Dependency-based Mixture Language Models

Zhixian Yang and **Xiaojun Wan**
Wangxuan Institute of Computer Technology, Peking University
Center for Data Science, Peking University
The MOE Key Laboratory of Computational Linguistics, Peking University
yangzhixian@stu.pku.edu.cn
wanxiaojun@pku.edu.cn

Abstract

Various models have been proposed to incorporate knowledge of syntactic structures into neural language models. However, previous works have relied heavily on elaborate components for a specific language model, usually recurrent neural network (RNN), which makes themselves unwieldy in practice to fit into other neural language models, such as Transformer and GPT-2. In this paper, we introduce the Dependency-based Mixture Language Models. In detail, we first train neural language models with a novel dependency modeling objective to learn the probability distribution of future dependent tokens given context. We then formulate the next-token probability by mixing the previous dependency modeling probability distributions with self-attention. Extensive experiments and human evaluations show that our method can be easily and effectively applied to different neural language models while improving neural text generation on various tasks.[1]

1 Introduction

Syntactic structures serve as the principle of how words are correctly combined to form sentences. It is widely acknowledged that learning syntactic structures should improve neural text generation (Shen et al., 2018; Peng et al., 2019; Du et al., 2020). Even though current neural language models, such as Transformer (Vaswani et al., 2017) and GPT-2 (Radford et al., 2019) have achieved outstanding performance without explicitly modeling latent syntactic structures, these models still fail to learn the long-range syntactic dependencies (Kuncoro et al., 2018; Xu et al., 2021).

To leverage explicit syntactic knowledge in natural language generation (NLG), many methods have been proposed (Wu et al., 2017; Shen et al., 2018; Zhang et al., 2019; Kim et al., 2019; Du et al., 2020). We conclude from previous works that knowledge of syntactic structures can bring four advantages to neural language models:

(1) Syntactic structures can be modeled to obtain better representations of natural language sentences (Jacob et al., 2018; Williams et al., 2018; Wang et al., 2019).

(2) Jointly training syntactic structure parsing and language modeling can contribute to each other (Shen et al., 2018; Dyer et al., 2016; Kim et al., 2019; Du et al., 2020; Shen et al., 2021b).

(3) Syntactic structures can be used to directly model the composition of language (Socher et al., 2013; Casas et al., 2020) and help with the long-range dependency problem by providing shortcuts for gradient backpropagation (Chung et al., 2017).

(4) Integrating syntactic structures into a neural network can improve generalization via a better inductive bias (Shen et al., 2019; Zhang et al., 2019).

Despite these advantages, it is not trivial to incorporate knowledge of syntactic structures into neural language models effectively and efficiently. Several practical problems arise:

(1) Previous works (Chung et al., 2017; Shen et al., 2018; Dyer et al., 2016; Kim et al., 2019; Shen et al., 2019) have relied heavily on elaborate components for a specific language model, usually recurrent neural network (RNN) (Sutskever et al., 2014). These methods are difficult to be adapted to other neural language models, such as Transformer and GPT-2.

(2) If jointly modeling language modeling and syntactic structure parsing, it will require much more time/memory during training or inference.

To address these problems while keeping the advantages, we explore incorporating knowledge of syntactic structures in a different manner. In this work, we propose a novel dependency modeling objective to train neural language models to directly predict the current token's *future dependent tokens* given the history. We define the *future dependent to-*

[1] Our code is available at https://github.com/FadedCosine/Dependency-Guided-Neural-Text-Generation

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 7758 - 7773
May 22-27, 2022 ©2022 Association for Computational Linguistics

Models	External Parameters?	External Networks?	Architecture Agnostic?
RNNG (Dyer et al., 2016)	Yes	Yes	No
PRPN (Shen et al., 2018)	Yes	Yes	No
URNNG (Kim et al., 2019)	Yes	Yes	No
ON-LSTM (Shen et al., 2019)	Yes	No	No
DMLM (Ours)	No or Negligible	No	Yes

Table 1: The difference between our DMLM and previous neural language models that incorporate knowledge of syntactic structures. Previous models often require external networks and external Parameters. For example, PRPN consists of three neural networks: Parsing Network, Reading Network and Predict Network. ON-LSTM is built upon a single LSTM, but it requires two additional gates in the LSTM cells, which leads to external parameters. All these previous models can only be built upon RNN architecture. However, as an architecture-agnostic method, DMLM needs no external parameters or networks when built upon Transformer, while it only needs negligible external parameters when built upon RNN.

kens of a specific token in a sentence as its children and parent in the dependency parse tree that will appear in the rest of the sentence. Further, we propose Dependency-based Mixture Language Models (DMLM) that, at each timestep, mixes the previous dependency modeling probability distributions with self-attention to get the next-token probability. As shown in Table 1, the proposed method can be adapted to any neural language model without adding external networks or parameters.

Our core idea can be illustrated in Figure 1 and Figure 2: when predicting the next-token "indicate" after reading "red figures on the screen", common language models are easy to predict an incorrect word, such as "indicates", since the prediction of these models relies heavily on the recent word, "screen" in this case. However, our propose DMLM will directly look back into the long-range context, and select the next-token from all the future dependent tokens predicted by previous tokens. According to the underlying dependency structure, DMLM pays different weights to different tokens' future dependent tokens. Thus, the model is more likely to predict "indicate" since DMLM tends to think of the next-token as a future dependent token of "figures" rather than "screen".

We conduct experiments with different neural language models including LSTM (Hochreiter and Schmidhuber, 1997), Transformer (Vaswani et al., 2017), and GPT-2 (Radford et al., 2019) across different tasks in conditional text generation, unconditional text generation, and language modeling. Through extensive experiments we demonstrate that DMLM consistently improves the generation quality according to both human evaluations and automatic metrics. Compared to other neural language models that incorporate syntactic knowledge, DMLM is architecturally simpler and easier to fit into any neural language model, while possessing wide applicability to different text generation tasks.

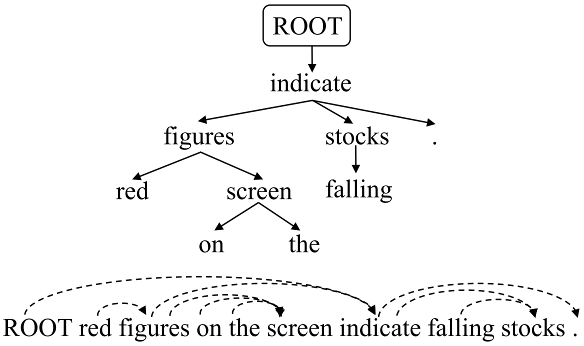

Figure 1: Example of dependency parse tree

2 Methodology

Our goal is to propose a simple yet effective method that can improve neural text generation by learning from the underlying syntactic structure, and can fit into any auto-regressive generation model without using additional elaborate components. We first introduce a novel dependency modeling objective to force the model to directly predict the future dependent tokens of the current token. Based on the dependency modeling, we then present the proposed DMLM.

2.1 Dependency Modeling

It has been a challenge to equip neural language models with the capability of modeling long-range dependency in text (Dai et al., 2019). In particular, previous works (Wu et al., 2017) observe that vanilla RNN can hardly capture many subtle long-range token dependencies effectively. On

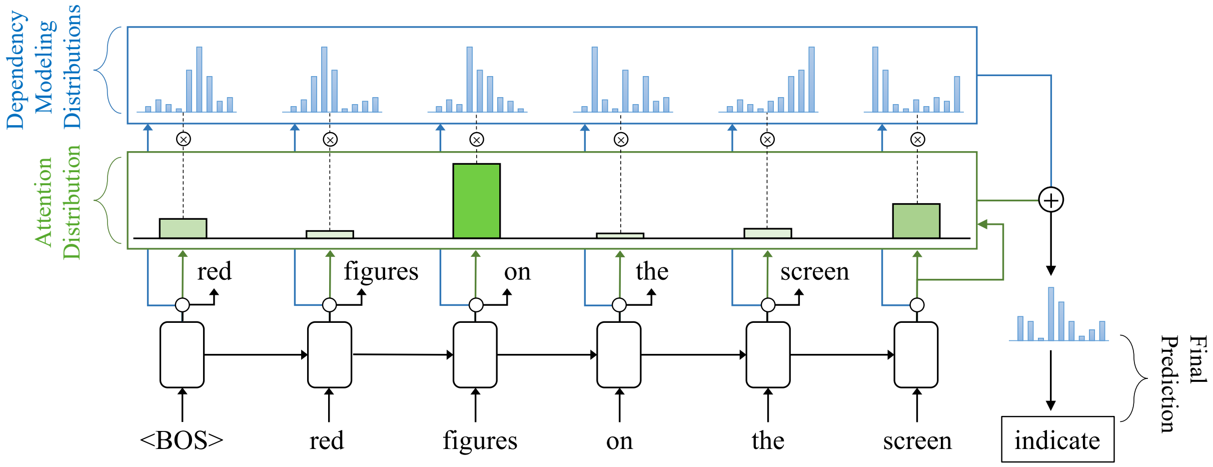

Figure 2: Illustration of DMLM. For each timestep, the language model outputs a dependency modeling distribution, while the self-attention produces a dependency attention distribution over the context. And then, the next-token probability is the sum of the context's dependency modeling probability distributions weighed by the dependency attention scores. Best viewed in color.

the other hand, though self-attention mechanisms can build direct connections between long-distance token pairs, it is still elusive for Transformer to be aware of syntactic dependency structures while also obtaining strong language modeling performance (Shen et al., 2021a).

The current neural language models are mostly trained purely using the language modeling objective with Maximum Likelihood Estimation (MLE). With the auto-regressive factorization, language modeling can be reduced to modeling the conditional distribution of the next-token x_t given the context $\mathbf{x}_{<t} = \{x_1, \ldots, x_{t-2}, x_{t-1}\}$. However, in order to make neural language models aware of long-range dependency and syntactic structures, we propose the dependency modeling objective to train models to learn the probability distribution of the future dependent tokens directly. Following Ahmed et al. (2019), we define the *future dependent tokens* of a specific token in a sentence as its children and parent in the dependency parse tree that will appear in the rest of the sentence. Taking Figure 1 as an example, the future dependent tokens of "figures" are "screen" and "indicate", since "red" does not appear after "figures" in this sentence.

Specifically, given a token sequence $\mathbf{x} = \{x_1, \ldots, x_{T-1}, x_T\}$ where $T \in \mathbb{N}$ denotes the sequence length, we first use dependency parser to generate a dependency tree. Then, we derive the future dependent tokens set Z_t for each token x_{t-1}, where $Z_t = \{x_i \mid i \geq t, x_i$ is the child or parent of $x_{t-1}\}$. We train a language model θ to maximize the log-likelihood sum of tokens in Z_t. This equals to minimize:

$$\mathcal{L}_{\mathrm{DM}}(\theta) = -\sum_{t=1}^{T} \sum_{z_t \in Z_t} \log p_\theta^{\mathrm{dep}}(z_t \mid \mathbf{x}_{<t}), \quad (1)$$

which is the dependency modeling objective.

2.2 Dependency-based Mixture Language Models

To give a categorical probability distribution over the next-token, a standard approach for the current neural language models is to encode the context into a fixed-size vector followed by an output embedding layer and a softmax function.

In our case, given the context $\mathbf{x}_{<t}$, we first train the language model to directly learn the probability distribution of x_{t-1}'s future dependent tokens $p_\theta^{\mathrm{dep}}(w \mid \mathbf{x}_{<t})$ by dependency modeling (Section 2.1). We then propose DMLM (depicted in Figure 2) that mixes dependency modeling probability distributions $P^{\mathrm{dep}} = \{p_\theta^{\mathrm{dep}}(w \mid \mathbf{x}_{<1}), \ldots, p_\theta^{\mathrm{dep}}(w \mid \mathbf{x}_{<t-1}), p_\theta^{\mathrm{dep}}(w \mid \mathbf{x}_{<t})\}$. All the probability distributions in P^{dep} are weighed by self-attention, and summed to obtain the final next-token probability distribution.

We can easily implement a self-attention in both Transformer-based and RNN-based language models. For example, in Transformer and GPT-2, the penultimate layer seems to naturally learn alignments (Garg et al., 2019), so we use its average attention weights over all the attentions heads as the dependency attention distribution. In RNN-based models, inspired by Merity et al. (2017) and

Vaswani et al. (2017), at each timestep, we linearly project the current hidden state $h_t \in \mathbb{R}^H$ to a query vector $q_t = W^Q h_t$ and a key vector $k_t = W^K h_t$, where $W^Q \in \mathbb{R}^{H \times H}$, $W^K \in \mathbb{R}^{H \times H}$, $q_t \in \mathbb{R}^H$, and $k_t \in \mathbb{R}^H$. To generate the dependency attention, we compute the match between the query q_t and the context's keys $\{k_1, \ldots, k_{t-1}, k_t\}$ by taking the inner product, followed by a softmax to obtain the dependency attention distribution:

$$\begin{aligned}
\mathbf{e}^{(\mathrm{t})} &= \{e_1^{(t)}, \ldots, e_{t-1}^{(t)}, e_t^{(t)}\}, \\
e_i^{(t)} &= q_t^T k_i, 1 \leq i \leq t, \\
\mathbf{a}^{(\mathbf{t})} &= \mathrm{softmax}(\frac{\mathbf{e}^{(\mathrm{t})}}{\sqrt{H}}), \\
\mathbf{a}^{(\mathbf{t})} &= \{a_1^{(t)}, \ldots, a_{t-1}^{(t)}, a_t^{(t)}\},
\end{aligned} \tag{2}$$

where $\mathbf{e}^{(\mathrm{t})} \in \mathbb{R}^t$, and $\mathbf{a}^{(\mathbf{t})} \in \mathbb{R}^t$. We scale the dot products by $\frac{1}{\sqrt{H}}$ following Vaswani et al. (2017).

The dependency attention distribution reveals which token in the context may have a strong dependency relation with the token to be predicted. Thus, the neural language model should pay more attention to previous tokens with high dependency attention scores, i.e., the next-token is more likely to be the future dependent token of those tokens in the context. Formally, the next-token probability is the sum of the context's dependency modeling probability distributions weighed by the dependency attention scores:

$$p_\theta\left(w \mid \mathbf{x}_{<t}\right) = \sum_{\tau=1}^{t} a_\tau^{(t)} p_\theta^{\mathrm{dep}}\left(w \mid \mathbf{x}_{<\tau}\right). \tag{3}$$

where $p_\theta^{\mathrm{dep}}\left(w \mid \mathbf{x}_{<\tau}\right)$ is the probability distribution of $x_{\tau-1}$'s future dependent tokens, since till now the neural language model is only trained by dependency modeling. Then, we further finetune the neural language model using MLE, but with respect to our modified probability distribution given in Equation 3:

$$\mathcal{L}_{\mathrm{LM}}(\theta) = -\sum_{t=1}^{T} \log p_\theta\left(x_t \mid \mathbf{x}_{<t}\right). \tag{4}$$

For each timestep during inference, DMLM outputs a dependency modeling distribution, and we store it in a list. To predict the next-token, DMLM applies self-attention in Equation 2 to produce a dependency attention distribution over the context, and then the next-token probability can be calculated by Equation 3, where the list preserves all the $p_\theta^{\mathrm{dep}}\left(w \mid \mathbf{x}_{<\tau}\right), 1 \leq \tau \leq t$.

3 Experiments

Despite previous works mainly focusing on language modeling, it has always been a thorny issue whether better language models lead to better performance in downstream tasks. Therefore, we showcase the performance of our proposed DMLM in three different tasks: conditional text generation (Section 3.1), unconditional text generation (Section 3.2), and language modeling (Section 3.3).

To verify the effectiveness and architecturally generalizability of our method, we conduct the generation tasks with three dominant neural language models, including LSTM, Transformer and GPT-2. We prefix the base model name with "**DM-**" to denote the corresponding Dependency-based Mixture language model. Specifically, we adopt AWD-LSTM (Merity et al., 2018) as our base LSTM, and further compare our DM-LSTM with **PRPN** (Shen et al., 2018) and **ON-LSTM** (Shen et al., 2019) which also incorporate knowledge of syntactic structures, and are built on LSTM. In the same task, we use exactly the same hyper-parameters and setups for the pairs of base models and corresponding DM-models. Other details of the experimental setup for each task can be seen in Appendix A.

For all the tasks, we use a state-of-the-art parser, HPSG Parser[2] (Zhou and Zhao, 2019) to get the dependency parse tree for each sentence in the datasets. We discuss the impact of the dependency parser in Appendix B.

3.1 Conditional Text Generation

Setup We take the story ending generation as the conditional text generation task, and evaluate our method on the ROCStories corpus (Mostafazadeh et al., 2016), which consists of 98,161 five-sentences. We follow the preprocessing[3] of Kong et al. (2021) to randomly split ROCStories by 8:1:1 for training/validation/test, respectively, and delexicalize stories by masking all the male/female/unknown names with "[MALE]"/"[FEMALE]"/"[NEUTRAL]". We finally get a word-level vocabulary with $31,216$ unique tokens. The conditional text generation task is to generate a reasonable ending given a four-sentence story context. For all models, we generate stories using nucleus sampling (Holtzman et al.,

[2] https://github.com/DoodleJZ/HPSG-Neural-Parser

[3] We use the preprocessed data in https://github.com/thu-coai/Stylized-Story-Generation-with-Style-Guided-Planning

Models	UNION ↑	BERTScore ↑	B-1 ↑	B-2 ↑	D2 ↑	D3 ↑	SB-2 ↓	SB-3 ↓
PRPN	83.37	29.11	21.45	6.84	13.22	33.50	95.17	86.76
ON-LSTM	82.18	29.41	22.16	7.33	13.93	35.71	94.98	85.80
AWD-LSTM	82.98	29.57	22.23	7.31	14.07	35.71	94.92	85.88
DM-LSTM	**83.97***	**29.93**	**22.54***	**7.63***	**14.92**	**37.44**	**94.47***	**84.77***
Transformer	81.39	27.64	21.28	7.01	17.48	**42.30**	93.18	81.52
DM-Transformer	**84.07***	**28.20***	**21.49**	**7.29***	**17.79**	42.08	**92.86***	**81.36***
GPT-2	84.41	29.02	21.79	7.45	17.09	40.74	93.51	82.55
DM-GPT-2	**85.31***	**30.18***	**22.81***	**8.02***	**17.98**	**43.29**	**93.18**	**81.41***

Table 2: Automatic evaluation results for the conditional text generation task on Rocstories dataset. * denotes that DM-model significantly outperforms the second best model for t-test (p-value<0.05).

Models	Grammaticality				Logicality			
	Win(%)	Lose(%)	Tie(%)	κ	Win(%)	Lose(%)	Tie(%)	κ
DM-LSTM vs. PRPN	36.2*	14.5	49.3	0.225	56.5*	17.5	26.0	0.306
DM-LSTM vs. ON-LSTM	12.8*	6.4	80.8	0.238	48.4*	24.4	27.2	0.409
DM-LSTM vs. AWD-LSTM	28.0*	14.5	57.5	0.224	43.0*	34.5	22.5	0.214
DM-Transformer vs. Transformer	18.2*	5.2	76.6	0.358	50.6*	18.6	30.8	0.342
DM-GPT-2 vs. GPT-2	20.4*	5.0	74.6	0.374	50.6*	18.8	30.6	0.224

Table 3: Human evaluation results for the conditional text generation task on Rocstories dataset. κ denotes the inter-annotator agreement Krippendorff's alpha (Hayes and Krippendorff, 2007) score. * means statistical significance for Wilcoxon signed-rank test (p-value<0.01). Note that, it is relatively easy for both models to generate a single sentence that is grammatically correct, so the rate of "tie" in Grammaticality is relatively high.

2020) with $p = 0.5$.

We measure the generated story endings by the following automatics metrics: (1) **UNION** (Guan and Huang, 2020): It is a learnable unreferenced metric for evaluating the quality of generated stories; (2) **BERTScore** (Zhang et al., 2020): The metric measures the semantic consistency between the generated and the referenced ones by BERT (Devlin et al., 2019); (3) **BLEU (B-n)** (Papineni et al., 2002): BLEU evaluates n-gram overlap between the generated stories and the references; (4) **Distinct (D-n)** (Li et al., 2016): The proportions of distinct n-grams in the outputs to evaluate the diversity of generated results. Since Distinct score will become extremely low for small n, we calculate it with $n = 2, 3$; (5) **Self-BLEU (SB-n)** (Zhu et al., 2018): The metric is calculated by computing n-grams ($n = 2, 3$) BLEU score of each generated text with all other generated ones as references. Smaller Self-BLEU scores indicate better diversity.

Results The experimental results of baselines and corresponding DM-models are shown in Table 2. Note that we do not conduct significant tests on Distinct since it is a document-level metric. We can see that, all the DM-models significantly outperform baseline models on almost all the metrics. Furthermore, compared with PRPN and ON-LSTM, our DM-LSTM performs significantly better in all the metrics. This indicates that incorporating knowledge of syntactic structures in our proposed way can effectively contribute to both the quality and diversity of the story ending generation. Moreover, no matter what the base model is, our DM-model can substantially improves the conditional text generation. This demonstrates that our method can be effectively adapted to different neural language models, such as the large scale language model, GPT-2, while previous models like ON-LSTM can only be built on LSTM.

Models	LM score ↓	RLM score ↓
PRPN	5.24	5.75
ON-LSTM	5.20	5.59
AWD-LSTM	5.18	5.64
DM-LSTM	**5.14**	**5.52**
Transformer	5.00	5.59
DM-Transformer	**4.97**	**5.49**
GPT-2	4.89	5.55
DM-GPT-2	**4.67**	**5.47**

Table 4: Results of global metrics for the unconditional text generation task on EMNLP2017 WMT News.

Human evaluation To further evaluate the fluency and logic of generated stories, following (Guan et al., 2020), we conduct pair-wise comparisons between DM-models and corresponding

Models	Nucleus-p							
	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
PRPN	41.48	45.77	55.32	64.23	83.98	109.3	172.09	302.57
ON-LSTM	37.46	42.98	**46.16**	56.69	72.36	98.06	152.60	274.43
AWD-LSTM	37.97	41.80	48.74	57.45	71.77	**94.22**	146.40	289.13
DM-LSTM	**36.11**	**39.53***	47.67	**55.30**	**69.38**	95.95	**136.98***	**256.51***
Transformer	45.37	46.36	50.90	60.27	70.74	91.65	125.46	222.27
DM-Transformer	**37.74***	**40.75***	**43.25***	**49.92***	**60.28***	**76.77***	**104.03***	**182.29***
GPT-2	41.19	44.05	47.86	53.97	63.18	81.45	112.81	192.10
DM-GPT-2	**36.41***	**40.99***	**41.75***	**46.18***	**55.36***	**67.97***	**92.22***	**152.98***

Table 5: GPT-2 Perplexity on $1,000$ random samples with various sampling hyper-parameters generated by models trained on EMNLP2017 WMT News dataset. Nucleus sampling is used here with various p. * denotes that DM-model significantly outperforms the second best model for t-test (p-value<0.05).

Models	Human score ↑
PRPN	0.380
ON-LSTM	0.278
AWD-LSTM	0.365
DM-LSTM	**0.444**
Transformer	0.400
DM-Transformer	**0.448**
GPT-2	0.468
DM-GPT-2	**0.512**
Real data	0.688

Table 6: Turing test results of the samples generated by models trained on EMNLP2017 WMT News dataset. To reach a good trade-off between quality and diversity, we adopt nucleus sampling with $p = 0.7$ for all the models to generate samples.

baselines. We randomly sample 100 story endings from each model. For each pair of stories (one by the DM-model and the other by the baseline, along with the beginning), five annotators are hired to give a preference (win, lose, or tie) from the following two aspects: (1) *Grammaticality*: whether a story ending is natural and fluent; (2) *Logicality*: whether a story is coherent to the given beginning and reasonable in terms of causal and temporal dependencies in the context. The detailed questionnaire and other details are shown in Appendix D.

The average win/lose/tie rates of the human evaluation are shown in Table 3. To measure the inter-annotator agreement, we calculate Krippendorff's alpha (Hayes and Krippendorff, 2007) for each pair-wise comparison, and all the results are fair agreement ($0.2 \leq \kappa \leq 0.4$) or moderate agreement ($0.4 \leq \kappa \leq 0.6$). The results show that our DM-models significantly outperform baseline models in both the grammaticality and logicality.

3.2 Unconditional Text Generation

Setup We perform experiments of unconditional text generation on EMNLP2017 WMT News dataset[4]. We use the preprocessed data of a recent work[5] (Caccia et al., 2020) that contains $5,268$ distinct words with maximum sentence length 51. The training/validation/test set consists of $268,586/10,000/10,000$ sentences.

Following Caccia et al. (2020), we evaluate the models with the *global metrics* (Semeniuta et al., 2018): (1) **Language Model score (LM score)**: We use the oracle Language Model to evaluate the negative log-likelihood of generated text as the metric to reflect quality; (2) **Reverse Language Model score (RLM score)** We train a new Language Model on the generated text, and then evaluate the negative log-likelihood of a held-out set of real text. This metric can measure text diversity since the generated text with better diversity would have a broader coverage over the real data space, and the new Language Model can be trained better, thus leading to lower RLM score. Both the LM score and RLM score are usually evaluated on the sentences generated by purely random sampling. Besides, to further measure the generation fluency, we directly use the public GPT-2 checkpoint of pretrained parameters without finetuning to calculate **GPT-2 Perplexity** of generated samples.

Results Table 4 shows the results of global metrics obtained by various models. All the DM-models again outperform the baselines. The consistently lower LM scores indicate that the generated

[4] http://statmt.org/wmt17/translation-task.htm

[5] https://github.com/pclucas14/GansFallingShort/tree/master/real_data_experiments/data/news

Models	#Params	Dev PPL	Test PPL
Pointer Sentinel-LSTM (Merity et al., 2017)	21M	72.4	70.9
RNNG (Dyer et al., 2016)	-	-	88.7
Variational RHN (Zilly et al., 2017)	23M	67.9	65.4
PRPN (Shen et al., 2018)	-	-	62.0
Fraternal dropout (Zolna et al., 2018)	24M	58.9	56.8
URNNG (Kim et al., 2019)	-	-	85.9
ON-LSTM (Shen et al., 2019)	25M	58.3	56.2
AWD-LSTM (Merity et al., 2018)	24M	60.0	57.3
DM-LSTM (Ours)	24M	58.6	56.2
AWD-LSTM-MoS(Yang et al., 2018)	22M	56.5	54.4
AWD-LSTM-DOC(Takase et al., 2018)	23M	54.1	52.4

Table 7: Various language models' perplexity evaluated on validation and test sets of Penn Treebank dataset. Yang et al. (2018) and Takase et al. (2018) focus on improving the softmax of LSTM LM, which are orthogonal to ours.

sentences of DM-models are of better quality, while the consistently lower RLM scores also demonstrate that DM-models can generate more diverse sentences meanwhile.

In addition, each model is used to generate $1,000$ sentences with various sampling hyper-parameters, and GPT-2 Perplexity is further calculated. As shown in Table 5, our proposed method can make neural language models perform significantly better in terms of generation fluency. In particular, Transformer-based models can gain more significant improvement from DMLM. We conjecture that this is because, in our implementation, we directly uses the penultimate multi-head attention layer of Transformer to obtain the dependency attention distribution of DMLM. Thus, it can easily inherit all the strengths of Transformer-based models.

Human evaluation Following previous work (Yu et al., 2017; Guo et al., 2018), we conduct a Turing test to further evaluate the generated text. In practice, we mix 100 randomly sampled sentences from each model, and another 100 sentences from the real test set. Five annotators are hired to judge whether each of the 900 sentences is created by human or machines. Each sentence gets $+1$ score when it is regarded as a real one, and 0 score otherwise. The detailed questionnaire and other details are shown in Appendix D.

The average score for each model is shown in Table 6, from which we can see all the DM-models surpass the baselines. Both automatic evaluations and human evaluations indicate that DMLM can help neural language models generate more readable, fluent, and natural sentences.

3.3 Language Modeling

Setup We evaluate the proposed method with the word-level language modeling task by measuring **Perplexity (PPL)** on the Penn Treebank (PTB) (Marcus et al., 1993; Mikolov et al., 2012) corpora. The PTB dataset has a vocabulary size of $10,000$ unique words, and the training/validation/test set consists of $42,068/3,370/3,761$ sentences.

For this task, we mainly implement the DMLM on the RNN-based language model, i.e., AWD-LSTM (Merity et al., 2018). For a fair comparison, our DM-LSTM uses exactly the same hyper-parameters and setups as AWD-LSTM. Since Transformer-based models' strong performance relies on training with large datasets, it will perform worse than random when trained on a small dataset (Shen et al., 2021a). We still report Transformer-based models' language modeling results on PTB in Appendix C.

Results We compare our method with its base model, AWD-LSTM, and we report the results along with other state-of-the-art models in Table 7. Compared with the AWD-LSTM, our DM-LSTM reduces the perplexity by 1.4 on the validation set and 1.1 on the test set, indicating that incorporating knowledge of syntactic structures in our proposed manner can substantially improve language modeling. Compared with other models that also leverage syntactic knowledge, our DM-LSTM strongly outperforms RNNG, PRPN, and URNNG. Moreover, though DM-LSTM does not make any changes to the architecture of the AWD-LSTM language model, it still achieves a comparable perplexity with ON-LSTM. Note that, since our method is model-agnostic, it can be harmonically

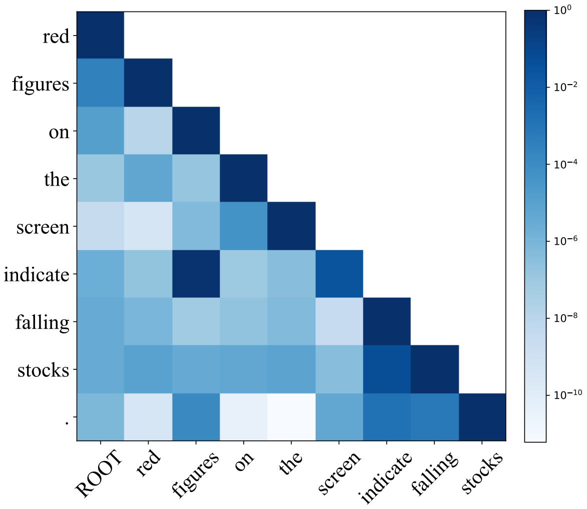

Figure 3: Visualization of dependency attention distributions. We left-shift the sentence by one step in the y-axis to better display the attention between the predicted next-token and the context in each row.

combined with other state-of-the-art models, such as MoS (Yang et al., 2018) and DOC (Takase et al., 2018).

4 Discussion

4.1 Visualization

We show how our proposed method works by visualizing the dependency attention distributions. We use DM-Transformer to generate a sentence: "red figures on the screen indicate falling stocks." For each generation step, we record this step's dependency attention distribution. When we finally generate the whole sentence, we get 9 distributions and plot Figure 3 from them. Each row in Figure 3 shows the dependency attention distribution of the model when generating the corresponding Y-axis token. When predicting the token "indicate", DMLM pays great attention to "figures". This is because these two tokens have a direct dependency connection in the dependency parse tree, and our method successfully captures this relationship. In addition, DMLM also helps the model better organize dependency information when the next-tokens, such as "screen" and "stocks", have dependencies on more than one token in the context.

4.2 Case Study

We perform case studies for a better understanding of the model performance. Table 8 provides examples of conditional text generation produced by our DM-models and other baselines. Obviously, all the DM-models can generate more reasonable and coherent story endings. Additionally, some examples of unconditional text generation are shown in Table 9 and Appendix E. These examples show that our DMLM can help base models generate more reasonable, readable, fluent, and natural sentences.

4.3 Computational Complexity

Compared with vanilla RNN, our DM-RNN indeed increases the computational complexity from $O(T)$ to $O(T^2)$. In practice, we can follow Merity et al. (2017) to set a context window that allows DMLM looks L timesteps into the past at most, where L is the context length. However, our DMLM can efficiently apply to Transformer-based models without additional computational complexity.

5 Related Works

Many previous studies have shown that leveraging the knowledge of syntactic structures can improve NLG (Chelba, 1997; Roark, 2001; Emami and Jelinek, 2005; Buys and Blunsom, 2015). Mirowski and Vlachos (2015) incorporated syntactic dependencies into the RNN formulation, but they limited the scope to the scoring of complete sentences, not to next word prediction. Some other efforts have been done to integrate dependency structure into neural machine translation (NMT) from both the source and target side. Eriguchi et al. (2016) proposed a tree-to-sequence attentional NMT model where source-side parse tree was used. Wu et al. (2017) involved target syntactic trees into NMT model to jointly learn target translation and dependency parsing. Casas et al. (2020) introduced a syntactic inductive bias to NLG in an iterative non-autoregressive way.

For neural language models, recently, Dyer et al. (2016) proposed recurrent neural network grammar (RNNG) to jointly model syntax and surface structure by incrementally generating a syntax tree and sentence. Subsequent work (Kim et al., 2019) extended the model to an unsupervised version. Shen et al. (2018) introduced the Parsing-Reading-Predict Networks (PRPN) to calculate syntactic distances among words and use self-attention to compose previous states. Its subsequent work (Shen et al., 2019) transferred the distance notion to LSTM cell, and introduced Ordered Neurons LSTM (ON-LSTM).

However, all these methods, mainly based on RNN (Sutskever et al., 2014), incorporate knowl-

Story context:	[FEMALE] bought packets of vegetable seeds from the store . she dug up the dirt in her garden . [FEMALE] planted onions , cilantro , and tomatoes . [FEMALE] watered the garden every night .
Golden Text:	by the end of the summer [FEMALE] had enough vegetables to make salsa .
PRPN:	she got to work in the morning and was happy to have a garden .
ON-LSTM:	[FEMALE] planted the plants and made it a huge success .
AWD-LSTM:	[FEMALE] was happy to be helping her plants .
DM-LSTM:	soon , [FEMALE] had enough vegetables to grow in her garden !
Transformer:	she went to the store to buy the seeds .
DM-Transformer:	soon , [FEMALE] had her garden full of vegetables !
GPT-2:	[FEMALE] 's garden grew very quickly and dry .
DM-GPT-2:	[FEMALE] now has fresh fruits and vegetables in her garden .

Table 8: Examples of conditional text generation on ROCStories dataset.

Golden Text:	what this group does is to take down various different websites it believes to be criminal and leading to terrorist acts .
PRPN:	the right point to pay for the purchase of a bike , that ' s all we want to do to build , build together the support that i need to get here .
ON-LSTM:	it ' s great to know that my experience has changed my mind because i ' m not going to work because i ' ve had to talk about that .
AWD-LSTM:	this is a tragic attack and it is understood that the pair will come up with a package of documents which may be possible .
DM-LSTM:	the win over bernie sanders was an emotional moment for clinton , who was running in the general election , though she lost their state of vermont .
Transformer:	' i ' ve just been in that position so i ' ve never seen anything like this before , but it ' s something i have to say and i ' m going to go to and win this series .
DM-Transformer:	in the second quarter of 2015 , the outlook for consumer spending rose 8 . 6 per cent , but for the fourth quarter , the company said it expects to expand by 0 . 7 per cent .
GPT-2:	if i had said a bit of pressure , i would probably be in a different position if i was a coach .
DM-GPT-2:	they ' ve also said that it ' s difficult to know how many emails clinton actually sent to her in recent weeks or whether she would be the nominee .

Table 9: Examples of unconditional text generation on EMNLP2017 WMT News dataset.

edge of syntactic structures by introducing complex architectural changes. Therefore, it can get very unwieldy to adapt them to other neural language models, such as Transformer and GPT-2.

6 Conclusion

In this paper, we introduce Dependency-based Mixture Language Models, which can incorporate knowledge of dependency structures into arbitrary auto-regressive generation models without any changes to the original architectures. Both automatic and human evaluation results in extensive experiments across different tasks and different architectures demonstrate the effectiveness and generalizability of our method.

In the future, we will explore to incorporate the dependency labels into our method, and combine our DMLM with more neural language models. Second, we would like to integrate other linguistic knowledge, such as constituency structures and semantic information, into neural language models in our manner.

Acknowledgements

This work was supported by National Key R&D Program of China (No.2018YFB1005100), Bejing Academy of Artificial Intelligence (BAAI) and State Key Laboratory of Media Convergence Production Technology and Systems. We appreciate the anonymous reviewers for their helpful comments. Xiaojun Wan is the corresponding author.

References

Mahtab Ahmed, Muhammad Rifayat Samee, and Robert E. Mercer. 2019. You only need attention to traverse trees. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 316–322.

Jan Buys and Phil Blunsom. 2015. Generative incremental dependency parsing with neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 2: Short Papers*, pages 863–869.

Massimo Caccia, Lucas Caccia, William Fedus, Hugo Larochelle, Joelle Pineau, and Laurent Charlin. 2020. Language gans falling short. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

Noe Casas, José A. R. Fonollosa, and Marta R. Costa-jussà. 2020. Syntax-driven iterative expansion language models for controllable text generation. In *Proceedings of the Fourth Workshop on Structured Prediction for NLP@EMNLP 2020, Online, November 20, 2020*, pages 1–10.

Ciprian Chelba. 1997. A structured language model. In *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, 7-12 July 1997, Universidad Nacional de Educación a Distancia (UNED), Madrid, Spain*, pages 498–500.

Junyoung Chung, Sungjin Ahn, and Yoshua Bengio. 2017. Hierarchical multiscale recurrent neural networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc Viet Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2978–2988.

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 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Wenyu Du, Zhouhan Lin, Yikang Shen, Timothy J. O'Donnell, Yoshua Bengio, and Yue Zhang. 2020. Exploiting syntactic structure for better language modeling: A syntactic distance approach. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 6611–6628.

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A. Smith. 2016. Recurrent neural network grammars. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 199–209.

Ahmad Emami and Frederick Jelinek. 2005. A neural syntactic language model. *Mach. Learn.*, 60(1-3):195–227.

Akiko Eriguchi, Kazuma Hashimoto, and Yoshimasa Tsuruoka. 2016. Tree-to-sequence attentional neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Sarthak Garg, Stephan Peitz, Udhyakumar Nallasamy, and Matthias Paulik. 2019. Jointly learning to align and translate with transformer models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 4452–4461.

Jian Guan, Fei Huang, Minlie Huang, Zhihao Zhao, and Xiaoyan Zhu. 2020. A knowledge-enhanced pretraining model for commonsense story generation. *Trans. Assoc. Comput. Linguistics*, 8:93–108.

Jian Guan and Minlie Huang. 2020. UNION: an unreferenced metric for evaluating open-ended story generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 9157–9166.

Jiaxian Guo, Sidi Lu, Han Cai, Weinan Zhang, Yong Yu, and Jun Wang. 2018. Long text generation via adversarial training with leaked information. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5141–5148.

Andrew F Hayes and Klaus Krippendorff. 2007. Answering the call for a standard reliability measure for coding data. *Communication methods and measures*, 1(1):77–89.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *In 8th International Conference on Learning Representations*.

Athul Paul Jacob, Zhouhan Lin, Alessandro Sordoni, and Yoshua Bengio. 2018. Learning hierarchical structures on-the-fly with a recurrent-recursive model for sequences. In *Proceedings of The Third*

Workshop on Representation Learning for NLP, Rep4NLP@ACL 2018, Melbourne, Australia, July 20, 2018, pages 154–158.

Yoon Kim, Alexander M. Rush, Lei Yu, Adhiguna Kuncoro, Chris Dyer, and Gábor Melis. 2019. Unsupervised recurrent neural network grammars. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 1105–1117.

Xiangzhe Kong, Jialiang Huang, Ziquan Tung, Jian Guan, and Minlie Huang. 2021. Stylized story generation with style-guided planning. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 2430–2436.

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 the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1426–1436.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 110–119.

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

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. 2018. Regularizing and optimizing LSTM language models. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Tomáš Mikolov et al. 2012. Statistical language models based on neural networks. *Presentation at Google, Mountain View, 2nd April*, 80:26.

Piotr Mirowski and Andreas Vlachos. 2015. Dependency recurrent neural language models for sentence completion. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 2: Short Papers*, pages 511–517.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James F. Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 839–849.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

Hao Peng, Roy Schwartz, and Noah A. Smith. 2019. Palm: A hybrid parser and language model. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3642–3649.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Brian Roark. 2001. Probabilistic top-down parsing and language modeling. *Comput. Linguistics*, 27(2):249–276.

Stanislau Semeniuta, Aliaksei Severyn, and Sylvain Gelly. 2018. On accurate evaluation of gans for language generation. *CoRR*, abs/1806.04936.

Yikang Shen, Zhouhan Lin, Chin-Wei Huang, and Aaron C. Courville. 2018. Neural language modeling by jointly learning syntax and lexicon. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.

Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron C. Courville. 2019. Ordered neurons: Integrating tree structures into recurrent neural networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Yikang Shen, Shawn Tan, Alessandro Sordoni, Siva Reddy, and Aaron C. Courville. 2021a. Explicitly modeling syntax in language models with incremental parsing and a dynamic oracle. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 1660–1672.

Yikang Shen, Yi Tay, Che Zheng, Dara Bahri, Donald Metzler, and Aaron C. Courville. 2021b. Structformer: Joint unsupervised induction of dependency and constituency structure from masked language modeling. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 7196–7209.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1631–1642.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

Sho Takase, Jun Suzuki, and Masaaki Nagata. 2018. Direct output connection for a high-rank language model. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4599–4609.

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 *In Advances in Neural Information Processing Systems*, pages 5998–6008.

Yau-Shian Wang, Hung-yi Lee, and Yun-Nung Chen. 2019. Tree transformer: Integrating tree structures into self-attention. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 1061–1070.

Adina Williams, Andrew Drozdov, and Samuel R. Bowman. 2018. Do latent tree learning models identify meaningful structure in sentences? *Trans. Assoc. Comput. Linguistics*, 6:253–267.

Shuangzhi Wu, Dongdong Zhang, Nan Yang, Mu Li, and Ming Zhou. 2017. Sequence-to-dependency neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 698–707.

Zenan Xu, Daya Guo, Duyu Tang, Qinliang Su, Linjun Shou, Ming Gong, Wanjun Zhong, Xiaojun Quan, Daxin Jiang, and Nan Duan. 2021. Syntax-enhanced pre-trained model. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 5412–5422.

Zhilin Yang, Zihang Dai, Ruslan Salakhutdinov, and William W. Cohen. 2018. Breaking the softmax bottleneck: A high-rank RNN language model. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 2852–2858.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

Xinyuan Zhang, Yi Yang, Siyang Yuan, Dinghan Shen, and Lawrence Carin. 2019. Syntax-infused variational autoencoder for text generation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2069–2078.

Junru Zhou and Hai Zhao. 2019. Head-driven phrase structure grammar parsing on penn treebank. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2396–2408.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1097–1100.

Julian Georg Zilly, Rupesh Kumar Srivastava, Jan Koutník, and Jürgen Schmidhuber. 2017. Recurrent highway networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70, pages 4189–4198.

Konrad Zolna, Devansh Arpit, Dendi Suhubdy, and Yoshua Bengio. 2018. Fraternal dropout. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.

A Experimental Setup

All the algorithms are implemented in Pytorch and trained on a machine with 8 NVIDIA GTX 2080Ti GPUs.

A.1 Conditional Text Generation

The dataset statistics of ROCStories dataset is reported in Table 10.

	Train	Validation	Test
#Stories	78,529	9,816	9,816

Table 10: Statistics of ROCStories dataset.

In this task, both the DM-LSTM and base LSTM are built on a AWD-LSTM language model with an embedding size of 400 and hidden layer units 1150. The dropout rates are $0.4, 0.25, 0.4$ for the output of the last layer, outputs between LSTM layers, and input embedding layers, respectively. The weight dropout for the RNN hidden to hidden matrix is 0.5, and the dropout rate to remove words from embedding layer is 0.1. The context length for DM-LSTM is set to 56. For PRPN and ON-LSTM, we keep their original settings.

In this task, all the models are trained on a singe GPU with learning rate 30, weight decay $1.2e-6$. LSTM baselines are trained for 500 epochs with batch size 100. DM-LSTM is first trained by dependency modeling objective for 100 epochs with batch size 80, and then by language modeling in Equation 4 for 400 epochs with batch size 60 due the computational budgets limit.

For both the DM-Transformer and base Transformer, we use a standard 6-layer Transformer language model with 8 attention heads, embedding dimension 512, projection dimension 2048 and dropout rate 0.1. During training, we use Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$, weight decay 0.01 and learning rate $5e-4$, and apply the dynamic batching provided by fairseq[6] to train both the models with 4 GPUs. Transformer is trained for 60 epochs, while DM-GPT-2 is first trained by dependency modeling for 30 epochs, and then trained by language modeling in Equation 4 for 30 epochs.

We use the pretrained GPT-2-base model for both the DM-GPT-2 and base GPT-2. In this comparison, we apply the same training settings with Transformer-base models except that learning rate is set to $5e-5$. GPT-2 is trained for 80 epochs, while DM-GPT-2 is first trained by dependency modeling for 40 epochs, and then trained by language modeling in Equation 4 for 40 epochs.

For all the models, we select the best checkpoint according to the loss of validation set for testing.

A.2 Unconditional Text Generation

The dataset statistics of EMNLP2017 WMT News dataset is reported in Table 11.

	Train	Validation	Test
#Stories	268,586	10,000	10,000

Table 11: Statistics of EMNLP2017 WMT News dataset.

The context length for DM-LSTM is set to 36. LSTM baselines are trained for 500 epochs with batch size 300. DM-LSTM is first trained by dependency modeling objective for 100 epochs with batch size 300, and then by language modeling for 400 epochs with batch size 200. Besides, all the other experimental setups are the same with those for the conditional text generation task.

A.3 Language Modeling

The dataset statistics of Penn Treebank dataset is reported in Table 12.

	Train	Validation	Test
#Stories	42,068	3,370	3,761

Table 12: Statistics of Penn Treebank dataset.

The context length for DM-LSTM is set to 16. DM-LSTM is trained for 1000 epochs with batch size 20, following (Merity et al., 2018). Besides, all the other experimental setups are the same with those for the conditional text generation task.

B Impact of the Dependency Parser

In our work, we use an off-the-shelf dependency parser to get the dependency parse trees for dependency modeling. Consequently, the better the quality of dependency parsing, the better the performance of our method. HPSG Parser (Zhou and Zhao, 2019), the dependency parser we use, is one of the state-state-of-the-art parsers. This ensures the high quality of parsing results. Zhou and Zhao (2019) trained HPSG Parser with the training set

[6] https://github.com/pytorch/fairseq

of PTB, and kept the test set held-out. So, when we do language modeling on PTB, the parser will not inject any future predictions that contribute to testing.

HPSG Parser maintains high-quality on out-of-domain text, as shown in its paper (Zhou and Zhao, 2019). Most importantly, even on the out-of-domain datasets, i.e., ROCStories and EMNLP2017 WMT News, our work can still obtain a significant improvement, as shown in Section 3.1 and Section 3.2.

C Language Modeling on Transformer-based Models

The language modeling results of Transformer-based models evaluated on PTB dataset are shown in following Table 13.

Models	#Params	Dev PPL	Test PPL
Transformer	24M	100.7	106.7
DM-Transformer	24M	**80.6**	**84.6**
GPT-2	163M	62.6	55.2
DM-GPT-2	163M	**58.8**	**51.6**

Table 13: Transformer-based models' perplexity evaluated on validation and test sets of Penn Treebank dataset.

The good performance of Transformer-based models often rely on training with large datasets, but PTB is a very small dataset. Therefore, Transformer-based models perform worse than LSTM-based models, as shown in Table 7 and Table 13. However, our DM-models still substantially reduce the perplexity compared with base models. DM-Transformer improves the base Transformer by over 20 perplexity points on both the validation and test set, and DM-GPT-2 also improves the base GPT-2 by almost 4 perplexity points. These results further confirm the effectiveness our method.

D Human Evaluation

We post the human evaluation questionnaire, as shown in Table 14 and Table 15, and then recruit five workers with sufficient high English skills. We pay each worker 45 US dollars, and let them complete the evaluation within a week.

E Generated Examples

For a more general comparison, we present more generated examples of unconditional text generation in Table 16.

Task Description
Each story contains about five sentences. For each story, we will put the first four sentences into two different systems, and then systems generate the last sentence. The requirement for this manual evaluation is to judge **which story better complies with the English grammar norm, and is more logically related to the first four sentences.**
NOTE that the names in all stories are replaced with "[MALE]" or "[FEMALE]" or "[NEUTRAL]", and all the sentences are preprocessed by lowercasing, separating punctuation, and splitting conjunctions. They are not grammar errors. Please ignore these when evaluating and do not allow them to affect your judgments.

Evaluation Criterion
You need to compare the stories from two metrics: **grammaticality** and **logicality**. And the two metrics are **independent** of each other. One of the judgments should not have any influence on the other one. Specific criteria for evaluating are as follows:
1. Grammaticality
In the process of evaluating grammaticality, it should be considered whether the statement itself complies with the English standard usage. Then annotate which story is better at grammaticality. You may not care about what the generated sentences are saying but **only if there are any grammatical problems in the sentence itself.**
2. Logicality
In the process of evaluating logicality, you need to carefully read the whole story including the first four sentences and the generated sentence, and compare stories in logicality. Then annotate which story is better at logicality in terms of the coherence to the given beginnings and the inter-sentence causal and temporal dependencies. In this process, you may encounter sentences that are not completely grammatical.**Please make a logical evaluation based on the main part of the sentence (such as some keywords, etc.) and what you can intuitively feel.** Under the circumstances, the story can be judged totally illogical only if the grammar is too poor to understand the meaning or the logic is unreasonable.

Notes
· Again, the grammaticality and logicality of the story are **two independent metrics**. Some very logically inappropriate generated stories are good in the grammaticality part, and there are some stories with obvious grammatical errors but they don't affect the respective judgment.
· Sometimes, there may be more than one kind of reasonable story for a beginning. Please do not limit your imagination. **As long as the story is logically reasonable, direct, and able to make sense, it can be judged good in logicality.**
· Some stories may not be accurately judged. In the process of determining the comparison of this type of two stories, according to your own understanding of the examples and the subjective feelings of the stories, choose a better story you think is the most appropriate. **Please ensure that your evaluation criterion for different stories is the same.**

Table 14: Human evaluation questionnaire for conditional text generation.

Task Description
In this review, you will read 900 sentences. For each sentence, you should determine **whether the sentence is written by human**. **Note**: All the sentences are preprocessed by lowercasing, separating punctuation, and splitting conjunctions. They are not grammar errors. Some sentences may have a specific context, or they may be talking about completely fictitious things. Please ignore these when evaluating and do not allow them to affect your judgments.
Evaluation Criterion
The judgment can mainly depend on your own understanding and the subjective feelings. But fluency, readability, engagement (whether you felt interested about the sentence), and anything else that you think is important can also help you make a decision.

Table 15: Human evaluation questionnaire for unconditional text generation.

Golden Text:	over 1 , 600 a day have reached greece this month , a higher rate than last july when the crisis was already in full swing .
	" we ’ re working through a legacy period , with legacy products that are 10 or 20 years old , " he says .
	’ the first time anyone says you need help , i ’ m on the defensive , but that ’ s all that i know .
	out of those who came last year , 69 per cent were men , 18 per cent were children and just 13 per cent were women .
PRPN:	as a mother , i can ’ t work to be working on some kind of stuff , but i ’ m not really sure that the single market is going to be as bad as i ’ m on .
	in fact , there is a good position to focus on this and that will be a clear opportunity for the us to make sure that we do not have any concerns .
	there ’ s still more opportunities than that , but this is what you ’ re talking about , but it ’ s not right .
	as well as a labour party , the former party member who claimed the vote in the referendum on whether to vote to leave the eu should be questioned .
ON-LSTM:	so they did that because we ’ ve been saying they ’ re going to be fighting for this state , but they ’ re going to keep going .
	the official said they were hoping to make a contribution in its strong inflation growth in the future , and that a more conservative leader could look for jobs and be stronger .
	it ’ s something that i think are a good team , the first place to do it and i ’ m really happy .
	’ there ’ s no question that the person we ’ re going to take is probably an important thing to be asked , " said john .
AWD-LSTM:	in this month ’ s election , the u . s . economy has fallen in the past few years , a higher than a decade ago .
	in the last year i had been an 18 - year - old woman in my two - year - old son .
	it was a great test for me to try to get back on the bench and be there , it ’ s a huge challenge for us .
	i just think it ’ s important for us to do something that would help them in the best way we can to do it .
DM-LSTM:	" the united states has to come to mind that the threat of climate change is less of a serious issue , " the pentagon said in a statement .
	in the event of an initial campaign for the democratic nomination , he had released some of the most controversial ads that they had been speaking about since he was a president .
	there is an example of a presidential candidate who has been on the debate trail for more than a year .
	the central bank of japan is set to raise its benchmark interest rate at its first time in nearly a decade .
Transformer:	you can ’ t get away with things that are better than you did at home and hopefully get better than not the first team .
	in the case of the cases , the nsw government said it would accept 10 , 000 additional emergency costs if it did not help the industry .
	if there is an oil price that is at stake , it is not as far as the price of oil .
	the country has promised to build a nationwide population of about 150 , 000 to more than 2 , 000 , with a budget to help in building more affordable housing .
DM-Transformer:	in this particular area , as in the modern world , he is seen as someone who takes the risk of suffering a heart attack .
	that ’ s why we ’ re talking about the second half of the year , and a lot of people have asked us to do the best we can .
	the vast majority of american voters , particularly those who chose trump , said that he had changed the result .
	so this is a big step , and i ’ m really excited to be part of the new york olympics .
GPT-2:	the reason is that the student community who doesn ’ t know what he ’ s talking about , or who ’ s not even a businessman , he ’ s going to take care of itself .
	the difference is that the reality of " brexit " has been the single largest trading partner in the world , and now is it .
	the game is now used to push for players to learn from them and learn from them and also play in the front of them .
	the first woman to run for president is to make a case for a woman she wants to make as president of the united states .
DM-GPT-2:	" i just thought that the whole picture was a strange story , " he said in a telephone interview on thursday .
	" the importance of local authorities is very strong , " she said in an interview on friday afternoon .
	we are working closely with the government to resolve this issue and have to work with local authorities to resolve the problem .
	a final verdict will be held on thursday at the supreme court in washington on march 15 , 2017 .

Table 16: Examples of unconditional text generation on EMNLP2017 WMT News dataset.