CAT-Gen: Improving Robustness in NLP Models via Controlled Adversarial Text Generation

# Tianlu Wang*十\** Xuezhi Wang*§* Yao Qin*§* Ben Packer*§* Kang Lee*§* Jilin Chen*§* Alex Beutel*§* Ed Chi*§*

*十*University of Virginia [tw8cb@virginia.edu](mailto:tw8cb@virginia.edu)

*§*Google Research *{*xuezhiw, yaoqin, bpacker, kanlig, jilinc, alexbeutel, edchi*}*@google.com

# Abstract

NLP models are shown to suffer from ro- bustness issues, i.e., a model’s prediction can be easily changed under small perturbations to the input. In this work, we present a Controlled Adversarial Text Generation (CAT- Gen) model that, given an input text, gener- ates adversarial texts through controllable at- tributes that are known to be irrelevant to task labels. For example, in order to attack a model for sentiment classiﬁcation over product re- views, we can use the product categories as the controllable attribute which should not change the sentiment of the reviews. Experiments on real-world NLP datasets demonstrate that our method can generate more diverse and ﬂu- ent adversarial texts, compared to many ex- isting adversarial text generation approaches. We further use our generated adversarial ex- amples to improve models through adversarial training, and we demonstrate that our gener- ated attacks are more robust against model re- training and different model architectures.

# Introduction

It has been shown that NLP models are often sen- sitive to random initialization ([Zhou et al.](#_bookmark31), [2020](#_bookmark31)), out-of-distribution data ([Hendrycks et al.](#_bookmark17), [2020](#_bookmark17); [Wang et al.](#_bookmark29), [2019](#_bookmark29)), and adversarially generated at- tacks ([Jia and Liang](#_bookmark21), [2017](#_bookmark21); [Jin et al.](#_bookmark23), [2020](#_bookmark23); [Alzan-](#_bookmark9) [tot et al.](#_bookmark9), [2018](#_bookmark9)). One line of research to improve models’ robustness to adversarial attacks is by gen- erating adversarial examples in either the input text space (discrete, e.g., [Alzantot et al.](#_bookmark9) ([2018](#_bookmark9)); [Jin et al.](#_bookmark23) ([2020](#_bookmark23))) or some intermediate representation space (continuous, e.g., [Zhao et al.](#_bookmark30) ([2018](#_bookmark30)); [Zhu et al.](#_bookmark32) ([2020](#_bookmark32))). However, existing adversarial text genera- tion approaches that try to perturb in the input text space might lead to generations *lacking diversity or*

*\**This research was conducted during the author’s intern- ship at Google Research.

*ﬂuency*. On the other hand, approaches focusing on perturbing in the intermediate representation space can often lead to generations that are not related to the input. We show some adversarial examples generated by existing works in Table [1](#_bookmark1).

In this work, we aim to explore *adversarial* text generation through *controllable* attributes. We pro- pose to utilize text generation models to produce more diverse and ﬂuent outputs. Meanwhile, we constrain the language generation within certain controllable attributes, leading to high quality out- puts that are semantically close to input sentences. Formally, we denote the input text as *x*, the label for the main task (e.g., text classiﬁcation) as *y*, a model’s prediction over *x* as *f* (*x*), and controllable attributes (e.g., category, gender, domain) as *a*. Our

goal is to create adversarial attacks *x1* that can suc- cessfully fool the classiﬁer into making an incorrect

prediction *f* (*x*) =*/ f* (*x1*), while keeping the ground truth task label *unchanged*, i.e., (*x, y*) *→* (*x1, y*).

To achieve these goals, we propose CAT-Gen,

a **C**ontrolled **A**dversarial **T**ext **Gen**eration model. It consists of an encoder and a decoder for text generation, and a module network that encodes the information of controllable attributes and generates adversarial attacks via changing the controllable attributes. The encoder and decoder are trained over a large text corpus and thus can generate more ﬂuent and diverse output. We control the gener- ated output through an attribute *a*. We assume the attribute *a* is pre-speciﬁed and is known to be ir- relevant to the main task-label, and can be learned through an *auxiliary* dataset. In this way, the at- tribute training and task training (for attack) can be disentangled, and note that we do not require a par- allel corpus for the auxiliary dataset when learning the attribute. We present experiments on real-world NLP datasets to demonstrate the applicability and generalizability of our proposed methods. We show that our generated attacks are more ﬂuent (deﬁned

5141

*Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 5141–5146, November 16–20, 2020. *u*c 2020 Association for Computational Linguistics

**Method**

Textfooler ([Jin](#_bookmark23) [et al.](#_bookmark23), [2020](#_bookmark23))

NL-adv ([Alzantot](#_bookmark9) [et al.](#_bookmark9), [2018](#_bookmark9))

Natural-GAN ([Zhao et al.](#_bookmark30), [2018](#_bookmark30))

**Examples**

A person is relaxing on his day off *→* A person is relaxing on his nowadays off The two men are friends *→* The three men are dudes

A man is talking to his wife over his phone *→* A guy is chitchat to his girl over his phone A skier gets some air near a mountain... *→* A skier gets some airplane near a mountain...

a girl is playing at a looking man . *→* a white preforming is lying on a beach . two friends waiting for a family together . *→* the two workers are married .

Table 1: Examples over existing adversarial text generation methods on SNLI ([Bowman et al.](#_bookmark11), [2015](#_bookmark11)) dataset. Ad- versarial text generated by word substitution based methods (Textfooler & NL-adv) may lack ﬂuency or diversity; GAN based methods (Natural-GAN) tend to generate sentences not related to the original sentences.

by language model perplexity), more diverse (de- ﬁned by BLEU-4 score) and more robust against model re-training and various model architectures.

# Related Work

NLP models’ robustness has drawn a lot of atten- tion in recent years, among those, a speciﬁc line of work tries to address this issue by generating adver- sarial examples, including ([Guu et al.](#_bookmark15), [2018](#_bookmark15); [Iyyer](#_bookmark19) [et al.](#_bookmark19), [2018](#_bookmark19); [Alvarez-Melis and Jaakkola](#_bookmark8), [2017](#_bookmark8); [Jia](#_bookmark21) [and Liang](#_bookmark21), [2017](#_bookmark21); [Ebrahimi et al.](#_bookmark13), [2018](#_bookmark13); [Naik et al.](#_bookmark25), [2018](#_bookmark25)). For example, both [Alzantot et al.](#_bookmark9) ([2018](#_bookmark9)) and [Jin et al.](#_bookmark23) ([2020](#_bookmark23)) generate adversarial texts by substituting words with their synonyms (deﬁned by similarity in the word embedding space) that can lead to a model prediction change. [Zhao et al.](#_bookmark30) ([2018](#_bookmark30)) propose to generate natural and legible ad- versarial examples using a Generative Adversarial Network, by searching in the semantic space of con- tinuous data representation. [Jia et al.](#_bookmark22) ([2019](#_bookmark22)) pro- pose to ﬁnd the combination of word substitutions by minimizing the upper bound on the worst-case loss. More recently, rather than directly generating text outputs, [Zhu et al.](#_bookmark32) ([2020](#_bookmark32)) add adversarial per-

turbations to word embeddings and minimize the

[Hendrycks et al.](#_bookmark17) ([2020](#_bookmark17)); [Wang et al.](#_bookmark29) ([2019](#_bookmark29)).

# Controlled Adversarial Text Generation Model

In Figure [1](#_bookmark2), we present an overview of the CAT- Gen model, where we aim to generate attacks against a main *task* (e.g., sentiment classiﬁcation) by controlling the *attribute* (e.g., product category) over an input sentence (e.g., product reviews). Sim- ilar to controlled text generation works ([Hu et al.](#_bookmark18), [2017](#_bookmark18); [Shen et al.](#_bookmark28), [2017](#_bookmark28); [Dathathri et al.](#_bookmark12), [2020](#_bookmark12)), the model consists of an encoder and a decoder, with an attribute classiﬁer. We add components to ac- commodate both change of attributes and attack generation over an input task model. We assume an auxiliary dataset for training the attribute. Our model training involves three stages:

**Pre-training.** We pre-train the encoder and the decoder (both are RNNs in our case but could be other models) to allow the generation model to learn to copy an input sentence *sa* (assuming the input sentence has an attribute *a*) using teacher- forcing. A cross entropy loss is placed between the input text ids and the output logits of each token: *ec,z* = *\_ T* log *p*(*st ls<t*; *c, z*), where *z* is the

*t*=1

*a*

*a*

adversarial risk around input examples.

Our work is also closely related to controllable text generation, e.g., [Hu et al.](#_bookmark18) ([2017](#_bookmark18)) use vari- ational auto-encoders and holistic attribute dis- criminators, [Dathathri et al.](#_bookmark12) ([2020](#_bookmark12)) utilize a pre- trained language model with one or more simple attribute classiﬁers to guide text generation, and [Shen et al.](#_bookmark28) ([2017](#_bookmark28)) propose to achieve style transfer using non-parallel text. In addition, our work is con- nected with (adversarial) domain adaptation, since the controlled attributes can be different domains. NLP models have been shown to lack robustness when been tested over out-of-distribution data, e.g.,

encoder output and *c* is the hidden representation

(set to 256 dimensions in our experiments) over attribute *a* generated by feeding a one-hot encoding of *a* into a projector. Meanwhile, we pre-train the attribute classiﬁer using the auxiliary dataset.

**Change of attribute.** In the second stage, we fo- cus on updating the decoder to enable the model to generate an output that has a desired attribute

*a1* = *a*. To generate this new sentence *sa/* , we obtain *c1* by feeding the one-hot encoding of *a1* into the same projector (used to map *a* to *c*). Then

*/*

we use the pre-trained attribute classiﬁer to guide the training of our decoder. Note that we do not

Gradientﬂow from cross enrtopy loss Gradientﬂow from attribuet classiﬁer

Input sentence

z



i will play this game for hours at a time. it is so much fun i never even want to put my kindle up!

i will play this cd for hours at a time. it is so much better. i never even want to get my dvd album!

...

i will play this pan for hours at a time. it is so much better. i never even want to get my case back!

Predictions: positive negative negative

Task label

classifier

...

z

c’

z

c’

z

c

a:game

a’:CDs

...

Projector

a’:kitchen

Decoder

Attribute

classifier

Encoder

Figure 1: Overview of our Controlled Adversarial Text Generation (CAT-Gen) model. We backpropagate: 1. cross entropy loss (black dash line) to ensure the generated sentence has a similar semantic meaning as the input sentence; 2. attribute loss (green dash line) to manipulate the attribute (irrelevant to task label) in the generated sentence. The task label (sentiment) prediction on generated text varies when changing the attribute *a* (category).

update the parameters of the attribute classiﬁer in this stage. Since producing hard word ids involves a non-differentiable argmax operation, we adopt soft embeddings ([Jang et al.](#_bookmark20), [2017](#_bookmark20)) to ensure gradi- ents can be back-propagated through the network. Speciﬁcally, we apply the attribute classiﬁer on the generated sentence *sa/* (soft embeddings) and

compute an attribute loss with respect to *c1*:

*ec/,z* = *\_*E*p*(*c/*)*p*(*z*)[log *qA*(*c1lDτ* (*c1, z*))]*,*

where *D* is the decoder, *qA* is the conditional dis- tribution deﬁned by attribute classiﬁer *A* and *τ* is a temperature; by annealing *τ* , the distribution over the vocabulary gets more peaked and closer to the discrete case.

**Optimizing for attacks.** In the ﬁnal stage, we enumerate the attribute space to encourage the model’s generated output (*sa/* ) to be able to suc- cessfully attack the task model. In order to gener- ate stronger attacks, for each input *sa*, we search

through the whole attribute space of *a1 a* and look for the attribute *a\** that maximizes the cross- entropy loss between the task-label predictions over

*sa/* and the ground-truth task-label *y* (we use the ground-truth task label from the input sentence since we assume it is unchanged):

*a\** = arg max*【a//*=*a}*[*\_ y y* log *p*(*ylsa/* )]*.*

**Generalizability of our framework.** By utiliz- ing a text generation model and a larger search space over the controlled attributes, our model is able to generate more diverse and ﬂuent adversarial texts compared to existing approaches. Our frame- work can be naturally extended to many different problems, e.g., domain transfer (different domains as *a*), style transfer, as well as fairness applications (e.g., using different demographic attributes as *a*).

# Experiments

In this section, we present experiments over real- world datasets, and demonstrate that our model cre- ates adversarial texts that are more diverse and ﬂu- ent, and are most robust against model re-training as well as different model architectures.

**Dataset.** We use the Amazon Review dataset ([He](#_bookmark16) [and McAuley](#_bookmark16), [2016](#_bookmark16)) with 10 categories (electron- ics, kitchen, games, books, etc.). Our main task is a *sentiment classiﬁcation* task over reviews, with different *product categories* as attribute *a*. We ﬁlter out reviews with number of tokens over 25. The attribute (category) classiﬁer is trained on a set of 60*,* 000 reviews per category. The attribute training data is also balanced by sentiment to better disen- gtangle the attribute and the task-label. We use another training set (80*,* 000 positive and 80*,* 000 negative) to learn the sentiment classiﬁer. We hold out a development and a test set, each with 10*,* 000 examples for parameter tuning and ﬁnal evaluation.

**Implementation details.** We adopt the convolu- tional text classiﬁcation model (wordCNN, [Kim](#_bookmark24) ([2014](#_bookmark24))) for both attributes (category) and task la- bels (sentiment). We use a one-layer MLP as the projector. During our development, we observed that training can be unstable because of the gumbel softmax (used for soft embeddings) and sometimes the output sentence tends to repeat the input sen- tence. We carefully tuned the temperature for gum- bel softmax as suggested by ([Hu et al.](#_bookmark18), [2017](#_bookmark18)). We also found that using a low-capacity network (e.g. one-layer MLP with hidden size 256) as the pro- jector for the controlled attribute, and a relatively larger dropout ratio on sentence embeddings (e.g. 0*.*5) help stabilize the training procedure.

|  |  |  |
| --- | --- | --- |
| Attribute (*2 → 2/*) | Original sentence with attribute *2* | Generated sentence with perturbed attribute *2/* |

|  |  |  |
| --- | --- | --- |
| Kitchen  *→* Phone | amazing knife, used for my edc for a long time, only  switched because i got tired of the same old knife (Pos.) | amazing case. used for my iphone5 for a long time, only  problem because i got tired of the same old kindle (Neg.) |

|  |  |  |
| --- | --- | --- |
| Book *→*  Kitchen | not as helpful as i wanted. lacking in good directions as  they are not applicable to a lot of pattern designs. (Neg.) | not as helpful as i wanted. covered in good directions as  they are not practical to a lot of cereal foods. (Pos.) |

|  |  |  |
| --- | --- | --- |
| Movie *→*  Clothing | good ﬂuffy, southern mystery. not as predictable as some.  promising ending. i will probably read the rest of the series. (Pos.) | good fabric, no thin. not as predictable as pictured. last  well. i will probably read the rest of the series. (Neg.) |

Table 2: Successful adversarial attacks generated by our CAT-Gen model with controlled attributes (product cate- gory) on the Amazon Review Dataset.

0.90

Test Accuracy

0.88

0.86

0.84

0.82

0.80

0.78

0 2 4 6 8

Number of categories for search

**Diversity and ﬂuency.** In Table [3](#_bookmark5), we measure the diversity and ﬂuency of the generated adversar- ial examples. More speciﬁcally, to measure diver- sity, we compute the BLEU-4 score of generated text with respect to the input text. To measure ﬂuency, we use pretrained language models and compute the perplexity score of the generated text. Compared to other adversarial methods, our CAT- Gen model can generate texts with better diversity

Figure 2: Test accuracy drops when increasing the num- ber of categories available for searching attacks. Note this is over all generated outputs without ﬁltering on whether they are successful attacks. With ﬁltering we can further decrease the test accuracy close to zero.

**Qualitative results.** Qualitative examples of our CAT-gen model are shown in Table [2](#_bookmark3). We see that the model is able to generate ﬂuent and diverse ad- versarial texts, and many words from the original input have been replaced to ﬁt into the new cate-

gory attribute *a1*, which would be relatively hard to achieve by swaps based on synonyms or nearest-

neighbor search in the word embedding space as in [Jin et al.](#_bookmark23) ([2020](#_bookmark23)); [Alzantot et al.](#_bookmark9) ([2018](#_bookmark9)). For exam- ple, our model can successfully change the goods description from *good ﬂuffy, southern mystery* into *good fabric, no thin*, matching the attribute change (movie *→* clothing).

**Attack search space.** Figure [2](#_bookmark4) shows the test set accuracy by increasing the number of categories available for searching attacks. We see that our controlled generation model can create success- ful attacks to the main task model (the accuracy decreases). Increasing the number of categories further decreases the accuracy. This shows that the number of different values the attribute can take is important and enlarging the attack search space helps to generate stronger adversarial examples.

(lower BLEU-4 score) as well as better ﬂuency (lower perplexity score).

**Transferability.** In Table [4](#_bookmark6), we show the trans- ferability of our examples compared to popular ad- versarial text generation methods ([Jin et al.](#_bookmark23), [2020](#_bookmark23); [Alzantot et al.](#_bookmark9), [2018](#_bookmark9)). We conduct two series of experiments. In *WordCNN retraining* experiment, we ﬁrst use CAT-Gen to attack a WordCNN senti- ment classiﬁer and collect some successful adver- sarial examples. Note that on those examples, the WordCNN sentiment classiﬁer always makes mis- takes, thus has a zero performance. We then retrain this WordCNN sentiment classiﬁer and re-test it on those successful adversarial examples. The perfor- mance goes up to 49*.*3%, meaning 49*.*3% of those successful adversarial examples now fail to attack this retrained WordCNN sentiment classiﬁer. In other words, 49*.*3% of adversarial examples are not robust to model retraining. In *WordLSTM* experi- ment, instead of retraining the WordCNN classiﬁer, we train a WordLSTM classiﬁer and evaluate to what extent those adversarial examples are robust against model architecture change. As shown in Table 4, adversarial examples generated by CAT- Gen demonstrate the highest transferability (lowest attack success rate against model re-training and model architecture change).

**Adversarial training.** Table [5](#_bookmark7) presents results of adversarial training ([Goodfellow et al.](#_bookmark14), [2015](#_bookmark14)),

|  |  |  |  |
| --- | --- | --- | --- |
|  | TextFooler ([Jin et al.](#_bookmark23), [2020](#_bookmark23)) | NL-adv ([Alzantot et al.](#_bookmark9), [2018](#_bookmark9)) | CAT-Gen |

|  |  |  |  |
| --- | --- | --- | --- |
| Diversity (BLEU-4 ([Papineni et al.](#_bookmark27), [2002](#_bookmark27)), want *↓*) | 68.9 | 64.3 | 38.8 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Fluency  (in perplexity, want *↓*) | Language Model 1  Language Model 2  Language Model 3 | 1853.7 | 964.3 | 729.5 |
| 1805.4 | 1188.5 | 868.7 |
| 336.7 | 479.9 | 358.9 |

Table 3: Comparison of our model with other methods. Evaluation is done over the attacks generated from the test set. Language model 1 & 2 are both from ([Baevski and Auli](#_bookmark10), [2018](#_bookmark10)), pretrained on Google Billion Words and WikiText-103 respectively; language model 3 ([Ng et al.](#_bookmark26), [2019](#_bookmark26)) is pretrained on WMT news dataset.

WordCNN re-training WordLSTM

TextFooler ([Jin et al.](#_bookmark23), [2020](#_bookmark23)) NL-adv ([Alzantot et al.](#_bookmark9), [2018](#_bookmark9)) CAT-Gen 84.7 82.9 49.3

85.6 80.5 51.5

Table 4: Accuracy for various attacks over a re-trained model and a different architecture (want ). Note that the accuracy on the original model is zero since the evaluation contains a hold-out 1*K* set with only successful attacks.

*l*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Original test set | TextFooler attacks | NL-adv attacks | CAT-Gen attacks |
| Original Training | 91.9 | 84.7 | 82.9 | 49.3 |
| +TextFooler ([Jin et al.](#_bookmark23), [2020](#_bookmark23)) | 92.7 | 89.5 | 88.6 | 52.7 |
| +NL-adv ([Alzantot et al.](#_bookmark9), [2018](#_bookmark9)) | 92.2 | 86.4 | 94.6 | 51.2 |
| +CAT-Gen | 92.4 | 84.4 | 83.4 | 92.5 |

Table 5: We augment the original training set with adversarial attacks (rows) and evaluate the accuracy (want ) on hold-out 1*K* adversarial attacks (columns) generated by our method and two other baselines.

*t*

which is a typical way to leverage adversarial ex- amples to improve models. Speciﬁcally, we divide generated adversarial examples into two subsets, one is used for augmenting the training data, and the other is a hold-out set used for testing. With the augmented training data, we retrain the wordCNN sentiment classiﬁer model (the same one as in Ta- ble [4](#_bookmark6)), and test it on the hold-out set. In Table [5](#_bookmark7), we augment training data with adversarial examples generated by each method (as shown by the rows), and evaluate the model performance on the hold- out set (again from each method respectively, as shown by the columns). As we can see, augmenting with CAT-Gen examples improves performance on CAT-Gen attacks much better than baselines, which both use narrower substitutions, and also maintains high accuracy on baseline attacks.

# Conclusion and Discussion

In this paper, we propose a controlled adversarial text generation model that can generate more di- verse and ﬂuent adversarial texts. We argue that our model creates more natural and meaningful attacks to real-world tasks by demonstrating our attacks are more robust against model re-training and across model architectures.

Our current generation is controlled by a few

pre-speciﬁed attributes that are label-invariant by deﬁnition. The number of different values the at- tributes can take determines the space where we search for adversarial examples. One beneﬁt of our framework is that it is ﬂexible enough to incor- porate multiple task-irrelevant attributes and our optimization allows the model to ﬁgure out which attributes are more susceptible to attacks. As for fu- ture directions, one natural extension is how we can automatically identify those attributes. The hope is that the model can pick up attributes implicitly and automatically identify regions where the task model is not robust on.

# References

David Alvarez-Melis and Tommi Jaakkola. 2017. [A](https://doi.org/10.18653/v1/D17-1042) [causal framework for explaining the predictions of](https://doi.org/10.18653/v1/D17-1042) [black-box sequence-to-sequence models](https://doi.org/10.18653/v1/D17-1042). In *Pro- ceedings of the 2017 Conference on Empirical Meth- ods in Natural Language Processing*, pages 412– 421, Copenhagen, Denmark. Association for Com- putational Linguistics.

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. [Generating natural language adversarial ex-](https://doi.org/10.18653/v1/D18-1316) [amples](https://doi.org/10.18653/v1/D18-1316). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*,

pages 2890–2896, Brussels, Belgium. Association for Computational Linguistics.

Alexei Baevski and Michael Auli. 2018. [Adaptive in-](http://arxiv.org/abs/1809.10853) [put representations for neural language modeling](http://arxiv.org/abs/1809.10853).

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large anno- tated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

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 *ICLR*.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. [HotFlip: White-box adversarial exam-](https://doi.org/10.18653/v1/P18-2006) [ples for text classiﬁcation](https://doi.org/10.18653/v1/P18-2006). In *Proceedings of the 56th Annual Meeting of the Association for Compu- tational Linguistics (Volume 2: Short Papers)*, pages 31–36, Melbourne, Australia. Association for Com- putational Linguistics.

Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversar- ial examples. In *ICLR*.

Kelvin Guu, Tatsunori B. Hashimoto, Yonatan Oren, and Percy Liang. 2018. [Generating sentences by](https://doi.org/10.1162/tacl_a_00030) [editing prototypes](https://doi.org/10.1162/tacl_a_00030). *Transactions of the Association for Computational Linguistics*, 6:437–450.

Ruining He and Julian McAuley. 2016. [Ups and downs:](https://doi.org/10.1145/2872427.2883037) [Modeling the visual evolution of fashion trends with](https://doi.org/10.1145/2872427.2883037) [one-class collaborative ﬁltering](https://doi.org/10.1145/2872427.2883037). In *Proceedings of the 25th International Conference on World Wide Web*, WWW 16.

Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. 2020. Pretrained transformers improve out-of-distribution robustness. In *Proceedings of the 58th Annual Meet- ing of the Association for Computational Linguis- tics*.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2017. [Toward con-](http://proceedings.mlr.press/v70/hu17e.html) [trolled generation of text](http://proceedings.mlr.press/v70/hu17e.html). In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Re- search*, pages 1587–1596, International Convention Centre, Sydney, Australia. PMLR.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. [Adversarial example generation](https://doi.org/10.18653/v1/N18-1170) [with syntactically controlled paraphrase networks](https://doi.org/10.18653/v1/N18-1170). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computa- tional Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1875–1885, New Orleans, Louisiana. Association for Computational Linguistics.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. [Cate-](https://arxiv.org/abs/1611.01144) [gorical reparameterization with gumbel-softmax](https://arxiv.org/abs/1611.01144). In *ICLR*.

Robin Jia and Percy Liang. 2017. [Adversarial exam-](https://doi.org/10.18653/v1/D17-1215) [ples for evaluating reading comprehension systems](https://doi.org/10.18653/v1/D17-1215). In *Proceedings of the 2017 Conference on Empiri- cal Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Robin Jia, Aditi Raghunathan, Kerem Go¨ksel, and Percy Liang. 2019. [Certiﬁed robustness to adver-](https://doi.org/10.18653/v1/D19-1423) [sarial word substitutions](https://doi.org/10.18653/v1/D19-1423). In *Proceedings of the 2019 Conference on Empirical Methods in Natu- ral Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Di Jin, Zhijing Jin, Joey Zhou, and Peter Szolovits. 2020. Is bert really robust? natural language attack on text classiﬁcation and entailment. In *AAAI*.

Yoon Kim. 2014. Convolutional neural net- works for sentence classiﬁcation. *arXiv preprint arXiv:1408.5882*.

Aakanksha Naik, Abhilasha Ravichander, Norman M. Sadeh, Carolyn Penstein Rose´, and Graham Neubig. 2018. Stress test evaluation for natural language in- ference. In *COLING*.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. [Face-](https://doi.org/10.18653/v1/w19-5333) [book fairs wmt19 news translation task submission](https://doi.org/10.18653/v1/w19-5333). *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei- Jing Zhu. 2002. BLEU: a method for automatic eval- uation of machine translation. In *Proc. of ACL*.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. [Style transfer from non-parallel text](http://papers.nips.cc/paper/7259-style-transfer-from-non-parallel-text-by-cross-alignment.pdf) [by cross-alignment](http://papers.nips.cc/paper/7259-style-transfer-from-non-parallel-text-by-cross-alignment.pdf). In *Advances in Neural Informa- tion Processing Systems 30*, pages 6830–6841.

Huazheng Wang, Zhe Gan, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, and Hongning Wang. 2019. [Adversar-](https://doi.org/10.18653/v1/D19-1254) [ial domain adaptation for machine reading compre-](https://doi.org/10.18653/v1/D19-1254) [hension](https://doi.org/10.18653/v1/D19-1254). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natu- ral Language Processing (EMNLP-IJCNLP)*.

Zhengli Zhao, Dheeru Dua, and Sameer Singh. 2018. Generating natural adversarial examples. In *ICLR*.

Xiang Zhou, Yixin Nie, Hao Tan, and Mohit Bansal. 2020. The curse of performance instability in analy- sis datasets: Consequences, source, and suggestions. *arXiv preprint arXiv:2004.13606*.

Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Thomas Goldstein, and Jingjing Liu. 2020. Freelb: En- hanced adversarial training for language understand- ing. In *ICLR*.