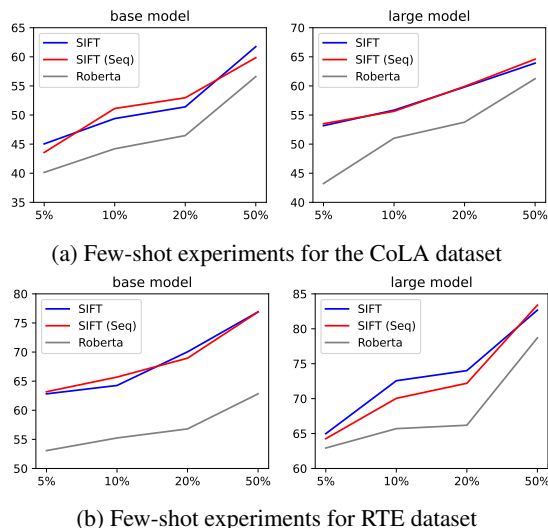
---

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

10066

(a) Few-shot experiments for the CoLA dataset

(b) Few-shot experiments for RTE dataset

Figure 2: Performance comparison of RoBERTa, SIFT and SIFT (SEQ) with different training data sizes. The rest of datasets follow similar trends.

3.2 Main results

Table 1 shows the main results of infusing with linguistic and trivial graphs. We can draw the following observations. First, sequential graphs and balanced trees achieve consistent improvements over corresponding linguistic ones. Also trivial graphs further improve RoBERTa baselines by more than 1.0 averaged point. Moreover, the random relation types in the trivial graphs have no effect on performance, again suggesting that linguistic structures and relations are not major contributors of the improvements. Second, skeleton graphs surprisingly improve the whole linguistic graphs, which indicates the fine-grained sentential semantics might not be necessary for language understanding tasks in GLUE. Note that skeleton graphs keep 75%-90% edges of parsed graphs on average (Detailed statistics in Appendix B).

To examine the impact of trivial graphs with different sizes of training data, especially in low-resource scenarios, we randomly sample 5%, 10%, 20% and 50% training data. The trivial graphs yield competitive or even better results for CoLA and RTE datasets in Fig 2 across different sizes.

3.3 Effect of Model Components

Given the strong performance of trivial graphs, we conduct the ablation study of model components in Section 2.3 besides graph structures. We first investigate whether pooling over transformer outputs can replace graph encoders, i.e., passing zero layer RGCN. The results in Table 2 show that the

Dataset	Method	Base	Large
CoLA	CLS (RoBERTa)	$63.1_{\pm 0.9}$	$68.0_{\pm 0.6}$
	CLS + RGCN (SIFT)	$64.8_{\pm 0.4}$	$69.7_{\pm 0.5}$
	CLS + 0 RGCN	$64.5_{\pm 0.9}$	$69.4_{\pm 0.9}$
	CLS + transformer	$64.8_{\pm 0.6}$	$70.1_{\pm 0.3}$
RTE	CLS (RoBERTa)	$79.0_{\pm 1.6}$	$85.1_{\pm 1.0}$
	CLS + RGCN (SIFT)	$81.0_{\pm 1.4}$	$87.0_{\pm 1.1}$
	CLS + 0 RGCN	$80.4_{\pm 0.9}$	$86.7_{\pm 0.4}$
	CLS + transformer	$81.2_{\pm 0.7}$	$87.1_{\pm 0.2}$

Table 2: Results of different variants for SIFT. "CLS + 0 RGCN" means direct pooling without graph encoders while "CLS + transformer" refers to replacing RGCN in SIFT with extra transformer layers.

concatenation of CLS embedding and direct pooled hidden states still falls behind graph fusion methods. But one interesting finding is that such simple way by introducing token embedding features can have large improvements over RoBERTa baselines.

Second, RGCN layers over sequential graphs achieve good performance in Table 1 . Considering the nature of sequential modeling ability of transformers, one question is whether additional transformer layers over sequences could learn better graph representations and improve the performance. The difference between transformer and RGCN is that the transformer captures complete graphs while RGCN takes sequential graphs as inputs. We stack more randomly-initialized transformer layers over the pretrained encoder outputs (comparable number of parameters with RGCN). Training with additional transformer layers yields similar results with RGCN encoders. From this perspective, structure biases make little difference and the gains of trivial graphs might be the results of additional token embedding features as well as their interactions via fusion modules (RGCN or transformer).

3.4 More Discussions

Besides the specific architecture discussed in §2.3, our study can be generalized to more knowledge fusion methods. For example, Zhang et al. (2020a) incorporated semantic role information by combining token embeddings and role type embeddings, which improved the performance of GLUE benchmark over BERT baselines. Similar experiments like replacing parsed role sequences with random sequences, are left for future work.

When it comes to entity knowledge-augmented methods, Raman et al. (2021) also observed similar findings as ours that using perturbed KGs can maintain the downstream performance of the original

KG though the perturbed KGs are significantly different in terms of semantics and graph structures. It again demonstrate that the way those methods use knowledge does not align with human priors. Both our findings can guide the future work on robust evaluation and explainability analysis of knowledge fusion methods.

4 Conclusion

Our study demonstrates that GLUE tasks can benefit from both trivial graphs and linguistic graphs, indicating that the performance gains of previous fusion methods should not be attributed to linguistic bias entirely. We argue that comparisons merely between methods with and without knowledge fusion may not be able to capture the whole picture. For example, without baselines considering trivial graph structures, the quality of various fused knowledge may not be accurately assessed. More careful evaluations of the effectiveness claims in existing work (Sachan et al., 2021; Liu et al., 2020; Peng et al., 2021, *inter alia*) may be encouraged in the same spirit. In addition, tasks and evaluation benchmarks are also crucial to investigate when linguistic structures help. Our study contributes to the broader question of how to accurately evaluate models that integrate external knowledge for downstream tasks, such as world or commonsense knowledge (Xu et al., 2021a; Zhu et al., 2022).

Limitations

We outline two limitations in our study. First, our comparison experiments require high-quality parsers to obtain accurate linguistic graphs. This assumption does not always hold especially for complex semantic parsing. In such case, it is rather difficult to quantify the effect of parsing results on downstream tasks' performance. Note that our constructed trivial graphs need no parser. Second, although we systematically evaluate on the GLUE benchmark, more diverse tasks like structure prediction tasks, more knowledge fusion methods, and more linguistic structures like phrase structures (Kong et al., 2015) remain to be explored in the future work.

Acknowledgements

References

Omri Abend 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, Sofia, Bulgaria. Association for Computational Linguistics.

Xuefeng Bai, Yulong Chen, and Yue Zhang. 2022. Graph pre-training for AMR parsing and generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6001–6015, Dublin, Ireland. Association for Computational Linguistics.

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

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 the Shared Task on Cross-Framework Meaning Representation Parsing at the 2019 Conference on Natural Language Learning*, pages 76–85, Hong Kong.

Noam Chomsky. 1957. Logical structure in language. *Journal of the American Society for Information Science*, 8(4):284.

Junqi Dai, Hang Yan, Tianxiang Sun, Pengfei Liu, and Xipeng Qiu. 2021. Does syntax matter? a strong baseline for aspect-based sentiment analysis with RoBERTa. In *Proceedings of the NAACL*, pages 1816–1829, Online.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the NAACL*, pages 4171–4186, Minneapolis, Minnesota.

Yoav Goldberg. 2019. Assessing bert's syntactic abilities. *ArXiv*, abs/1901.05287.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the NAACL*, pages 4129–4138, Minneapolis, Minnesota.

Yifan Hou and Mrinmaya Sachan. 2021. Bird's eye: Probing for linguistic graph structures with a simple information-theoretic approach. In *Proceedings of ACL*, pages 1844–1859, Online.

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

Alexander Koller, Stephan Oepen, and Weiwei Sun. 2019. Graph-based meaning representations: Design and processing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 6–11, Florence, Italy.

Lingpeng Kong, Alexander M. Rush, and Noah A. Smith. 2015. Transforming dependencies into phrase structures. In *Proceedings of NAACL*, pages 788–798, Denver, Colorado.

Adhiguna Kuncoro, Lingpeng Kong, Daniel Fried, Dani Yogatama, Laura Rimell, Chris Dyer, and Phil Blunsom. 2020. Syntactic structure distillation pretraining for bidirectional encoders. *Transactions of the Association for Computational Linguistics*, 8:776–794.

Tao Liu, Xin Wang, Chengguo Lv, Ranran Zhen, and Guohong Fu. 2020. Sentence matching with syntax- and semantics-aware BERT. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3302–3312, Barcelona, Spain (Online).

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

Xuan-Phi Nguyen, Shafiq Joty, Steven Hoi, and Richard Socher. 2019. Tree-structured attention with hierarchical accumulation. In *International Conference on Learning Representations*.

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 the Shared Task on Cross-Framework Meaning Representation Parsing at the 2019 Conference on Natural Language Learning*, pages 1–27, Hong Kong. Association for Computational Linguistics.

Qiwei Peng, David Weir, and Julie Weeds. 2021. Structure-aware sentence encoder in bert-based Siamese network. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*, pages 57–63, Online. Association for Computational Linguistics.

Jakob Prange, Nathan Schneider, and Lingpeng Kong. 2022. Linguistic frameworks go toe-to-toe at neuro-symbolic language modeling. In *Proceedings of the NAACL*.

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.

Mrigank Raman, Siddhant Agarwal, Peifeng Wang, Aaron Chan, Hansen Wang, Sungchul Kim, Ryan A. Rossi, Handong Zhao, Nedim Lipka, and Xiang Ren. 2021. Learning to deceive knowledge graph augmented models via targeted perturbation. *ArXiv*, abs/2010.12872.

Devendra Sachan, Yuhao Zhang, Peng Qi, and William L. Hamilton. 2021. Do syntax trees help pre-trained transformers extract information? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2647–2661, Online. Association for Computational Linguistics.

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*, pages 593–607. Springer.

Haoyue Shi, Hao Zhou, Jiaze Chen, and Lei Li. 2018. On tree-based neural sentence modeling. In *Proceedings of the EMNLP*, pages 4631–4641, Brussels, Belgium.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of ACL*, pages 1556–1566, Beijing, China.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R Bowman, Dipanjan Das, et al. 2018. What do you learn from context? probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019*.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Jianshu Ji, Guihong Cao, Daxin Jiang, and Ming Zhou. 2021. K-Adapter: Infusing Knowledge into Pre-Trained Models with Adapters. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1405–1418, Online. Association for Computational Linguistics.

Zhaofeng Wu, Hao Peng, and Noah A Smith. 2021. Infusing finetuning with semantic dependencies. *Transactions of the Association for Computational Linguistics*, 9:226–242.

Zhiyong Wu, Yun Chen, Ben Kao, and Qun Liu. 2020. Perturbed masking: Parameter-free probing for analyzing and interpreting BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4166–4176, Online.

Ruochen Xu, Yuwei Fang, Chenguang Zhu, and Michael Zeng. 2021a. Does knowledge help general nlu? an empirical study. *arXiv preprint arXiv:2109.00563*.

Zenan Xu, Daya Guo, Duyu Tang, Qinliang Su, Linjun Shou, Ming Gong, Xiaojun Quan, Daxin Jiang, and Nan Duan. 2021b. Syntax-enhanced pre-trained model. In *Proceedings of ACL*, pages 5412–5422, Online. Association for Computational Linguistics.

Zhuosheng Zhang, Yuwei Wu, Hai Zhao, Zuchao Li, Shuailiang Zhang, Xi Zhou, and Xiang Zhou. 2020a. Semantics-aware bert for language understanding. In *Proceedings of the AAAI*, volume 34, pages 9628–9635.

Zhuosheng Zhang, Yuwei Wu, Junru Zhou, Sufeng Duan, Hai Zhao, and Rui Wang. 2020b. SG-Net: Syntax-guided machine reading comprehension. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*.

Chenguang Zhu, Yichong Xu, Xiang Ren, Bill Yuchen Lin, Meng Jiang, and Wenhao Yu. 2022. Knowledge-augmented methods for natural language processing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 12–20, Dublin, Ireland.

A Statistics of GLUE Benchmark

In Table 3, we list the tasks as well as dataset statistics of the GLUE benchmark. We also show the averaged degrees of parsed semantic graphs for each dataset.

Dataset	Task	#Train	#Dev	Avg |D|
CoLA	Acceptability	8.5K	1K	1.74
MRPC	Paraphrase	2.7K	409	1.91
STS-B	Similarity	5.8K	1.5k	1.81
SST-2	Sentiment	67K	873	1.79
RTE	Entailment	2.5K	278	1.94
QNLI	Entailment	105k	5.5K	1.91
QQP	Paraphrase	363K	40K	1.82
MNLI	Entailment	392k	9.8K	1.89

Table 3: The statistics of evaluation datasets in the GLUE benchmark. #Train and #Dev refer to the size of training set and development set respectively.

Dataset	Node Rate	Edge Rate
CoLA	93.2%	89.0%
MRPC	81.9%	75.4%
STS-B	84.8%	79.7%
SST-2	87.8%	80.6%
RTE	81.5%	74.7%
QNLI	84.4%	78.6%
QQP	84.2%	78.9%
MNLI	87.2%	80.6%

Table 4: The ratios of nodes and edges in skeleton graph compared with original parsed graphs.

B Statistics of Skeleton Graphs

We present the remained ratios of nodes and edges in the skeleton graphs in Table 4.