Syntactic Structure Distillation Pretraining for Bidirectional Encoders

Adhiguna Kuncoro[*♠♢] **Lingpeng Kong**[*♠] **Daniel Fried**[*♣]
Dani Yogatama[♠] **Laura Rimell**[♠] **Chris Dyer**[♠] **Phil Blunsom**[♠♢]
[♠]DeepMind, London, UK
[♢]Department of Computer Science, University of Oxford, UK
[♣]Computer Science Division, University of California, Berkeley, CA, USA
{akuncoro,lingpenk,dyogatama,laurarimell,cdyer,pblunsom}@google.com
dfried@cs.berkeley.edu

Abstract

Textual representation learners trained on large amounts of data have achieved notable success on downstream tasks; intriguingly, they have also performed well on challenging tests of syntactic competence. Hence, it remains an open question whether scalable learners like BERT can become fully proficient in the syntax of natural language by virtue of data scale alone, or whether they still benefit from more explicit **syntactic biases**. To answer this question, we introduce a knowledge distillation strategy for injecting syntactic biases into BERT pretraining, by distilling the syntactically informative predictions of a hierarchical—albeit harder to scale—syntactic language model. Since BERT models masked words in bidirectional context, we propose to distill the approximate marginal distribution over words in context from the syntactic LM. Our approach reduces relative error by 2–21% on a diverse set of structured prediction tasks, although we obtain mixed results on the GLUE benchmark. Our findings demonstrate the benefits of syntactic biases, even for representation learners that exploit large amounts of data, and contribute to a better understanding of where syntactic biases are helpful in benchmarks of natural language understanding.

1 Introduction

Large-scale textual representation learners trained with variants of the language modeling (LM) objective have achieved remarkable success on downstream tasks (Peters et al., 2018; Devlin et al., 2019; Yang et al., 2019). Furthermore, these models have also been shown to perform remarkably well at syntactic grammaticality judgment tasks (Goldberg, 2019), and encode substantial amounts of syntax in their learned representations (Liu et al., 2019a; Tenney et al., 2019a,b; Hewitt and Manning, 2019; Jawahar et al., 2019). Intriguingly, success on these syntactic tasks has been achieved by Transformer architectures (Vaswani et al., 2017) that lack explicit notions of **hierarchical** syntactic structures.

Based on such evidence, it would be tempting to conclude that data scale alone is all we need to learn the syntax of natural language. Nevertheless, recent findings that systematically compare the syntactic competence of models trained at varying data scales suggest that model *inductive biases* are in fact more important than data scale for acquiring syntactic competence (Hu et al., 2020). Two natural questions, therefore, are the following: Can representation learners that work well at scale still benefit from explicit *syntactic biases*? And where exactly would such syntactic biases be helpful in different language understanding tasks? Here we work towards answering these questions by devising a new pretraining strategy that injects syntactic biases into a BERT (Devlin et al., 2019) learner that works well at scale. We hypothesize that this approach can improve the competence of BERT on various tasks, which provides evidence for the benefits of syntactic biases in large-scale models.

Our approach is based on the prior work of Kuncoro et al. (2019), who devised an effective knowledge distillation (KD; Bucilă et al., 2006; Hinton et al., 2015) procedure for improving the syntactic competence of scalable LMs that lack explicit syntactic biases. More concretely, their KD procedure utilized the predictions of an explicitly hierarchical (albeit hard to scale) syntactic LM, recurrent neural network grammars

[*]Equal contribution.

Transactions of the Association for Computational Linguistics, vol. 8, pp. 776–794, 2020. https://doi.org/10.1162/tacl_a_00345
Action Editor: James Henderson. Submission batch: 6/2020; Revision batch: 8/2020; Published 12/2020.

(RNNGs; Dyer et al., 2016) (§2) as a syntactically informed learning signal for a sequential LM that works well at scale.

Our setup nevertheless presents a new challenge: Here the BERT student is a denoising autoencoder that models a collection of conditionals for words in *bidirectional* context, while the RNNG teacher is an autoregressive LM that predicts words in a *left-to-right* fashion, that is $t_\phi(x_i|\mathbf{x}_{<i})$. This mismatch crucially means that the RNNG's estimate of $t_\phi(x_i|\mathbf{x}_{<i})$ may fail to take into account the right context $\mathbf{x}_{>i}$ that is accessible to the BERT student (§3). Hence, we propose an approach where the BERT student distills the RNNG's marginal distribution over words in context, $t_\phi(x_i|\mathbf{x}_{<i}, \mathbf{x}_{>i})$. We develop an efficient yet effective approximation for this quantity, since exact inference is expensive owing to the RNNG's left-to-right parameterization.

Our structure-distilled BERT model differs from the standard BERT model only in its pretraining objective, and thus retains the scalability afforded by Transformer architectures and specialized hardwares like TPUs. In fact, our approach maintains compatibility with standard BERT pipelines; the structure-distilled BERT models can simply be loaded as pretrained BERT weights, which can then be fine-tuned in the exact same fashion.

We hypothesize that the stronger syntactic biases from our new pretraining procedure are useful for a variety of natural language understanding (NLU) tasks that involve *structured output spaces*—including tasks like semantic role labeling (SRL) and coreference resolution that are not explicitly syntactic in nature. We thus evaluate our models on six diverse structured prediction tasks, including phrase-structure parsing (in-domain and out-of-domain), dependency parsing, SRL, coreference resolution, and a combinatory categorial grammar (CCG) supertagging probe, in addition to the GLUE benchmark (Wang et al., 2019). On the structured prediction tasks, our structure-distilled $\text{BERT}_{\text{BASE}}$ reduces relative error by 2% to 21%. These gains are more pronounced in the low-resource scenario, suggesting that stronger syntactic biases help improve sample efficiency (§4).

Despite the gains on the structured prediction tasks, we achieve mixed results on GLUE: Our approach yields improvements on the corpus of linguistic acceptability (Warstadt et al., 2018, CoLA), but performs slightly worse on the rest of GLUE. These findings allude to a partial dissociation between model performance on GLUE, and on structured prediction benchmarks of NLU.

Altogether, our findings: (i) showcase the benefits of syntactic biases, even for representation learners that leverage large amounts of data, (ii) help better understand where syntactic biases are most helpful, and (iii) make a case for designing approaches that not only work well at scale, but also integrate stronger notions of syntactic biases.

2 Recurrent Neural Network Grammars

Here we briefly describe the RNNG (Dyer et al., 2016) that we use as the teacher model. An RNNG is a syntactic LM that defines the joint probability of surface strings $\mathbf{x}$ and phrase-structure nonterminals $\mathbf{y}$, henceforth denoted as $t_\phi(\mathbf{x}, \mathbf{y})$, through a series of structure-building actions that traverse the tree in a top–down, left-to-right fashion. Let N and Σ denote the set of phrase-structure non-terminals and word terminals, respectively. At each time step, the decision over the next action $a_t \in \{\text{NT}(n), \text{GEN}(w), \text{REDUCE}\}$, where $n \in N$ and $w \in \Sigma$, is parameterized by a stack LSTM (Dyer et al., 2015) that encodes partial constituents. The choice of a_t yields these transitions:

- $a_t \in \{\text{NT}(n), \text{GEN}(w)\}$ would push the corresponding non-terminal or word embeddings —$\mathbf{e}_n$ or $\mathbf{e}_w$—onto the stack;
- $a_t = \text{REDUCE}$ would pop the top k elements up to the last incomplete non-terminal, **compose** these elements with a separate bidirectional LSTM, and lastly push the composite phrase embedding $\mathbf{e}_{\text{phrase}}$ back onto the stack. The hierarchical inductive bias of RNNGs can be attributed to this *composition function*,[1] which recursively combines smaller units into larger ones.

RNNGs attempt to maximize the probability of correct action sequences relative to each gold tree.[2]

[1] Not all syntactic LMs have hierarchical biases; Choe and Charniak (2016) modeled strings and phrase structures *sequentially* with LSTMs. This model can be understood as a special case of RNNGs without the composition function.

[2] Unsupervised RNNGs (Kim et al., 2019) exist, although they perform worse on measures of syntactic competence.

Extension to Subwords. Here we extend the RNNG to operate over subword units (Sennrich et al., 2016) to enable compatibility with the BERT student. As each word can be split into an arbitrary-length sequence of subwords, we preprocess the phrase-structure trees to include an additional nonterminal symbol that represents a word sequence, as illustrated by the example ''(S (NP (WORD *the*) (WORD *d ##og*)) (VP (WORD *ba ##rk ##s*)))'', where tokens prefixed by ''##'' are subword units.[3]

3 Approach

We begin with a brief review of the BERT objective, before outlining our structure distillation approach.

3.1 BERT Pretraining Objective

The aim of BERT pretraining is to find model parameters $\hat{\theta}_{\text{B}}$ that would maximize the probability of reconstructing parts of $\mathbf{x} = x_1, \cdots, x_k$ conditional on a corrupted version $c(\mathbf{x}) = c(x_1), \cdots, c(x_k)$, where $c(\cdot)$ denotes the stochastic corruption protocol of Devlin et al. (2019) that is applied to each word $x_i \in \mathbf{x}$. Formally:

$$\hat{\theta}_{\text{B}} = \arg\min_{\theta} \sum_{i \in M(\mathbf{x})} -\log p_\theta(x_i | c(x_1), \cdots, c(x_k)), \tag{1}$$

where $M(\mathbf{x}) \subseteq \{1, \cdots, k\}$ denotes the indices of *masked tokens* that serve as reconstruction targets.[4] This masked LM objective is then combined with a next-sentence prediction loss that predicts whether the two segments in $\mathbf{x}$ are contiguous sequences.

3.2 Motivation

Because the RNNG teacher is an expert on syntactic generalizations (Kuncoro et al., 2018; Futrell et al., 2019; Wilcox et al., 2019), we adopt a structure distillation procedure (Kuncoro et al., 2019) that enables the BERT student to learn from the RNNG's syntactically informative predictions. Our setup nevertheless means that the two models here crucially differ in nature: The BERT student is *not* a left-to-right LM like the RNNG, but rather a denoising autoencoder that models a collection of conditionals for words in **bidirectional** context (Eq. 1).

Figure 1: An example of the masked LM task, where [MASK] = *chase*, and *window* is an attractor (red). We suppress phrase-structure annotations and corruptions on the context tokens for clarity.

We now present two strategies for dealing with this challenge. The first, naïve approach is to *ignore* this difference, and let the BERT student distill the RNNG's marginal next-word distribution for each $w \in \Sigma$ based on the left context alone, that is $t_\phi(w|\mathbf{x}_{<i})$. Although this approach is surprisingly effective (§4.3), we illustrate an issue in Figure 1 for ''*The dogs by the window* [MASK=*chase*] *the cat*''.

The RNNG's strong syntactic biases mean that we can expect $t_\phi(w|\textit{The dogs by the window})$ to assign high probabilities to plural verbs like *bark*, *chase*, *fight*, and *run* that are consistent with the agreement controller *dogs*—despite the presence of a singular attractor (Linzen et al., 2016), *window*, that can distract the model into predicting singular verbs like *chases*. Nevertheless, some plural verbs that are favored based on the left context alone, such as *bark* and *run*, are in fact poor alternatives when considering the right context (e.g., ''*The dogs by the window bark/run the cat*'' are syntactically illicit). Distilling $t_\phi(w|\mathbf{x}_{<i})$ thus fails to take into account the right context $\mathbf{x}_{>i}$ that is accessible to the BERT student, and runs the risk of encouraging the student to assign high probabilities for words that fit poorly with the bidirectional context.

Hence, our second approach is to learn from teacher distributions that not only: (i) reflect the strong syntactic biases of the RNNG teacher, but also (ii) consider both the left and right context when predicting $w \in \Sigma$. Formally, we propose to distill the RNNG's marginal distribution over words in bidirectional context, $t_\phi(w|\mathbf{x}_{<i}, \mathbf{x}_{>i})$, henceforth referred to as the **posterior** probability for generating w under all available information. We now demonstrate that this quantity can, in fact, be computed from left-to-right LMs like RNNGs.

[3] An alternative here is to represent each phrase as a flat sequence of subwords, although our preliminary experiments indicate that this approach yields worse perplexity.

[4] In practice, the corruption protocol $c(\cdot)$ and the reconstruction targets $M(\mathbf{x})$ are intertwined; $M(\mathbf{x})$ denotes the indices of tokens in $\mathbf{x}$ ($\sim 15\%$) that were altered by $c(\mathbf{x})$.

3.3 Posterior Inference

Given a *pretrained* autoregressive, left-to-right LM that factorizes $t_\phi(\mathbf{x}) = \prod_{i=1}^{k} t_\phi(x_i|\mathbf{x}_{<i})$, we discuss how to infer an estimate of $t_\phi(x_i|\mathbf{x}_{<i}, \mathbf{x}_{>i})$. By definition of conditional probabilities:[5]

$$t_\phi(x_i|\mathbf{x}_{<i}, \mathbf{x}_{>i}) = \frac{t_\phi(\mathbf{x}_{<i}, x_i, \mathbf{x}_{>i})}{\sum_{w\in\Sigma} t_\phi(\mathbf{x}_{<i}, \tilde{x}_i = w, \mathbf{x}_{>i})},$$
$$= \frac{t_\phi(\mathbf{x}_{<i})\, t_\phi(x_i|\mathbf{x}_{<i})\, t_\phi(\mathbf{x}_{>i}|x_i, \mathbf{x}_{<i})}{t_\phi(\mathbf{x}_{<i}) \sum_{w\in\Sigma} t_\phi(w|\mathbf{x}_{<i})\, t_\phi(\mathbf{x}_{>i}|\tilde{x_i} = w, \mathbf{x}_{<i})}, \quad (2)$$
$$= \frac{t_\phi(x_i|\mathbf{x}_{<i}) \prod_{j=i+1}^{k} t_\phi(x_j|\mathbf{x}_{<j})}{\sum_{w\in\Sigma} t_\phi(w|\mathbf{x}_{<i}) \prod_{j=i+1}^{k} t_\phi(x_j|\tilde{\mathbf{x}}_{<j}(w,i))},$$

where $\tilde{\mathbf{x}}_{<j}(w,i) = [\mathbf{x}_{<i}; w; \mathbf{x}_{i+1:j-1}]$ is an alternate left context where x_i is replaced by $w \in \Sigma$.

Intuition. After cancelling common factors $t_\phi(\mathbf{x}_{<i})$, the posterior computation in Eq. 2 is decomposed into two terms: (i) the likelihood of producing x_i given its prefix—$t_\phi(x_i|\mathbf{x}_{<i})$, and (ii) conditional on the fact that we have generated x_i and its prefix $\mathbf{x}_{<i}$, the likelihood of producing the observed continuations $\mathbf{x}_{>i}$—$t_\phi(\mathbf{x}_{>i}|x_i, \mathbf{x}_{<i})$. In our running example (Figure 1), the posterior would assign low probabilities to plural verbs like *bark* that are nevertheless probable under the left context alone (i.e., t_ϕ(*bark* | *The dogs by the window*) would be probable), because they are unlikely to generate the continuations $\mathbf{x}_{>i}$ (i.e., we expect t_ϕ(*the cat* | *The dogs by the window bark*) to be low because it is syntactically illicit). In contrast, the posterior would assign high probabilities to plural verbs like *fight* and *chase* that are consistent with the bidirectional context, because we expect both t_ϕ(*fight* | *The dogs by the window*) and t_ϕ(*the cat* | *The dogs by the window fight*) to be probable.

Computational Cost. Let k denote the maximum length of $\mathbf{x}$. Our KD approach requires computing the posterior distribution (Eq. 2) for every masked token x_i in the dataset D, which (excluding marginalization cost over $\mathbf{y}$) necessitates $O(|\Sigma| * k * |D|)$ operations, where each operation returns the RNNG's estimate of $t_\phi(x_j|\mathbf{x}_{<j})$. In the standard BERT setup,[6] this procedure leads to a prohibitive number of operations ($\sim 5 * 10^{+16}$).

3.4 Posterior Approximation

Because exact inference of the posterior is computationally expensive, here we propose an efficient approximation procedure. Approximating $t_\phi(\mathbf{x}_{>i}|x_i, \mathbf{x}_{<i}) \approx t_\phi(\mathbf{x}_{>i}|x_i)$ in Eq. 2 yields:[7]

$$t_\phi(x_i|\mathbf{x}_{<i}, \mathbf{x}_{>i}) \approx \frac{t_\phi(x_i|\mathbf{x}_{<i})\, t_\phi(\mathbf{x}_{>i}|x_i)}{\sum_{w\in\Sigma} t_\phi(w|\mathbf{x}_{<i})\, t_\phi(\mathbf{x}_{>i}|w)}. \quad (3)$$

Although Eq. 3 is still expensive to compute, it enables us to apply the Bayes rule to compute $t_\phi(\mathbf{x}_{>i}|x_i)$:

$$t_\phi(\mathbf{x}_{>i}|x_i) = \frac{t_\phi(x_i|\mathbf{x}_{>i})\, t_\phi(\mathbf{x}_{>i})}{q(x_i)}, \quad (4)$$

where $q(\cdot)$ denotes the unigram distribution. For efficiency, we replace $t_\phi(x_i|\mathbf{x}_{>i})$ through a separately trained ''reverse'', **right-to-left** RNNG, denoted as $r_\omega(x_i|\mathbf{x}_{>i})$. We now apply Eq. 4 and the right-to-left parameterization $r_\omega(x_i|\mathbf{x}_{>i})$ into Eq. 3, and cancel common factors $t_\phi(\mathbf{x}_{>i})$:

$$t_\phi(x_i|\mathbf{x}_{<i}, \mathbf{x}_{>i}) \approx \frac{\frac{t_\phi(x_i|\mathbf{x}_{<i})\, r_\omega(x_i|\mathbf{x}_{>i})}{q(x_i)}}{\sum_{w\in\Sigma} \frac{t_\phi(w|\mathbf{x}_{<i})\, r_\omega(w|\mathbf{x}_{>i})}{q(w)}}. \quad (5)$$

Our approximation in Eq. 5 crucially reduces the required number of operations from $O(|\Sigma| * k * |D|)$ to $O(|\Sigma| * |D|)$, although the actual speedup is much more substantial in practice, since Eq. 5 involves easily batched operations that considerably benefit from specialized hardwares like GPUs.

Notably, our proposed approach here is a general one; it can approximate the posterior

[5] In this setup, we assume that $\mathbf{x}$ is a fixed-length sequence. We aim to infer the LM's estimate for generating a *single* token x_i, relative to all potential single tokens $w \in \Sigma$ (denominator in Eq. 2), conditional on the bidirectional context.

[6] In our BERT pretraining setup, $|\Sigma| \approx 29,000$ (vocabulary size of BERT-cased), $|D| \approx 3 * 10^9$, and $k = 512$.

[7] This approximation preserves the intuition explained in §3.3. Concretely, verbs like *bark* would also be assigned low probabilities under this posterior approximation, since t_ϕ(*the cat* | *bark*) would be low since it is syntactically illicit—the alternative ''*bark **at** the cat*'' would be syntactically licit.

over x_i from *any* left-to-right LM, which can be used as a learning signal for BERT through KD, irrespective of the LM's parameterization. It does, however, necessitate a separately trained right-to-left LM.

Connection to a Product of Experts. Eq. 5 has a similar form to a product of experts (PoE; Hinton, 2002) between the left-to-right and right-to-left RNNGs' next-word distributions, albeit with extra unigram terms $q(w)$. If we replace the unigram distribution with a uniform one, namely, $q(w) = 1/|\Sigma|\ \forall w \in \Sigma$, Eq. 5 reduces to a standard PoE.

Approximating the Marginal. The approximation in Eq. 5 requires estimates of $t_\phi(x_i|\mathbf{x}_{<i})$ and $r_\omega(x_i|\mathbf{x}_{>i})$ from the left-to-right and right-to-left RNNGs, respectively, which necessitate expensive marginalizations over all possible tree prefixes $\mathbf{y}_{<i}$ and $\mathbf{y}_{>i}$. Following Kuncoro et al. (2019), we approximate this marginalization using a one-best predicted tree $\hat{\mathbf{y}}(\mathbf{x}) = \text{argmax}_{\mathbf{y}\in Y(\mathbf{x})}$ $s_\psi(\mathbf{y}|\mathbf{x})$, where $s_\psi(\mathbf{y}|\mathbf{x})$ is parameterized by the transition-based parser of Fried et al. (2019), and $Y(\mathbf{x})$ denotes the set of all possible trees for $\mathbf{x}$. Formally:

$$t_\phi(x_i|\mathbf{x}_{<i}) \approx t_\phi(x_i|\mathbf{x}_{<i}, \hat{\mathbf{y}}_{<i}(\mathbf{x})), \qquad (6)$$

where $\hat{\mathbf{y}}_{<i}(\mathbf{x})$ denotes the non-terminal symbols in $\hat{\mathbf{y}}(\mathbf{x})$ that occur before x_i.[8] The marginal next-word distributions $r_\omega(x_i|\mathbf{x}_{>i})$ from the right-to-left RNNG is approximated similarly.

Preliminary Experiments. Before proceeding with the KD experiments, we assess the quality and feasibility of our approximation through preliminary LM experiments on the Penn Treebank (PTB; Marcus et al., 1993). We find that our approximation is much faster than exact inference by a factor of more than 50,000, at the expense of a slightly worse average posterior negative log-likelihood (2.68 rather than 2.5 for exact inference). More details are provided in Appendix A.

[8]Our approximation of $t_\phi(x_i|\mathbf{x}_{<i})$ relies on a tree prefix $\hat{\mathbf{y}}_{<i}(\mathbf{x})$ from a separate discriminative parser, which has access to yet unseen words $\mathbf{x}_{>i}$. This non-incremental procedure is justified, however, because we aim to design the most informative teacher distributions for the non-incremental BERT student, which also has access to bidirectional context.

Model	KL Div. with Posterior Approx.
Left-to-right LM	2.27±1.84
Right-to-left LM	2.04±1.87
Product of Experts	1.12±1.08

Table 1: Preliminary experiments reporting the *mean*±*stdev.* of the KL divergence (in nats) between the proposed posterior approximation (Eq. 5) and: (i) the left-to-right LM, (ii) the right-to-left LM, and (iii) a simple product of experts baseline (Eq. 5, but with the uniform distribution for $q(w)$).

Differences Between the Models. We now empirically validate our motivating intuition in Figure 1: A model that takes into account the bidirectional context (as is the case for our proposed posterior approximation in Eq. 5) *should* make different predictions compared with the unidirectional left-to-right and right-to-left models.[9] To ascertain whether this is truly the case, we compute the mean Kullback-Leibler (KL) divergence between the distributions from the proposed posterior approximation (Eq. 5) and the distributions from: (i) the left-to-right model, (ii) the right-to-left model, and (iii) a simple product of experts baseline (i.e., Eq. 5, but where $q(w)$ is the uniform distribution). The findings in Table 1 suggest that our proposed posterior approximation approach indeed yields quantifiably different distributions from the left-to-right and right-to-left baselines. To a lesser extent, it also differs from a simple product of experts baseline that similarly incorporates *both* the left-to-right and right-to-left models' predictions, albeit with the uniform distribution for $q(w)$.

3.5 Objective Function

In our structure distillation pretraining, we aim to find BERT parameters $\hat{\theta}_{\text{KD}}$ that emulate our approximation of $t_\phi(w|\mathbf{x}_{<i}, \mathbf{x}_{>i})$ through a word-level cross-entropy loss (Hinton et al., 2015; Kim and Rush, 2016; Furlanello et al., 2018, *inter alia*):

$$\hat{\theta}_{\text{KD}} = \arg\min_\theta \frac{1}{|D|} \sum_{\mathbf{x}\in D} \ell_{\text{KD}}(\mathbf{x};\theta), \text{ where}$$

$$\ell_{\text{KD}}(\mathbf{x};\theta) = -\sum_{i\in M(\mathbf{x})} \sum_{w\in\Sigma} \Big[\tilde{t}_{\phi,\omega}(w|\mathbf{x}_{<i}, \mathbf{x}_{>i}) \log p_\theta\left(\tilde{x}_i = w|c(x_1), \cdots, c(x_k)\right)\Big],$$

[9]We use the same setup as **Preliminary Experiments**.

where $\tilde{t}_{\phi,\omega}(w|\mathbf{x}_{<i}, \mathbf{x}_{>i})$ is our approximation of $t_\phi(w|\mathbf{x}_{<i}, \mathbf{x}_{>i})$, as defined in Eqs. 5 and 6.

Interpolation. The RNNG teacher is an expert on syntax, although in practice it is only feasible to train it on a much smaller dataset. Hence, we not only want the BERT student to learn from the RNNG's syntactic expertise, but also from the rich common-sense and semantics knowledge contained in large text corpora by virtue of predicting the true identity of the masked token x_i,[10] as done in the standard BERT setup. We thus interpolate the KD loss and the original BERT masked LM objective:

$$\hat{\theta}_{\text{B-KD}} = \arg\min_{\theta} \frac{1}{|D|} \sum_{\mathbf{x} \in D} \Big[\alpha \ell_{\text{KD}}(\mathbf{x}; \theta) + (1 - \alpha) \sum_{i \in M(\mathbf{x})} -\log p_\theta(x_i | c(x_1), \cdots, c(x_k)) \Big], \tag{7}$$

omitting the next-sentence prediction for brevity. We henceforth set $\alpha = 0.5$ unless stated otherwise.

4 Experiments

Here we outline the evaluation setup, present our results, and discuss the implications of our findings.

4.1 Evaluation Tasks and Setup

We conjecture that the improved syntactic competence from our approach would benefit a broad range of tasks that involve structured output spaces, including tasks that are not explicitly syntactic. We thus evaluate our structure-distilled BERTs on six diverse structured prediction tasks that encompass syntactic, semantic, and coreference resolution tasks, in addition to the GLUE benchmark that is largely composed of classification tasks.

Phrase-structure Parsing - PTB. We first evaluate our model on phrase-structure parsing on the WSJ section of the PTB. Following prior work, we use sections 02–21 for training, section 22 for validation, and section 23 for testing. We apply our approach on top of the BERT-augmented in-order (Liu and Zhang, 2017) transition-based parser of Fried et al. (2019), which approaches the current state of the art. Because the RNNG teacher that we distill into BERT also uses phrase-structure trees, this setup is related to self-training (Yarowsky, 1995; Charniak, 1997; Zhou and Li, 2005; McClosky et al., 2006; Andor et al., 2016, *inter alia*).

Phrase-structure Parsing - OOD. Still in the context of phrase-structure parsing, we evaluate how well our approach generalizes to three out-of-domain (OOD) treebanks: Brown (Francis and Kučera, 1979), Genia (Tateisi et al., 2005), and the English Web Treebank (Petrov and McDonald, 2012). Following Fried et al. (2019), we test the PTB-trained parser on the test splits[11] of these OOD treebanks *without* any retraining, to simulate the case where no in-domain labeled data are available. We use the same codebase as above.

Dependency Parsing - PTB. Our third task is PTB dependency parsing with Stanford Dependencies (De Marneffe and Manning, 2008) v3.3.0. We use the BERT-augmented joint phrase-structure and dependency parser of Zhou and Zhao (2019), which is inspired by head-driven phrase-structure grammar (HPSG; Pollard and Sag, 1994).

Semantic Role Labeling. Our fourth evaluation task is span-based (SRL) on the English CoNLL 2012 (OntoNotes) dataset (Pradhan et al., 2013). We apply our approach on top of the BERT-augmented model of Shi and Lin (2019), as implemented on AllenNLP (Gardner et al., 2018).

Coreference Resolution. Our fifth evaluation task is coreference resolution, also on the English OntoNotes dataset (Pradhan et al., 2012). For this task, we use the BERT-augmented model of Joshi et al. (2019), which extends the higher-order coarse-to-fine model of Lee et al. (2018).

CCG Supertagging Probe. All proposed tasks thus far necessitate either fine-tuning the entire BERT model, or training a task-specific model on top of the BERT embeddings. Hence, it remains unclear how much of the gains are due to better structural representations from our new *pretraining* strategy, rather than the available supervision at the *fine-tuning* stage. To better understand the gains from our approach, we evaluate on CCG (Steedman, 2000) supertagging

[10] The KD loss $\ell_{\text{KD}}(\mathbf{x}; \theta)$ is defined independently of x_i.

[11] We use the Brown test split of Gildea (2001), the Genia test split of McClosky et al. (2008), and the EWT test split from SANCL 2012 (Petrov and McDonald, 2012).

(Bangalore and Joshi, 1999; Clark and Curran, 2007) through a **classifier probe** (Shi et al., 2016; Adi et al., 2017; Belinkov et al., 2017, *inter alia*), where no BERT fine-tuning takes place.[12]

CCG supertagging is a compelling probing task because it necessitates an understanding of bidirectional context information; the per-word classification setup also lends itself well to classifier probes. Nevertheless, it remains unclear how much of the accuracy can be attributed to the information encoded in the representation, as opposed to the classifier probe itself. We thus adopt the **control task** protocol of Hewitt and Liang (2019) that assigns each word type to a random control category,[13] which assesses the memorisation capacity of the classifier. In addition to the probing accuracy, we report the probe *selectivity*,[14] where higher selectivity denotes probes that faithfully rely on the linguistic knowledge encoded in the representation. We use linear classifiers to maintain high selectivities.

Commonality. All our structured prediction experiments are conducted on top of publicly available repositories of BERT-augmented models, with the exception of the CCG supertagging task that we evaluate as a probe. This setup means that obtaining our results is as simple as changing the pretrained BERT weights to our structure-distilled BERT, and applying the exact same steps as for fine-tuning the baseline model. The fine-tuning hyperparameters are summarized in Appendix C.

GLUE. Beyond the six structured prediction tasks above, we evaluate our approach on the classification[15] tasks of the GLUE benchmark except the Winograd NLI (Levesque et al., 2012) for consistency with the BERT paper (Devlin et al., 2019). The BERT GLUE fine-tuning hyperparameters are based on the fine-tuning configurations of Joshi et al. (2020); we summarize these in Appendix C.

[12] A similar CCG probe was explored by Liu et al. (2019a); we obtain comparable results for the no distillation baseline.

[13] Following Hewitt and Liang (2019), the cardinality of this control category is the same as the number of supertags.

[14] A probe's selectivity is defined as the difference between the probing task accuracy and the control task accurary.

[15] This setup excludes the semantic textual similarity benchmark (STS-B), which is formulated as a regression task.

4.2 Experimental Setup and Baselines

Here we describe the key aspects of our empirical setup, and outline the baselines for assessing the efficacy of our approach.

RNNG Teacher. We implement the subword-augmented RNNG teachers (§2) on DyNet (Neubig et al., 2017a), and obtain ''silver-grade'' phrase-structure annotations for the entire BERT training set using the transition-based parser of Fried et al. (2019). These trees are used to train the RNNG (§2), and to approximate its marginal next-word distribution at inference (Eq. 6). We use the same WordPiece tokenization and vocabulary as BERT-Cased; Appendix B summarizes the complete list of RNNG hyperparameters. Because our approximation (Eq. 5) makes use of a right-to-left RNNG, we train this variant with the same hyperparameters and data as the left-to-right model. We train each directional RNNG teacher on a shared subset of 3.6M sentences ($\sim$3%) from the BERT training set with automatic dynamic batching (Neubig et al., 2017b), which takes three weeks on a V100 GPU.

BERT Student. We first apply our structure distillation pretraining protocol to $\text{BERT}_{\text{BASE}}$-Cased. We use the exact same training dataset, model configuration, WordPiece tokenization, vocabulary, and hyperparameters (Appendix C) as in the standard pretrained BERT model.[16] The sole exception is that we use a larger initial learning rate of $3e^{-4}$ based on preliminary experiments,[17] which we apply to all models (including the no distillation/standard BERT baseline) for fair comparison.

Baselines and Comparisons. We compare the following set of models in our experiments:

- A standard $\text{BERT}_{\text{BASE}}$-Cased without any structure distillation loss, which benefits from scalability but lacks syntactic biases (**''No-KD''**);
- Four variants of structure-distilled BERTs that: (i) only distill the left-to-right RNNG (**''L2R-KD''**), (ii) only distill the right-to-left RNNG (**''R2L-KD''**), (iii) distill the RNNG's approximated marginal for

[16] https://github.com/google-research/bert.

[17] We find this larger learning to perform better on most of our evaluation tasks. Liu et al. (2019b) have similarly found that tuning BERT's initial learning rate leads to better results.

Task		Validation Set						Test Set		
		Baselines		Structure-distilled BERTs				No-KD	Best-KD	Err. Red.
		No-KD	Seq-KD	L2R-KD	R2L-KD	UF-KD	UG-KD			
Parsing	Const. PTB - F1	95.38	95.33	95.55	95.55	95.58	**95.59**	95.35	**95.70**	7.6%
	Const. PTB - EM	55.33	55.41	55.92	56.18	56.39	**56.59**	55.25	**57.77**	5.63%
	Const. OOD - F1†	86.76	86.54	87.43	**87.53**	87.23	87.40	89.04	**89.76**	6.55%
	Dep. PTB - UAS	96.48	96.40	**96.70**	96.64	96.60	96.66	96.79	**96.86**	2.18%
	Dep. PTB - LAS	94.65	94.56	**94.90**	94.80	94.79	94.83	95.13	**95.23**	1.99%
	SRL - OntoNotes	86.17	86.09	86.34	86.29	86.30	**86.46**	86.08	**86.39**	2.23%
	Coref. - OntoNotes	72.53	69.27	73.74	73.49	**73.79**	73.33	72.71	**73.69**	3.58%
	CCG supertag. probe	93.69	91.59	93.97	**95.21**	95.13	**95.21**	93.88	**95.2**	21.57%
	Probe selectivity	24.79	23.77	23.3	23.57	27.28	**28.3**	23.15	**26.07**	N/A

Table 2: Validation and test results for the structured prediction tasks; each entry reflects the mean of three random seeds. To preserve test set integrity, we only obtain test set results for the no distillation baseline and the best structure-distilled BERT on the validation set; **''Err. Red.''** reports the test error reductions *relative* to the **No-KD** baseline. We report F1 and exact match (EM) for PTB phrase-structure parsing; for dependency, we report unlabeled (UAS) and labeled (LAS) attachment scores. The ''Const. OOD'' (†) row indicates the mean F1 from three out-of-domain corpora: Brown, Genia, and the English Web Treebank (EWT), although the validation results exclude the Brown Treebank that has no validation set.

generating x_i under the bidirectional context, where $q(w)$ (Eq. 5) is the *uniform* distribution (**''UF-KD''**), and lastly (iv) a similar variant as (iii), but where $q(w)$ is the *unigram* distribution (**''UG-KD''**). All these BERT models crucially benefit from the syntactic biases of RNNGs, although only variants (iii) and (iv) learn from teacher distributions that consider *bidirectional context* for predicting x_i; and

- A $\text{BERT}_{\text{BASE}}$ model that distills the approximate posterior for generating x_i under the bidirectional context, but from *sequential* LSTM teachers (**''Seq-KD''**) in place of RNNGs.[18] This baseline crucially isolates the importance of learning from hierarchical teachers, because it utilizes the exact same approximation technique and KD loss as the structure-distilled BERTs.

Learning Curves. Given enough labeled data, BERT can acquire the relevant structural information from the fine-tuning (as opposed to pretraining) procedure, although better pretrained representations can nevertheless facilitate *sample-efficient* generalizations (Yogatama et al., 2019). We thus additionally examine the models' fine-tuning learning curves, as a function of varying amounts of training data, on phrase-structure parsing and SRL.

Random Seeds. Because fine-tuning the same pretrained BERT with different random seeds can lead to varying results, we report the mean performance from three random seeds on the structured prediction tasks, and from five random seeds on GLUE.

Test Results. To preserve the integrity of the test sets, we first report all performance on the validation sets, and only report the test set results for: (i) the **No-KD** baseline, and (ii) the best structure-distilled model on the validation set (**''Best-KD''**).

4.3 Findings and Discussion

We report the validation and test set results for the structured prediction tasks in Table 2. The validation set learning curves for phrase-structure parsing and SRL that compare the **No-KD** baseline with the **UG-KD** variant are provided in Figure 2.

General Discussion. We summarize several key observations from Table 2 and Figure 2.

- All four structure-distilled BERT models consistently outperform the **No-KD** baseline, including the **L2R-KD** and **R2L-KD** variants

[18] For fair comparison, we train the LSTM on the exact same subset as the RNNG, with comparable number of model parameters. An alternative here is to use Transformers, although we elect to use LSTMs to facilitate fair comparison with RNNGs, which are also based on LSTM architectures.

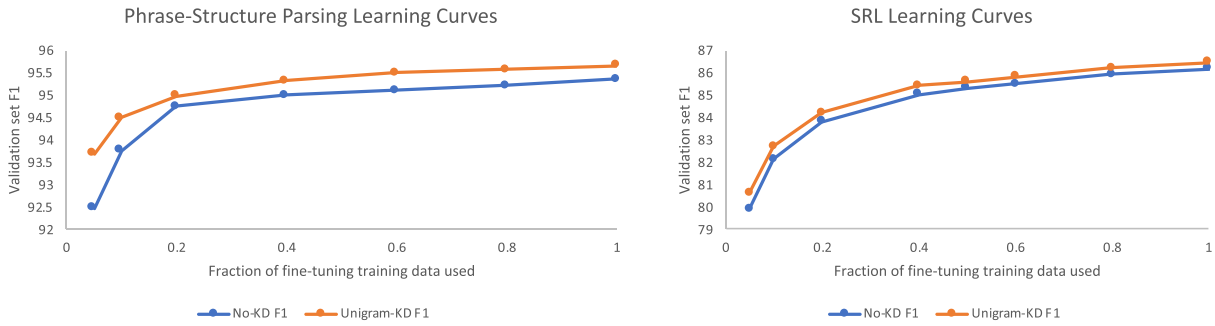

Figure 2: The fine-tuning learning curves that examine how the number of fine-tuning instances (from 5% to 100% of the full training sets) affect validation set F1 scores in the case of phrase-structure parsing and SRL. We compare the **No-KD**/standard $\text{BERT}_{\text{BASE}}$-Cased and the **UG-KD** structure-distilled BERT.

that only distill the syntactic knowledge of unidirectional RNNGs. Remarkably, this pattern holds true for *all six* structured prediction tasks. In contrast, we observe no such gains for the **Seq-KD** baseline, which largely performs worse than the **No-KD** model. We conclude that the gains afforded by our structure-distilled BERTs can be attributed to the **syntactic biases** of the RNNG teacher.

- We conjecture that the surprisingly strong performance of the **L2R-KD** and **R2L-KD** models, which distill the knowledge of *unidirectional* RNNGs, can be attributed to the interpolated objective in Eq. 7 ($\alpha = 0.5$). This interpolation means that the target distribution assigns a probability mass of at least 0.5 to the true masked word x_i, which is guaranteed to be consistent with the bidirectional context. However, the syntactic knowledge contained in the unidirectional RNNGs' predictions can still provide a structurally informative learning signal, via the rest of the probability mass, for the BERT student.

- Although all structure-distilled variants outperform the baseline, models that distill our approximation of the RNNG's distribution for words in bidirectional context (**UF-KD** and **UG-KD**) yield the best results on four out of six tasks (PTB phrase-structure parsing, SRL, coreference resolution, and the CCG supertagging probe). This finding confirms the efficacy of our approach.

- We observe the largest gains for the syntactic tasks, particularly for phrase-structure parsing and CCG supertagging. However, the improvements are not at all confined to purely syntactic tasks: we reduce relative error from strong BERT baselines by 2.2% and 3.6% on SRL and coreference resolution, respectively. While the RNNG's syntactic biases are derived from phrase-structure grammar, the strong improvement on CCG supertagging, in addition to the smaller improvement on dependency parsing, suggests that the RNNG's syntactic biases generalize well across different syntactic formalisms.

- We observe larger improvements in a low-resource scenario, where the model is exposed to fewer fine-tuning instances (Figure 2), suggesting that syntactic biases are helpful for enabling more **sample-efficient** generalizations. This pattern holds for both tasks that we investigated: phrase-structure parsing (syntactic in nature) and SRL (not explicitly syntactic in nature). With only 5% of the fine-tuning data, the **UG-KD** model improves F1 score from 79.9 to **80.6** for SRL (a 3.5% error reduction relative to the **No-KD** baseline, as opposed to 2.2% on the full data). For phrase-structure parsing, the **UG-KD** model achieves a remarkable **93.68** F1 (a 16% relative error reduction, as opposed to 7.6% on the full data) with only 5% of the PTB—this performance is notably better than past state of the art phrase-structure parsers trained on the *full* PTB c. 2017 (Kuncoro et al., 2017).

GLUE Results and Discussion. We report the GLUE validation and test results for $\text{BERT}_{\text{BASE}}$-Cased in Table 3. Because we observe a different pattern of results on the Corpus of Linguistic Acceptability (CoLA; Warstadt et al., 2018) than

	No-KD	UG-KD
Validation Set (Per-task average / 1-best random seed)		
CoLA	50.7 / 60.2	**54.3 / 60.6**
7-task avg. (excl. CoLA)	**85.4 / 87.8**	84.8 / 86.9
Overall 8-task avg.	**81.1 / 84.4**	81.0 / 83.6
Test set (Per-task 1-best random seed on validation set)		
CoLA	53.1	**55.3**
7-task avg. (excl. CoLA)	**84.2**	83.5
Overall 8-task avg.	**80.3**	80.0

Table 3: Summary of the validation and test set results on GLUE. The validation results are derived from the average of five random seeds for each task, which accounts for variance, and the 1-best random seed, which does not. The test results are derived from the 1-best random seed on the validation set.

on the rest of GLUE, we henceforth report: (i) the CoLA results, (ii) the seven task average that excludes CoLA, and (iii) the average across all eight tasks. We select the **UG-KD** model because it achieved the best validation set eight task average among the structure-distilled BERTs; the per-task GLUE breakdown is provided in Appendix D.

The results on GLUE provide an interesting contrast to the consistent improvements we observed on the structured prediction tasks. More concretely, our **UG-KD** model outperforms the baseline on CoLA, but performs slightly worse on the other GLUE tasks in aggregate, leading to a slightly lower overall test set accuracy (80.0 for the **UG-KD** as opposed to 80.3 for the **No-KD** baseline).

The improvement on the syntax-sensitive CoLA provides additional evidence—beyond the improvement on the syntactic tasks (Table 2)—that our approach indeed yields improved syntactic competence. We conjecture that these improvements do not transfer to the other GLUE tasks because they rely more on lexical and semantic properties, and less on syntactic competence (McCoy et al., 2019).

We defer a more thorough investigation of how much syntactic competence is necessary for solving most of the GLUE tasks to future work, but make two remarks. First, the findings on GLUE are consistent with the hypothesis that our approach yields improved structural competence, albeit at the expense of a slightly less rich meaning representation, which we attribute to the smaller dataset used to train the RNNG teacher. Second, human-level natural language understanding includes the ability to predict structured outputs, for example, to decipher ''*who did what to whom*'' (SRL). Succeeding in these tasks necessitates inference about structured output spaces, which (unlike most of GLUE) cannot be reduced to a single classification decision. Our findings indicate a partial dissociation between model performance on these two types of tasks; hence, supplementing GLUE evaluation with some of these structured prediction tasks can offer a more holistic assessment of progress in NLU.

CCG Probe Example. The CCG supertagging probe is a particularly interesting test bed, because it clearly assesses the model's ability to use contextual information in making its predictions—*without* introducing additional confounds from the BERT fine-tuning procedure. We thus provide a representative example of four different BERT variants' predictions on the CCG supertagging probe in Table 4, based on which we discuss two observations. First, the different models make different predictions, where the **No-KD** and **L2R-KD** models produce (coincidentally the same) incorrect predictions, while the **R2L-KD** and **UG-KD** models are able to predict the correct supertag. This finding suggests that different teacher models are able to impose different biases on the BERT students.[19]

Second, the mistakes of the **No-KD** and **L2R-KD** BERTs belong to the broader category of challenging argument-adjunct distinctions (Palmer et al., 2005; Fowlie, 2017). Here both models fail to subcategorize for the prepositional phrase (PP) ''*as screens*'', which serves as an argument of the verb ''*use*'', as opposed to the noun phrase ''*TV sets*''. Distinguishing between these two potential dependencies naturally requires syntactic information from the right context; hence the **R2L-KD** BERT, which is trained to emulate the predictions of an RNNG teacher that observes the right context, is able to make the correct prediction. This advantage is crucially retained by the **UG-KD** model that distills the RNNG's approximate distribution over words in bidirectional context (Eq. 5), and further confirms the efficacy of our proposed approach.

[19] All four BERTs have access to the full *bidirectional* context at test time, although some are trained to mimic the predictions of *unidirectional* RNNGs (**L2R-KD** and **R2L-KD**).

Sentence Input	No-KD & L2R-KD Pred.	R2L-KD & UG-KD Pred.
*‘‘Apple II owners , for example , had to **use** their TV sets as screens and stored data on audiocassettes’’*	(S[b]\NP)/NP	((S[b]\NP)/PP)/NP

Table 4: An example of the CCG supertag predictions for the verb ‘‘***use***’’ from four different BERT variants. The correct answer is ‘‘((S[b]\NP)/PP)/NP’’, which both the **R2L-KD** and **UG-KD** predict correctly (blue). However, the **No-KD** baseline and the **L2R-KD** model produce (the same) incorrect predictions (red); both models fail to subcategorize for the prepositional phrase ‘‘*as screens*’’ as a dependent of the verb ‘‘***use***’’. Beyond this, all four models predict the correct supertags for all other words (not shown).

Measuring the Models' Differences. Beyond the qualitative example in Table 4, we further quantify the extent to which the different BERT models produce different predictions. To this end, we compute *pairwise model agreement* for the phrase-structure parsing task, as measured by exact match accuracy. We present the full experimental setup and findings in Appendix E, but summarize two key findings here.

First, the highest exact match agreement between any pair of different models is fairly low at 44.92%, further supporting our conjecture that different teacher models indeed impose different biases on the BERT student, as evidenced by the different model predictions. Second, all four structure-distilled BERT variants have the lowest pairwise agreement score with the **No-KD** baseline ($< 39\%$ pairwise model agreement), suggesting that all variants of our structure distillation objectives yield quantifiably different outputs compared to the no distillation alternative, which does not learn from the syntactic knowledge of RNNGs.

$\text{BERT}_{\text{LARGE}}$ Results. Having evaluated our structure-distilled $\text{BERT}_{\text{BASE}}$-Cased, we now apply our approach on top of $\text{BERT}_{\text{LARGE}}$-Cased, and present the results on the structured prediction tasks in Table 5. Overall, we observe a similar pattern of results with $\text{BERT}_{\text{LARGE}}$ as we do with $\text{BERT}_{\text{BASE}}$: On the structured prediction tasks, our best structure distillation approach reduces error by 1.5% to 5.5% relative to the **No-KD** baseline. Furthermore, our structure-distilled $\text{BERT}_{\text{LARGE}}$ models establish new state of the art single model results—among models pretrained on the original BERT training set[20]—on phrase-structure parsing (PTB and OOD), PTB dependency parsing, and SRL.

[20]This comparison excludes other models like XLNet and RoBERTa, which are trained on more data.

	Task	Test Set - $\text{BERT}_{\text{LARGE}}$-Cased			
		No-KD	Best-KD	Error Red.	BERT SoTA
Parsing	Const. PTB – F1	95.80	**95.95**	3.73%	95.84†
	Const. PTB – EM	56.87	**57.74**	2.02%	–
	Const. OOD – F1	89.63	**90.20**	5.48%	89.91‡
	Dep. PTB – UAS	96.91	**97.03**	3.78%	97.0†
	Dep. PTB – LAS	95.33	**95.49**	3.43%	95.43†
	SRL – OntoNotes	87.59	**87.77**	1.45%	86.5◇
	Coref. – OntoNotes	74.03	**74.69**	2.55%	79.6♦

Table 5: Test set results for the structured prediction tasks with $\text{BERT}_{\text{LARGE}}$-Cased; each entry reflects the mean of three random seeds. We compare the no distillation baseline (**‘‘No-KD’’**) with the best structure-distilled model, as selected on the validation set (**‘‘Best-KD’’**); **‘‘Error Red.’’** reports the test error reductions *relative* to the **No-KD** baseline. We also report the previous state of the art among non-ensemble models pretrained on the original BERT training set (‘‘**BERT SoTA**’’).[21]

4.4 Limitations

We outline two limitations to our approach. First, we assume the existence of decent-quality ‘‘silver-grade’’ phrase-structure trees to train the RNNG teacher. Although this assumption holds true for English because of the existence of accurate phrase-structure parsers, this is not necessarily the case for other languages. Second, pretraining the BERT student in our naïve implementation is about half as fast on TPUs compared with the baseline due to I/O bottleneck. This overhead only applies at pretraining, and can be reduced through parallelization.

5 Related Work

Earlier work has proposed a few ways for introducing notions of hierarchical structures into

[21]†Zhou and Zhao (2019), ‡Fried et al. (2019), ◇Shi and Lin (2019), and ♦Joshi et al. (2020).

BERT, for instance, through designing structurally motivated auxiliary losses (Wang et al., 2020), or including syntactic information in the embedding layers that serve as inputs for the Transformer (Sundararaman et al., 2019). In contrast, we use a different technique for injecting syntactic biases, which is based on the structure distillation technique of Kuncoro et al. (2019), although our work features two key differences. First, Kuncoro et al. (2019) put a sole emphasis on cases where both the teacher and student models are autoregressive, left-to-right LMs; here we extend this objective for when the student model is a representation learner that has access to bidirectional context. Second, Kuncoro et al. (2019) only evaluated their structure-distilled LMs in terms of perplexity and grammatical judgment (Marvin and Linzen, 2018). In contrast, we evaluate our structure-distilled BERT models on six diverse structured prediction tasks and the GLUE benchmark. It remains an open question whether, and how much, syntactic biases are helpful for a broader range of NLU tasks beyond grammatical judgment; our work represents a step towards answering this question.

Substantial progress has recently been made in improving the performance of BERT and other masked LMs (Lan et al., 2020; Liu et al., 2019b; Raffel et al., 2019; Sun et al., 2020, *inter alia*). Our structure distillation technique is orthogonal, and can be applied for these approaches. Lastly, our findings on the benefits of syntactic knowledge for structured prediction tasks that are not explicitly syntactic in nature, such as SRL and coreference resolution, are consistent with those of prior work (He et al., 2017; Swayamdipta et al., 2018; He et al., 2018; Strubell et al., 2018, *inter alia*).

6 Conclusion

Given the remarkable success of textual representation learners trained on large amounts of data, it remains an open question whether syntactic biases are still relevant for these models that work well at scale. Here we present evidence to the affirmative: our structure-distilled BERT models outperform the baseline on a diverse set of six structured prediction tasks. We achieve this through a new pretraining strategy that enables the BERT student to learn from the predictions of an explicitly hierarchical, but much less scalable, RNNG teacher model. Because the BERT student is a bidirectional model that estimates the conditional probabilities of masked words in context, we propose to distill an efficient yet surprisingly effective approximation of the RNNG's posterior estimate for generating each word conditional on its bidirectional context.

Our findings suggest that syntactic inductive biases are beneficial for a diverse range of structured prediction tasks, including for tasks that are not explicitly syntactic in nature. In addition, these biases are particularly helpful for improving fine-tuning sample efficiency on these tasks. Lastly, our findings motivate the broader question of how we can design models that integrate stronger notions of structural biases—and yet can be easily scalable at the same time—as a promising (if relatively underexplored) direction of future research.

Acknowledgments

We would like to thank Mandar Joshi, Zhaofeng Wu, Rui Zhang, Timothy Dozat, and Kenton Lee for answering questions regarding the evaluation of the model. We also thank Sebastian Ruder, John Hale, Kris Cao, Stephen Clark, and the three anonymous reviewers for their helpful suggestions. A. K. is supported by an EPSRC Doctoral Training Partnership studentship and a Balliol Mark Sadler scholarship; D. F. is supported by a Google PhD Fellowship.

References

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*.

Daniel Andor, Chris Alberti, David Weiss, Aliaksei Severyn, Alessandro Presta, Kuzman Ganchev, Slav Petrov, and Michael Collins. 2016. Globally normalized transition-based neural networks. In *Proceedings of ACL*. **DOI:** https://doi.org/10.18653/v1/P16-1231

Srinivas Bangalore and Aravind K. Joshi. 1999. Supertagging: An approach to almost parsing. *Computational Linguistics*.

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017. What do

neural machine translation models learn about morphology? In *Proceedings of ACL*. **DOI:** https://doi.org/10.18653/v1/P17-1080

Cristian Bucilă, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *Proceedings of KDD*. **DOI:** https://doi.org/10.1145/1150402.1150464

Eugene Charniak. 1997. Statistical parsing with a context-free grammar and word statistics. In *Proceedings of AAAI*.

Do Kook Choe and Eugene Charniak. 2016. Parsing as language modeling. In *Proceedings of EMNLP*. **DOI:** https://doi.org/10.18653/v1/D16-1257

Stephen Clark and James R. Curran. 2007. Wide-coverage efficient statistical parsing with CCG and log-linear models. *Computational Linguistics*. **DOI:** https://doi.org/10.1162/coli.2007.33.4.493

Marie-Catherine De Marneffe and Christopher D. Manning. 2008. Stanford typed dependencies manual. **DOI:** https://doi.org/10.3115/1608858.1608859

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*.

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. 2015. Transition-based dependency parsing with stack long short-term memory. In *Proceedings of ACL*. **DOI:** https://doi.org/10.3115/v1/P15-1033

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A. Smith. 2016. Recurrent neural network grammars. In *Proceedings of NAACL*. **DOI:** https://doi.org/10.18653/v1/N16-1024, **PMID:** 26993434

Meaghan Fowlie. 2017. *Slaying the Great Green Dragon: Learning and Modelling Iterable Ordered Optional Adjuncts.* Ph.D. thesis, UCLA.

Winthrop Nelson Francis and Henry Kučera. 1979. *Manual of information to accompany a standard corpus of present-day edited American English, for use with digital computers*. Brown University, Department of Linguistics.

Daniel Fried, Nikita Kitaev, and Dan Klein. 2019. Cross-domain generalization of neural constituency parsers. In *Proceedings of ACL*. **DOI:** https://doi.org/10.18653/v1/P19-1031

Tommaso Furlanello, Zachary Chase Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. 2018. Born-again neural networks. In *Proceedings of ICML*.

Richard Futrell, Ethan Wilcox, Takashi Morita, Peng Qian, Miguel Ballesteros, and Roger Levy. 2019. Neural language models as psycholinguistic subjects: Representations of syntactic state. In *Proceedings of NAACL*. **DOI:** https://doi.org/10.18653/v1/N19-1004

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew E. Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. *CoRR*, abs/1803.07640. **DOI:** https://doi.org10.18653/v1/W18-2501

Daniel Gildea. 2001. Corpus variation and parser performance. In *Proceedings of EMNLP*.

Yoav Goldberg. 2019. Assessing BERT's syntactic abilities. *CoRR*, abs/1901.05287.

Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. 2017. Deep semantic role labeling: What works and what's next. In *Proceedings of ACL*. **DOI:** https://doi.org/10.18653/v1/P17-1044, **PMCID:** PMC5961228

Shexia He, Zuchao Li, Hai Zhao, and Hongxiao Bai. 2018. Syntax for semantic role labeling, to be, or not to be. In *Proceedings of ACL*. **DOI:** https://doi.org/10.18653/v1/P18-1192, **PMCID:** PMC6010685

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. In *Proceedings of EMNLP*. **DOI:** https://doi.org/10.18653/v1/D19-1275

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

Geoffrey E Hinton. 2002. Training products of experts by minimizing contrastive divergence.

Neural Computation. **DOI:** https://doi.org/10.1162/089976602760128018, **PMID:** 12180402

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531.

Jennifer Hu, Jon Gauthier, Peng Qian, Ethan Wilcox, and Roger P. Levy. 2020. A systematic assessment of syntactic generalization in neural language models. In *Proceedings of ACL*.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of ACL*. **DOI:** https://doi.org/10.18653/v1/P19-1356

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *TACL*. **DOI:** https://doi.org/10.1162/tacl_a_00300

Mandar Joshi, Omer Levy, Luke Zettlemoyer, and Daniel Weld. 2019. BERT for coreference resolution: Baselines and analysis. In *Proceedings of EMNLP*.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of EMNLP*. **DOI:** https://doi.org/10.18653/v1/D16-1139

Yoon Kim, Alexander M. Rush, Lei Yu, Adhiguna Kuncoro, Chris Dyer, and Gabor Melis. 2019. Unsupervised recurrent neural network grammars. In *Proceedings of NAACL*. **DOI:** https://doi.org/10.18653/v1/N19-1114

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of EMNLP System Demonstrations*. **DOI:** https://doi.org/10.18653/v1/D18-2012, **PMID:** 29382465

Adhiguna Kuncoro, Miguel Ballesteros, Lingpeng Kong, Chris Dyer, Graham Neubig, and Noah A. Smith. 2017. What do recurrent neural network grammars learn about syntax? In *Proceedings of EACL*. **DOI:** https://doi.org/10.18653/v1/E17-1117

Adhiguna Kuncoro, Chris Dyer, John Hale, Dani Yogatama, Stephen Clark, and Phil Blunsom. 2018. LSTMs can learn syntax-sensitive dependencies well, but modeling structure makes them better. In *Proceedings of ACL*. **DOI:** https://doi.org/10.18653/v1/P18-1132

Adhiguna Kuncoro, Chris Dyer, Laura Rimell, Stephen Clark, and Phil Blunsom. 2019. Scalable syntax-aware language modelling with knowledge distillation. In *Proceedings of ACL*. **DOI:** https://doi.org/10.18653/v1/P19-1337

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *Proceedings of ICLR*.

Kenton Lee, Luheng He, and Luke Zettlemoyer. 2018. Higher-order coreference resolution with coarse-to-fine inference. In *Proceedings of NAACL*.

Hector J. Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Proceedings of KR*.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *TACL*. **DOI:** https://doi.org/10.1162/tacl_a_00115

Jiangming Liu and Yue Zhang. 2017. In-order transition-based constituent parsing. *TACL*.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. 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. *CoRR*, abs/1907.11692.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19:313–330.

DOI: https://doi.org/10.21236/ADA273556

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. In *Proceedings of EMNLP*. **DOI:** https://doi.org/10.18653/v1/D18-1151

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Effective self-training for parsing. In *Proceedings of NAACL*. **DOI:** https://doi.org/10.3115/1220835.1220855

David McClosky, Eugene Charniak, and Mark Johnson. 2008. When is self-training effective for parsing? In *Proceedings of COLING*. **DOI:** https://doi.org/10.3115/1599081.1599152

Tom 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

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Proceedings of Interspeech*. **DOI:** https://doi.org/10.1109/ICASSP.2011.5947611

Graham Neubig, Chris Dyer, Yoav Goldberg, Austin Matthews, Waleed Ammar, Antonios Anastasopoulos, Miguel Ballesteros, David Chiang, Daniel Clothiaux, Trevor Cohn, Kevin Duh, Manaal Faruqui, Cynthia Gan, Dan Garrette, Yangfeng Ji, Lingpeng Kong, Adhiguna Kuncoro, Gaurav Kumar, Chaitanya Malaviya, Paul Michel, Yusuke Oda, Matthew Richardson, Naomi Saphra, Swabha Swayamdipta, and Pengcheng Yin. 2017a. DyNet: The Dynamic Neural Network Toolkit. *arXiv preprint arXiv:1701.03980*.

Graham Neubig, Yoav Goldberg, and Chris Dyer. 2017b. On-the-fly operation batching in dynamic computation graphs. In *Proceedings of NeurIPS*.

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

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

Slav Petrov and Ryan McDonald. 2012. Overview of the 2012 shared task on parsing the web. In *Notes of the First Workshop on Syntactic Analysis of Non-Canonical Language (SANCL)*.

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

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using OntoNotes. In *Proceedings of CoNLL*.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Proceedings of CoNLL*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of ACL*. **DOI:** https://doi.org/10.18653.18653/v1/P16-1162

Peng Shi and Jimmy Lin. 2019. Simple BERT models for relation extraction and semantic role labeling. *CoRR*, abs/1904.05255.

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

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*.

Mark Steedman. 2000. *The Syntactic Process*, MIT Press. **DOI:** https://doi.org/10.7551/mitpress/6591.001.0001

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

Yu Sun, Shuohuan Wang, Yu-Kun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2020. ERNIE 2.0: A continual pre-training framework for language understanding. In *Proceedings of AAAI*. **DOI:** https://doi.org/10.1609/aaai.v34i05.6428

Dhanasekar Sundararaman, Vivek Subramanian, Guoyin Wang, Shijing Si, Dinghan Shen, Dong Wang, and Lawrence Carin. 2019. Syntax-infused transformer and bert models for machine translation and natural language understanding. *arXiv preprint arXiv:1911.06156*.

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

Yuka Tateisi, Akane Yakushiji, Tomoko Ohta, and Jun'ichi Tsujii. 2005. Syntax annotation for the GENIA corpus. In *Proceedings of IJCNLP*.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019a. BERT rediscovers the classical NLP pipeline. In *Proceedings of ACL*. **DOI:** https://doi.org/10.18653/v1/v1/P19-1452

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of NeurIPS*.

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 *Proceedings of ICLR*. **DOI:** https://doi.org/10.18653/v1/W18-5446

Wei Wang, Bin Bi, Ming Yan, Chen Wu, Zuyi Bao, Liwei Peng, and Luo Si. 2020. StructBERT: Incorporating language structures into pre-training for deep language understanding. In *Proceedings of ICLR*.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2018. Neural network acceptability judgments. *arXiv preprint arXiv:1805.12471*.

Ethan Wilcox, Peng Qian, Richard Futrell, Miguel Ballesteros, and Roger Levy. 2019. Structural supervision improves learning of non-local grammatical dependencies. In *Proceedings of NAACL*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In *Proceedings of NeurIPS*.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of ACL*. **DOI:** https://doi.org/10.3115/981658.981684

Dani Yogatama, Cyprien de Masson d'Autume, Jerome Connor, Tomás Kociský, Mike Chrzanowski, Lingpeng Kong, Angeliki Lazaridou, Wang Ling, Lei Yu, Chris Dyer, and Phil Blunsom. 2019. Learning and evaluating general linguistic intelligence. *CoRR*, abs/1901.11373.

Junru Zhou and Hai Zhao. 2019. Head-driven phrase structure grammar parsing on Penn treebank. In *Proceedings of ACL*. **DOI:** https://doi.org/10.18653/v1/P19-1230, **PMCID:** PMC6593428

Zhi-Hua Zhou and Ming Li. 2005. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Transactions on Knowledge and Data Engineering*. **DOI:** https://doi.org/10.1109/TKDE.2005.186

A Preliminary Experiments

Here we discuss the preliminary experiments to assess the quality and computational efficiency of our posterior approximation procedure (§3.4). Recall that this approximation procedure only applies at *inference*; the LM is still *trained* in a typical autoregressive, left-to-right fashion.

Model. Because exactly computing the RNNG's next-word distributions $t_\phi(x_i|\mathbf{x}_{<i})$ involves an intractable marginalization over all possible tree prefixes $\mathbf{y}_{<i}$, we run our experiments in the context of sequential LSTM language models, where $t_{\text{LSTM}}(x_i|\mathbf{x}_{<i})$ can be computed exactly. This setup crucially enables us to isolate the impact of approximating the posterior distribution over x_i under the bidirectional context (Eq. 2) with our proposed approximation (Eq. 5), without introducing further confounds stemming from the RNNG's marginal approximation procedure (Eq. 6).

Dataset and preprocessing. We train the LSTM LM on an *open-vocabulary* version of the PTB,[22] in order to simulate the main experimental setup where both the RNNG teacher and BERT student are also open-vocabulary by virtue of byte-pair encoding (BPE) preprocessing. To this end, we preprocess the dataset with SentencePiece (Kudo and Richardson, 2018) BPE tokenization, where $|\Sigma| = 8,000$; we preserve all case information. We follow the empirical setup of the parsing (§4.1) experiments, with Sections 02–21 for training, Section 22 for validation, and Section 23 for testing.

Model Hyperparameters. We train the LM with 2 LSTM layers, 250 hidden units per layer, and a dropout (Srivastava et al., 2014) rate of 0.2. Model parameters are optimized with stochastic gradient descent (SGD), with an initial learning rate of 0.25 that is decayed exponentially by a factor of 0.92 for every epoch after the tenth. Because our approximation relies on a separately trained right-to-left LM (Eq. 5), we train this variant with the exact same hyperparameters and dataset split as the left-to-right model.

[22] Our open-vocabulary setup means that our results are not directly comparable to prior work on PTB language modeling (Mikolov et al., 2010, *inter alia*), which mostly utilize a special ''UNK'' token for infrequent or unknown words.

Model	Posterior NLL	Posterior Ppl.
MoE	3.28	26.58
Uniform Approx.	3.18	24.17
Unigram Approx.	**2.68**	**14.68**
Exact Inference	2.50	12.25

Table 6: The findings from the preliminary experiments that assess the quality of our posterior approximation procedure. We compare three variants against exact inference (bottom row; Eq. 2) as computed from the left-to-right model.

Evaluation and Baselines. We evaluate the models in terms of the average posterior negative log likelihood (NLL) and perplexity.[23] Because exact inference of the posterior is expensive, we evaluate the model only on the first 400 sentences of the test set. We compare the following variants:

- A mixture of experts baseline that simply mixes ($\alpha = 0.5$) the probabilities from the left-to-right and right-to-left LMs in an *additive* fashion, as opposed to *multiplicative* as in the case of our PoE-like approximation in Eq. 5 (**''MoE''**);
- Our approximation of the posterior over x_i (Eq. 5), where $q(w)$ is the *uniform* distribution (**''Uniform Approx.''**);
- Our approximation of the posterior over x_i (Eq. 5), but where $q(w)$ is the *unigram* distribution (**''Unigram Approx.''**); and
- Exact inference of the posterior as computed from the left-to-right model, as defined in Eq. 2 (**''Exact Inference''**).

Discussion. We summarize the findings in Table 6, based on which we remark on two observations. First, the posterior NLL of our approximation procedure that makes use of the unigram distribution (**Unigram Approx.**; third row) is not much worse than that of exact inference, in exchange for a more than 50,000 times speedup[24]

[23] In practice, this perplexity is derived from simply exponentiating the average posterior negative log likelihood.

[24] All three approximations in Table 6 have similar runtimes.

in computation time. Nevertheless, using the uniform distribution (second row) on $q(w)$ in place of the unigram one (Eq. 5) results in a much worse posterior NLL. Second, combining the left-to-right and right-to-left LMs using a mixture of experts—a baseline which is not well-motivated by our theoretical analysis—yields the worst result.

B RNNG Hyperparameters

To train the subword-augmented RNNG teacher (§2), we use the following hyperparameters that achieve the best validation perplexity from a grid search: 2-layer stack LSTMs (Dyer et al., 2015) with 512 hidden units per layer, optimized by standard SGD with an initial learning rate of 0.5 that is decayed exponentially by a factor of 0.9 for every epoch after the tenth. We use dropout with $p = 0.3$.

C BERT Hyperparameters

Here we outline the hyperparameters of the BERT student in terms of pretraining data creation, masked LM pretraining, and fine-tuning.

Pretraining Data Creation. We use the same codebase[25] and pretraining data as Devlin et al. (2019), which are derived from a mixture of Wikipedia and Books text corpora. To train our structure-distilled BERTs, we sample a masking from these corpora following the same hyperparameters used to train the original $\text{BERT}_{\text{BASE}}$-Cased model: a maximum sequence length of 512, a per-word masking probability of 0.15 (up to a maximum of 76 masked tokens in a 512-length sequence), and a dupe factor of 10. We apply a random seed of 12345. We preprocess the raw dataset using NLTK tokenizers, and then apply the same (BPE-augmented) vocabulary and WordPiece tokenization as in the original BERT model. All other hyperparameters are set to the same values as in the publicly released original BERT model.

Masked LM Pretraining. We train all model variants (including the no distillation/standard BERT baseline for fair comparison) using a batch size of 256 sequences. We use an initial Adam learning rate of $3e^{-4}$ for the $\text{BERT}_{\text{BASE}}$ models (as opposed to $1e^{-4}$ in the original BERT model) and $1e^{-4}$ for the $\text{BERT}_{\text{LARGE}}$ models. Following Devlin et al. (2019), we pretrain our models for 1M steps. All other hyperparameters are similarly set to their default values.

GLUE Fine-tuning. For each GLUE task, we fine-tune the BERT model by running a grid search over learning rates of $\{5e^{-6}, 1e^{-5}, 2e^{-5}, 3e^{-5}, 5e^{-5}\}$ and batch sizes of $\{16, 32\}$, with 5 random seeds. Following Joshi et al. (2020), we train each fine-tuning configuration for 10 epochs, except for CoLA, where we train for 4 epochs.

Structured Prediction Fine-tuning. For the structured prediction tasks, we use the following hyperparameters for learning rate and batch size. These hyperparameters are either the default for a given codebase, or lightly tuned on the No-KD models. We use the same hyperparameters across all models (No-KD and KD) of a given size (BASE or LARGE) on a given task.

- In-order phrase-structure parser: a BERT learning rate of $2e^{-5}$, a batch size of 32, and a warmup period of 160 updates.
- HPSG dependency parser: a BERT learning rate of $5e^{-5}$, a batch size of 150, and a warmup period of 160 updates.
- Coreference resolution: for the $\text{BERT}_{\text{BASE}}$ models: a learning rate of $1e^{-5}$ and a maximum segment length of 128 word pieces. For the $\text{BERT}_{\text{LARGE}}$ models: a learning rate of $5e^{-6}$ and a maximum segment length of 512 word pieces. Both sizes use a learning rate warmup period of 2 epochs and a batch size of 1 document.
- Semantic role labeling: for the $\text{BERT}_{\text{BASE}}$ models: a learning rate of $5e^{-5}$. For the $\text{BERT}_{\text{LARGE}}$ models: a learning rate of $1e^{-5}$. Both sizes use a batch size of 32.

D Full GLUE Results

We present the full GLUE results for the **No-KD** baseline and the **UG-KD** BERT in Table 7.

E Quantifying Model Differences

We quantify the extent to which learning from different teacher models results in BERT models that make different predictions. To this end, we

[25] https://github.com/google-research/bert.

	Model	CoLA	SST-2	MRPC	QQP	MNLI (M/MM)	QNLI	RTE	GLUE Avg
Dev	No-KD	60.2	92.2	90.0	89.4	90.3/90.9	90.7	71.1	84.4
	UG-KD	60.6	92.0	88.9	89.3	89.6/90.0	89.9	68.6	83.6
Test	No-KD	53.1	92.5	88.0	88.8	82.8/81.8	89.9	65.4	80.3
	UG-KD	55.3	91.2	87.6	88.7	81.9/80.8	89.5	65.0	80.0

Table 7: Summary of the full results on GLUE, comparing the **No-KD** baseline with the **UG-KD** structure-distilled BERT (§4.2). All results are based on a single random seed: we select the 1-best fine-tuning hyperparameters (including random seed) on the validation set, which we then evaluate on the test set.

Pairwise Exact Match Agreement	No-KD	L2R-KD	R2L-KD	UF-KD	UG-KD
No-KD	–	36.20	36.56	38.01	37.05
L2R-KD	36.20	–	42.25	43.58	44.43
R2L-KD	36.56	42.25	–	39.95	41.53
UF-KD	38.01	43.58	39.95	–	44.92
UG-KD	37.05	44.43	41.53	44.92	–

Table 8: Pairwise model agreement scores for phrase-structure parsing, as measured by average exact match on trees from the validation set of the PTB for which some pair of models produced different trees. Self-agreement (diagonals) are 100%. Exact match is symmetric; hence the table is also symmetric.

compute *pairwise model agreement*,[26] in terms of phrase-structure parsing exact match, between each pair of five model variants (**No-KD**, **L2R-KD**, **R2L-KD**, **UF-KD**, and **UG-KD**). We compute this pairwise model agreement score on the PTB dev set (§22). To better understand the *differences* between the models, we exclude sentences where all five models produce the exact same phrase-structure trees, leaving 826 out of 1700 sentences; further analysis indicates that the excluded sentences tend to be shorter and less ambiguous.

We present the findings in Table 8, and summarize two key observations. First, the highest exact match agreement between any pair of models is fairly low at 44.92%. This finding supports our conjecture that different teacher models indeed impose different biases for the BERT students, as evidenced by the different model predictions. Second, each RNNG-distilled BERT model has the lowest agreement rate with the **No-KD** baseline. This finding suggests that *all* variants of our structure distillation approach produce quantifiably different predictions (<39% pairwise model agreement) from the **No-KD**/standard BERT baseline that does not learn from the syntactic knowledge of RNNGs.

[26]For instance, when comparing the agreement between the **No-KD** and the **UG-KD** models, we treat the **UG-KD** model's output as ''gold reference'', and compute the exact match from the **No-KD** model's output with respect to that.