Sentence Bottleneck Autoencoders from Transformer Language Models

Ivan Montero♣ **Nikolaos Pappas**♣ **Noah A. Smith**♣◇

♣Paul G. Allen School of Computer Science & Engineering, University of Washington
◇Allen Institute for Artificial Intelligence
{ivamon,npappas,nasmith}@cs.washington.edu

Abstract

Representation learning for text via pretraining a language model on a large corpus has become a standard starting point for building NLP systems. This approach stands in contrast to autoencoders, also trained on raw text, but with the objective of learning to encode each input as a vector that allows full reconstruction. Autoencoders are attractive because of their latent space structure and generative properties. We therefore explore the construction of a sentence-level autoencoder from a pretrained, frozen transformer language model. We adapt the masked language modeling objective as a generative, denoising one, while only training a sentence bottleneck and a single-layer modified transformer decoder. We demonstrate that the sentence representations discovered by our model achieve better quality than previous methods that extract representations from pretrained transformers on text similarity tasks, style transfer (an example of controlled generation), and single-sentence classification tasks in the GLUE benchmark, while using fewer parameters than large pretrained models.[1]

1 Introduction

Recent research has focused on devising new unsupervised pretraining methods from unlabeled data that involves some form of language modeling, primarily autoregressive (Peters et al., 2018; Radford et al., 2019), masked (Devlin et al., 2019; Liu et al., 2019; Conneau et al., 2020) and generalized (Radford et al., 2019; Brown et al., 2020; Song et al., 2019), with much success on downstream tasks. Under the hood, most of these methods use transformers (Vaswani et al., 2017) for encoding text sequences, which allows them to learn powerful contextual word representations that have been used widely for building models in NLP. However, this does not hold for sentence representations derived from pretrained transformer language models based on a special token or basic pooling operations. To this end, representation learning methods have been designed to better capture semantic information from pretrained transformer language models, e.g., using Siamese networks trained with a triplet loss (Reimers and Gurevych, 2019) or transforming the desired sentence distribution to a Gaussian distribution through normalizing flows (Li et al., 2020).

Existing sentence representations directly derived from pretrained language models or learned by specialized methods cannot guarantee perfect reconstruction of the input, a property that can enhance the structure of their semantic space and enable their use for controlled generation tasks. For the latter, a few recent studies have looked into ways to steer generation of pretrained language models towards a particular style (Dathathri et al., 2020; Krause et al., 2021), although they require following the gradient during the sampling process and rely on style text classifiers which might not be always available. The latent space of a text autoencoder allows one to perform controlled text generation by directly manipulating sentence representations using basic numerical operations (Shen et al., 2020a). Yet, how to convert pretrained transformer language models to autoencoders with such properties still remains unexplored.

To fill in this gap, we introduce AUTOBOT, a new autoencoder model for learning sentence "bottleneck" (i.e., fixed-size) representations from pretrained transformers that is useful for similarity, generation, and classification, displayed in Figure 1. Our model has two unique components: (i) a transformation that uses dot product attention to dynamically pool semantic information from the pretrained model's hidden states into a sentence bottleneck representation, and (ii) a shallow transformer decoder that is modified to operate based on the bottleneck representation. Instead of training our autoencoder from scratch, we directly finetune

---

[1]Our code is available at: https://github.com/ivanmontero/autobot

it using an input reconstruction objective on the unlabeled data on which the original pretrained transformer was trained. We keep the underlying pretrained transformer encoder fixed, which makes it more efficient than training from scratch and proves beneficial even if we compare to pretrained transformers trained for an equal number of steps.

Our evaluation on representative sentence similarity, classification, and generation tasks demonstrates that the resulting sentence representations are compact, better capture semantic similarity at the sentence-level than strong sentence representation methods (Reimers and Gurevych, 2019), and can be used for controlled generation tasks. Lastly, our model performs almost on par with the large RoBERTa model (Liu et al., 2019) even though it only introduces 1.6% additional parameters relative to the base RoBERTa model.

2 Model: AUTOBOT

Taking inspiration from recent research on text autoencoders (Bowman et al., 2016b; Shen et al., 2020b; Mai et al., 2020), we extend standard autoregressive text autoencoders, which have been predominantly based on recurrent networks, to a transformer-based architecture and integrate them with pretrained language models; here we focus on RoBERTa (Liu et al., 2019).

Autoencoders generally follow the encoder-decoder model structure to reconstruct their input with the constraint that the encoder produces a single, fixed-length hidden representation $\mathrm{enc}(\boldsymbol{x}) = \mathbf{z}$:

$$\mathrm{AE}(\boldsymbol{x}) = \mathrm{dec}(\mathrm{enc}(\boldsymbol{x})) = \boldsymbol{x}'. \quad (1)$$

Here, we focus on denoising autoencoders that aim to reconstruct a perturbed version of the input (Vincent et al., 2010; Shen et al., 2020b), which is compatible with many of the pretrained language models that are based on masked language modeling. In our experiments, we use the same masking procedure as Devlin et al. (2019) to perturb the input.

2.1 Encoder

Standard approaches use encoders that reduce the input to a single representation $\mathbf{z}$. To use a pretrained transformer for this purpose we need to reduce its output hidden representations $\mathbf{H}$ after processing the input to a single vector. Since using the special token representation or basic pooling methods have been shown sub-optimal in prior work (Reimers and Gurevych, 2019), here we opt
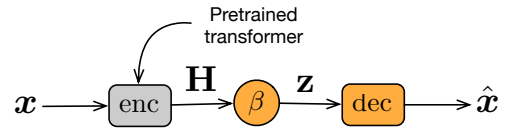

Figure 1: Our autoencoder consists of a pretrained transformer encoder enc, a function β that compresses the encoder's final representations $\mathbf{H}$ of size $T \times d$ to a sentence bottleneck representation $\mathbf{z}$ of size d, and a transformer decoder dec that is trained to fully reconstruct the training sentence $\boldsymbol{x}$.

to keep the original encoder fixed and train a transformation β that will learn to compress $\mathbf{H}$ into a single representation $\mathbf{z} = \beta(\mathbf{H}; \theta)$, with θ being an additional set of parameters to be learned during finetuning. We choose β to be a multi-head attention mechanism that takes as input the keys $\mathbf{K}$ and values $\mathbf{V}$ corresponding to the final representations $\mathbf{H}$ from the pretrained model and a query vector $\mathbf{q}$ corresponding to a context vector $\mathbf{u}$ that we choose to be the CLS vector from the pretrained model:

$$\beta(\mathbf{H}; \theta) = \mathrm{MultiHead}(\mathbf{q}, \mathbf{K}, \mathbf{V}) \quad (2)$$

where the parameters to be learned, θ, include the weights that are used to transform the query, keys, and values which amount to $3d^2$ with d being the dimensionality of each head ($d = 64$ in our experiments).

2.2 Decoder

The cross-attention layer in the Transformer decoder architecture by Vaswani et al. (2017) expects hidden representations for every token input from the encoder in order for each output candidate to attend to each input token. In the situation where only a single representation comes from the encoder, we have

$$\mathrm{Attention}(\mathbf{Q}, \mathbf{z}^\top \mathbf{W}_K, \mathbf{z}^\top \mathbf{W}_V) = \mathbf{z}^\top \mathbf{W}_V \quad (3)$$

Note that the queries $\mathbf{Q}$, which come from the previous masked self-attention layer, are not taken into account, and each step in the decoder will receive the exact same $\mathbf{z}^\top \mathbf{W}_V$ as a result. In order to mitigate this, we propose a gating method inspired by Hochreiter and Schmidhuber (1997). Concretely, let $\mathbf{Q_t}$ be the tth query representation. Then, the tth output $\mathbf{o}_t$ of the cross-attention layer is computed as follows

$$\mathbf{g}_t = \sigma(\mathbf{G}\mathbf{Q}_t + \mathbf{G}'\mathbf{z}); \quad \mathbf{o_t} = \mathbf{g}_t \odot \mathbf{z}^\top \mathbf{W}_V \quad (4)$$

where $\sigma(\cdot)$ is the *sigmoid* activation function and **G** and **G**$'$ are the parameters of the transformation for the gate. One can view the role of the gate as determining the amount of per-element information from the linear transformation of the latent representation to keep for the current layer and timestep. Preliminary experiments found this method beneficial for generation.

Training considerations To avoid training our model from scratch, we finetune it for 100K optimization steps on a pretraining dataset using the base RoBERTa model (Liu et al., 2019) on the encoder side and a single layer decoder side for efficiency purposes (Kasai et al., 2021). The model is trained using an input reconstruction loss by minimizing the negative log-likelihood computed over the reconstructed inputs. Note that only the parameters of the sentence bottleneck and the decoder are learned; the encoder parameters are kept fixed.

3 Experiments

To assess the quality of the sentence representations learned by our model we evaluate on sentence similarity (Section 3.2), classification (Section 3.3), and generation tasks (Section 3.4).

3.1 Settings

Datasets Since the RoBERTa dataset is not publicly available, we use for pretraining the exact same dataset as BERT (Devlin et al., 2019), which is composed of BooksCorpus (Zhu et al., 2015) and English Wikipedia. For sentence similarity, we use the Natural Language Inference (NLI) dataset (Bowman et al., 2015) for finetuning and evaluate on the Semantic Textual Similarity (STS) dataset (Cer et al., 2017), following Conneau et al. (2017). For classification, we use mainly single-sentence datasets from the GLUE benchmark (Wang et al., 2018), namely Stanford Sentiment Treebank (SST) and Corpus of Linguistic Acceptability (CoLA) datasets, but we also report the average performance on the remaining datasets. For generation, we use the Yelp reviews dataset (Shen et al., 2017).

Baselines For sentence similarity, we compare to SBERT which is a competitive method for deriving informative sentence representations from pretrained language models (Reimers and Gurevych, 2019). They obtain sentence representations by using simple pooling methods over BERT representations such as mean and max (instead of the CLS

token representation) then finetuning the whole pretrained model using Siamese networks on a combination of natural language inference data. To compare with them on sentence similarity, we incorporate our model within their framework and follow their settings and training/evaluation protocol (details in Appendix A.2).

For sentence classification, we compare our model to RoBERTa-base and RoBERTa-large models (Liu et al., 2019). Note that BART (Lewis et al., 2019) achieves similar results to RoBERTa, so a similar comparison can be made.

For sentence generation tasks, we compare to a strong and efficient style transfer method by Shen et al. (2020b), which is a recurrent network-based denoising text autoencoder on in domain data. The style transfer is achieved through vector arithmetic, namely computing a "sentiment vector" **v** by taking the vector difference between 100 negative and positive sentences, then evaluating by taking an input sentence, encoding it, adding a multiple of the sentiment vector to the sentence representation, then decoding the resulting representation. In addition to the denoising auto encoder (DAE) of Shen et al. (2020b), we include more sophisticated methods for style transfer that are more computationally expensive such as fast gradient iterative modification (FGIM) of Wang et al. (2019) and Emb2Emb of Mai et al. (2020) for reference.

3.2 Sentence Similarity

The results on the sentence similarity task are displayed in Table 1. Due to resource constraints and unreported results by prior work, we report our model only with RoBERTa-base. We can observe that AUTOBOT applied to RoBERTa-base significantly outperforms other supervised base transformer methods. Additionally, AUTOBOT approaches the performance of large transformers while having a minimal parameter overhead of 1.6%.

We also find that AUTOBOT without any supervision (AUTOBOT-base unsup.) outperforms all of the unsupervised methods, and most notably improves upon average BERT embeddings by 26.1%. This demonstrates that our approach is effective in both supervised and unsupervised settings.

We find in Table 2 that using the proposed sentence bottleneck based on learned context provides noticeable gains over using simpler pooling methods from prior work. We suspect this is due to the

Model	Spearman	Parameters
Unsupervised		
Avg. GloVe embeddings	58.02	-
Avg. BERT embeddings	46.35	-
AUTOBOT-base unsup.	**58.49**	-
Supervised		
InferSent - GloVe	68.03	-
Universal Sentence Encoder	74.92	-
RoBERTa-base	75.37	125M
SRoBERTa-base	76.89	125M
AUTOBOT-base (ours)	**78.59**	127M
RoBERTa-large	80.16	355M

Table 1: On semantic textual similarity (STS), AU-TOBOT outperforms previous sentence representation methods and reaches a score similar to RoBERTa-large while having fewer parameters. We report Spearman's rank correlation on the test set and the model sizes are reported in terms of trained parameter size.

Pooling	Spearman
MEAN	80.78
MAX	78.76
CLS	79.67
β (ours)	**81.88**

Table 2: Performance of sentence representations from RoBERTa trained with different pooling methods on NLI data and then evaluated on STS benchmark's development set in terms of Spearman's rank correlation.

additional flexibility provided by our bottleneck acting as "weighted pooling" by attending over all tokens to compute the final representation, as opposed to equal contribution of all tokens regardless of the input.

3.3 Sentence Classification

The results on single-sentence classification tasks and other tasks from the GLUE benchmark are displayed in Table 3. We find that AUTOBOT provides a noticable performance increase on single-sentence tasks, specifically on the CoLA datasets when using both the RoBERTa-base and RoBERTa-large models. Additionally, we also find that AU-TOBOT, when fed both sentences concatenated for dual sentence GLUE tasks, maintains the original performance of the underlying pretrained encoder.

Hence, our model improves the quality of the sentence representations from pretrained transformer models without hurting their performance.

Model	SST	CoLA	Others (avg)
RoBERTa-base	94.8	63.6	88.7
AUTOBOT-base	**95.0**	**66.0**	88.7
RoBERTa-large	96.4	68.0	91.1
AUTOBOT-large	**96.9**	**70.2**	91.1

Table 3: Single-sentence GLUE classification dev. results. Median accuracy is reported over over three random seeds. Our model improves performance on single-sentence classification tasks over both base and large RoBERTa models while maintaining their performance on the remaining multi-sentence tasks.

3.4 Sentence Generation

For sentence generation, we focus on the sentiment transfer task proposed by Shen et al. (2020b) both with and without further training on in-domain data from Yelp. When finetuning, we perform an additional 10K optimization steps using the Yelp dataset. Note that all the baselines require training on in-domain data, while this is optional for our model. In Figure 2 we find that the AUTO-BOT model not exposed to the Yelp dataset during finetuning performed on par with the DAE that was trained specifically on Yelp. Additionally, AU-TOBOT outperforms the DAE in the above-40 percent accuracy range when finetuned on in-domain data. We include AUTOBOT results with partial finetuning of the encoder in the appendix, which we find considerably improves the Self-BLEU metric. Since AUTOBOT uses vector arithmetic, inference is as fast as the DAE and over twice that of other methods.
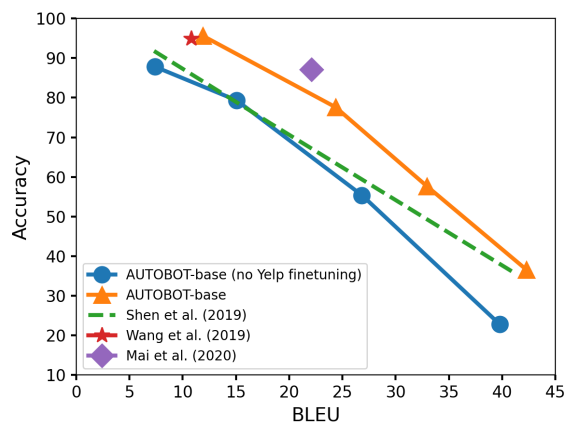

Figure 2: Automatic evaluations of vector arithmetic for sentiment transfer, plotted as accuracy vs. self-BLEU. Accuracy (ACC) is measured by a sentiment classifier, and values for varying multiples of the sentiment vector are plotted. Upper right is better.

4 Related Work

Reconstructing text with autoencoders is an active area of research that has lead to several advancements such as denoising (Vincent et al., 2010), variational (Kingma and Welling, 2014; Higgins et al., 2017; Dai and Wipf, 2019), adversarial (Makhzani et al., 2016; Zhao et al., 2018), and regularized (Ghosh et al., 2020) autoencoders. They have been found especially useful in controlled text generation (Hu et al., 2017; Logeswaran et al., 2018; Bowman et al., 2016a), especially in sentiment style transfer (Mai et al., 2020; Shen et al., 2017).

The encoder-decoder structure for obtaining representations has been used in pretraining (Lewis et al., 2019), sentence infilling (Huang et al., 2020), and multilingual (Artetxe and Schwenk, 2019) scenarios. In particular, Lewis et al. (2019) treat denoising as translation task to perform pretraining from scratch, but their approach does not induce a sentence representation space with generative properties. In contrast, our method makes use of a frozen pretrained transformer to learn a shallow, sentence bottleneck autoencoder on top.

5 Conclusion

We proposed an approach that converts a pretrained transformer language model into a sentence-level autoencoder that is able to reconstruct its pretraining data. The resulting model improves the performance of the pretrained model on sentence-level tasks while maintaining its performance on multi-sentence tasks. In addition, the new sentence representations are suitable for efficient conditional text generation such as sentiment transfer without the need for training on in-domain data.

Acknowledgments

References

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Samuel Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016a. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21.

Samuel R. 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 (EMNLP)*. Association for Computational Linguistics.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016b. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, Berlin, Germany. Association for Computational Linguistics.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *EMNLP*.

Bin Dai and David Wipf. 2019. Diagnosing and enhancing VAE models. In *International Conference on Learning Representations*.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2020. Plug and play language models: A simple approach to controlled text generation. In *International Conference on Learning Representations*.

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.

Partha Ghosh, Mehdi S. M. Sajjadi, Antonio Vergari, Michael Black, and Bernhard Scholkopf. 2020. From variational to deterministic autoencoders. In International Conference on Learning Representations.

Irina Higgins, Loïc Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew M Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. beta-vae: Learning basic visual concepts with a constrained variational framework. In International Conference on Learning Representations.

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

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward controlled generation of text. In ICML.

Yichen Huang, Yizhe Zhang, Oussama Elachqar, and Yu Cheng. 2020. Inset: Sentence infilling with inter-sentential transformer. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 2502–2515.

Jungo Kasai, Nikolaos Pappas, Hao Peng, James Cross, and Noah Smith. 2021. Deep encoder, shallow decoder: Reevaluating non-autoregressive machine translation. In International Conference on Learning Representations.

Diederik P Kingma and Max Welling. 2014. Auto-encoding variational bayes. In International Conference on Learning Representations.

Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq Joty, richard socher, and Nazneen Rajani. 2021. Gedi: Generative discriminator guided sequence generation.

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

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 9119–9130, Online. Association for Computational Linguistics.

Y. Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, M. Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. ArXiv, abs/1907.11692.

Lajanugen Logeswaran, Honglak Lee, and Samy Bengio. 2018. Content preserving text generation with attribute controls. Advances in Neural Information Processing Systems, 31.

Florian Mai, Nikolaos Pappas, Ivan Montero, Noah A Smith, and James Henderson. 2020. Plug and play autoencoders for conditional text generation. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 6076–6092.

Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, and Ian Goodfellow. 2016. Adversarial autoencoders. In International Conference on Learning Representations.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. 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), pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

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

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In EMNLP/IJCNLP.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In Proceedings of the 31st International Conference on Neural Information Processing Systems, pages 6833–6844.

Tianxiao Shen, Jonas Mueller, Dr.Regina Barzilay, and Tommi Jaakkola. 2020a. Educating text autoencoders: Latent representation guidance via denoising. In Proceedings of the 37th International Conference on Machine Learning, volume 119 of Proceedings of Machine Learning Research, pages 8719–8729. PMLR.

Tianxiao Shen, Jonas Mueller, Regina Barzilay, and Tommi Jaakkola. 2020b. Educating text autoencoders: Latent representation guidance via denoising. In International Conference on Machine Learning, pages 8719–8729. PMLR.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. MASS: Masked sequence to sequence pre-training for language generation. In Proceedings of the 36th International Conference on Machine Learning, volume 97 of Proceedings of Machine Learning Research, pages 5926–5936. PMLR.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(12).

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.

Ke Wang, Hang Hua, and Xiaojun Wan. 2019. Controllable unsupervised text attribute transfer via editing entangled latent representation. In *Advances in Neural Information Processing Systems*, pages 11034–11044.

Junbo Jake Zhao, Yoon Kim, Kelly Zhang, Alexander M Rush, Yann LeCun, et al. 2018. Adversarially regularized autoencoders. In *Proceedings of the 35th International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. 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.

A   Reproducibility

A.1   Experimental Setup

Computing Infrastructure   For all of our experiments involving base models, we use a computation cluster with 5 NVIDIA RTX 2080 TI GPU, 11GB GPU memory, and 128GB RAM. For large models, we use a computation cluster with 4 NVIDIA TITAN RTX GPUs, 24GB GPU memory and 256GB RAM.

Implementation   We will make our implementation available on Github.[2] We used Python 3.7, PyTorch 1.6.0, and Sentence Transformers 0.3.7. We use modified versions of Fairseq 0.9.0 and Transformers 3.3.1. We obtain our datasets from the citations specified in the main paper.

AUTOBOT Training   We extract the sentences from the BooksCorpus and English Wikipedia datasets to recreate the BERT dataset, and use RoBERTa-base's pretrained tokenizer for tokenization. We report our hyperparameters for AUTOBOT-base in Table 4. Our decoder only has one single layer, and RoBERTa-base remains fixed during finetuning.

MODEL PARAMETERS	VALUE
Fixed Parameters	
Transformer Encoder	RoBERTa-base
Transformer Encoder Fixed	True
Warmup Steps	4000
Dropout	0.1
Optimizer	Adam
Learning Rate Schedule	Linear Decay
Max Sequence Length	128
Max Tokens	24576
Bottleneck Heads	12
Hidden Size	768
Decoder Layers	1
Tuned Parameters	
Num Steps	$\{1k, 10k, 100k\}$
Learning Rate	$\{$1e-3, 1e-4, 1e-5$\}$
Optimal Parameters	
Num Steps	$100k$
Learning Rate	$1e-3$
Extra Info	
Model Size (# params)	$127M$

Table 4: Hyperparamters for AUTOBOT-base

A.2   Sentence Representations

We use the Sentence Transformers framework for training and evaluation of AUTOBOT. We use the default settings in their framework to train on NLI, and evaluate using the Spearman correlation of the cosine similarity. During NLI finetuning, we only use the encoder and bottlneck, with the bottleneck representation used as the sentence representation, and allow for all parameters to be finetuned.

A.3   Sentence Generation

We use a modified version of Fairseq's generation code for encoder-decoder models to perform vector arithmetic for sentiment transfer. We follow the instructions of Mai et al. (2020) to finetune a sentiment classifier using DistilBERT from the Huggingface transformers library.

For the AUTOBOT models finetuned to the Yelp dataset, we follow the exact same steps as Appendix A.1 except beginning with the AUTOBOT-base model, using the Yelp training set, and performing 10k optimization steps.

A.4   Sentence Classification

We use the Huggingface library to perform sentence classification using AUTOBOT. During finetuning, we only use the encoder and bottleneck, with the bottleneck representation used as a `CLS` representation, and allow for all parameters to be finetuned. We perform a hyperparameter search similar to that of RoBERTa by comparing development performances when using $\{$1e-5, 2e-5, 3e-5$\}$ for the learning rate.

B   Additional Results

We provide additional results in addition to our experiments below.

B.1   Autoencoding Steps

We perform an ablation study on the effect of autoencoding finetuning steps of the underlying pretrained encoder during autoencoding on the downstream sentence representation performance. We provide the detailed performances of performing Table 4 when using a learning rate of 1e-3 in Table 5.

B.2   Finetunable Encoder Layers

We perform an ablation study on the effect of finetuning the underlying pretrained encoder during

Training Steps	Spearman
1	74.38
1k	75.45
10k	78.01
100k	**78.59**
baseline	77.03

Table 5: AUTOBOT pretraining steps vs. sentence representation performance when training on NLI and evaluating on STS

autoencoding on downstream sentence representation performance. We provide the detailed performances of performing Table 4 with the optimal parameters, but varying how many of the last layers of RoBERTa-base to finetune. Results are in Table 6

Finetunable Layers	Spearman
None	**78.59**
1	77.24
2	76.17
3	76.20
baseline	77.03

Table 6: AUTOBOT finetunable layers vs. sentence representation performance when training on NLI and evaluating on STS

B.3   Finetunable Encoder Generation

We provide an appended generation table from Section 3.4 to include the generation results we obtained by allowing the top three layers of RoBERTa-base to be finetuned during autoencoding on the style generation task. The results are shown in Figure 3. The same model as used in Appendix B.2 is used.

B.4   Style Transfer Results

We provide Table 7 that reports results on the Yelp sentiment transfer test set from the generation table in Section 3.4, appending to the table (Mai et al., 2020). We outline the relative time differences during inference. We can observe that our model not only provides competitive speed-quality tradeoff.

B.5   Detailed Sentence Classification Results

Section 3.3 provides a summary of the GLUE results, while outlining the specific single-sentence classification performances. We provide the results for each task in Table 8
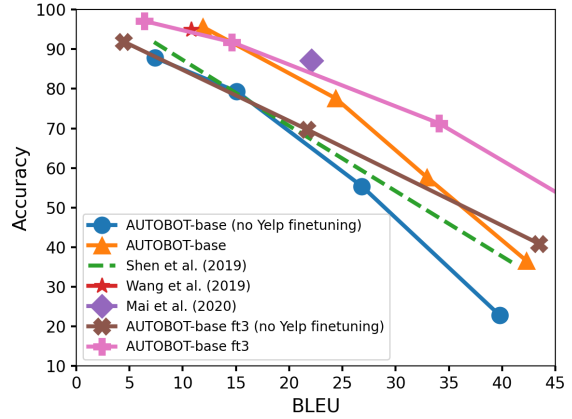

Figure 3: Automatic evaluations of vector arithmetic for sentiment transfer, plotted as accuracy vs. self-BLEU. Accuracy is measured by a sentiment classifier, and values for varying multiples of the sentiment vector are plotted. Upper right is better.

Model	Acc.	BLEU	+Time
FGIM	94.9	10.8	70.0×
Mai et al. 2020 + FGIM	93.1	18.1	2820.0×
Mai et al. 2020	87.1	22.1	1.0×
Shen et al. (2019)	96.8	6.5	0.5×
AUTOBOT-base (ours)	95.6	11.90	0.5×

Table 7: Self BLEU on the Yelp sentiment transfer test set with highest transfer accuracy ("Acc."). "+Time" reports the inference-time slowdown factor due to each method's additional computation relative to the method by Mai et al. (2020).

	MNLI	QNLI	QQP	RTE	SST	MRPC	CoLA	STS
RoBERTa-base	87.6	92.8	91.9	78.7	94.8	90.2	63.6	91.2
AUTOBOT-base	88.0	92.7	91.9	79.5	95.0	88.4	66.0	91.4
RoBERTa-large	90.2	94.7	92.2	86.6	96.4	90.9	68.0	92.4
AUTOBOT-large	90.5	94.6	92.2	87.6	96.9	89.0	70.2	92.4

Table 8: Dev. results on GLUE. For RTE, STS and MRPC we finetune starting from the MNLI model instead of the baseline pretrained model.