Disentangling Semantics and Syntax in Sentence Embeddings with Pre-trained Language Models

James Y. Huang and **Kuan-Hao Huang** and **Kai-Wei Chang**
University of California, Los Angeles
{jyhuang, khhuang, kwchang}@cs.ucla.edu

Abstract

Pre-trained language models have achieved huge success on a wide range of NLP tasks. However, contextual representations from pre-trained models contain entangled semantic and syntactic information, and therefore cannot be directly used to derive useful semantic sentence embeddings for some tasks. Paraphrase pairs offer an effective way of learning the distinction between semantics and syntax, as they naturally share semantics and often vary in syntax. In this work, we present ParaBART, a semantic sentence embedding model that learns to disentangle semantics and syntax in sentence embeddings obtained by pre-trained language models. ParaBART is trained to perform syntax-guided paraphrasing, based on a source sentence that shares semantics with the target paraphrase, and a parse tree that specifies the target syntax. In this way, ParaBART learns disentangled semantic and syntactic representations from their respective inputs with separate encoders. Experiments in English show that ParaBART outperforms state-of-the-art sentence embedding models on unsupervised semantic similarity tasks. Additionally, we show that our approach can effectively remove syntactic information from semantic sentence embeddings, leading to better robustness against syntactic variation on downstream semantic tasks.

1 Introduction

Semantic sentence embedding models encode sentences into fixed-length vectors based on their semantic relatedness with each other. If two sentences are more semantically related, their corresponding sentence embeddings are closer. As sentence embeddings can be used to measures semantic relatedness without requiring supervised data, they have been used in many applications, such as semantic textual similarity (Agirre et al., 2016a), question answering (Nakov et al., 2017), and natural language inference (Artetxe and Schwenk, 2019a).

Recent years have seen huge success of pre-trained language models across a wide range of NLP tasks (Devlin et al., 2019; Lewis et al., 2020). However, several studies (Reimers and Gurevych, 2019; Li et al., 2020) have found that sentence embeddings from pre-trained language models perform poorly on semantic similarity tasks when the models are not fine-tuned on task-specific data. Meanwhile, Goldberg (2019) shows that BERT without fine-tuning performs surprisingly well on syntactic tasks. Hence, we posit that these contextual representations from pre-trained language models without fine-tuning capture entangled semantic and syntactic information, and therefore are not suitable for sentence-level semantic tasks.

Ideally, the semantic embedding of a sentence should not encode its syntax, and two semantically similar sentences should have close semantic embeddings regardless of their syntactic differences. While various models (Conneau et al., 2017; Cer et al., 2018; Reimers and Gurevych, 2019) have been proposed to improve the performance of sentence embeddings on downstream semantic tasks, most of these approaches do not attempt to separate syntactic information from sentence embeddings.

To this end, we propose ParaBART, a semantic sentence embedding model that learns to disentangle semantics and syntax in sentence embeddings. Our model is built upon BART (Lewis et al., 2020), a sequence-to-sequence Transformer (Vaswani et al., 2017) model pre-trained with self-denoising objectives. Parallel paraphrase data is a good source of learning the distinction between semantics and syntax, as paraphrase pairs naturally share the same meaning but often differ in syntax. Taking advantage of this fact, ParaBART is trained to perform syntax-guided paraphrasing, where a source sentence containing the desired semantics and a parse tree specifying the desired syntax are given as inputs. In order to generate a paraphrase that follows the given syntax, ParaBART uses separate encoders to learn disentangled semantic and syntactic representations from their respective inputs. In this way, the disentangled representations capture sufficient semantic and syntactic information needed for paraphrase generation. The semantic encoder is also encouraged to ignore the syntax of the source sentence, as the desired syntax is already provided by the syntax input.

ParaBART achieves strong performance across unsupervised semantic textual similarity tasks. Furthermore, semantic embeddings learned by ParaBART contain significantly less syntactic information as suggested by probing results, and yield robust performance on datasets with syntactic variation.

Our source code is available at https://github.com/uclanlp/ParaBART.

2 Related Work

Various sentence embedding models have been proposed in recent years. Most of these models utilize supervision from parallel data (Wieting and Gimpel, 2018; Artetxe and Schwenk, 2019b; Wieting et al., 2019, 2020), natural language inference data (Conneau et al., 2017; Cer et al., 2018; Reimers and Gurevych, 2019), or a combination of both (Subramanian et al., 2018).

Many efforts towards controlled text generation have been focused on learning disentangled sentence representations (Hu et al., 2017; Fu et al., 2018; John et al., 2019). In the context of disentangling semantics and syntax, Bao et al. (2019) and Chen et al. (2019) utilize variational autoencoders to learn two latent variables for semantics and syntax. In contrast, we use the outputs of a constituency parser to learn purely syntactic representations, and facilitate the usage of powerful pre-trained language models as semantic encoders.

Our approach is also related to prior work on syntax-controlled paraphrase generation (Iyyer et al., 2018; Kumar et al., 2020; Goyal and Durrett, 2020; Huang and Chang, 2021). While these approaches focus on generating high-quality paraphrases that conform to the desired syntax, we are interested in how semantic and syntactic information can be disentangled and how to obtain good semantic sentence embeddings.

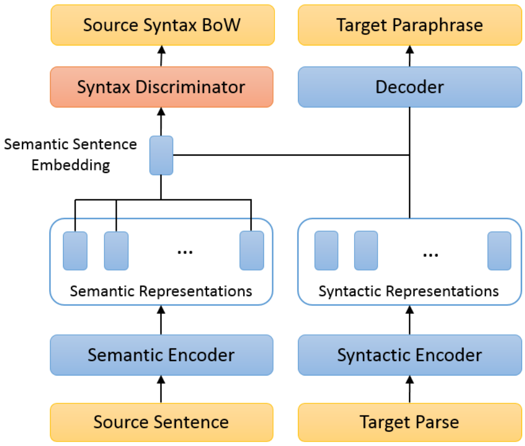

Figure 1: An overview of ParaBART. The model extracts semantic and syntactic representations from a source sentence and a target parse respectively, and uses both the semantic sentence embedding and the target syntactic representations to generate the target paraphrase. ParaBART is trained in an adversarial setting, with the syntax discriminator (red) trying to decode the source syntax from the semantic embedding, and the paraphrasing model (blue) trying to fool the syntax discriminator and generate the target paraphrase at the same time.

3 Proposed Model – ParaBART

Our goal is to build a semantic sentence embedding model that learns to separate syntax from semantic embeddings. ParaBART is trained to generate syntax-guided paraphrases, where the model attempts to only extract the semantic part from the input sentence, and combine it with a different syntax specified by the additional syntax input in the form of a constituency parse tree.

Figure 1 outlines the proposed model, which consists of a semantic encoder that learns the semantics of a source sentence, a syntactic encoder that encodes the desired syntax of a paraphrase, and a decoder that generates a corresponding paraphrase. Additionally, we add a syntax discriminator to adversarially remove syntactic information from the semantic embeddings.

Given a source sentence S_1 and a target constituency parse tree P_2, ParaBART is trained to generate a paraphrase S_2 that shares the semantics of S_1 and conforms to the syntax specified by P_2. Semantics and syntax are two key aspects that determine how a sentence is generated. Our model learns purely syntactic representations from the output trees generated by a constituency parser, and extracts the semantic embedding directly from the source sentence. The syntax discriminator and the syntactic encoder are designed to remove source syntax and provide target syntax, thus encouraging the semantic encoder to only capture source semantics.

Semantic Encoder The semantic encoder E_{sem} is a Transformer encoder that embeds a sentence $S = (s^{(1)}, ..., s^{(m)})$ into contextual semantic representations:

$$U = (\mathbf{u}^{(1)}, ..., \mathbf{u}^{(m)}) = E_{sem}\left((s^{(1)}, ..., s^{(m)})\right).$$

Then, we take the mean of these contextual representations $\mathbf{u}^{(i)}$ to get a fixed-length semantic sentence embedding

$$\bar{\mathbf{u}} = \frac{1}{m}\sum_{i=1}^{m}\mathbf{u}^{(i)}.$$

Syntactic Encoder The syntactic encoder E_{syn} is a Transformer encoder that takes a linearized constituency parse tree $P = (p^{(1)}, ..., p^{(n)})$ and converts it into contextual syntactic representations

$$V = (\mathbf{v}^{(1)}, ..., \mathbf{v}^{(n)}) = E_{syn}\left((p^{(1)}, ..., p^{(n)})\right).$$

For example, the linearized parse tree of the sentence "This book is good." is "(S (NP (DT) (NN)) (VP (VBZ) (ADJP)) (.))". Such input sequence preserves the tree structure, allowing the syntactic encoder to capture the exact syntax needed for decoding.

Decoder The decoder D_{dec} uses the semantic sentence embedding $\bar{\mathbf{u}}$ and the contextual syntactic representations V to generate a paraphrase that shares semantics with the source sentence while following the syntax of the given parse tree. In other words,

$$(y^{(1)}, ..., y^{(l)}) = D_{dec}\left(\text{Concat}(\bar{\mathbf{u}}, V)\right).$$

During training, given a source sentence S_1, a target parse tree P_2 and a target paraphrase $S_2 = (s_2^1, ..., s_2^l)$, we minimize the following *paraphrase generation loss*:

$$\mathcal{L}_{para} = -\sum_{i=1}^{l}\log P(y^{(i)} = s_2^{(i)}|S_1, P_2).$$

Since the syntactic representations do not contain semantics, the semantic encoder needs to accurately capture the semantics of the source sentence for a paraphrase to be generated. Meanwhile, the full syntactic structure of the target is provided by the syntactic encoder, thus encouraging the semantic encoder to ignore the source syntax.

Syntax Discriminator To further encourage the disentanglement of semantics and syntax, we employ a syntax discriminator to adversarially remove syntactic information from semantic embeddings. We first train the syntax discriminator to predict the syntax from its semantic embedding, and then train the semantic encoder to "fool" the syntax discriminator such that the source syntax cannot be predicted from the semantic embedding.

More specifically, we adopt a simplified approach similar to John et al. (2019) by encoding source syntax as a Bag-of-Words vector $\mathbf{h}$ of its constituency parse tree. For any given source parse tree, this vector contains the count of occurrences of every constituent tag, divided by the total number of constituents in the parse tree. Given the semantic sentence embedding $\bar{\mathbf{u}}$, our linear syntax discriminator D_{dis} predicts $\mathbf{h}$ by

$$\mathbf{y}_h = D_{dis}(\bar{\mathbf{u}}) = \text{softmax}(\mathbf{W}\bar{\mathbf{u}} + \mathbf{b})$$

with the following *adversarial loss*:

$$\mathcal{L}_{adv} = -\sum_{t \in T}\mathbf{h}(t)\log(\mathbf{y}_h(t)),$$

where T denotes the set of all constituent tags.

Training We adversarially train E_{sem}, E_{syn}, D_{dec}, and D_{dis} with the following objective:

$$\min_{E_{sem}, E_{syn}, D_{dec}}\left(\max_{D_{dis}}\left(\mathcal{L}_{para} - \lambda_{adv}\mathcal{L}_{adv}\right)\right),$$

where λ_{adv} is a hyperparameter to balance loss terms. In each iteration, we update the D_{dis} by considering the inner optimization, and then update E_{sem}, E_{syn} and D_{dec} by considering the outer optimization.

4 Experiments

In this section, we demonstrate that ParaBART is capable of learning semantic sentence embeddings that capture semantic similarity, contain less syntactic information, and yield robust performance against syntactic variation on semantic tasks.

4.1 Setup

We sample 1 million English paraphrase pairs from ParaNMT-50M (Wieting and Gimpel, 2018), and split this dataset into 5,000 pairs as the validation set and the rest as our training set. The constituency parse trees of all sentences are obtained from Stanford CoreNLP (Manning et al., 2014). We fine-tune a 6-layer BART$_{\text{base}}$ encoder as the semantic

Model	STS12	STS13	STS14	STS15	STS16	STS-B	Avg.
Avg. BERT embeddings (Devlin et al., 2019)	46.9	52.8	57.2	63.5	64.5	47.9	55.5
Avg. BART embeddings (Lewis et al., 2020)	50.8	42.8	56.1	63.9	59.5	52.0	54.2
InferSent (Conneau et al., 2017)	59.3	59.0	70.0	71.5	71.5	70.0	66.9
VGVAE (Chen et al., 2019)	61.8	62.2	69.2	72.5	67.8	74.2	68.0
USE (Cer et al., 2018)	61.4	63.5	70.6	74.3	73.9	74.2	69.7
Sentence-BERT (Reimers and Gurevych, 2019)	64.6	67.5	73.2	74.3	70.1	74.1	70.6
BGT (Wieting et al., 2020)	**68.9**	62.2*	75.9	79.4	79.3	-	-
ParaBART	68.4	**71.1**	**76.4**	80.7	**80.1**	78.5	**75.9**
- w/o adversarial loss	67.5	70.0	75.8	**80.9**	80.0	**78.7**	75.5
- w/o adversarial loss and syntactic guidance	66.4	65.3	73.6	80.0	78.6	75.4	73.2

Table 1: Pearson's r (in percentage) between cosine similarity of sentence embeddings and gold labels on STS tasks from 2012 to 2016 and STS Benchmark test set. BGT results are taken from Wieting et al. (2020). *BGT is evaluated on an additional dataset from STS13, which is not included in the standard SentEval toolkit.

encoder and the first BART$_{\text{base}}$ decoder layer as the decoder for our model.

We train ParaBART on a GTX 1080Ti GPU using AdamW (Loshchilov and Hutter, 2019) optimizer with a learning rate of 2×10^{-5} for the encoder and syntax discriminator, and 1×10^{-4} for the rest of the model. The batch size is set to 64. All models are trained for 10 epochs, which takes about 2 days to complete. The maximum length of input sentences and linearized parse trees are set to 40 and 160 respectively. We set the weight of adversarial loss to 0.1. Appendix A shows more implementation details.

Baselines We compare our model with other sentence embeddings models, including InferSent (Conneau et al., 2017), Universal Sentence Encoder (USE) (Cer et al., 2018), Sentence-BERT$_{\text{base}}$ (Reimers and Gurevych, 2019), VGVAE (Chen et al., 2019), and BGT (Wieting et al., 2020). We also include mean-pooled BERT$_{\text{base}}$ and BART$_{\text{base}}$ embeddings. In addition to ParaBART, we consider two model ablations: ParaBART without adversarial loss, and ParaBART without syntactic guidance and adversarial loss.

4.2 Semantic Textual Similarity

We evaluate our semantic sentence embeddings on the unsupervised Semantic Textual Similarity (STS) tasks from SemEval 2012 to 2016 (Agirre et al., 2012; 2013; 2014; 2015; 2016b) and STS Benchmark test set (Cer et al., 2017), where the goal is to predict a continuous-valued score between 0 and 5 indicating how similar the meanings of a sentence pair are. For all models, we compute the cosine similarity of embedding vectors as the semantic similarity measure. We use the standard SentEval toolkit (Conneau and Kiela, 2018) for evaluation and report average Pearson correlation over all domains.

Model	BShift	TreeDepth	TopConst
Avg. BART embed.	90.5	47.8	80.1
ParaBART	**72.4**	**33.9**	**67.2**
- w/o AL	75.4	36.6	71.7
- w/o AL and SG	83.3	46.5	83.1

Table 2: Results on syntactic probing tasks. Semantic embeddings with lower accuracy on downstream syntactic tasks contain less syntactic information, suggesting better disentanglement of semantics and syntax. AL and SG denote adversarial loss and syntactic guidance, respectively.

As shown in Table 1, both average BERT embeddings and average BART embeddings perform poorly on STS tasks, as the entanglement of semantic and syntactic information leads to low correlation with semantic similarity. Training ParaBART on paraphrase data substantially improves the correlation. With the addition of syntactic guidance and adversarial loss, ParaBART achieves the best overall performance across STS tasks, showing the effectiveness of our approach.

4.3 Syntactic Probing

To better understand how well our model learns to disentangle syntactic information from semantic embeddings, we probe our semantic sentence embeddings with downstream syntactic tasks. Following Conneau et al. (2018), we investigate to what degree our semantic sentence embeddings can be used to identify bigram word reordering (BShift), estimate parse tree depth (TreeDepth), and predict parse tree top-level constituents (TopConst). Top-level constituents are defined as the group of constituency parse tree nodes immediately below the sentence (S) node. We use the datasets provided by SentEval (Conneau and Kiela, 2018) to train a Multi-Layer Perceptron classifier with a single 50-neuron hidden layer on top of semantic sentence embeddings, and report accuracy on all

QQP-Easy
What are the essential skills of the project management? What are the essential skills of a project manager?
QQP-Hard
Is there a reason why we should travel alone? What are some reasons to travel alone?

Table 3: Examples of paraphrase pairs from *QQP-Easy* and *QQP-Hard*.

tasks.

As shown in Table 2, sentence embeddings pooled from pre-trained BART model contain rich syntactic information that can be used to accurately predict syntactic properties including word order and top-level constituents. The disentanglement induced by ParaBART is evident, lowering the accuracy of downstream syntactic tasks by more than 10 points compared to pre-trained BART embeddings and ParaBART without adversarial loss and syntactic guidance. The results suggest that the semantic sentence embeddings learned by ParaBART indeed contain less syntactic information.

4.4 Robustness Against Syntactic Variation

Intuitively, semantic sentence embedding models that learn to disentangle semantics and syntax are expected to yield more robust performance on datasets with high syntactic variation. We consider the task of paraphrase detection on Quora Question Pairs (Iyer et al., 2017) dev set as a testbed for evaluating model robustness. We categorize paraphrase pairs based on whether they share the same top-level constituents. We randomly sample 1,000 paraphrase pairs from each of the two classes, combined with a common set of 1,000 randomly sampled non-paraphrase pairs, to create two datasets *QQP-Easy* and *QQP-Hard*. Paraphrase pairs from *QQP-Hard* are generally harder to identify as they are much more syntactically different compared to those from *QQP-Easy*. Table 3 shows some examples from these two datasets. We evaluate semantic sentence embeddings on these datasets in an unsupervised manner by computing the cosine similarity as the semantic similarity measure. We search for the best threshold between -1 and 1 with a step size of 0.01 on each dataset, and report the highest accuracy. The results are shown in Table 4.

Model	**QQP-Easy**	**QQP-Hard**
Avg. BART embed.	72.3	64.1
InferSent	72.1	67.5
VGVAE	71.5	67.1
USE	**80.7**	72.4
Sentence-BERT	74.3	70.7
ParaBART	76.5	**72.7**
- w/o AL	76.8	72.1
- w/o AL and SG	76.1	69.9

Table 4: Results on *QQP-Easy* and *QQP-Hard*. For every model we report the highest accuracy after finding the best threshold. AL and SG denote adversarial loss and syntactic guidance, respectively.

While Universal Sentence Encoder scores much higher than other models on *QQP-Easy*, its performance degrades significantly on *QQP-Hard*. In comparison, ParaBART demonstrates better robustness against syntactic variation, and surpasses USE to become the best model on the more syntactically diverse *QQP-Hard*. It is worth mentioning that even pre-trained BART embeddings give decent results on *QQP-Easy*, suggesting large overlaps between paraphrase pairs from *QQP-Easy*. On the other hand, the poor performance of pre-trained BART embeddings on a more syntactically diverse dataset like *QQP-Hard* clearly shows its incompetence as semantic sentence embeddings.

5 Conclusion

In this paper, we present ParaBART, a semantic sentence embedding model that learns to disentangle semantics and syntax in sentence embeddings from pre-trained language models. Experiments show that our semantic sentence embeddings yield strong performance on unsupervised semantic similarity tasks. Further investigation demonstrates the effectiveness of disentanglement, and robustness of our semantic sentence embeddings against syntactic variation on downstream semantic tasks.

Acknowledgments

We thank anonymous reviewers for their helpful feedback. We thank UCLA-NLP group for the valuable discussions and comments. This work is supported in part by Amazon Research Award.

Ethics Considerations

Our sentence embeddings can potentially capture bias reflective of the training data we use, which is a common problem for models trained on large annotated datasets. While the focus of our work is to disentangle semantics and syntax, our model can potentially generate offensive or biased content learned from training data if it is used for paraphrase generation. We suggest carefully examining the potential bias exhibited in our models before deploying them in any real-world applications.

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Iñigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. 2015. SemEval-2015 task 2: Semantic textual similarity, English, Spanish and pilot on interpretability. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 252–263.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. SemEval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 81–91.

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016a. SemEval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 497–511.

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016b. SemEval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 497–511.

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. SemEval-2012 task 6: A pilot on semantic textual similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *SEM 2013 shared task: Semantic textual similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 32–43.

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

Mikel Artetxe and Holger Schwenk. 2019b. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7(0).

Yu Bao, Hao Zhou, Shujian Huang, Lei Li, Lili Mou, Olga Vechtomova, Xin-yu Dai, and Jiajun Chen. 2019. Generating sentences from disentangled syntactic and semantic spaces. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6008–6019.

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.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174.

Mingda Chen, Qingming Tang, Sam Wiseman, and Kevin Gimpel. 2019. A multi-task approach for disentangling syntax and semantics in sentence 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 2453–2464.

Alexis Conneau and Douwe Kiela. 2018. SentEval: An evaluation toolkit for universal sentence representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136.

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.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style transfer in text: Exploration and evaluation. In *Proceedings of The*

Thirty-Second Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence (AAAI).

Yoav Goldberg. 2019. Assessing bert's syntactic abilities. *CoRR*, abs/1901.05287.

Tanya Goyal and Greg Durrett. 2020. Neural syntactic preordering for controlled paraphrase generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 238–252.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2017. Toward controlled generation of text. Proceedings of Machine Learning Research.

Kuan-Hao Huang and Kai-Wei Chang. 2021. Generating syntactically controlled paraphrases without using annotated parallel pairs. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*.

Shankar Iyer, Nikhil Dandekar, and Kornél Csernai. 2017. First quora dataset release: Question pairs.

Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. 2015. Deep unordered composition rivals syntactic methods for text classification. In *Association for Computational Linguistics*, pages 1681–1691.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. 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 1875–1885.

Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. 2019. Disentangled representation learning for non-parallel text style transfer. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 424–434.

Ashutosh Kumar, Kabir Ahuja, Raghuram Vadapalli, and Partha Talukdar. 2020. Syntax-guided controlled generation of paraphrases. *Transactions of the Association for Computational Linguistics*, pages 329–345.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pretraining for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

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.

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

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.

Preslav Nakov, Doris Hoogeveen, Lluís Màrquez, Alessandro Moschitti, Hamdy Mubarak, Timothy Baldwin, and Karin Verspoor. 2017. SemEval-2017 task 3: Community question answering. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 27–48.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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 3982–3992.

Sandeep Subramanian, Adam Trischler, Yoshua Bengio, and Christopher J Pal. 2018. Learning general purpose distributed sentence representations via large scale multi-task learning. In *International Conference on Learning Representations*.

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

John Wieting and Kevin Gimpel. 2018. ParaNMT-50M: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462.

John Wieting, Kevin Gimpel, Graham Neubig, and Taylor Berg-Kirkpatrick. 2019. Simple and effective paraphrastic similarity from parallel translations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4602–4608.

John Wieting, Graham Neubig, and Taylor Berg-Kirkpatrick. 2020. A bilingual generative transformer for semantic sentence embedding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1581–1594.

A Implementation Details

Datasets We use the ParaNMT-50M dataset released by Wieting and Gimpel (2018), which can be obtained from https://github.com/jwieting/para-nmt-50m. We sample 1 million English paraphrase pairs from ParaNMT-50M, and split this dataset into 5000 pairs as the validation set and the rest as our training set. STS and syntactic probing datasets are directly taken from SentEval, which can be accessed from https://github.com/facebookresearch/SentEval. Quora Question Pairs are downloaded from the official GLUE Benchmark website (https://gluebenchmark.com/).

Word Dropout We observe that some paraphrase pairs in our training set contain many overlapping words, which means our model can learn to generate the target paraphrase by just copying words from a source sentence without fully understanding the semantics of the sentence. To alleviate this issue, we apply word dropout (Iyyer et al., 2015) that randomly masks a portion of the input tokens. We don't apply word dropout to syntactic inputs, as these inputs are designed to provide the exact syntactic structure of the paraphrase and encourage disentanglement of syntactic and semantic representations. We set the word dropout probability to 0.2 for all our models.

Hyperparameter Search Hyperparameters of ParaBART are tuned manually based on the paraphrase generation loss on the validation set. Specifically, the weight of adversarial loss is tuned within $\{0.1, 0.2, 0.5, 1.0\}$. Word dropout is selected from $\{0.0, 0.1, 0.2, 0.4\}$. Learning rate is tuned within $\{1,2,5,10\}\times10^{-5}$.

None of the previous models we compare in this work involves any hyperparameter search. The results for BGT are taken from Wieting et al. (2020). For all other sentence embedding models, we use the trained model provided by their respective authors. These models include InferSent (https://github.com/facebookresearch/InferSent, USE (https://tfhub.dev/google/universal-sentence-encoder-large/2), Sentence-BERT$_{\text{base}}$ (https://github.com/UKPLab/sentence-transformers) and VGVAE (https://github.com/mingdachen/syntactic-template-generation).

Performance on STS and QQP are evaluated under unsupervised settings. For syntactic probing tasks that involve training classifiers, we report the accuracy on the validation set provided by SentEval in Table 5.

Model	BShift	TreeDepth	TopConst
Avg. BART embed.	90.4	47.5	80.2
ParaBART	73.0	34.8	67.6
- w/o AL	75.4	36.7	72.1
- w/o AL and SG	84.0	46.7	82.7

Table 5: Validation accuracy on syntactic probing tasks.