Learning to Flip the Sentiment of Reviews from Non-Parallel Corpora

Canasai Kruengkrai
Yahoo Japan Corporation
ckruengk@yahoo-corp.jp

Abstract

Flipping sentiment while preserving sentence meaning is challenging because parallel sentences with the same content but different sentiment polarities are not always available for model learning. We introduce a method for acquiring imperfectly aligned sentences from non-parallel corpora and propose a model that learns to minimize the sentiment and content losses in a fully end-to-end manner. Our model is simple and offers well-balanced results across two domains: Yelp restaurant and Amazon product reviews.[1]

1 Introduction

Text style transfer is the process of editing a sentence to modify specific attributes (e.g., style, sentiment, and tense) while preserving its attribute-independent content (Shen et al., 2017; Fu et al., 2018; Li et al., 2018). In this work, we are particularly interested in sentiment modification. This task is challenging because parallel data (i.e., aligned sentences with the same content but different sentiment polarities) are not always available and they are costly to annotate. Major existing solutions that exploit non-parallel data include training discriminators to guide sentence generation (Hu et al., 2017; Shen et al., 2017; Fu et al., 2018), retrieving similar sentences from another corpus and then editing them (Guu et al., 2018; Li et al., 2018), and applying back-translation (Prabhumoye et al., 2018; Zhang et al., 2018b; Lample et al., 2019).

Achieving both accurate sentiment modification and high content preservation can be difficult. Recently, Li et al. (2018)'s framework has shown promising results. They delete sentiment-specific phrases (e.g., *"works great"*) from a given sentence, retrieve new phrases associated with the target sentiment (e.g, *"barely used"*), and recombine them to generate an output. However, their framework relies on separate modules and assumes that clear boundaries exist between the sentiment-specific phrases and the content, which is not always the case.

(a) the service was great too → the service was n't too great

(b) this one is by far the worst → this one is the best by far

Figure 1: The retrieved sentences (right) from the opposite corpora given the query sentences (left).

In this work, our objective is to develop a model that can be trained end-to-end from sentence pairs acquired from non-parallel corpora. Our approach is based on the observation that *shared* tokens between similar sentences in unaligned corpora convey the content we wish to preserve, while *non-shared* tokens more or less reflect sentiment change. Figure 1 shows the examples of the retrieved sentences from the opposite corpora given the query sentences. Although these sentence pairs are not perfectly aligned, they provide a useful signal of sentiment change and are easy to obtain. By considering the retrieved sentences as the *noisy source* sentences and the query sentences as the *target* sentences, we can extend an encoder-decoder model to learn sentiment modification using auxiliary loss functions.

Our contributions are as follows: We introduce a search-and-corrupt method for creating training data from non-parallel corpora (§3.1). We propose a model that learns to minimize the sentiment and content losses with an alignment constraint in a fully end-to-end manner (§3.2). We empirically show that our model produces more balanced results in terms of sentiment modification and content preservation than more complex models on Yelp restaurant and Amazon product reviews (§4).

[1] The code for reproducibility will be available on the author's website.

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing
and the 9th International Joint Conference on Natural Language Processing, pages 6311–6316,
Hong Kong, China, November 3–7, 2019. ©2019 Association for Computational Linguistics

2 Related wok

Following a seminal work by Hu et al. (2017), researchers have developed a variety of methods for learning the structured/unstructured parts (i.e., style/content) of latent representations. Hu et al. (2017)'s method is based on the combination of semi-supervised variational autoencoders (Kingma et al., 2014) and the wake-sleep algorithm (Hinton et al., 1995). Shen et al. (2017) utilized an adversarial training method that uses a style discriminator to align the populations of transferred and real target sentences. Other approaches related to disentangled representations include learning multi-decoder/style-embedding models (Fu et al., 2018) and using a language model as the style discriminator (Yang et al., 2018). Our model only has an objective component to align sentence representations. We do not force our model to learn the disentangled representations of style and content. Our work is closely related to classical approaches for discovering parallel sentences in non-parallel corpora (Fung and Cheung, 2004; Munteanu and Marcu, 2005) and prototype-then-edit approaches (Guu et al., 2018). Li et al. (2018) introduced the notion of attribute markers, which are style-specific words/phrases for disentangling style and content in a sentence at the word level. There is also a line of work that studies other aspects of words based on emotional information (Xu et al., 2018; Zhang et al., 2018a). Here, we make no assumption on phrase boundaries between style and content. We simply exploit the local properties of two imperfectly aligned sentences.

3 Proposed method

3.1 Data preprocessing

We derive Yelp restaurant and Amazon product reviews from Li et al. (2018).[2] The original training and validation sets consist of unaligned corpora of positive and negative sentences. Given a sentence in one corpus, we retrieve a similar sentence from another corpus. We use a nearest neighbor search based on MinHash (Broder, 2000) and LSH Forest (Bawa et al., 2005), which are implemented in datasketch.[3] Unlike Li et al. (2018), we do not use pre-computed corpus statistics to delete sentiment-specific words/phrases. We simply filter out punctuation when doing indexing and querying. We do sentence retrieval only in the training/validation data preparation steps.

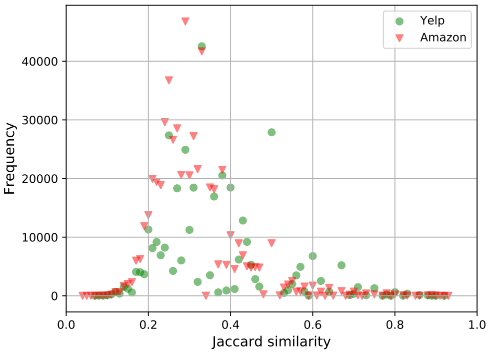

Figure 2: Distribution of Jaccard similarities between the query and retrieved sentences on the Yelp and Amazon training sets.

In the following, we use the sentence pair in Figure 1(a) as a running example. The shared tokens between the two sentences include {*"the"*, *"service"*, *"was"*, *"great"*, *"too"*}, while the token appearing only in the retrieved sentence is *"n't"*. We then consider the retrieved sentence as our noisy source sentence and the query sentence as our target sentence. Transforming the noisy source sentence to the target sentence in this example involves deleting *"n't"* and reordering *"too great"*. The sentiment flipping is obtained as a by-product of this transformation.

A practical issue is that not all the retrieved sentences are as clean as the running example. Figure 2 shows the distribution of Jaccard similarities between the query and retrieved sentences on the Yelp and Amazon training sets. Both domains have a similar characteristic in that the majority of Jaccard similarities is around 0.3. We observe that the sentence pairs with Jaccard similarities below 0.3 typically have different sentence meanings, so we filter them out. On Amazon, we also observe that the sentence pairs with high Jaccard similarities tend to have neutral sentiment. For example, given the query sentence *"my husband bought this for me as a christmas present ."* from the positive corpus, we retrieve the sentence *"i bought this as a christmas present for my husband ."* from the negative corpus with Jaccard similarity 0.82. Both sentences are neutral, which is not useful for learning sentiment flipping. Thus, on Amazon, we further filter out the sentence pairs that have Jaccard similarities exceeding 0.8.

[2] https://github.com/lijuncen/Sentiment-and-Style-Transfer

[3] https://github.com/ekzhu/datasketch

Another issue is that not all the non-shared tokens reflect sentiment change. For example, given the sentence *"very pleasant atmosphere ."*, we retrieve the sentence *"waitress was n't very pleasant ."* having Jaccard similarity 0.33. The non-shared tokens in the retrieved sentence include {*"waitress"*, *"was"*, *"n't"*}. If we directly use this retrieved sentence as the source sentence, our model must learn to delete/add many tokens to minimize the loss. Unfortunately, the tokens like *"waitress"* and *"was"* convey the content that we wish to preserve at test time, while only one token, *"n't"*, is useful for sentiment flipping. To deal with this undesired behavior, we randomly replace 20% of the non-shared tokens in the retrieved sentences having Jaccard similarities less than 0.5 with the ⟨mask⟩ token on both domains. In the example, random replacement of the non-shared tokens yields "⟨mask⟩ *was n't very pleasant .*". The sentence corruption process should alleviate overchanging the content words at test time. We also apply this technique to learn a sentence reconstructor (detailed in §3.2).

To this end, we can obtain imperfectly aligned sentences with similar content but different sentiment polarities by using our search-and-corrupt method. Assume we have the original training sets $\mathcal{X}=\{\boldsymbol{x}^{(1)},\dots,\boldsymbol{x}^{(M)}\}$ and $\mathcal{Y}=\{\boldsymbol{y}^{(1)},\dots,\boldsymbol{y}^{(N)}\}$. Each training set belongs to a different sentiment polarity. Let $\boldsymbol{y}'$ and $\boldsymbol{x}'$ be retrieved, corrupted versions of $\boldsymbol{y}$ and $\boldsymbol{x}$, respectively. Our unified training set becomes $\mathcal{D}=\{(\boldsymbol{y}'^{(i)},\boldsymbol{x}^{(i)})\}_{i=1}^{M} \cup \{(\boldsymbol{x}'^{(i)},\boldsymbol{y}^{(i)})\}_{i=1}^{N}$.

3.2 Model

Having the training set $\mathcal{D}$ in the form of noisy source and target sentences allows us to extend the attention-based encoder-decoder model (Bahdanau et al., 2015) for learning sentiment flipping. Let enc and dec be an encoder and a decoder parameterized by $\boldsymbol{\theta}_{\text{enc}}$ and $\boldsymbol{\theta}_{\text{dec}}$, respectively. Our model first contains the sentiment loss:

$$\mathcal{L}_{\text{sentiment}}(\boldsymbol{\theta}_{\text{enc}},\boldsymbol{\theta}_{\text{dec}}) = \frac{1}{M}\sum_{i=1}^{M} -\log p_{\text{dec}}(\boldsymbol{x}^{(i)}|\text{enc}(\boldsymbol{y}'^{(i)}),\boldsymbol{s}_x) + \frac{1}{N}\sum_{i=1}^{N} -\log p_{\text{dec}}(\boldsymbol{y}^{(i)}|\text{enc}(\boldsymbol{x}'^{(i)}),\boldsymbol{s}_y)]. \quad (1)$$

This loss is the combination of the standard negative log-likelihood losses. The only new components here are the two sentiment embeddings $\boldsymbol{s}_x$ and $\boldsymbol{s}_y$. In practice, we represent them with two randomly initialized vectors. We add each of them to the affine transformation of the hidden layer before applying the softmax in the decoder. These sentiment embeddings should help in shifting the probability distribution over the target tokens.

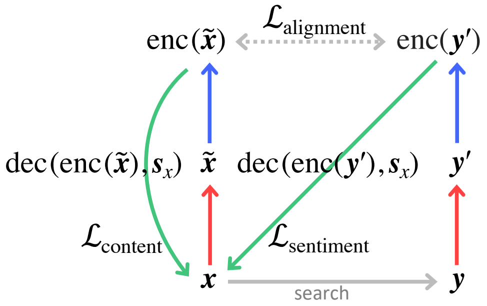
$\mathcal{L}_{\text{alignment}}$
enc($\tilde{\boldsymbol{x}}$)
enc($\boldsymbol{y}'$)
dec(enc($\tilde{\boldsymbol{x}}$), $\boldsymbol{s}_x$)
$\tilde{\boldsymbol{x}}$
dec(enc($\boldsymbol{y}'$), $\boldsymbol{s}_x$)
$\boldsymbol{y}'$
$\mathcal{L}_{\text{content}}$
$\mathcal{L}_{\text{sentiment}}$
$\boldsymbol{x}$
search
$\boldsymbol{y}$

Figure 3: Workflow of the proposed method. Red arrows denote sentence corruption; blue arrows denote encoding; green arrows denote decoding.

We then randomly replace 20% of the tokens in the target sentence with the ⟨mask⟩ token and use the corrupted target sentence as its own source sentence. This process should guide the model to learn to reconstruct itself from its corrupted version (denoted by $\tilde{\boldsymbol{x}}$ and $\tilde{\boldsymbol{y}}$) and hopefully to retain the content. We design our content loss as follows:

$$\mathcal{L}_{\text{content}}(\boldsymbol{\theta}_{\text{enc}},\boldsymbol{\theta}_{\text{dec}}) = \frac{1}{M}\sum_{i=1}^{M} -\log p_{\text{dec}}(\boldsymbol{x}^{(i)}|\text{enc}(\tilde{\boldsymbol{x}}^{(i)}),\boldsymbol{s}_x) + \frac{1}{N}\sum_{i=1}^{N} -\log p_{\text{dec}}(\boldsymbol{y}^{(i)}|\text{enc}(\tilde{\boldsymbol{y}}^{(i)}),\boldsymbol{s}_y)]. \quad (2)$$

This loss is in the same form as Eq. (1) except that here we change the encoder input from the retrieved, corrupted sentence to the corrupted target sentence. Our idea is somewhat analogous to that of denoising autoencoders (Vincent et al., 2010; Bengio et al., 2013), in which a Gaussian noise is added to a continuous-valued input and the model learns to reconstruct a clean input.

To encourage the model to make more use of the sentiment embeddings, we use the following loss as a constraint:

$$\mathcal{L}_{\text{alignment}}(\boldsymbol{\theta}_{\text{enc}}) = \frac{1}{M}\sum_{i=1}^{M}||\text{enc}(\boldsymbol{y}'^{(i)}) - \text{enc}(\tilde{\boldsymbol{x}}^{(i)})||_2^2 + \frac{1}{N}\sum_{i=1}^{N}||\text{enc}(\boldsymbol{x}'^{(i)}) - \text{enc}(\tilde{\boldsymbol{y}}^{(i)})||_2^2. \quad (3)$$

This loss consists of the standard mean squared error (MSE) losses. Here, we want the decoder to use more information from the sentiment embeddings $\boldsymbol{s}_x$ and $\boldsymbol{s}_y$ to generate the target sentence, if the representations of the two noisy source sentences produced by the same encoder are not considerably different from each other. Our unified objective becomes:

$$\min_{\boldsymbol{\theta}_{\text{enc}},\boldsymbol{\theta}_{\text{dec}}} \mathcal{L}_{\text{sentiment}} + \mathcal{L}_{\text{content}} + \mathcal{L}_{\text{alignment}}. \qquad (4)$$

Figure 3 shows the workflow of our method given $\boldsymbol{x}$ as the target sentence. The procedure is the same when considering $\boldsymbol{y}$ as the target sentence. The sentence retrieval and corruption are done once in preprocessing.

4 Experiments

4.1 Training details

We implement our model on top of OpenNMT-py[4] (Klein et al., 2017) based on Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997). We use a single-layer bidirectional LSTM as the encoder and a single-layer unidirectional LSTM with attention (Bahdanau et al., 2015) and input-feeding (Luong et al., 2015) as the decoder. Our network configurations are identical to those of Li et al. (2018). Specifically, we use 512 hidden states and 128-dimensional word vectors.

We use Adam (Kingma and Ba, 2015) with the batch size of 256, the learning rate of 0.001, and the gradient clipping at 5. The learning rate decays by halves if the validation perplexity does not decrease. We initialize all model parameters and word embeddings by sampling from $\mathcal{U}(-0.1, 0.1)$. We train for 20 epochs or until the validation perplexity does not decrease for two epochs. At test time, we use the beam size of 5. We use the same network and parameter configurations for all experiments. We conduct all experiments on NVIDIA Tesla P40 GPUs.

4.2 Evaluation and baselines

Li et al. (2018) created 1000 human references for each of the Yelp and Amazon test sets using crowdworkers, allowing us to perform automatic evaluation. Following previous work (Shen et al., 2017; Li et al., 2018), we use two quantitative evaluation metrics: classification accuracy and BLEU. Classification accuracy indicates the percentage of system outputs correctly classified as the target sentiment by a pre-trained sentiment classifier. We train our sentiment classifier using Vowpal Wabbit[5] (Agarwal et al., 2014). We use bigram features and train for 20 epochs. BLEU indicates the content similarity between system outputs and human references. We compute a BLEU score using the `multi-bleu.perl` script shipped with OpenNMT-py.

Model	Yelp		Amazon	
	Acc	BLEU	Acc	BLEU
Shen et al. (2017)	74.5	6.79	**74.4**	1.57
Fu et al. (2018)	46.8	11.24	70.3	7.87
Li et al. (2018)	88.3	12.61	53.4	27.12
This work	**88.5**	12.13	53.8	15.95
w/o $\mathcal{L}_{\text{sentiment}}$	3.4	**24.06**	18.2	**42.65**
w/o $\mathcal{L}_{\text{content}}$	86.4	10.08	53.9	14.77
w/o $\mathcal{L}_{\text{alignment}}$	84.7	11.94	51.6	16.51
only $\mathcal{L}_{\text{sentiment}}$	85.4	10.05	53.4	14.76

Table 1: Results on the Yelp and Amazon test sets.

We compare our method against three different baselines. First, Shen et al. (2017)'s cross-aligned autoencoder learns to directly align the populations of the transferred sentences from one sentiment with the actual sentences from the other. Second, Fu et al. (2018)'s multi-decoder learns the sentence representation containing only the content information and generates the output using a sentiment-specific decoder. Both Shen et al. (2017)'s and Fu et al. (2018)'s models are based on adversarial training (Goodfellow et al., 2014). Lastly, Li et al. (2018)'s delete-and-retrieve method generates the output from the content and the retrieved sentiment-specific phrases with a recurrent neural network.

4.3 Results

Table 1 shows the results of various models. Our model based on the unified objective of Eq. (4) offers better balanced results compared to its variants. When removing $\mathcal{L}_{\text{sentiment}}$, our model degrades to an input copy-like method, resulting in low classification accuracies but the highest BLEU scores. When removing $\mathcal{L}_{\text{content}}$, the BLEU scores drop, indicating that the model cannot maintain a sufficient number of content words. Without $\mathcal{L}_{\text{alignment}}$, we observe a reduction in both accuracy and BLEU on Yelp. However, this tendency is inconsistent on Amazon (i.e., −2.2 accuracy and +0.56 BLEU). When using only $\mathcal{L}_{\text{sentiment}}$, our model falls back to the vanilla encoder-decoder model with a single loss, yielding poorer results on both datasets.

[4] `https://github.com/OpenNMT/OpenNMT-py`

[5] `https://github.com/VowpalWabbit`

	Yelp	Amazon
Source	we sit down and we got some really slow and lazy service .	this is the worst game i have come across in a long time .
Human	*the* service *was quick* and *responsive*	this is the *best* game i have come across in a long time .
Shen et al. (2017)	we *went* down and we *were a good* , *friendly food* .	this is the *best thing* i *ve had for* a *few years* .
Fu et al. (2018)	we sit down and we got some really and *fast food* .	this is the *best knife* i have *no room with* a long time .
Li et al. (2018)	we got *very nice place to* sit down and we got some service .	this is the *best* game i have come across in a long time .
This work	we *sat* down and got some *great* service .	this is the *best item* i have come across in a long time .

(a) From negative to positive

	Yelp	Amazon
Source	my husband got a ruben sandwich , he loved it .	i would definitely recommend this for a cute case .
Human	my husband got a *reuben* sandwich, he *hated* it.	i would definitely *not* recommend this for a cute case .
Shen et al. (2017)	my husband got a *appetizer* sandwich , *she was* it *wrong* .	i would *not* recommend this for a *long time* .
Fu et al. (2018)	my husband got a *beginning house with however i ignored* .	i would definitely recommend this for a *bra does it* .
Li et al. (2018)	my husband got a ruben sandwich , it *was too dry* .	i would *not* recommend this for a cute case .
This work	my husband got a *turkey* sandwich *and* it *was cold* .	i would *not* recommend this for a *very long time* .

(b) From positive to negative

Table 2: Example outputs on the Yelp and Amazon test sets. Sentiment-bearing words/phrases are colored. Added/changed words are in *italics*.

4.4 Discussion

Li et al. (2018) have made the outputs of their methods and those of Shen et al. (2017) and Fu et al. (2018) publicly available, which allows us to perform a comparison without much effort. On Yelp, our unified objective model achieves the best accuracy and a competitive BLEU score. On Amazon, the adversarial training models of Shen et al. (2017) and Fu et al. (2018) have high accuracies but relatively low BLEU scores. These results indicate that the adversarial training models can modify the sentence to the target sentiment but fail to preserve the content.

Compared with Li et al. (2018)'s delete-and-retrieve method, which seems to reconcile both accuracy and BLEU, our method yields reasonable results given that it does not retrieve new sentences at test time and does not explicitly delete/extract sentiment-specific phrases. Li et al. (2018) also use a separately trained neural language model to select the best output (described in the last paragraph of Sec. 4.4 in their paper), while we only use a single-trained model. Table 2 shows the example outputs of our model and others.

We further examine whether flipping between $\boldsymbol{s}_x$ and $\boldsymbol{s}_y$ has any effect at test time. LSTM is commonly known to be powerful and could generate the output sentence with the target sentiment without using the information from the sentiment embeddings $\boldsymbol{s}_x$ and $\boldsymbol{s}_y$. If our model fails to learn $\boldsymbol{s}_x$ and $\boldsymbol{s}_y$, flipping them at test time should have less or no effect. In other words, given the same input sentence, changing $\boldsymbol{s}_x$ to $\boldsymbol{s}_y$ and vice versa would result in nearly the same outputs. To test this, we compose simple, neutral sentences and feed them to our unified objective model trained on Yelp with different sentiment embeddings. Table 3 shows the generated outputs, confirming that our model indeed makes use of $\boldsymbol{s}_x$ and $\boldsymbol{s}_y$.

Source	i have been there .
$\boldsymbol{s}_x$	i have *enjoyed eating* there .
$\boldsymbol{s}_y$	i have *n't* been there .
Source	i had experience .
$\boldsymbol{s}_x$	i had *great* experience .
$\boldsymbol{s}_y$	*worst* experience i *ever* had .

Table 3: Outputs of the proposed model given the same source sentences with different sentiment embeddings ($\boldsymbol{s}_x$ = positive; $\boldsymbol{s}_y$ = negative).

5 Conclusion

We have shown that our unified objective model successfully learns from the noisy training sentence pairs acquired from non-parallel corpora using our search-and-corrupt method. Even though our model is conceptually simpler, it produces competitive results against the strong baselines across two domains.

Acknowledgments

We thank the anonymous reviewers of all versions of this paper for their useful comments.

References

Alekh Agarwal, Olivier Chapelle, Miroslav Dudík, and John Langford. 2014. A reliable effective terascale linear learning system. *J. Mach. Learn. Res.*, 15(1).

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of ICLR*.

Mayank Bawa, Tyson Condie, and Prasanna Ganesan. 2005. LSH forest: Self-tuning indexes for similarity search. In *Proceedings of WWW*.

Yoshua Bengio, Li Yao, Guillaume Alain, and Pascal Vincent. 2013. Generalized denoising autoencoders as generative models. In *Proceedings of NIPS*.

Andrei Z. Broder. 2000. Identifying and filtering near-duplicate documents. In *Proceedings of the 11th Annual Symposium on Combinatorial Pattern Matching*.

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

Pascale Fung and Percy Cheung. 2004. Mining very-non-parallel corpora: Parallel sentence and lexicon extraction via bootstrapping and em. In *Proceedings of EMNLP*.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Proceedings of NIPS*.

Kelvin Guu, Tatsunori B. Hashimoto, Yonatan Oren, and Percy Liang. 2018. Generating sentences by editing prototypes. *Transactions of the Association for Computational Linguistics (TACL)*.

Geoffrey E. Hinton, Peter Dayan, Brendan J. Frey, and Radford M. Neal. 1995. The wake-sleep algorithm for unsupervised neural networks. *Science*, 268.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8).

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

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of ICLR*.

Diederik P. Kingma, Danilo J. Rezende, Shakir Mohamed, and Max Welling. 2014. Semi-supervised learning with deep generative models. In *Proceedings of NIPS*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL*.

Guillaume Lample, Sandeep Subramanian, Eric Michael Smith, Ludovic Denoyer, Marc'Aurelio Ranzato, and Y-Lan Boureau. 2019. Multiple-attribute text rewriting. In *Proceedings of ICLR*.

Juncen Li, Robin Jia, He He, and Percy Liang. 2018. Delete, retrieve, generate: A simple approach to sentiment and style transfer. In *Proceedings of NAACL*.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of EMNLP*.

Dragos Stefan Munteanu and Daniel Marcu. 2005. Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics*, 31(4).

Shrimai Prabhumoye, Yulia Tsvetkov, Ruslan Salakhutdinov, and Alan W. Black. 2018. Style transfer through back-translation. In *Proceedings of ACL*.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *Proceedings of NIPS*.

Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.*, 11.

Jingjing Xu, Xu Sun, Qi Zeng, Xuancheng Ren, Xiaodong Zhang, Houfeng Wang, and Wenjie Li. 2018. Unpaired sentiment-to-sentiment translation: A cycled reinforcement learning approach. In *Proceedings of ACL*.

Zichao Yang, Zhiting Hu, Chris Dyer, Eric P Xing, and Taylor Berg-Kirkpatrick. 2018. Unsupervised text style transfer using language models as discriminators. In *Proceedings of NIPS*.

Yi Zhang, Jingjing Xu, Pengcheng Yang, and Xu Sun. 2018a. Learning sentiment memories for sentiment modification without parallel data. In *Proceedings of EMNLP*.

Zhirui Zhang, Shuo Ren, Shujie Liu, Jianyong Wang, Peng Chen, Mu Li, Ming Zhou, and Enhong Chen. 2018b. Style transfer as unsupervised machine translation. *CoRR*, abs/1808.07894.