Identifying the Limits of Cross-Domain Knowledge Transfer for Pretrained Models

Zhengxuan Wu
Stanford University
wuzhengx@stanford.edu

Nelson F. Liu
Stanford University
nfliu@cs.stanford.edu

Christopher Potts
Stanford University
cgpotts@stanford.edu

Abstract

There is growing evidence that pretrained language models improve task-specific fine-tuning even where the task examples are radically different from those seen in training. We study an extreme case of transfer learning by providing a systematic exploration of how much transfer occurs when models are denied any information about word identity via random scrambling. In four classification tasks and two sequence labeling tasks, we evaluate LSTMs using GloVe embeddings, BERT, and baseline models. Among these models, we find that only BERT shows high rates of transfer into our scrambled domains, and for classification but not sequence labeling tasks. Our analyses seek to explain why transfer succeeds for some tasks but not others, to isolate the separate contributions of pretraining versus fine-tuning, to show that the fine-tuning process is not merely learning to unscramble the scrambled inputs, and to quantify the role of word frequency. Furthermore, our results suggest that current benchmarks may overestimate the degree to which current models actually understand language.

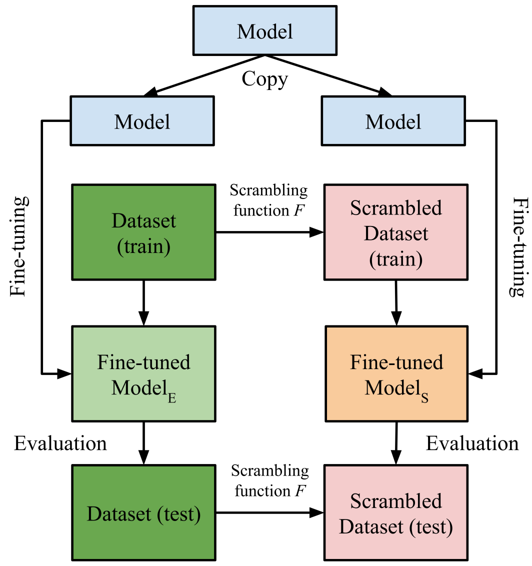

Figure 1: An overview of our experiment paradigm. Starting with a model (e.g., pretrained BERT, GloVe-initialized LSTM, etc.), we copy it and fine-tune it on the regular and scrambled train set using a scrambling function F. The model is then evaluated on regular and scrambled test sets. Our paper explores different options for F and a number of variants of our models to try to quantity the amount of transfer and identify its sources.

1 Introduction

Fine-tuning pretrained language models has proven to be highly effective across a wide range of NLP tasks; the leaderboards for standard benchmarks are currently dominated by models that adopt this general strategy (Rajpurkar et al., 2016, 2018; Wang et al., 2018; Yang et al., 2018; Wang et al., 2019). Recent work has extended these findings in even more surprising ways: Artetxe et al. (2020), Karthikeyan et al. (2019), and Tran (2020) find evidence of transfer between natural languages, and Papadimitriou and Jurafsky (2020) show that pretraining language models on non-linguistic data such as music and computer code can improve test performance on natural language.

Recently, Tamkin et al. (2020) show that BERT's performance on downstream GLUE tasks suffers only marginally even if some layers are reinitialized before fine-tuning, and Gauthier and Levy (2019), Zanzotto et al. (2020), Pham et al. (2020), and Sinha et al. (2021) show that BERT-like models are largely insensitive to word order changes. In this work, we extend this line of research by providing a systematic exploration of how much cross-domain transfer we see when the model is denied any information about word identity.

Figure 1 gives an overview of our core experimental paradigm: starting with two identical copies of a single pretrained model for English, we fine-tune one on English examples and the other on scrambled English sentences, using a scrambling function F (Section 3), and then we evaluate the resulting models. We apply this paradigm to four classification tasks and two sequence modeling tasks, and we evaluate bag-of-words baselines, LSTMs with GloVe initialization and rich attention mechanisms, and BERT. Our central finding is that, for BERT, high rates of transfer occur on classification tasks, but not sequence labeling tasks

To better understand why such transfer is successful for some tasks but not others, we pursue a number of hypotheses. First, we assess whether the transfer occurs if only word identities are scrambled among words with similar frequencies. Second, we assess whether our matching methods might actually be inserting semantic consistency into the scrambling process by matching synonyms. Third, we analyze the learning dynamics behind such transfer by studying the effects of model pretraining.

Our findings suggest that performance on existing tasks may be less informative than previously thought about the degree to which a model understands language. Our pretrained models transfer knowledge in our tasks even when they are denied any information about word identity. Thus, a large percentage of their success might trace to factors that have nothing to do with communication or understanding. After all, our scrambled data do not have the semantics of English, or indeed of any language.

2 Related work

2.1 Studies of Why Transfer Happens

There are diverse efforts underway to more deeply understand why transfer occurs. Probing tests often involve fitting supervised models on internal representations in an effort to determine what they encode. Such work suggests that BERT representations encode non-trivial information about morphosyntax and semantics (Tenney et al., 2019; Liu et al., 2019; Hewitt and Manning, 2019; Manning et al., 2020) and perhaps weakly encode world knowledge such as relations between entities (Da and Kasai, 2019; Petroni et al., 2019), but that they contain relatively little information about pragmatics or role-based event knowledge (Ettinger, 2020). Newer feature attribution methods (Sundararajan et al., 2017) and intervention methods (McCoy et al., 2019; Vig et al., 2020; Geiger et al., 2020) are corroborating these findings while also yielding a picture of the internal causal dynamics of these models.

Another set of strategies for understanding transfer involves modifying network inputs or internal representations and studying the effects of such changes on task performance, as in the above-cited work by Tamkin et al. (2020), Gauthier and Levy (2019), Zanzotto et al. (2020), Pham et al. (2020), and Sinha et al. (2021).

Scrambling Method	Sentence
Original English (No Scrambling)	"the worst titles in recent cinematic **history**"
Similar Frequency	"a engaging semi is everyone dull **dark**"
Random	"kitsch theatrically tranquil andys loaf shorty **lauper**"

Table 1: An example from the SST-3 dataset and its two scrambled variants.

2.2 Extreme Cross-Domain Transfer

Cross-domain transfer is not limited to monolingual cases (Karthikeyan et al., 2019). With modifications to its tokenizer, English-pretrained BERT improves performance on downstream multilingual NLU tasks (Artetxe et al., 2020; Tran, 2020). Papadimitriou and Jurafsky (2020) show that pretraining language models on structured non-linguistic data (e.g., MIDI music or Java code) improves test performance on natural language. Our work complements and advances these efforts along two dimensions. First, we challenge models with extremely ambitious cross-domain settings and find that BERT shows a high degree of transfer, and we conduct a large set of follow-up experiments to help identify the sources and limitations of such transfer.

3 Experimental Paradigm

We now describe the evaluation paradigm summarized in Figure 1 (Section 3.1), with special attention to the scrambling functions F that we consider (Sections 3.2–3.3).

3.1 Evaluation Pipeline

Figure 1 shows our main evaluation paradigm for testing the transfer abilities of a model without word identity information. On the left side, we show the classic fine-tuning pipeline (i.e., we fine-tune on the original English training set and evaluate on the original English test set). On the right side, we show our new evaluation pipeline: starting from a single model, we (1) fine-tune it with a corrupted training split where regular English word identities are removed and then (2) evaluate the model on a version of the evaluation set that is corrupted in the same manner. The paradigm applies equally to models without any pretraining and with varying degrees of pretraining for their model parameters.

3.2 Scrambling with Similar Frequency

To remove word identities, we scrambled each sentence in each dataset by substituting each word w with a new word w' in the vocabulary of the dataset. For Scrambling with Similar Frequency, we use the following rules:

1. w and w' must have the same sub-token length according to the BERT tokenizer; and
2. w and w' must have similar frequency.

The first rule is motivated by the concern that sub-token length may correlate with word frequency, given that rarer and longer words may be tokenized into more sub-tokens. The second rule is the core of the procedure. The guiding idea is that word frequency is often reflected in learned embeddings (Gong et al., 2018), so this scrambling procedure might preserve useful information and thus help to identify the source of transfer. Table 1 shows an example, and our supplementary materials provide details on the matching algorithm and additional examples of scrambled sentences.

3.3 Random Scrambling

To better understand the role of frequency in domain transfer, we also consider a word scrambling method that does not seek to match word frequencies. For this, we simply shuffle the vocabulary and match each word with another random word in the vocabulary without replacement.[1] We include the distributions of the difference in frequency for every matched word pair in our supplementary materials, to show that each word is paired with a new word with drastically different frequency in the dataset.

4 Models

In this section, we describe the models we evaluated within our paradigm. Our supplementary materials provide additional details about how the models were designed, optimized, and evaluated.

BERT For our BERT model (Devlin et al., 2019), we import weights from the pretrained BERT-base model through the HuggingFace `transformers` library (Wolf et al., 2020). For sequence classification tasks, we append a classification head after the `[CLS]` token embedding in the last layer of the BERT model. If an input example contains a pair of sentences, we concatenate them using a `[SEP]` token in between. For sequence labeling tasks, we append a shared classification head to each token embedding in the last layer of the BERT model.

Our supplementary materials provide results for `DeBERTa` models (He et al., 2021) as well.

LSTM We contextualize our results by comparing them against a strong LSTM-based model (Hochreiter and Schmidhuber, 1997). We lowercase each input sentence and tokenize it by separating on spaces and punctuation. We then use 300-dimensional GloVe embeddings (Pennington et al., 2014)[2] as inputs to a single-layer recurrent neural network with LSTM cells, with a hidden size of 64. We use dot-product attention (Luong et al., 2015) to formulate a context vector for each sentence. Finally, we pass the context vector through a multilayer perceptron (MLP) layer with a hidden size of 64 to get the final prediction. For an input example with a pair of sentences, we concatenate two sentences together with a separator token before feeding them into our LSTM encoder. For sequence labeling tasks, we directly feed the hidden state at each position to the MLP layer to get the final prediction.

Bag-of-Words (BoW) Model We compare against a BoW classifier, which provides an estimate of model performance when only word co-occurrence information is available. For each sentence in a dataset, we first formulate a BoW vector that uses unigram representations. Then, we feed the BoW vector through a softmax classifier. For examples with a pair of sentences, we create two BoW vectors for each sentence, and concatenate them together before feeding them into the linear layer for predicting labels. For sequence labeling tasks, we use conditional random fields (Lafferty et al., 2001) with character-level unigram BoW features.

Dummy Model We include a random classifier that generates predictions randomly proportional to the

[1] We also tried to pair words by the reverse order of frequencies, which yielded similar results, so we report only random scrambling results here.

[2] We use the Common Crawl cased version: `http://nlp.stanford.edu/data/glove.840B.300d.zip`

Dataset	Standard Models (Train and Test on English)				Scrambled Models (Train and Test on Scrambled English)			
					BERT-Scrambled		LSTM-Scrambled	
	BERT	LSTM	BoW	Dummy	Similar Frequency	Random	Similar Frequency	Random
SST-3	.71 (.02)	.62 (.01)	.59 (.00)	.33 (.02)	.65 (.01)	.64 (.02)	.57 (.02)	.56 (.02)
SNLI	.91 (.02)	.78 (.02)	66 (.02)	.33 (.01)	.84 (.01)	.82 (.02)	.72 (.00)	.71 (.01)
QNLI	.91 (.02)	.68 (.02)	.62 (.01)	.50 (.01)	.82 (.01)	.79 (.02)	.62 (.01)	.61 (.01)
MRPC	.86 (.01)	.72 (.02)	.70 (.02)	.50 (.02)	.82 (.02)	.78 (.02)	.69 (.00)	.68 (.00)
EN-EWT	.97 (.01)	.85 (.02)	.65 (.01)	.09 (.01)	.86 (.01)	.81 (.02)	.80 (.01)	.72 (.01)
CoNLL-2003	.95 (.01)	.75 (.01)	.28 (.02)	.02 (.01)	.74 (.01)	.72 (.02)	.61 (.02)	.56 (.01)

Table 2: Model performance results for models trained on original English and on scrambled English. Standard deviations are reported for all entries.

class distribution of the training set. We use this model to further contextualize our results.

5 Tasks

Sequence Classification We select four NLU datasets for sequence classification. We consider sentiment analysis (SST-3; Socher et al., 2013), where SST-3 is a variant of the Stanford Sentiment Treebank with positive/negative/neutral labels; we train on the phrase- and sentence-level sequences in the dataset and evaluate only on its sentence-level labels. Additionally, we include natural language inference (QNLI; Demszky et al., 2018 and SNLI; Bowman et al., 2015) and paraphrase (MRPC; Dolan and Brockett, 2005). QNLI is derived from a version of the Stanford Question Answering Dataset (SQuAD; Rajpurkar et al. 2016). For sequence classification tasks, we use Macro-F1 scores for SST-3, and accuracy scores for the other NLU tasks.

Our supplementary materials provide results for the full GLUE benchmark (Wang et al., 2018).

Sequence Labeling In contrast to sequence classification, where the classifier only considers the `[CLS]` token of the last layer and predicts a single label for a sentence, sequence labeling requires the model to classify all tokens using their contextualized representations. We select two datasets covering distinct tasks: part-of-speech detection (POS) and named entity recognition (NER). We used the Universal Dependencies English Web Treebank (EN-EWT; Silveira et al. 2014) for POS and CoNLL-2003 (Tjong Kim Sang and De Meulder, 2003) for NER. For sequence labeling tasks, we used Micro-F1 (i.e., accuracy with full labels) for POS and F1 scores for NER.

6 Results

In this section, we analyze the fine-tuning performance of BERT on scrambled datasets. Table 2 shows performance results. We focus for now on the results for Scrambling with Similar Frequency. Additionally, we also include baseline models trained with original sentences for comparison purposes. When training models on each task, we select models based on performance on the dev split during fine-tuning. We report average performance results across runs with three different random seeds.

6.1 Sequence Classification

Comparing the second column (BERT models trained and tested on English) with the sixth column (BERT models trained and tested on Scrambled English with Similar Frequency Scrambling) in Table 2, we see that BERT maintains strong performance for all sequence classification tasks even when the datasets are scrambled. More importantly, we find that BERT fine-tuned with a scrambled dataset performs significantly better than the LSTM model (with GloVe embeddings) trained and evaluated on standard English data

For example, on the MRPC task, BERT evaluated with scrambled data experiences a less than 5% performance drop, and shows significantly better performance than the best LSTM model (a 13.9% improvement). BERT evaluated with scrambled QNLI experiences the biggest drop (a 9.89% decrease). However, this still surpasses the best LSTM performance by a large margin (a 20.6% improvement).

Table 2 also presents performance results for other baseline models, which can be used to assess the intrinsic difficulty of each task. Our results suggest that BERT models fine-tuned with scrambled tasks remain very strong across the board, and they remain stronger than the best LSTM baseline models (those trained and tested on regular English) in all the classification tasks.

The overall performance of the LSTM models is worth further attention. The LSTMs are far less successful at our tasks than the BERT models. However, it seems noteworthy that scrambling does not lead to catastrophic failure for these models. Rather, they maintain approximately the same performance in the scrambled and unscrambled conditions. This might seem at first like evidence of some degree of transfer. However, as we discuss in Section 7.3, the more likely explanation is that the LSTM is simply being retrained more or less from scratch in the two conditions.

6.2 Sequence Labeling

For a more complex setting, we fine-tuned BERT on sequence labeling tasks, and evaluated its transfer abilities without word identities (i.e., using datasets that are scrambled in the same way as in our sequence classification tasks). The bottom two rows of Table 2 show performance results for these tasks, where the goal of the BERT model is to classify every token correctly. As shown in Table 2, BERT experiences a significant drop when evaluated with a scrambled dataset for a sequence labeling task. For LSTMs trained with scrambled sequence labeling tasks, we also observe bigger drops compared with sequence classification tasks. For CoNLL-2003, the LSTM with GloVe embeddings drops from its baseline counterpart (a 18.7% decrease). Our results suggest that transfer learning without word identities is much harder for sequence labeling tasks. One intuition is that sequence labeling tasks are more likely to rely on word identities given the fact that classification (i.e., labeling) is at the token-level.

7 Analysis

7.1 Frequency Effects

Preserving word frequencies during scrambling may lead to higher performance when training and evaluating on scrambled datasets. To assess how much of the observed transfer relates to this factor, we can compare Scrambling with Similar Frequency (`SSF`) with Random Scrambling (`RS`), as described in Section 3. As shown in Table 2, performance drops slightly if we use `RS`. For sequence classification tasks, `RS` experiences 1–5% drops in performance compared with `SSF`. For sequence labeling tasks, the difference is slightly larger: about 2–6%. This suggests that word frequency is indeed one of the factors that affects transfer, though the differences are relatively small, indicating that this is not the only contributing factor. This is consistent with similar findings due to Karthikeyan et al. 2019 for multilingual BERT.

7.2 Does Scrambling Preserve Meaning?

Another potential explanation is that our scrambling methods tend to swap words that are predictive of the same labels. For example, when we are substituting words with similar frequencies in SST-3, "good" may be swapped with "great" since they may have similar frequencies in a sentiment analysis dataset. To rule this out, we conducted zero-shot evaluation experiments with our BoW model on sequence classification tasks. The rationale here is that, to the extent that our swapping preserved the underlying connection between features and class labels, this should show up directly in the performance of the BoW model. For example, just swapping of "good" for "great" would hardly affect the final scores for each class. If there are a great many such invariances, then it would explain the apparent transfer.

Figure 2 shows the zero-shot evaluation results of our BoW model on all sequence classification datasets. Our results show that both scrambling methods result in significant performance drops, which suggests that word identities are indeed destroyed by our procedure, which again shines the spotlight on BERT as the only model in our experiments to find and take advantage of transferable information.

7.3 Transfer or Simple Retraining?

Our results on classification tasks show that English-pretrained BERT can achieve high performance when fine-tuned and evaluated on scrambled data. Is this high performance uniquely enabled by transfer from BERT's pretrained representations, or is BERT simply re-learning the token identities from its scrambled fine-tuning data?

To distinguish between these two hypotheses, we first examine whether randomly-initialized BERT models can also achieve high performance when fine-tuned and evaluated on scrambled data. We study models of varying capacity by modulating the number of BERT Transformer blocks. We use datasets scrambled with `SSF`.

Dataset	LSTM-Baseline	LSTM-Scrambled Similar Frequency GloVe	LSTM-Scrambled Similar Frequency No GloVe
SST-3	.62 (.01)	.57 (.02)	.58 (.01)
SNLI	.78 (.02)	.72 (.00)	.71 (.00)
QNLI	.68 (.02)	.62 (.01)	.61 (.01)
MRPC	.72 (.02)	.69 (.00)	.69 (.00)
EN-EWT	.85 (.02)	.80 (.01)	.79 (.01)
CoNLL-2003	.75 (.01)	.61 (.02)	.60 (.01)

Table 3: Performance results for LSTM models trained on regular English and on English with Scrambling with Similar Frequency, with GloVe embeddings and with randomly initialized embeddings.

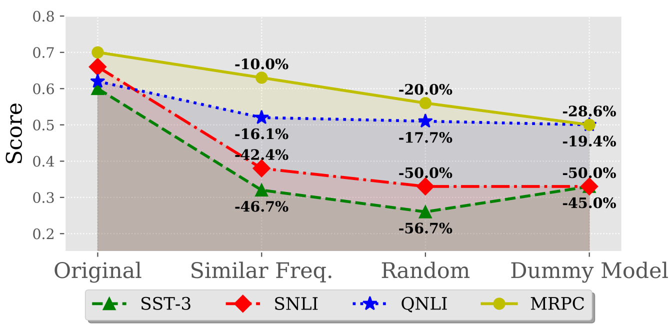

Figure 2: Zero-shot evaluation with the Bag-of-Word (BoW) model on scrambled datasets and the dummy model. Numbers are the differences between the current points and the first points in percentages.

We compare these varying-depth randomly-initialized models against BERT models pretrained on English. To modulate the capacity of these pretrained models, we progressively discard the later Transformer layers (i.e., we make predictions from intermediate layers). Comparing these models is a step toward disentangling the performance gains of pretraining from the performance gains relating to model capacity.

Figure 3 summarizes these experiments. The red line represents our fine-tuning results, across different model sizes. The shaded area represents the performance gain from pretraining when training and testing on scrambled data. Pretraining yields consistent gains across models of differing depths, with deeper models seeing greater gains.

For sequence labeling tasks, the patterns are drastically different: the areas between the two lines are small. Since the randomly-initialized and pretrained models achieve similar performance when fine-tuned and tested on scrambled data, pretraining is not beneficial. This suggests that BERT hardly transfers knowledge when fine-tuned for sequence labeling with scrambled data.

Table 3 shows our results when training LSTMs without any pretrained embeddings. Unlike with BERT, GloVe initialization (a pretraining step) hardly impacts model performance across all tasks. Our leading hypothesis here is that the LSTMs may actually relearn all weights without taking advantage of pretraining. All of our LSTM models have parameter sizes around 1M, whereas the smallest BERT model (i.e., with a single Transformer layer) is around 3.2M parameters. Larger models may be able to rely more on pretraining.

Overall, these results show that we do see transfer of knowledge, at least for classification tasks, but that there is variation between tasks in how much transfer actually happens.

7.4 Assessing Transfer with Frozen BERT Parameters

We can further distinguish the contributions of pretraining versus fine-tuning by freezing the BERT parameters and seeing what effect this has on cross-domain transfer. Ethayarajh (2019) provides evidence that early layers are better than later ones for classifier fine-tuning, so we explore the effects of this freezing for all the layers in our BERT model. We use datasets scrambled with SSF.

As shown in Figure 4, performance scores drop significantly if we only fine-tune the classifier head and freeze the rest of the layers in BERT, across three of our tasks. However, we find that performance scores change significantly depending on which layer we append the classifier head to. Consistent with Ethayarajh's findings, contextualized embeddings in lower layers tend to be more predictive. For example, if we freeze BERT weights and use the contextualized embeddings from the second layer for SST-3, the model reaches peak performance compared with contextualized embeddings from other layers. More importantly, the trend of the green line follows the red line in Figure 4, especially for SST-3 and QNLI. The only exception is MRPC, where the red line plateaus but the green line keeps increasing. This could be an artifact of the size of the dataset, since MRPC only contains around 3.7K training examples. Our results suggest that pretrained weights in successive self-attention layers provide a good initial point for the fine-tuning process.

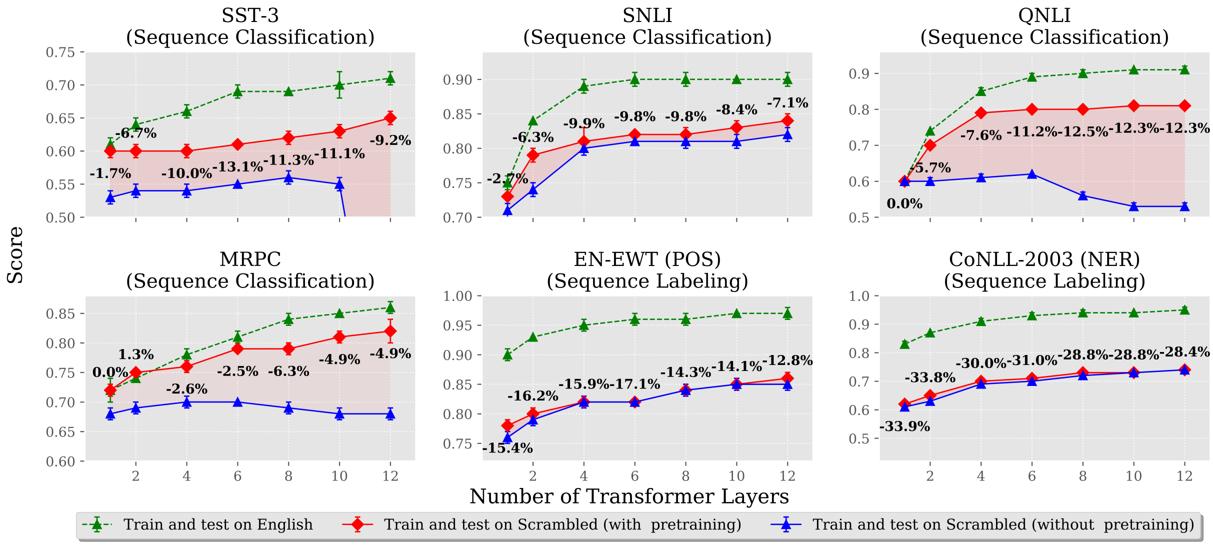

Figure 3: Performance results when fine-tuning end-to-end for different number of Transformer layers. Annotated numbers are the differences between the red lines and the green lines in percentages. Scoring for each task is defined in Section 5.

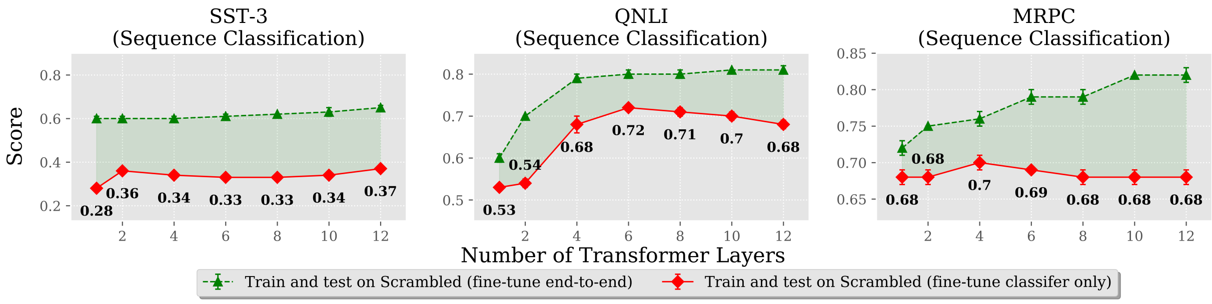

Figure 4: Performance results when fine-tuning only the classifier head by freezing all preceeding layers in BERT (red line) vs. fine-tuning end-to-end, which includes the classifier head and BERT with different numbers of layers (green line). Numbers are scores for the red lines. Scoring for each task is defined in Section 5.

7.5 Probing for Word Identity Reassociations

We further investigate the learning dynamics of our fine-tuned models. Specifically, we study whether our fine-tuned models reassociate word identities with tokens for our sequence classification tasks. To do this, we measure the cosine similarities between words and their scrambled counterparts before and after the fine-tuning process.[3] To the extent that these similarities are increased after fine-tuning, we have evidence that fine-tuning has learned to ressociate words with their scrambled counterparts. We use datasets scrambled with SSF.

We find essentially no evidence for such reassociations. As shown in Figure 5, the correlation distributions before fine-tuning and after are extremely similar. This suggests that our fine-tuned models rarely reassociate word identities in the embedding layer.

To push this analysis a step further, we probe whether word identities are recovered through Transformer layers by adapting the probing method with control task from Hewitt and Liang (2019). Formally, we use an MLP classifier to predict the word identity for w using the contextualized hidden representations of its scrambled counterpart w'. For our control task, we ask the probe to predict random word identities. The difference in performance between these two conditions is know as *selectivity*, and it estimates the degree to which the word identities are recoverable, taking the power of the probe model into account. As shown in Figure 5, our results suggest that relatively little information about the scrambling map is latent in these representations, across tasks and model layers.

[3]We only consider shared words in the model vocabulary and our scrambling maps, which includes 30% of words in the model vocabulary.

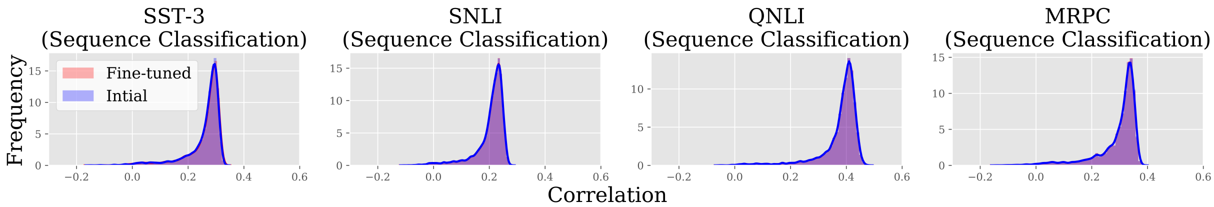

Figure 5: Correlations between cosine similarities of word embeddings before fine-tuning v.s. fine-tuning with scrambled datasets. Measurements of correlations are defined in Section 7.5.

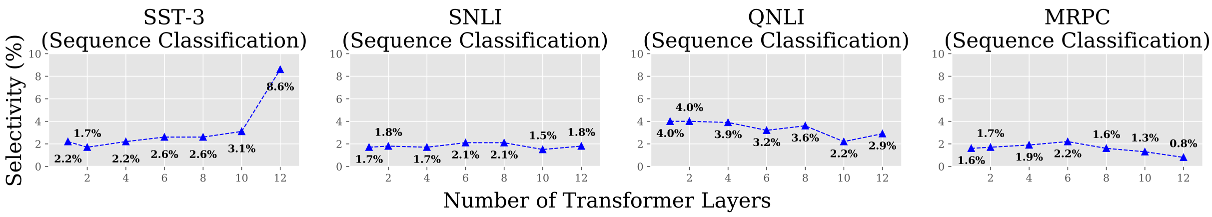

Figure 6: Accuracy of word identity probes when applied to hidden states of each layer comparing to the control task introduced by Hewitt and Liang (2019). Measurements of accuracies are defined in Section 7.5.

8 Conclusion

In this paper, we propose an evaluation pipeline for pretrained models by testing their transfer abilities when they are denied all information about word identity. Specifically, we take an English pretrained BERT off-the-shelf and fine-tune it with a scrambled English dataset. We conduct analyses across six tasks covering both classification and sequence labeling. By evaluating performance against multiple baselines, we aim to assess where BERT can transfer knowledge even without word identities. We find considerable transfer for BERT as compared to even powerful baselines, but only for classification tasks.

What is the source of successful cross-domain transfer with BERT? We find that word frequency contributes, but only to a limited extent: scrambling with matched word frequencies consistently outperforms scrambling with unmatched word frequencies, but transfer still occurs robustly even with random scrambling. We are also able to determine that both pretraining and fine-tuning are important and interacting factors in this transfer; freezing BERT weights during task-specific training leads to much less transfer, but too much task-specific training erodes the benefits of pretraining and in turn reduces the amount of transfer observed.

These analyses begin to piece together a full account of these surprising transfer results for BERT, but they do not fully explain our experimental results. Recent literature suggests at least two new promising avenues to explore. First, Sinha et al. (2021) seek to help characterize the rich distributional prior that models like BERT may be learning, which suggests that higher-order notions of frequency play a significant role in transfer. Second, the findings of Ethayarajh (2019) may be instructive: through successful layers, BERT seems to perform specific kinds of dimensionality reduction that help with low-dimensional classification tasks. Our results concerning layer-wise variation are consistent with this.

Our results are also highly relevant to questions of benchmarking in NLP. It is widely assumed that the benchmark tasks we considered here can help illuminate the capacity of modern NLP systems to process and understand language. However, in our experiments, fine-tuned BERT models are successful at these tasks even in scrambled conditions that render all the examples meaningless, which should lead us to think critically about whether success in the usual unscrambled conditions is reliable evidence of understanding.

References

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637.

Samuel Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical*

Methods in Natural Language Processing, pages 632–642.

Jeff Da and Jungo Kasai. 2019. Cracking the contextual commonsense code: Understanding commonsense reasoning aptitude of deep contextual representations. *EMNLP 2019*, page 1.

Dorottya Demszky, Kelvin Guu, and Percy Liang. 2018. Transforming question answering datasets into natural language inference datasets.

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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Kawin Ethayarajh. 2019. 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)*, pages 55–65.

Allyson Ettinger. 2020. What bert is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

Jon Gauthier and Roger Levy. 2019. Linking artificial and human neural representations of language. 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)*, pages 529–539, Hong Kong, China. Association for Computational Linguistics.

Atticus Geiger, Kyle Richardson, and Christopher Potts. 2020. Neural natural language inference models partially embed theories of lexical entailment and negation. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 163–173, Online. Association for Computational Linguistics.

Chengyue Gong, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2018. Frage: Frequency-agnostic word representation. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

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

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control 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)*, pages 2733–2743.

John Hewitt and Christopher D Manning. 2019. 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)*, pages 4129–4138.

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

K Karthikeyan, Zihan Wang, Stephen Mayhew, and Dan Roth. 2019. Cross-lingual ability of multilingual bert: An empirical study. In *International Conference on Learning Representations*.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, page 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Nelson F Liu, Matt Gardner, Yonatan Belinkov, Matthew E Peters, and Noah A Smith. 2019. 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)*, pages 1073–1094.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421.

Christopher D Manning, Kevin Clark, John Hewitt, Urvashi Khandelwal, and Omer Levy. 2020. Emergent linguistic structure in artificial neural networks trained by self-supervision. *Proceedings of the National Academy of Sciences*, 117(48):30046–30054.

R. Thomas McCoy, Tal Linzen, Ewan Dunbar, and Paul Smolensky. 2019. RNNs implicitly implement tensor product representations. In *In Proceedings of the 7th International Conference on Learning Representations*, New Orleans, USA.

Isabel Papadimitriou and Dan Jurafsky. 2020. Learning music helps you read: Using transfer to study linguistic structure in language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6829–6839.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. 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)*, pages 2463–2473.

Thang M Pham, Trung Bui, Long Mai, and Anh Nguyen. 2020. Out of order: How important is the sequential order of words in a sentence in natural language understanding tasks? *arXiv preprint arXiv:2012.15180*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. 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)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Christopher D. Manning. 2014. A gold standard dependency corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*.

Koustuv Sinha, Robin Jia, Dieuwke Hupkes, Joelle Pineau, Adina Williams, and Douwe Kiela. 2021. Masked language modeling and the distributional hypothesis: Order word matters pre-training for little. ArXiv:2104.06644.

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*, pages 1631–1642.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328, International Convention Centre, Sydney, Australia. PMLR.

Alex Tamkin, Trisha Singh, Davide Giovanardi, and Noah Goodman. 2020. Investigating transferability in pretrained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1393–1401, Online. Association for Computational Linguistics.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. Bert rediscovers the classical nlp pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. 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*, pages 142–147.

Ke Tran. 2020. From english to foreign languages: Transferring pre-trained language models. *arXiv preprint arXiv:2002.07306*.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. Causal mediation analysis for interpreting neural nlp: The case of gender bias.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Fabio Massimo Zanzotto, Andrea Santilli, Leonardo Ranaldi, Dario Onorati, Pierfrancesco Tommasino, and Francesca Fallucchi. 2020. Kermit: Complementing transformer architectures with encoders of explicit syntactic interpretations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 256–267.