Semantic-Augment: Augmenting the Semantic Space of Transformers Improves Generalization

Stanford CS224N Custom Project

Emirhan Kurtulus
Department of Computer Science
Stanford University
emirhank@stanford.edu

Abstract

Self-supervised pretraining has been a critical component for quality of the natural language processing methods. Because of the abundance of text data, the simple noising-denoising paradigm allowed the models to be scaled to trillions of parameters. This paradigm can be extended by adding consistency maximization and noise invariance. In this work, we propose a novel and simple framework to improve representation learning in language model pretraining by maximizing the similarity between two noised views of the same sequence in addition to the conventional masked language modeling loss. The proposed method improves performance on downstream tasks (0.9% average GLUE score, and 1.1% SWAG accuracy), while enlarging the representation space (10 to 20% higher feature alignment) of the model without incurring additional computational costs, and can be generalized to both encoder-only, encoder-decoder, and decoder-only model pretraining.

1 Key Information to include

- Mentor: Ekin Dogus Cubuk (Google Brain)
- External Collaborators: N/A
- Sharing project: N/A

2 Introduction

Large language models have been one of the driving factors in recent deep learning advances [1, 2, 3, 4]. Since the introduction of the Transformer architecture [5], large language models (LLMs) have been all we need. Language models show great performance on a variety of tasks, from question answering [6] to common sense reasoning [7]. One of the main reasons for its effectiveness on a wide range of tasks [8] is the pre-training phase, in which the model can learn from a huge corpus of up to 500 billion tokens of unlabeled text [9]. The independence from labeled data allows training trillion-scale models [10].

Although they offer great advantages due to their performance on downstream tasks and their zero-shot generalization capabilities, they still have various problems with language understanding, ranging from representation deficiency[11] to alignment problems with human language understanding [12]. These problems can be mitigated by improving the pre-training phase to teach LLMs better language representations and improve their generalization.

For this purpose, we propose a new pretraining framework, which we call Sementic-Augment, that is easily applicable to encoder-only models [13], decoder-only models [14], and encoder-decoder models [15]. Here, we combine the advantages of denoising approaches with consistency regularization through noise invariance. Our framework performs two forward passes for different noised views of the same sequence, and applies feature similarity loss to these views in addition to the conventional

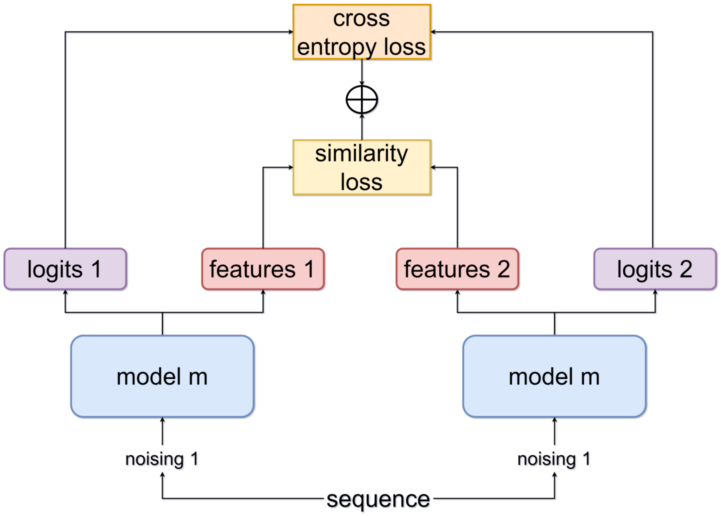

Figure 1: Semantic-Augment framework.

language modeling loss. Semantic-Augment improves the generalization of the models to downstream tasks (+0.9% GLUE score and 1.1% SWAG accuracy), while expanding the feature space of the models (approximately 10% higher alignment on STSB) and reducing representation deficiency.

3 Background / Related Work

Many studies focused on pretraining different LLM architectures. Therefore, the field is largely dominated by the Transformer architecture [5]. As one of the first works on pretraining Transformers on-scale, BERT [13] proposes training an encoder-only model to denoise the masked words in a given sentence while predicting, for given two sentences, if the second entails the first. This approach is later extended by BART [16] to encoder-decoder models by adding more noising strategies to the pretraining stage while using a unidirectional encoder. They show that under right conditions, a unidirectional encoder is as good as bidirectional for representation learning. Later, A whole series of models for the sole purpose of next-token prediction were proposed by the GPT family [14, 17, 1]. GPT family shows us that scale enables many features such as few-shot learning and zero-shot task generalization which motivated a set of works on scaling laws [18, 9], instruction tuning [19, 20], and in-context learning [21, 22].

Another set of works is on sequence-level representation learning where the goal is to learn the best features for a given sentence. Sentence-BERT [23] shows that using BERT embeddings for semantic search is infeasible and proposes a self-supervised representation learning method for sequences. ConSERT [24] improves this line of work by using contrastive learning [25] through adding a projection head on top of BERT which enforces feature-invariance against augmentations such as adversarial perturbations, token/feature cutoff, and dropout. SimCSE [26] further simplifies the pretraining stage by using different dropout masks for two branches and simply relying on dropout features for invariance-maximization. DiffCSE [27] improves the method by incorporating generative learning into self-supervised pretraining through replaced token detection similar to Electra [28]. PaSer [29] uses a fully generative pretraining objective by encoding different views of the same sentence which are later used to recover the masked words.

At the intersection of pretraining and representation learning, there are a set of works that try to combine language modeling and representation learning. In pre-transformer era, CVT [30] increased the supervision of the models by adding auxiliary tasks for different views of which an LSTM model makes predictions. COCO-LM [31] shows that Transformer models have a squeezed representation space and mitigates the problem by jointly doing language modeling and sentence-level representation learning. TaCL [32] takes this a step further by learning a token-based features, opposed to sentence level, while doing language modeling. DialogueCSE [33] extends such work to the dialogue domain

```
# model: a neural network that returns features and logits
# tw: tied-weight
# noise1: a stochastic data augmentation module (e.g. random masking)
# noise2: a stochastic data augmentation module (e.g. random masking)
# note that noise1 and noise2 must be different

# ce = cross entropy loss
# mse = mean squared error loss

for x,y in loader:
   # generate two augmented views of the same sequence of tokens
   x1 = noise1(x) # noise randomly
   x2 = noise2(x) # noise randomly

   # extract logits and features
   f1, l1 = model(x1)
   f2, l2 = model(x2)

   # calculate loss
   ce_loss = (ce(l1, y) + ce(l2, y)) / 2
   feature_loss = mse(f1, f2)
   loss = ce_loss + tw * feature_loss
```

Figure 2: Python code for Semantic-Augment based on NumPy.

by learning dialogue-level features through encoding features at a sentence level but aggregating the features at individual turns level and applying contrastive loss between turns.

4 Approach

Semantic-Augment combines representation learning and language modeling. During training, our framework enforces feature invariance over different noised view of the same sequence. As shown in Figure 1, our framework consists of three parts:

- **Two stochastic noising methods** generate two different noised views of the same sequence (e.g., "Semantic-Augment rules NLP" → ("Semantic-Augment rules [MASK]", "Semantic-Augment [MASK] NLP). This is applicable to all Transformer-based architectures, since the main difference between them in pretraining is noising method.
- **A language model** that generates pre-logits features used for feature similarity loss.
- **Pairwise feature similarity and supervised loss functions** are used to enforce feature invariance in learning conventional language modeling.

In each optimization step, we take a sample of size N, where each sample is a sequence of the same length. Then, using random noising, we generate two views of each sequence, resulting in total of 2N samples. For all sequences, we extract logits and features from the neural network. We apply a similarity loss (L2) between the features of the augmented views of the same sequence to enforce invariance, and we also compute cross entropy loss. We provide a general overview of our framework in NumPy [34] in Figure 2. Given input x, logits $f(x)$, labels y, features of the first augmented views $v_1 = v_1(x)$, features of the second augmented views $v_2 = v_2(x)$, supervised loss ℓ, and feature similarity loss weight w (we call this hyperparameter tied-weight), the loss function of Semantic-Augment is:

$$\mathcal{L}_{\text{Semantic-Aug}} = \sum_i \ell(f(v_i(x)), y) + w\|v_1(x) - v_2(x)\|^2 \tag{1}$$

There are several ways to apply feature similarity loss, since language models output a feature vector for all words in a given sequence. In this work, we only consider the use of the loss of similarity

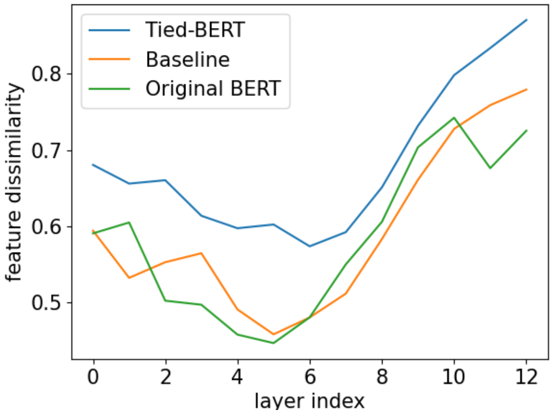

Figure 3: Feature dissimilarity comparison between the proposed model, original BERT, and our baseline. In this comparison, we compare 10,000 random sentences from the Bookcorpus. Visualized feature dissimilarity is (1 - cosine). Layer index denotes the index of the hidden state used to calculate the score among all the intermediate hidden states of the model.

between the features of CLS tokens only, at the word level, where the individual features of all words are included in the L2 loss, and at the sentence level, where a sentence-level feature is constructed by an average of the word-level features before applying the loss. We refer to the models trained using our method as Tied-X, where X is the name of the model (e.g., Tied- BERT). Since our approach can be viewed as maximizing feature similarity in semantic space, we call our method Semantic-Augment.

4.1 Baselines

Pre-training BERT is an expensive task (about 40 Nvidia V100 days). Therefore, we perform our experiments using a simplified Cramming [35] setting where the models are limited by a single GPU day, in our case 1 Nvidia A100 day. Since the proposed method almost doubles the runtime, we consider two baselines. The first is the model trained for twice the number of steps, while the second is the proposed setting where the similarity loss is zero. The version with zero similarity loss also shows the benefits of our method over Batch Augmentation [36], i.e., the noise invariance caused by different augmented views of the same sequence is examined in tw=0 model.

Our main comparison metric is the mean GLUE [37] and SWAG [38] performance as well as the feature space distribution. Also, we do not compare our method to COCO-LM [31], even if it is really relevant, since their computational budget is significantly higher than ours (20 A100 days versus 1 A100 day).

4.2 Implementation Details

All of our experiments were performed using the Pytorch [39] deep learning framework. Training[1]

5 Experiments and Analysis

In this section, we present our baselines and compare our results to the baseline, analyzing the benefits of Semantic-Augment.

5.1 Experimental Setting

Following [13], we pretrain our models using the Bookcorpus [41] and a 2022 Wikipedia dump [2] datasets, where the Wikipedia dump is approximately the same size as the original model. We do not apply any preprocessing, except for tokenization, which is done uncased.

[2] downloaded from `https://huggingface.co/datasets/wikipedia`

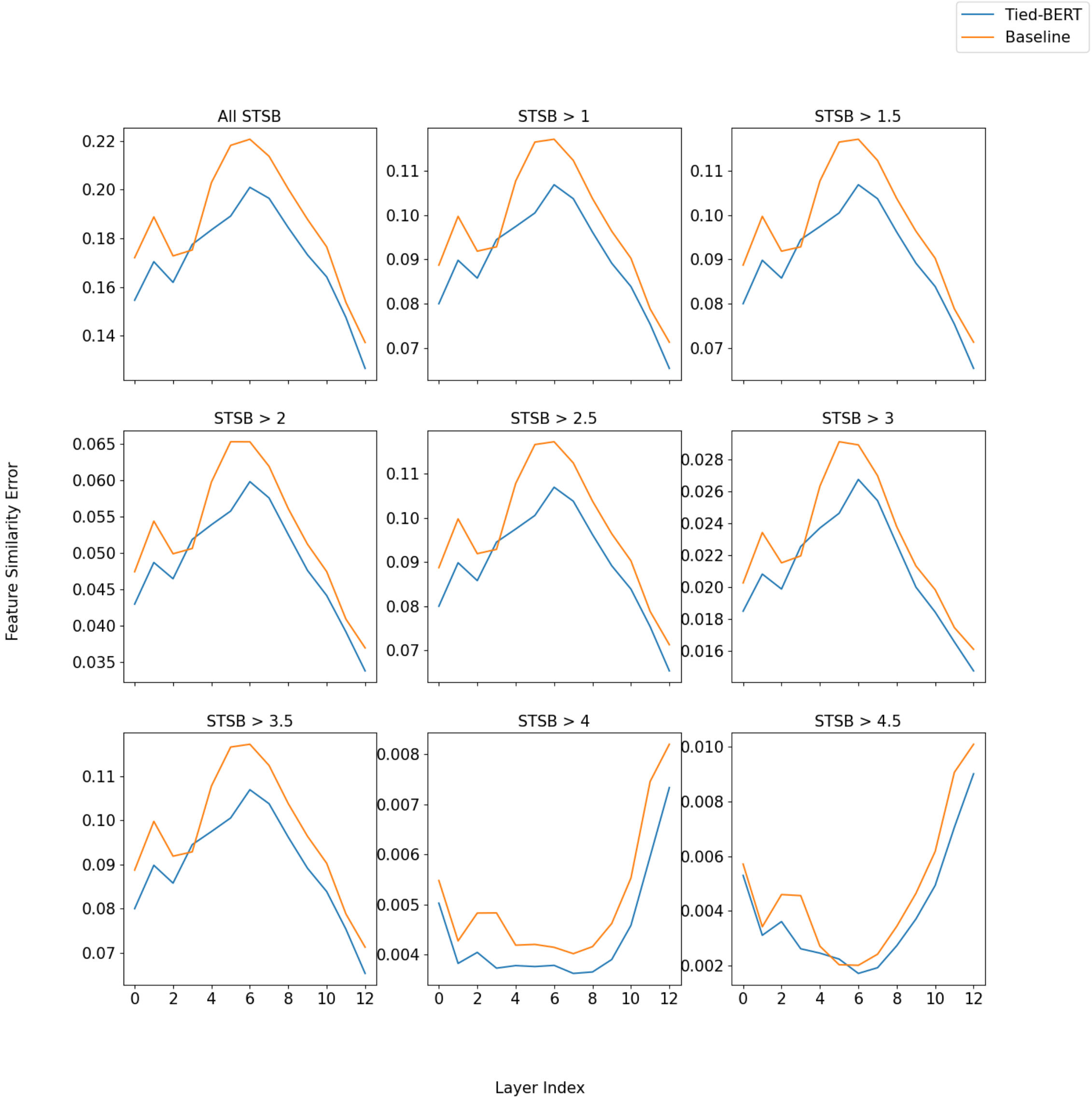

Figure 4: Feature similarity error comparison between our model and our baseline. All STSB means evaluation on both train and dev set. STSB > X denotes the case where samples with semantic overlap bigger than X are sampled. Feature similarity error is the L2 error between the cosine similarity output by the model and the STSB label.

	0.0 / 0.89	0.0 / 0.86	0.08 / 0.88
Sentence-1	A woman is dancing.	Someone is feeding an animal.	Someone is feeding an animal.
Sentence-2	A man is eating.	Someone is playing a piano.	Someone is playing a piano.

Table 1: Some sentences with low semantic similarity sampled from STSB [40] dataset. On the top row, the first number is the semantic similarity label given in the dataset, while the second is the sentence-level cosine similarity between the features output by the BERT-base model for sentence-1 and sentence-2.

	MNLI	SST-2	STSB	RTE	QNLI	QQP	MRPC	CoLA	Average
Baseline	79.6	90.4	85.2	58.1	89.1	86.3	87.9	44.8	77.7
Tied-BERT (tw=0)	80.4	90.0	85.1	59.9	87.8	86.0	88.0	**46.5**	78.0
Tied-BERT	**80.7**	**90.8**	**85.7**	**60.3**	**89.4**	**86.9**	**89.1**	45.7	**78.6**

Table 2: Results on GLUE dataset for the baseline models and Tied-BERT. We report the designated metrics given in [37]. All compared models are trained for 1 Nvidia A100 day. Reported results are single-task single-model finetuning.

	baseline	Tied-BERT (tw=0)	Tied-BERT
Accuracy	63.5	63.9	**64.6**

Table 3: Results on SWAG question-answering dataset. All compared models are pretrained for a single A100 day.

In pre-training, we train a BERT-base model for 80,000 steps over 800, of which a linear learning rate warm-up is applied. We use a sequence length of 128, a learning rate of 1e-4, a weight decay of 0.01, a global batch size of 1024, and for increased stability we clip the gradient to ensure the norm is 1 using a linear learning rate scheduler. Following [42], we omit the next sentence prediction loss and only do masked language modeling for all our experiments. We use no special attention mechanisms or implementation tricks, and make no changes to the BERT architecture to ensure that our gains are not due to the software lottery [43]. We omit curriculum learning and batch warmup in Cramming setting.

We evaluate the downstream performance of our models using two settings: GLUE [37] and SWAG [38]. Due to a limited computational budget, for pretraining,we run a hyperparameter search over 1, 3, 5, 7, 10 for tied-weight. For the downstream tasks, we perform a hyperparameter search over learning rates {5e-5, 4e-5, 3e-5, 2e-5}, the batch sizes {16, 32}, and weight decay 0.1 to GLUE for 3 epochs, while for SWAG we only train using batch size of 16, a weight decay of 0.01, and a learning rate of 2e-5 for 2 epochs.

5.2 GLUE

GLUE tasks are fine-tuning datasets that contain benchmarks for domains ranging from sentiment classification to semantic similarity. In our evaluation, which follows the original BERT evaluation, we exclude the WNLI task because of its problematic train/dev/set split [3]. In Table 2, we present the performances of our model and the baselines. We see that our model generalizes better to the downstream tasks, outperforming the baseline by 0.9% of the average GLUE score. This is to be expected in the sense that our approach increases noise invariance and implicitly introduces better semantic understanding. This clearly shows that the benefits of Semantic-Augment.

5.3 Question-Answering

We evaluate the question-answering capabilities of our model on SWAG [38] dataset. It inclues 113k adversarially-curated multiple choice questions covering a rich set of domains. The goal of this task is to choose one of the given four choices that is semantically the most compatible with the given sentence. For this task, Semantic-Augment significantly outperforms its baseline by 1.1% accuracy.

5.4 Feature Distribution

Representation deficiency is the problem where the representation space of the model is significantly limited [31]in the sense that the model uses only a small range of its output distribution. This results in features for different sentences being very close to each other. For two completely random sentences, the sentence-level features output from BERT baseline have a high cosine similarity. We provide examples for this case in Table 1.

[3] https://gluebenchmark.com/faq

Since Semantic-Augment aims to enlarge the feature space of networks, we evaluate the features of our model using a benchmark we developed. For our benchmark, we use the STSB [40] dataset, which contains sentence pairs and a manually annotated sentence similarity label. We evaluate the STSB examples without fine-tuning on STSB examples to see the differences caused by the pretraining stage. Since we do not fine-tune the model on train set, we evaluate on both train and dev sets.

Our benchmark consists of two different scenarios. The first is to compare the feature similarity error for the sentence pairs from the STSB dataset for a variety of sentence similarity brackets {all, above {0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5}. In addition to examining fine-grained feature similarity, we also examine feature dissimilarity between random sentences samples from the Bookcorpus [41] dataset. We perform this analysis for all layers considering that for a sentence-pair, feature similarity should be aligned for all layers.

In Figure 3, we show the feature dissimilarity for 10,000 randomly sampled sentence pairs from the Bookcorpus dataset. We report the feature dissimilarity results for all layers. Since we are aiming to distinguish between two unrelated sentences, we use feature dissimilarity as a metric (1 - cosine similarity). As expected, the last layer is closer to the golden truth, but for all layers the features of Tied-BERT are significantly closer to the golden truth. For the first layer the difference is about 9%, while for the last layer the difference is almost 10%. This is a clear indication that Semantic-Augment prevents the model from assigning a high semantic overlap score to unrelated sentences.

our results on the STSB train and dev set are depicted in Figure 4. We not only evaluate our models on the entire STSB dataset, but also create subsets with different semantic overlap regimes. We show that the Semantic Augment model has significantly more matching features than its baseline model for all layers and all regimes. It is worth noting that the alignment benefits of our model also apply to the early layers, showing that there is a general feature alignment throughout the internal representation.

6 Conclusion

Currently, language models assign extremely high scores to totally unrelated sentences and, unlike their image processing counterparts, pre-training language models does not maximize noise invariance, resulting in suboptimal performance. In this work, we present Semantic-Augment, a simple framework for unifying representation learning and language modeling. It significantly improves the performance of the models on GLUE and SWAG datasets while allowing the model to create a broader and more expressive feature space, thus avoiding representation deficiency. Moreover, Semantic-Augment can be implemented with only a few lines of additional code, making it an easy-to-use framework.

Our framework is easily applicable to all Transformer-based pretraining tasks, as it simply depends on the noising methods already used in pretraining. In the future, we aim to investigate the use of this framework for encoder-decoder and decoder-only architectures and its effectiveness when applied to Instruction-Tuning [19]. Given that our model creates a better representation space, another research direction would be to analyze whether our method minimizes model biases and thus potential harms.

References

[1] 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*, 33:1877–1901, 2020.

[2] Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. The flan collection: Designing data and methods for effective instruction tuning. *arXiv preprint arXiv:2301.13688*, 2023.

[3] Srinivasan Iyer, Xi Victoria Lin, Ramakanth Pasunuru, Todor Mihaylov, Dániel Simig, Ping Yu, Kurt Shuster, Tianlu Wang, Qing Liu, Punit Singh Koura, et al. Opt-iml: Scaling language model instruction meta learning through the lens of generalization. *arXiv preprint arXiv:2212.12017*, 2022.

[4] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. Lit: Zero-shot transfer with locked-image text tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18123–18133, 2022.

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

[6] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

[7] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.

[8] Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*, 2022.

[9] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

[10] Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. Glam: Efficient scaling of language models with mixture-of-experts. In *International Conference on Machine Learning*, pages 5547–5569. PMLR, 2022.

[11] Yu Meng, Jitin Krishnan, Sinong Wang, Qifan Wang, Yuning Mao, Han Fang, Marjan Ghazvininejad, Jiawei Han, and Luke Zettlemoyer. Representation deficiency in masked language modeling. *arXiv preprint arXiv:2302.02060*, 2023.

[12] Ethan A Chi, John Hewitt, and Christopher D Manning. Finding universal grammatical relations in multilingual bert. *arXiv preprint arXiv:2005.04511*, 2020.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[14] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding with unsupervised learning. 2018.

[15] 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. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

[16] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

[17] 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.

[18] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[19] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021, 2020.

[20] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[21] Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models. *arXiv preprint arXiv:2211.15661*, 2022.

[22] Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*, 2022.

[23] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

[24] Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. Consert: A contrastive framework for self-supervised sentence representation transfer. *arXiv preprint arXiv:2105.11741*, 2021.

[25] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006.

[26] Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*, 2021.

[27] Yung-Sung Chuang, Rumen Dangovski, Hongyin Luo, Yang Zhang, Shiyu Chang, Marin Soljačić, Shang-Wen Li, Wen-tau Yih, Yoon Kim, and James Glass. Diffcse: Difference-based contrastive learning for sentence embeddings. *arXiv preprint arXiv:2204.10298*, 2022.

[28] Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*, 2020.

[29] Bohong Wu and Hai Zhao. Generative or contrastive? phrase reconstruction for better sentence representation learning. *arXiv preprint arXiv:2204.09358*, 2022.

[30] Kevin Clark, Minh-Thang Luong, Christopher D Manning, and Quoc V Le. Semi-supervised sequence modeling with cross-view training. *arXiv preprint arXiv:1809.08370*, 2018.

[31] Yu Meng, Chenyan Xiong, Payal Bajaj, Paul Bennett, Jiawei Han, Xia Song, et al. Coco-lm: Correcting and contrasting text sequences for language model pretraining. *Advances in Neural Information Processing Systems*, 34:23102–23114, 2021.

[32] Yixuan Su, Fangyu Liu, Zaiqiao Meng, Tian Lan, Lei Shu, Ehsan Shareghi, and Nigel Collier. Tacl: Improving bert pre-training with token-aware contrastive learning. *arXiv preprint arXiv:2111.04198*, 2021.

[33] Che Liu, Rui Wang, Jinghua Liu, Jian Sun, Fei Huang, and Luo Si. Dialoguecse: Dialogue-based contrastive learning of sentence embeddings. *arXiv preprint arXiv:2109.12599*, 2021.

[34] Charles R. Harris, K. Jarrod Millman, Stéfan J van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585:357–362, 2020.

[35] Jonas Geiping and Tom Goldstein. Cramming: Training a language model on a single gpu in one day. *arXiv preprint arXiv:2212.14034*, 2022.

[36] Elad Hoffer, Tal Ben-Nun, Itay Hubara, Niv Giladi, Torsten Hoefler, and Daniel Soudry. Augment your batch: Improving generalization through instance repetition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8129–8138, 2020.

[37] 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. *arXiv preprint arXiv:1804.07461*, 2018.

[38] Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. Swag: A large-scale adversarial dataset for grounded commonsense inference. *arXiv preprint arXiv:1808.05326*, 2018.

[39] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[40] Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*, 2017.

[41] 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*, pages 19–27, 2015.

[42] 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*, 2019.

[43] Sara Hooker. The hardware lottery. *Communications of the ACM*, 64(12):58–65, 2021.