LANGUAGE MODEL PRE-TRAINING WITH LINGUISTICALLY MOTIVATED CURRICULUM LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Pre-training serves as a foundation of recent NLP models, where language modeling task is performed over large texts. It has been shown that data affects the quality of pre-training, and curriculum has been investigated regarding sequence length. We consider a linguistic perspective in the curriculum, where frequent words are learned first and rare words last. This is achieved by replacing hierarchical phrases that contain infrequent words by their constituent labels. By such syntactic substitutions, a curriculum can be made by gradually introducing words with decreasing frequency levels. Without modifying model architectures or introducing external computational overhead, our data-centric method gives better performances over vanilla BERT on various downstream benchmarks.

1 INTRODUCTION

Pre-trained language models (PLM) have gained much attention and achieved strong results in various NLP tasks (Devlin et al., 2019; Radford et al., 2019; Brown et al., 2020). Based on self-supervised learning objectives such as causal language modeling (Peters et al., 2018; Radford et al., 2019), masked language modeling (Devlin et al., 2019), and text-to-text generation (Lewis et al., 2020; Raffel et al., 2020), PLM can learn task-agnostic transferable features from large-scale unlabeled corpora. It has also been shown that PLM can encode syntactic (Hewitt & Manning, 2019; Goldberg, 2019; Wu et al., 2020), semantic (Tenney et al., 2019; Jawahar et al., 2019), and factual (Petroni et al., 2019; Dai et al., 2022) knowledge.

For improving the representation power of PLM, much research has been done on setting different training objectives (Zhang et al., 2019; Yang et al., 2019; Liu et al., 2019b), modifying model architectures (Dong et al., 2019; Clark et al., 2020; He et al., 2021) and scaling up the parameter count (Shoeybi et al., 2019; Rae et al., 2021; Fedus et al., 2022; Chowdhery et al., 2022). However, relatively less work considers on the way of using pre-training corpus, where most of the methods leverage the raw text as a whole (from millions to billions of tokens) and train for multiple epochs, given sufficient data, the training strategy may have reduced effect.

Recent work has shown the influence of a curriculum for pre-training. Li et al. (2021) propose a sequence length warmup strategy for GPT-2 pre-training, which can improve training stability and efficiency. Similarly, Nagatsuka et al. (2021) split corpus into blocks with specified sizes for BERT pre-training. These methods focus on changing the sequence length instead of the content and emphasize the convergence speed. Beyond text length, there is a more salient discrepancy between the current PLM training and the language learning process of humans. In particular, we only learn limited but the most common and useful words at the beginning, then we grasp some basic syntactic concepts such as part-of-speech, set phrase, and clause, before recognizing a large number of uncommon words via generalization or their specific usages.

Inspired by psycho-linguistic curriculum learning (Elman, 1990; 1993; Bengio et al., 2009), we propose a data-centric approach that progressively pre-trains a language model using a curriculum that involves reconstructed data. An example contrast of the masked language model pre-training and our multi-stage curriculum training is shown in Figure 1. Our curriculum consists of m stages ($m = 2$ in Figure 1(b)), with each having a incrementally larger vocabulary. Specifically, we first use constituent (and part-of-speech) labels from Penn Treebank (Marcus et al., 1993) to replace the lower frequency words, and the model updates using the text composed of the most frequent words and the constituent labels. In this stage, all words are at the same frequency level, and thus

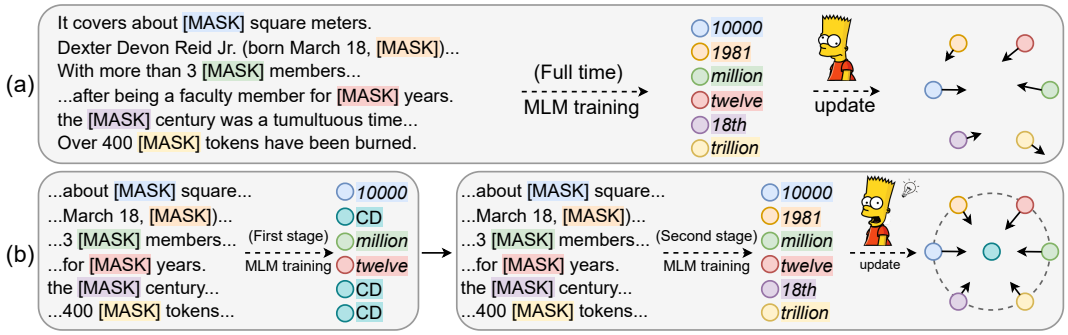

Figure 1: An example of (a) vanilla masked language modeling, and (b) our method using two-stage curriculum training. In the first stage, we replace the original lower frequency target word such as "*trillion*" by a constituent label `CD`, which stands for the cardinal number. The representations can be better updated in the latter training stage after acquiring the "concept" of `CD`.

trained equally thoroughly. Then we gradually introduce less frequent words, letting the model further improve based on previously acquired knowledge. During this stage, the previously learned constituent labels can serve as categorical knowledge to guide the learning of infrequent words.

Experimental results using BERT show that our method can improve pre-training, showing better performance across tasks including general language understanding, named entity recognition, question answering, part-of-speech tagging, and parsing. Through empirical analysis, we find that our curriculum training can mitigate the representation degeneration problem (Gao et al., 2019) in PLM, and the injected constituent labels can encode meaningful linguistic features that bridge the word representations across different frequencies. Code and model will be released for further research.

2 Method

We take BERT (Devlin et al., 2019) as our baseline, which is trained using masked language modeling, one of the most successful self-supervised learning objectives for pre-training (§2.1). Our method leverages linguistically motivated curriculum learning based on vanilla masked language modeling (§2.2), with a dedicated data-centric method for stage-wise corpus reconstruction (§2.3).

2.1 Masked Language Modeling

Masked language modeling (MLM) aims to predict the original target word w_i through modeling the contextualized representation of a randomly masked word $\widetilde{w}_i$ in its context:

$$\mathcal{L}_{\text{MLM}} = -\sum_i \log P_\theta(w_i|\widetilde{w}_i) = -\sum_i \log \frac{\exp(E(w_i)^\top \widetilde{h}_i)}{\sum_{j=1}^{|V|} \exp(E(w_j)^\top \widetilde{h}_i)}, \tag{1}$$

where $\widetilde{w}_i$ is the masked symbol `[MASK]` in a context, $\widetilde{h}_i$ is the corresponding contextualized output.

2.2 Motivation for Data-centric Curriculum Training

During vanilla MLM pre-training shown in Figure 1(a), the model needs to predict the corresponding word surface independently. Although the more common words such as "*10000*", "*million*", and "*twelve*" could be updated frequently, lower frequency words such as "*1981*", "*18th*", and "*trillion*" may receive far less training signal. The discrepancy in word frequency may do harm to model training in tasks such as text classification and machine translation (Gong et al., 2018). For language modeling, previous studies have shown that lower frequency words are learned poorly (Schick & Schütze, 2020), which can also degenerate the training process for all other words (Yu et al., 2022).

We consider a psycho-linguistically motivated curriculum learning method by proposing two main rules to address the issues: 1) common words first, rare words next, and 2) models are learned with structural constraints, or syntax. To build a curriculum schedule that satisfies the above rules, we inject constituent labels (Marcus et al., 1993) into raw text, replacing different words in different training stages by their corresponding constituent structures.

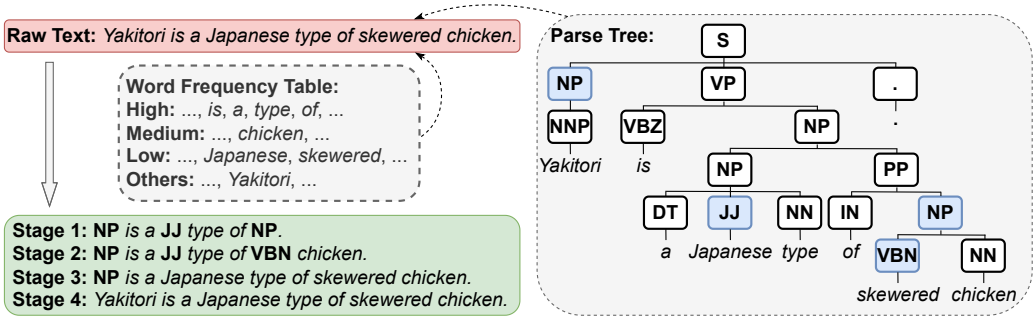

Figure 2: Illustration for reconstructing the sentence "*Yakitori is a Japanese type of skewered chicken.*" in our four stages of curriculum training. At each stage, we use the corresponding constituent labels (colored in blue) to replace the original words according to their overall frequency.

In Figure 1(b), in contrast to the baseline method, we mitigate the influence of infrequent words in the initial stages by predicting the `[MASK]` symbol as a constituent label "`CD`" (cardinal number) instead of the original word, based on the fact that these words share a unified constituent structure across texts. Then we train our model using the original target after it acquires some "concept" of what `CD` is. Since the contextualized representation is built for predicting the unified virtual target `CD` in the first stage, the update of lower frequency word representations could be better guided through such previously learned category knowledge.

2.3 Reconstructing Data with a Constituency Parser

Based on the above motivation, we use a mixup strategy that injects multiple constituent labels into the raw text and progressively incorporates more infrequent words. Our final curriculum includes four stages, where a different number of words in the word frequency table are used in each stage, according to Algorithm 1.

Algorithm 1 Injecting constituent labels for language model pre-training.

Input: Raw text s_t, a constituency Parser, a word frequency table TopList
Output: Reconstructed text $s_t^\dagger$

$s_t^\dagger = [\,]$, Tree = Parser.parse(s_t)
function PROCESS(Tree)
  if Tree has no SubTree **then**
    if Tree.word in TopList **then**
      $s_t^\dagger \leftarrow$ Tree.word
    else
      $s_t^\dagger \leftarrow$ Tree.tag
  else
    if all Tree.leaves.word not in TopList **then**
      $s_t^\dagger \leftarrow$ Tree.tag
    else
      for each SubTree in Tree **do**
        PROCESS(SubTree)

Figure 2 shows an example of injecting constituent labels into a sentence. In the first stage, we only keep the most frequent words ("*is*", "*a*", "*type*", "*of*") while replacing the others by their corresponding constituent labels, these labels are directly used as normal words in the corpus, which can also be randomly masked and predicted. In the second stage, we allow medium frequency words ("*chicken*") to appear together with the most frequent words and remaining labels. In the third stage, we add the low-frequency words ("*Japanese*", "*skewered*") and the data is close to the original format, except for some labels that indicate rare words ("*Yakitori*"). In the last stage, we use the original corpus for training. In practice, we set the word frequency ranking intervals of ∼0.5K, 0.5K∼3K, 3K∼18K, and 18K∼ as the high, medium, low frequency, and other rare words, respectively. The number of training stages and the frequency intervals are set roughly according to the word distribution and vocabulary size of the embedding table, we leave the optimization of these settings to future work.

To simplify the implementation of our curriculum, we directly use the wordfreq library from (Speer et al., 2018) and ignore the statistics of sub-words (Sennrich et al., 2016; Wu et al., 2016) after tokenization. Thus we can directly process the corpus without considering the distinction of word distribution for different domains, and avoid selecting among tokenizers.

3 Experiments

3.1 Pre-training

We follow the setup of BERT-base-cased architecture from Devlin et al. (2019). The model is a 12 layers Transformer encoder, with a 768 hidden size and 12 attention heads. English Wikipedia

and the BOOKCORPUS (Zhu et al., 2015) are used as the pre-training data. We train our model with AdamW (Loshchilov & Hutter, 2019) optimizer for 1M steps with a learning rate 1e-4, batch size 256, warmup ratio 0.01, and with mixed precision using 8×32GB V100 GPUs. Following the recipe from Liu et al. (2019c) and Izsak et al. (2021), we do not use the next sentence prediction objective.

For offline data reconstruction, we use the Benepar (Kitaev & Klein, 2018) for parsing. In each of the first three stages, we train our model using the reconstructed corpus for 200K steps (*i.e.*, a total of 600K steps). Then we use the raw corpus for the 600K∼1M training steps. Since we add some constituent labels such as `NP`, `VP`, and `JJ` (see the full list in Appendix A) in the text, we enlarge the embedding table by treating them as normal tokens, thus making our vocabulary size slightly larger (from 28,996 to 29,051)[1]. The 55 externally added embeddings can be discarded after pre-training.

3.2 DOWNSTREAM TASKS AND DATASET

We evaluate on general tasks including natural language understanding, named entity recognition, and question answering. Since our method uses text mixed with syntax-related labels during pre-training, we also evaluate on syntax-related tasks such as part-of-speech tagging and parsing. Statistics of the datasets are shown in Appendix B.

GLUE. The GLUE benchmark (Wang et al., 2019a) is used for evaluating general language understanding, we compare on sub-tasks including MNLI (Williams et al., 2018), QQP (Chen et al., 2018), QNLI (Rajpurkar et al., 2016), SST-2 (Socher et al., 2013), CoLA (Warstadt et al., 2019), STS-B (Cer et al., 2017), MRPC (Dolan & Brockett, 2005), and RTE (Bentivogli et al., 2009).

Named Entity Recognition. We use the CoNLL2003 datasets (Tjong Kim Sang & De Meulder, 2003) for named entity recognition, the entity labels include `PER`, `LOC`, `ORG`, and `MISC`.

Question Answering. Two versions of the Stanford Question Answering Dataset (SQuAD) (Rajpurkar et al., 2016; 2018) are used. SQuAD 1.1 aims to predict the text span in the passage. SQuAD 2.0 allows the possibility that no answer exists in the paragraph.

Part-of-Speech Tagging. The Wall Street Journal (WSJ) portion of the Penn Treebank (Marcus et al., 1993) is used for POS tagging. We follow Manning (2011) by selecting sections 0-18 as the training set, 19-21 as the development set, and 22-24 as the test set.

Constituency Parsing. WSJ is also used for constituency parsing, where we use the standard splits with sections 02-21 as the training set, 22 as the development set, and 23 as the test set.

3.3 FINE-TUNING

For constituency parsing, we use the self-attentive encoder (Kitaev & Klein, 2018) and initialize it with different pre-trained models. For sentence-level classification (GLUE), token-level labeling (NER and POS tagging), and span-based question answering (SQuAD), we follow BERT for fine-tuning. Following Liu et al. (2019c) and Lan et al. (2020), we fine-tune STS-B, MRPC, and RTE by starting from a trained MNLI checkpoint. For other tasks, we train separately using single model and single task without data augmentation. Hyperparameter settings are shown in Appendix C.

We compare fine-tuned results using different pre-trained models: 1) The BERT-base-cased checkpoint released by Google, denoted as **BERT**; 2) Our model trained from scratch, denoted as **BERT-reimp**. The main difference between BERT-reimp and BERT is that we do not use the next sentence prediction objective; 3) Our model trained from scratch with curriculum learning, we denote it as **BERT-CL**. The only difference between BERT-CL and BERT-reimp is the training corpus. For our results, we report by averaging five runs with different seeds.

3.4 RESULTS

Table 1 shows the results for the GLUE benchmark. We find that BERT-CL consistently outperforms BERT-reimp across all tasks, showing that our curriculum pre-training is useful. Among all models, BERT-CL gives the best averaged results for both development and test set. We find that

[1]There is another option to avoid increasing vocabulary size by using the `[unused1]` to `[unused99]` tokens for replacing the added constituent labels.

Model	MNLI (m/mm) Acc.	QQP F1	QNLI Acc.	SST-2 Acc.	CoLA Mcc.	STS-B Spear.	MRPC F1	RTE Acc.	*Avg.*
				(*Development Set*)					
BERT	84.09/83.82	**87.53**	90.84	92.31	57.27	88.24	89.41	**65.69**	82.13
BERT-reimp	83.58/83.62	87.12	90.06	92.31	57.43	88.22	89.55	63.28	81.68
BERT-CL	**84.95/84.77**	87.15	**91.06**	**93.00**	**62.06**	**88.41**	**91.14**	65.61	**83.12**
				(*Test Set via Leaderboard*)					
BERT	84.5/83.6	71.1	90.1	93.6	53.3	**84.9**	88.3	**68.4**	79.7
BERT-reimp	83.9/82.6	71.1	90.2	93.7	51.9	82.4	88.7	62.0	78.5
BERT-CL	**85.4/84.4**	**71.2**	**90.6**	**93.9**	**59.8**	83.9	**89.5**	66.5	**80.5**

Table 1: Results on GLUE benchmark dev set and test set. The best results are in bold.

Model	P	R	F1
BERT	90.91	92.24	91.57
BERT-reimp	90.94	91.93	91.43
BERT-CL	**91.63**†	**92.31**†	**91.97**†

Table 2: Results on CoNLL2003 test set. †: Statistically significant compared BERT-reimp with $p < 0.01$ by t-test.

Model	SQuAD 1.1 F1/EM	SQuAD 2.0 F1/EM
BERT	88.96/81.61	75.69/72.51
BERT-reimp	89.38/82.71	78.19/75.20
BERT-CL	**89.87/83.07**	**80.02/77.06**

Table 3: Results on SQuAD dev set.

the CoLA task shows the largest improvement (+12.2% compared with BERT), which aims to judge the linguistic acceptability of sentences. This task benefits from our curriculum since we offer some syntactic labels during pre-training, nevertheless, our method can also improve the capability for general language understanding tasks such as natural language inference and sentiment analysis.

Table 2 shows the results for NER. BERT-CL gives a 91.97 F1 score, better than both BERT (+0.40) and BERT-reimp (+0.54). Note that the reported result on the CoNLL2003 test set from Devlin et al. (2019) is 92.4 F1 score, however, we did not achieve it with the current library, as discussed in Stanislawek et al. (2019) and Gui et al. (2020). The improvement over BERT and BERT-reimp may come from the fact that the named entity usually forms a `NN` and `NP` structure in the constituency tree and such knowledge can be better acquired in our preliminary curriculum training stages.

Table 3 shows the results for question answering. Our model gives the best results on both datasets (+0.49/+0.36 and +1.83/+1.86 over BERT-reimp). Compared with the sentence-level classification and token-level labeling tasks, question answering is more challenging since it requires understanding both query and passage with long-term dependency. We hypothesize that the improvement is due to that span-based answers usually form common constituent structures such as `NP` and `CD`, where these features are quite useful for answering the majority of questions like "*what...*", "*where...*", and "*how many...*", "*when...*".

Table 4 shows the results for POS tagging. By using the full training set, our model gives better results than BERT and BERT-reimp. Since WSJ POS tagging is a less complicated task with rich training resources, we also 1) use fewer training data with 2% to 75% samples, and 2) fix the model parameters while only training a linear classifier upon the contextualized output, which is also called probing (Conneau et al., 2018; Liu et al., 2019a). We find that BERT-CL still consistently gives better results under low-resource and probing settings.

Table 5 shows the results for constituency parsing. Compared with BERT and BERT-reimp, our model gives an absolute improvement with +0.30 and +0.37 F1 scores, respectively. The advantage of using the complete match metric is more significant, where BERT-CL gives +2.47 and +1.30 absolute improvement, respectively. Note that although we leverage a constituency parser for building reconstructed data mixed with constituent labels, the improvement is non-trivial since we use the general purposed MLM training instead of specifically augmenting training data for parsing.

To analyze the influence of corpus, we evaluate on GLUE and CoNLL2003 NER test set by 1) using only the WIKIPEDIA, and 2) adding CC-NEWS (Hamborg et al., 2017) for pre-training. Results are shown in Table 6. We can see that BERT-CL still outperforms BERT-reimp when changing the pre-training corpus, showing that the curriculum is useful when applied to different corpora.

	Model	Accuracy by using p% Training Set					
		p=2	p=5	p=25	p=50	p=75	p=100
Fine-tuning	BERT	96.92	96.93	97.55	97.59	97.66	97.70
	BERT-reimp	96.71	96.96	97.60	97.61	97.66	97.71
	BERT-CL	**96.95**†	**96.97**	**97.62**	**97.63**	**97.70**‡	**97.75**‡
Probing	BERT	94.73	95.15	95.92	96.00	96.05	96.06
	BERT-reimp	94.72	95.20	95.82	95.88	95.91	95.92
	BERT-CL	**94.96**†	**95.34**†	**96.02**†	**96.07**†	**96.12**†	**96.17**†

Table 4: Results on WSJ POS tagging test set by model fine-tuning and linear probing. $\dagger, \ddagger$: Statistically significant compared BERT-reimp with $p < 0.01$ and $p < 0.05$ by t-test, respectively.

Model	LR	LP	F1	CM
BERT	95.20	95.32	95.26	52.06
BERT-reimp	94.98	95.41	95.19	53.23
BERT-CL	**95.54**	**95.58**	**95.56**	**54.53**

Table 5: Results on WSJ parsing test set. CM means complete matching the constituency tree of the whole sentence.

Model	GLUE	CoNLL2003
(*wikipedia only*)		
BERT-reimp	78.2	91.49
BERT-CL	**79.9**	**91.90**
(*wikipedia*, *bookcorpus*, *cc-news*)		
BERT-reimp	78.9	91.52
BERT-CL	**80.7**	**91.98**

Table 6: Results on GLUE and CoNLL2003 by using different pre-training corpus.

To compare with existing work, we reimplement the method from Nagatsuka et al. (2021) which use a four-stage curriculum with increasing block-size (64, 128, 256, and 512) for BERT pre-training. We follow their setting with 250K training steps for each stage and compare it with our method in Table 7. We find that the length-based curriculum does not help much in downstream tasks and our method significantly performs better on both GLUE and CoNLL2003 NER tasks. This shows that our content-based curriculum is not only closer to the process of language learning of humans, but also more helpful for model training.

Model	GLUE	CoNLL2003
Nagatsuka et al. (2021)	78.2	90.91
BERT-CL	**80.5**	**91.97**

Table 7: Comparison with an existing method that leverages curriculum learning for pre-training.

To further evaluate the generalizability of our method, we also try our method to 1) different model settings including larger model RoBERTa-large, and the generative-style language model GPT-2; 2) different curriculum training schedule. We find that our method can also generalize to larger model or GPT-2-style causal language model training, and the curriculum schedule can also affect the overall performance. Detailed results and discussion can be found in Appendix D.

4 Analysis

We analyze the possible reasons behind the performance advantage of BERT-CL, discussing how data-centric curriculum training helps language model pre-training.

4.1 Representation Degeneration of Language Model

Existing work shows the representation degeneration problem of language models (Gao et al., 2019; Ethayarajh, 2019; Wang et al., 2019b; Cai et al., 2020; Biś et al., 2021; Yu et al., 2022), where the word embeddings or contextualized output are highly anisotropic and may limit the representation power. This problem also exists when training static word embeddings (Mu et al., 2018). Figure 3 visualizes the evolution of word embeddings over training iterations using PCA. In the top row, we find that the embeddings of BERT degenerate quickly in the early stage, where the overall distribution falls into a relatively narrow angle. The low-frequency words are significantly separated from others and the overall shape does not change much during all training iterations.

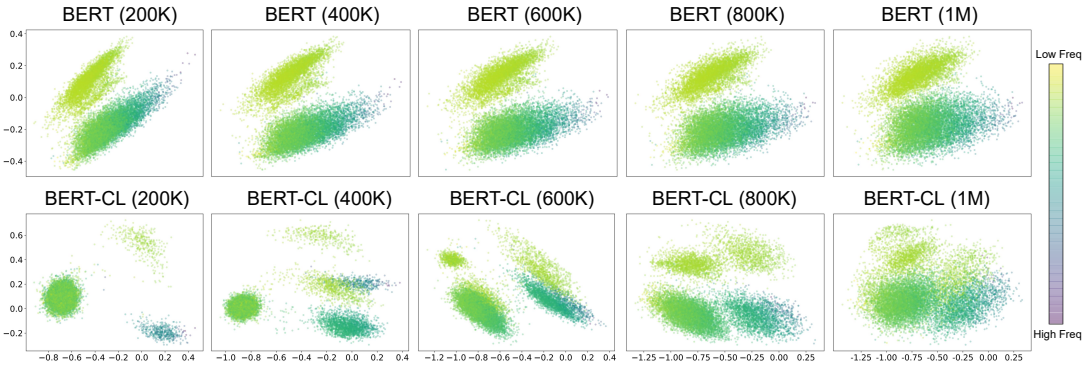

Figure 3: Visualization of word embeddings during pre-training, the numbers in the parentheses denote the training steps. Different colors mean different word frequencies.

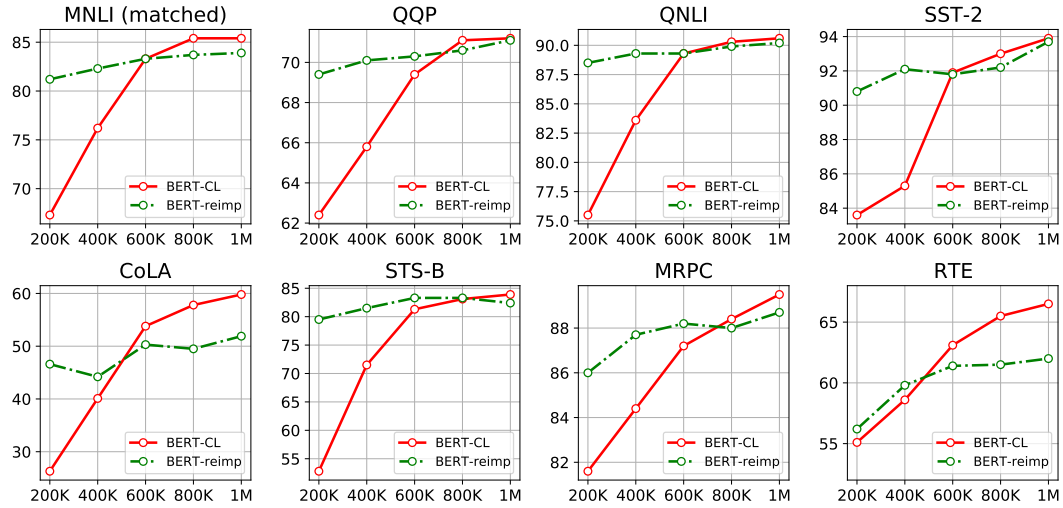

Figure 4: The GLUE benchmark test set results from intermediate checkpoints during pre-training.

The evolution of word distribution in BERT-CL is highly different. Since we incorporate words in different frequency intervals at different stages, they gradually form their clusters stage-by-stage until 600K steps. When using the raw corpus for the last 400K steps of training, we find that the low-frequency words are not represented separately, and different clusters come close to each other. Finally, there is no significant borderline for words with different frequencies, and the overall distribution is also more uniform than BERT. In addition to visualization, we use a measure of isotropy in Mu et al. (2018) and Rajaee & Pilehvar (2021) for evaluation. Details and results are shown in Appendix E, we find that our curriculum training leads to a more isotropic representation space quantitatively.

One of the reasons behind the representation degeneration problem is the frequency discrepancy between words. For example, Gong et al. (2018) find that the word embeddings are heavily biased towards word frequency, proposing an adversarial training method to learn frequency-agnostic representations. Gao et al. (2019) theoretically show that it could be caused by a large number of rarely appeared tokens, and they use a cosine regularization term to enforce normalizing the distribution. Yu et al. (2022) leverage an adaptive gradient gating mechanism for rare tokens training. Although these methods alleviate the degeneration problem and improve performance in pure language modeling or machine translation task, they require additional computational costs and are not used in pre-training. In contrast, we use a data-centric curriculum pre-training approach that introduces the constituent labels and decouples the words with different frequencies by reformulating the corpus.

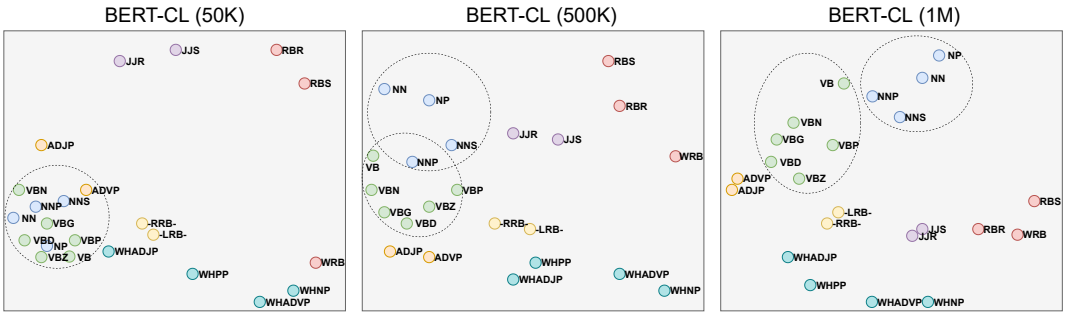

Figure 5: Visualization of the learned constituent label embeddings using t-SNE.

4.2 Intermediate Results of Curriculum Training

In Figure 4, we show the GLUE benchmark test set results using different checkpoints during BERT pre-training with and without curriculum training. We find that the BERT model gives a relatively higher performance in the initial stages for almost all tasks. However, the improvement is limited as the training step increases and also inconsistent for tasks such as SST-2, CoLA, STS-B, and MRPC.

For BERT-CL, the performance is lower at the very beginning, but significantly and stably increases then, especially for the first 600K training steps. This shows that curriculum training can boost the capability of the model constantly. We also find that, for MNLI, QNLI, CoLA, and RTE, BERT-CL (600K) already performs on par with or better than BERT (1M). This shows that the reconstructed data can also steer PLM capability for certain tasks with less training cost.

4.3 Visualization of Constituent Label Embeddings

The added constituent labels are heavily used during the early pre-training stages for data reconstructing, and their corresponding representations are also updated according to their parameters in the extended embedding table. Although the external added parameters can be discarded during fine-tuning, we are still interested in the role of these constituent labels during curriculum training.

Figure 5 visualizes the 2D distributions from the embeddings of constituent labels. In the very early stages (steps 0$\sim$50K), constituent structures can be learned quickly, where some small meaningful clusters emerge and some relationships are also mined. For example, the groups of nouns and verbs, and the similarity between `JJR`/`JJS` and `RBR`/`RBS` pairs. The discriminating distribution becomes clearer when the training steps increases, where the nouns and verbs (also the most common constituent structures) are gradually separated from each other. These show that our model could learn some meaningful concepts of these constituent structures during pre-training.

4.4 Constituent Labels Serve as Anchors to Bridge Token Learning

In the initial stage, the constituent labels are combined with the most frequent words so that the model can learn some syntax rules, the basic usage of frequent words or terms, and their interactions. Since we only leave the high-frequency words and a bunch of labels, the vocabulary size is much smaller, the training signal received for each token is enriched and also more uniform.

The initial results can offer the fundamental capability for language understanding, we then allow more medium and low frequency words to participate in pre-training, combined with constituent labels and most frequent words that have been well learned. When adding infrequent words to replace the constituent labels, since the words and labels share a similar context, the learned contextualized knowledge from constituent labels can also serve as guidance to the latter learning process of the upcoming words. From this perspective, the curriculum settings of using constituent labels can bridge the gap between words across different frequencies.

Table 8 shows the most similar words to some constituent labels in the trained embedding table. We find that 1) The neighboring words can reflect the fine-grained linguistic characteristics of the constituent labels. For example, the plural number ("*games*", "*states*", "*children*") of `NNS`, the 3rd person singular present ("*is*", "*does*", "*has*") of `VBZ`, and the cardinal number ("*five*", "*million*", "*00*") of `CD`; 2) Different words are captured reasonably according to each constituent label, for

Lables	High (Top 1∼500)	Medium (Top 500∼3000)	Low (Top 3000∼)
NNS	*games, states, children*	*teeth, victims, units*	*boxers, responses, bands*
NP	*him, she, he*	*steps, himself, teeth*	*descent, ghosts, witches*
VBZ	*is, does, has*	*becomes, feels, gets*	*receives, saves, serves*
WHNP	*who, what, where*	*whom, whatever, whose*	*whoever, wherever, wherein*
CD	*five, three, four*	*twenty, ten, hundred*	*fifty, fifteen, twelve*
	million, two, 00	*thousand, zero, decade*	*forty, 9th, 8th*

Table 8: The most similar words to each constituent label according to their dot product.

high, medium, and low frequency intervals. This shows that the constituent label can serve as an anchor or prototype to help model words of different frequency ranges in the curriculum; 3) Phrase-level constituent labels that are usually composed of multiple tokens such as noun phrase of NP and wh-adjective phrase of WHNP are also well encoded with meaningful similar words such as "*him*", "*she*", "*he*", and "*who*", "*what*", "*where*". See Appendix F for more examples.

5 Related Work

Knowledge Enhanced LM. It has been shown that PLM are capable of encoding syntax and semantic knowledge (Hewitt & Manning, 2019; Tenney et al., 2019; Pérez-Mayos et al., 2021). There are also a line of work explicitly integrating such knowledge to enhance model representation (Lauscher et al., 2020; Sachan et al., 2021; Xu et al., 2021b). In particular, Levine et al. (2020) leverage word sense prediction task into BERT pre-training, Bai et al. (2022) propose hypernym class prediction for causal language modeling. These methods focus on word-level external knowledge stored in WordNet. Instead, we uses the hierarchical syntactic tree to inject word-, phrase- and clause-level knowledge. Moreover, through the underlying syntax structure of texts, our method can tackle more situations during pre-training where words are not in WordNet (*e.g.*, url/email address, new words).

Curriculum Learning. Curriculum learning has been extensively studied in a range of tasks (Wang et al., 2022). In natural language processing, Bengio et al. (2009) first show that it can help generalization and speed up the convergence of language modeling. For general-purposed language model pre-training, Campos (2021) defines some sentence difficulty metrics based on sentence length, n-gram probability, and part-of-speech diversity for curriculum settings on LSTM-based model. Zhang et al. (2021) group sequences with similar length during pre-training and find that it can help downstream tasks. Nagatsuka et al. (2021) propose progressively increasing the block-size of input text, *i.e.*, using sentences of increasing lengths for pre-training. Li et al. (2021) propose a regularization method for GPT-2 curriculum training, which is also based on sentence length. Unlike these methods, we build a linguistically motivated curriculum based on the learning content.

Data-centric AI. Data-centric method become an emerging topic for modern AI systems (Ng, 2021; Hajij et al., 2021; Xu et al., 2021a; Huang et al., 2022; Eyuboglu et al., 2022). The main idea is to use an established model off-the-shelf, but engineer the data for stronger results, including data collection, annotation, augmentation, cleaning, reordering, and deduplicating (Russell et al., 2008; Krishnan et al., 2016; Wei & Zou, 2019; Press et al., 2021; Agrawal et al., 2021; Lee et al., 2022). To better leverage the language model for downstream tasks, there is also a trend to build cloze-style samples for fine-tuning (Schick & Schütze, 2021; Gao et al., 2021b). Reconstructed training data such as prompts or instructions are also being studied for better using large language models in few/zero-shot scenarios (Sanh et al., 2022; Wei et al., 2022; Yuan & Liu, 2022). In this paper, we attempt to reformulate the data using a syntax-guided mixup strategy for language model pre-training.

6 Conclusion

We investigate curriculum learning for language model pre-training and focus on a purely data-centric method, without setting multiple training tasks, modifying model architecture, or introducing external computational overhead during pre-training. Particularly, we propose a data mixup strategy that injects constituent labels into the text and progressively increases the vocabulary on the corpus from high-frequency to low-frequency words. Experiments on multiple downstream tasks show that our method leads to better performance compared with baselines.

REFERENCES

Ameeta Agrawal, Suresh Singh, Lauren Schneider, and Michael Samuels. On the role of corpus ordering in language modeling. In *Proceedings of the Second Workshop on Simple and Efficient Natural Language Processing*, 2021.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. A Latent Variable Model Approach to PMI-based Word Embeddings. *Transactions of the Association for Computational Linguistics*, 4:385–399, 2016.

He Bai, Tong Wang, Alessandro Sordoni, and Peng Shi. Better language model with hypernym class prediction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, 2009.

Luisa Bentivogli, Ido Dagan, Hoa Trang Dang, Danilo Giampiccolo, and Bernardo Magnini. The fifth pascal recognizing textual entailment challenge. In *Proceedings of Text Analysis Conference (TAC'09)*, 2009.

Daniel Biś, Maksim Podkorytov, and Xiuwen Liu. Too much in common: Shifting of embeddings in transformer language models and its implications. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 2020.

Xingyu Cai, Jiaji Huang, Yuchen Bian, and Kenneth Church. Isotropy in the contextual embedding space: Clusters and manifolds. In *International Conference on Learning Representations*, 2020.

Daniel Campos. Curriculum learning for language modeling. *arXiv preprint arXiv:2108.02170*, 2021.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, 2017.

Zihan Chen, Hongbo Zhang, Xiaoji Zhang, and Leqi Zhao. Quora question pairs, 2018.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*, 2020.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019.

William B. Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, 2005.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*, 2019.

Jeffrey L. Elman. Finding structure in time. *Cognitive Science*, 14(2):179–211, 1990.

Jeffrey L Elman. Learning and development in neural networks: The importance of starting small. *Cognition*, 48(1):71–99, 1993.

Kawin Ethayarajh. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. 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.

Sabri Eyuboglu, Bojan Karlaš, Christopher Ré, Ce Zhang, and James Zou. Dcbench: A benchmark for data-centric AI systems. In *Proceedings of the Sixth Workshop on Data Management for End-To-End Machine Learning*, 2022.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. Representation degeneration problem in training natural language generation models. In *International Conference on Learning Representations*, 2019.

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, Jason Phang, Laria Reynolds, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, 2021a.

Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021b.

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

Chengyue Gong, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. Frage: Frequency-agnostic word representation. In *Advances in Neural Information Processing Systems*, 2018.

Tao Gui, Jiacheng Ye, Qi Zhang, Zhengyan Li, Zichu Fei, Yeyun Gong, and Xuanjing Huang. Uncertainty-aware label refinement for sequence labeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

Mustafa Hajij, Ghada Zamzmi, Karthikeyan Natesan Ramamurthy, and Aldo Guzman Saenz. Data-centric AI requires rethinking data notion. *arXiv preprint arXiv:2110.02491*, 2021.

Felix Hamborg, Norman Meuschke, Corinna Breitinger, and Bela Gipp. news-please: A generic news crawler and extractor. In *Proceedings of the 15th International Symposium of Information Science*, 2017.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Wei Chen. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*, 2021.

John Hewitt and Christopher D. Manning. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019.

Yizheng Huang, Huaizheng Zhang, Yuanming Li, Chiew Tong Lau, and Yang You. Active-learning-as-a-service: An efficient mlops system for data-centric AI. *arXiv preprint arXiv:2207.09109*, 2022.

Peter Izsak, Moshe Berchansky, and Omer Levy. How to train BERT with an academic budget. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

Lu Jiang, Deyu Meng, Qian Zhao, Shiguang Shan, and Alexander Hauptmann. Self-paced curriculum learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2015.

Nikita Kitaev and Dan Klein. Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018.

Sanjay Krishnan, Jiannan Wang, Eugene Wu, Michael J Franklin, and Ken Goldberg. Activeclean: Interactive data cleaning for statistical modeling. *Proceedings of the VLDB Endowment*, 9(12): 948–959, 2016.

M. Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*, 2010.

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

Anne Lauscher, Ivan Vulić, Edoardo Maria Ponti, Anna Korhonen, and Goran Glavaš. Specializing unsupervised pretraining models for word-level semantic similarity. In *Proceedings of the 28th International Conference on Computational Linguistics*, December 2020.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. Deduplicating training data makes language models better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022.

Yoav Levine, Barak Lenz, Or Dagan, Ori Ram, Dan Padnos, Or Sharir, Shai Shalev-Shwartz, Amnon Shashua, and Yoav Shoham. SenseBERT: Driving some sense into BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proc. of ACL*, 2020.

Conglong Li, Minjia Zhang, and Yuxiong He. Curriculum learning: A regularization method for efficient and stable billion-scale gpt model pre-training. *arXiv preprint arXiv:2108.06084*, 2021.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. Linguistic knowledge and transferability of contextual representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019a.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019b.

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

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Christopher D Manning. Part-of-speech tagging from 97% to 100%: is it time for some linguistics? In *International conference on intelligent text processing and computational linguistics*, 2011.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of english: The penn treebank. *Comput. Linguist.*, 19(2):313–330, jun 1993. ISSN 0891-2017.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

Jiaqi Mu, Suma Bhat, and Pramod Viswanath. All-but-the-top: Simple and effective postprocessing for word representations. In *International Conference on Learning Representations*, 2018.

Koichi Nagatsuka, Clifford Broni-Bediako, and Masayasu Atsumi. Pre-training a BERT with curriculum learning by increasing block-size of input text. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, 2021.

Andrew Ng. Mlops: From model-centric to data-centric AI, 2021.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016.

Laura Pérez-Mayos, Miguel Ballesteros, and Leo Wanner. How much pretraining data do language models need to learn syntax? In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. Language models as knowledge bases? 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.

Ofir Press, Noah A. Smith, and Mike Lewis. Shortformer: Better language modeling using shorter inputs. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021.

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

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*, 2021.

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

Sara Rajaee and Mohammad Taher Pilehvar. A cluster-based approach for improving isotropy in contextual embedding space. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 2021.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016.

Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2018.

Bryan C Russell, Antonio Torralba, Kevin P Murphy, and William T Freeman. Labelme: a database and web-based tool for image annotation. *International journal of computer vision*, 77(1):157–173, 2008.

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

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*, 2022.

Timo Schick and Hinrich Schütze. Rare words: A major problem for contextualized embeddings and how to fix it by attentive mimicking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

Timo Schick and Hinrich Schütze. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 2021.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 2013.

Robyn Speer, Joshua Chin, Andrew Lin, Sara Jewett, and Lance Nathan. Luminosoinsight/wordfreq: v2.2, 2018.

Tomasz Stanislawek, Anna Wróblewska, Alicja Wójcicka, Daniel Ziembicki, and Przemyslaw Biecek. Named entity recognition - is there a glass ceiling? In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, 2019.

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

Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, 2003.

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

Lingxiao Wang, Jing Huang, Kevin Huang, Ziniu Hu, Guangtao Wang, and Quanquan Gu. Improving neural language generation with spectrum control. In *International Conference on Learning Representations*, 2019b.

Xin Wang, Yudong Chen, and Wenwu Zhu. A survey on curriculum learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):4555–4576, 2022.

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

Jason Wei and Kai Zou. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. 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.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2022.

Alexander Wettig, Tianyu Gao, Zexuan Zhong, and Danqi Chen. Should you mask 15% in masked language modeling? *arXiv preprint arXiv:2202.08005*, 2022.

Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

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

Liang Xu, Jiacheng Liu, Xiang Pan, Xiaojing Lu, and Xiaofeng Hou. Dataclue: A benchmark suite for data-centric nlp. *arXiv preprint arXiv:2111.08647*, 2021a.

Zenan Xu, Daya Guo, Duyu Tang, Qinliang Su, Linjun Shou, Ming Gong, Wanjun Zhong, Xiaojun Quan, Daxin Jiang, and Nan Duan. 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 (Volume 1: Long Papers)*, 2021b.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, 2019.

Sangwon Yu, Jongyoon Song, Heeseung Kim, Seongmin Lee, Woo-Jong Ryu, and Sungroh Yoon. Rare tokens degenerate all tokens: Improving neural text generation via adaptive gradient gating for rare token embeddings. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022.

Weizhe Yuan and Pengfei Liu. restructured pre-training. *arXiv preprint arXiv:2206.11147*, 2022.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.

Wei Zhang, Wei Wei, Wen Wang, Lingling Jin, and Zheng Cao. Reducing bert computation by padding removal and curriculum learning. In *2021 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, pp. 90–92, 2021.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, 2015.

A LIST OF CONSTITUENT LABELS

We injected into the corpus with a total of 55 constituent labels defined in Penn Treebank, including:

Labels = [-LRB-, -RRB-, ADJP, ADVP, CONJP, DT, EX, FRAG, FW, INTJ, JJ, JJR, JJS, LS, LST, NAC, NN, NNP, NNPS, NNS, NP, NX, PDT, POS, PRN, PRP, PRP$, PRT, QP, RBR, RBS, RP, RRC, SBAR, SBARQ, SINV, SQ, SYM, TOP, UCP, UH, VB, VBD, VBG, VBN, VBP, VBZ, WDT, WHADJP, WHADVP, WHNP, WHPP, WP, WP$, WRB]

A brief description of these labels can be found in http://surdeanu.cs.arizona.edu/mihai/teaching/ista555-fall13/readings/PennTreebankConstituents.html. After data reconstruction, we treat these labels as normal tokens and enlarge our embedding table, making them involved in the masked language model pre-training:

```
tokenizer_kwarg={"additional_special_tokens":Labels}
tokenizer=AutoTokenizer.from_pretrained(tokenizer_name,**tokenizer_kwarg)
model=AutoModelForMaskedLM.from_pretrained(model_name)
model.resize_token_embeddings(len(tokenizer))
```

B STATISTICS OF DATASETS

Statistics of the datasets are shown in Table 9.

Dataset	Task	#Train	#Dev	#Test	#Label
MNLI	Natural language inference	393k	20k	20k	3
QQP	Paraphrase	364k	40k	391k	2
QNLI	Natural language inference	108k	5.7k	5.7k	2
SST-2	Sentiment	67k	872	1.8k	2
CoLA	Acceptability	8.5k	1k	1k	2
STS-B	Similarity	7k	1.5k	1.4k	1
MRPC	Paraphrase	3.7k	408	1.7k	2
RTE	Natural language inference	2.5k	276	3k	2
CoNLL2003	Named entity recognition	14.9k	3.4k	3.6k	8
SQuAD1.1	Reading comprehension	87.6k	10.5k	9.5k	-
SQuAD2.0	Reading comprehension	130.3k	11.9k	8.9k	-
WSJ POS tagging	Part-of-speech tagging	38.2k	5.5k	5.4k	45
WSJ parsing	Constituency parsing	39.8k	1.7k	2.4k	52

Table 9: Statistics of datasets in our experiments.

C PARAMETER SETTINGS FOR FINE-TUNING

Fine-tuning parameters for GLUE, NER, SQuAD, POS tagging, and parsing are given in Table 10.

	GLUE	NER	SQuAD1.1&2.0
Epochs	{3, 5, 10, 20}	20	2
Learning Rate	{2e-5, 3e-5, 5e-5}	2e-5	3e-5
Batch Size	{16, 32}	16	12
Warmup Ratio	{0.06, 0.1}	-	-
	POS Tagging (fine-tuning)	**POS Tagging (probing)**	**Parsing**
Epochs	20	20	terminated if no improvement on dev set for 60 epochs
Learning Rate	1e-5	2e-3	3e-5
Batch Size	16	{8, 16}	32
Warmup Steps	-	-	160

Table 10: Parameter settings for fine-tuning downstream tasks.

D MORE EXPERIMENTAL RESULTS

D.1 DIFFERENT MODEL SETTINGS

For larger model settings, we pre-train RoBERTa-large model (355M parameters) on the same corpus and compared on GLUE tasks. In particular, we use the recipe from Wettig et al. (2022) for efficient pre-training by using 40% masking ratio with 500K training steps.

Model	**MNLI**	**QQP**	**QNLI**	**SST-2**	**CoLA**	**STS-B**	**MRPC**	**RTE**	*Avg.*
RoBERTa-large	83.9/84.9	87.8	91.5	93.2	55.7	87.4	75.7	64.3	80.4
RoBERTa-large-reimp	83.3/84.1	88.1	91.8	93.6	51.3	**88.0**	75.5	64.9	80.0
RoBERTa-large-CL	**85.5/85.5**	**88.5**	**92.4**	**94.0**	**56.8**	87.4	**80.0**	**66.1**	**81.8**

Table 11: Comparison between larger models with RoBERTa-large setting.

Results on GLUE dev sets are shown in Table 11. Overall, compared with RoBERTa-large-reimp, our method lead to large improvement on CoLA and MRPC, and also giving close performance or minor improvement across other tasks.

Beyond autoencoding-style model like BERT, our method can apply to auto-regressive model like GPT2 where the reconstructed data is used for left-to-right language modeling. Specifically, we use GPT2-base as our backbone and pre-train on the same corpus. We evaluate them on LAMBADA (Paperno et al., 2016), WikiText2 (Merity et al., 2016) and SWAG (Zellers et al., 2018) without any fine-tuning (*i.e.*, zero-shot) using the LM evaluation framework from Gao et al. (2021a).

Model	**LAMBADA**		**WikiText2**			**SWAG**
	ppl.	Acc.	ppl.	byte ppl.	bpb	Acc.
GPT2	40.06	32.54	37.30	1.96	0.97	53.78
GPT2-reimp	36.95	33.46	31.61	1.91	0.93	54.06
GPT2-CL	**32.97**	**35.67**	**30.13**	**1.89**	**0.91**	**54.56**

Table 12: Comparison between auto-regressive language models with GPT2-base setting. ppl: perplexity, bpb: bits per byte.

Results are shown in Table 12. We find that the results of auto-regressive LMs are still in favor of the proposed technique, where better results such as lower perplexity are achieved.

D.2 DIFFERENT CURRICULUM TRAINING STEPS SETTINGS

We investigate the different training steps setting in our four-stage curriculum training, results are shown in Table 13.

Model / CL Settings	**MNLI**	**QQP**	**QNLI**	**SST-2**	**CoLA**	**STS-B**	**MRPC**	**RTE**	*Avg.*
RoBERTa-large-reimp	83.3/84.1	88.1	91.8	93.6	51.3	88.0	75.5	64.9	80.0
RoBERTa-large-CL									
20-20-20-40 (%)	85.5/85.5	88.5	92.4	94.0	56.8	87.4	80.0	66.1	81.8
10-20-30-40 (%)	85.5/85.8	88.6	92.1	93.6	56.2	87.6	79.5	66.8	81.7
25-25-25-25 (%)	84.4/84.8	87.8	91.9	93.2	56.0	86.8	77.7	67.1	81.0
10-10-10-70 (%)	84.2/84.6	87.9	91.6	93.6	55.2	87.0	77.5	66.1	80.8

Table 13: Comparison on different four-stage curriculum training schedule.

Overall, the proposed curriculum training can improve the performance stably. We find that different schedule settings affect the results, where a moderate amount of training using both mixup and raw training data is necessary. For example, training with only 30% steps in the first three stages with the mixup data is not much sufficient. In future work, techniques such as self-paced learning (Kumar et al., 2010; Jiang et al., 2015) can also be considered for setting a better schedule.

E MEASURE THE ISOTROPY OF REPRESENTATION SPACE

We measure the isotropy of representation space using the metric defined by Mu et al. (2018), which is also used for measuring recent language models (Rajaee & Pilehvar, 2021):

$$I(W) = \frac{\min_{u \in U} Z(u)}{\max_{u \in U} Z(u)}, \tag{2}$$

where W is the set of representation vectors, U is the set of eigenvectors of $W^\top W$, $Z(u)$ is a partition function define in Arora et al. (2016):

$$Z(u) = \sum_{w_i \in W} \exp(u^\top w_i). \tag{3}$$

The perfect isotropic space would have $I(W)$ close to 1. We calculated the $I(W)$ scores for BERT, BERT-reimp, and BERT-CL, the results are shown in Table 14.

	BERT	BERT-reimp	BERT-CL
$I(W)$	1.05e-5	6.15e-7	**1.15e-4**

Table 14: Measuring the isotropy of representation space of different models.

We find that the $I(W)$ score of BERT-reimp is lower that that of BERT, the reason can be that the original BERT leverage multi-task training (masked language modeling and next sentence prediction). Compared with these two models, BERT-CL gives a higher $I(W)$ score of 1.15e-4, showing that curriculum training can lead to a more isotropic representation space.

F MORE EXAMPLES FOR THE MOST SIMILAR WORDS

More examples for the most similar words to each constituent label are shown in Table 15.

Lables	**High (Top 1∼500)**	**Medium (Top 500∼3000)**	**Low (Top 3000∼)**
NN	*light, service, group*	*present, mark, mission*	*seed, concentration, penalty*
NNP	*from, for, the*	*present, steel, opposition*	*clay, audio, miniature*
NNPS	*others, children, team*	*crown, lights, figures*	*minors, blues, blacks*
VB	*keep, tell, let*	*promote, kill, develop*	*convert, recover, minimize*
VBP	*are, were, am*	*re, ve, themselves*	*traded, overlap, dwell*
VBN	*taken, given, done*	*broken, combined, dropped*	*torn, risen, divided*
VBD	*took, had, could*	*spoke, closed, ran*	*tore, rolled, slid*
VBG	*taking, saying, looking*	*passing, putting, turning*	*advancing, returning, protecting*
WHADVP	*where, when, why*	*whenever, whom, till*	*wherein, whereby, wherever*
WRB	*how, where, whether*	*whenever, lets, forgot*	*wherever, whereby, wherein*
RBS	*far, ago, least*	*highest, worst, anywhere*	*hardest, fastest, shortest*
RBR	*less, oh, ago*	*wanna, faster, longer*	*sooner, hotter, warmer*
JJS	*best, least, most*	*worst, highest, largest*	*lowest, deepest, smallest*
JJR	*better, more, less*	*greater, stronger, larger*	*warmer, happier, thicker*

Table 15: More examples for the most similar words to the constituent labels. NN: noun; NNP: proper noun, singular; NNPS: proper noun, plural; VB: verb; VBP: non-3rd person singular present; VBN: past participle; VBD: past tense; VBG: gerund or present participle; WHADVP: wh-adverb phrase; WRB: wh-adverb; RBS: adverb, superlative; RBR: adverb, comparative; JJS: adjective, superlative; JJR: adjective, comparative.