From Disjoint Sets to Parallel Data to Train Seq2Seq Models for Sentiment Transfer

Paulo Cavalin, Marisa Vasconcelos, Marcelo Grave, Claudio Pinhanez, Victor Henrique Alves Ribeiro
IBM Research
São Paulo, SP, Brazil
pcavalin@br.ibm.com

Abstract

We present a method for creating parallel data to train Seq2Seq neural networks for sentiment transfer. Most systems for this task, which can be viewed as monolingual machine translation (MT), have relied on unsupervised methods, such as Generative Adversarial Networks (GANs)-inspired approaches, for coping with the lack of parallel corpora. Given that the literature shows that Seq2Seq methods have been consistently outperforming unsupervised methods in MT-related tasks, in this work we exploit the use of semantic similarity computation for converting non-parallel data onto a parallel corpus. That allows us to train a transformer neural network for the sentiment transfer task, and compare its performance against unsupervised approaches. With experiments conducted on two well-known public datasets, i.e. Yelp and Amazon, we demonstrate that the proposed methodology outperforms existing unsupervised methods very consistently in fluency, and presents competitive results in terms of sentiment conversion and content preservation. We believe that this works opens up an opportunity for seq2seq neural networks to be better exploited in problems for which they have not been applied owing to the lack of parallel training data.

1 Introduction

Sentiment transfer can be considered as a subset of the style transfer task, the main goal of which is to convert a text that presents a style τ_1 to another style τ_2, while keeping its original meaning μ. Given the increasing number of applications that currently make use of natural language user interfaces, style transfer can be useful in many real-world applications, for instance, chatbot personality transformation for fitting chatbot language to a specific public, bias removal (such as gender and racial bias), offensive and hate speech-language filtering, and thus forth.

Previous studies on style transfer focused mostly on unsupervised methods, for instance Generative Adversarial Networks (GANs), owing to the lack of parallel corpora. However, given that style transfer can be viewed as a monolingual machine translation (MT) task, and that seq2seq models such as the transformer have shown to outperform unsupervised methods in multi-lingual MT when a sufficiently large parallel corpus is available (Lample et al., 2018; Artetxe et al., 2019; Subramanian et al., 2018), in our opinion it is expected that seq2seq would outperform unsupervised approaches if parallel data is available for style transfer. However, to the best of our knowledge, a parallel corpus for style transfer currently does not exist. But considering that semantic similarity metrics are becoming more and more effective (Schwenk and Douze, 2017; Wu et al., 2018), and that considerably large non-parallel data exist for some style transfer tasks, for instance, sentiment transfer and the Yelp and Amazon review datasets, one could take advantage of such metrics to build parallel corpora (Shen et al., 2017; Li et al., 2018).

Given these standpoints, we propose and evaluate an approach to create parallel training data from non-parallel sets of data, on sentiment transfer datasets as a use-case for style transfer[1], and compare the resulting transferred outputs of a Transformer Seq2Seq neural network (Vaswani et al., 2017) against those of state-of-the-art unsupervised methods. Considering the Yelp and Amazon data sets for sentiment transfer, we take advantage of semantic similarity using Universal Sentence Encoders (USE) (Cer et al., 2018) to represent sentences and the euclidean distance, which is scalable to large sets of data. Our results show that our proposed method can generate more fluently-written texts than unsupervised approaches, and that is well

[1]The method can be easily applied to other tasks, provided disjoint style-related sets of texts are available.

Findings of the Association for Computational Linguistics: EMNLP 2020, pages 689–698
November 16 - 20, 2020.

balanced in terms of sentiment conversion and content preservation.

The remainder of this work is organized as follows: Section 2 introduces the related work; Section 3 details the methodology for building parallel corpora, the used seq2seq model, and the evaluation metrics; Section 4 presents the experiments and results. Finally, the paper is concluded with some final remarks.

2 Related Work

Several methods have been proposed for converting one text to another, which is usually referred to as machine translation (MT). In recent years, great progress has been made with deep learning for multi-lingual MT (Nguyen Le et al., 2017), where a text in a given input language needs to be converted to another text in the desired output language. Much of the progress made in that field is owned to the possibility of mining large corpora of parallel sentences from the web (Uszkoreit et al., 2010; Morishita et al., 2019).

Mono-lingual MT (Ghosh et al., 2017; Shen et al., 2017; dos Santos et al., 2018) also has emerged in recent years, given the potential set of applications, such as the conversion of offensive language to non-offensive (dos Santos et al., 2018) and the generation of customizable affective text (Ghosh et al., 2017). In this case, the input text should be converted to another one in the same language, keeping its main content, but being transformed in some aspects such as language style, tone, or sentiment. Differently from multi-lingual MT, mono-lingual MT generally suffers from the lack of parallel corpora (that is, different versions of the same text rephrased in different tones) to train end-to-end deep learning methods. Efforts to create corpora have been made only on limited domains, such as formality transfer for informal texts (Rao and Tetreault, 2018). As a consequence, both corpora and approaches proposed for the task are generally non-parallel, and unsupervised systems have emerged, mostly making use of text generation models and adversarial samples (Ghosh et al., 2017; Shen et al., 2017; dos Santos et al., 2018; Li et al., 2018; Zhang et al., 2018; Luo et al., 2019).

Recent work in multi-lingual MT has shown that supervised methods tend to achieve better results than unsupervised approaches when the number of parallel sentences is larger than 100,000 (Lample et al., 2018). Considering that the non-parallel data used by unsupervised methods for mono-lingual MT tasks, in special sentiment transfer, large non-parallel set of samples are available, and that textual semantic similarity and representation methods are evolving considerably (Kusner et al., 2015; Wu et al., 2018; Cer et al., 2018; Turc et al., 2019), one could build a sufficiently large corpus of parallel data to train Seq2Seq models for mono-lingual MT.

For this reason, the main contribution of this work is to present an investigation of training Seq2Seq neural networks for sentiment transfer, considering as training data parallel corpora generated from non-parallel disjoint sets, by making use of state-of-the-art semantic representations.

3 Methodology

In this section, we first describe the proposed method for building a style transfer parallel corpora, followed by the seq2seq Transformer neural network and selected metrics for the evaluation methodology.

3.1 Parallel corpora building method

Consider two disjoint sets of textual data $X^1 = \{x_0^1, \ldots, x_N^1\}$ and $X^2 = \{x_0^2, \ldots, x_M^2\}$, with N and M samples, related to two distinct styles τ_1 and τ_2, respectively. The task of creating a parallel corpus consists of creating a third set, namely $\bar{X}^{1,2} = \{\ldots, (x_i^1, x_j^2), \ldots\}$, where $1 \leq i \leq N$, $1 \leq j \leq M$, and x_i^1 has been found to be semantically similar to x_j^2 according to a similarity metric Ψ.

In this work, we implement the aforementioned idea in the following manner. We initialize $\bar{X}^{1,2}$ as an empty set. Then, by iterating in the samples of one set, we compute the similarity of each sample against all samples on the other set, adding a new pair in $\bar{X}^{1,2}$ comprising the current sample in the iteration and its corresponding most similar one from the other set. More formally, for each $x_i^1 \in X^1$, we compute the semantic similarity $\psi_{i,j}$ to each $x_j^2 \in X^2$, resulting in the set $\Psi_i^1 = \{\psi_{i,0}^1, \ldots, \psi_{i,M}^1\}$. Next, we include in $\bar{X}^{1,2}$ the new pair (x_i^1, x_j^2), where $j = \text{argmax}(\Psi_i^1)$.

Since one cannot rely on the assumption that each pair (x_i^1, x_j^2) are actually parallel samples, a post-filtering is applied on $\bar{X}^{1,2}$ considering two thresholds, i.e. θ_{min} and θ_{max}. While the first aims at reducing the effect of noise that can be presented in the input data, such as samples that are too similar, the second is used to eliminate pairs

with not enough similarity between the samples.

To compute the semantic similarity, we take into account Universal Sentence Encoders (USE) sentence embeddings (Cer et al., 2018). Such an approach consist of a Transformer Neural Network (Vaswani et al., 2017), trained on varied sources of data. That approach has been designed not only to serve as a baseline model to take advantage of transfer learning when little data is available, but also as a means to encode textual information, i.e., sentences, into real-valued N-dimensional embedding vector.

Thus, after pre-processing, normalizing, and tokenizing all samples in X^1 and X^2, we compute the USE embedding vector for each of these samples, resulting in sets $V^1 = \{v_0^1, \ldots, v_N^1\}$ and $V^2 = \{v_0^2, \ldots, v_M^2\}$. As a consequence, to compute the set of similarities Ψ_i^1, we compute the Euclidean distance[2] between the sentence embedding vectors in V^1 and V^2. Note that, this method can be costly in terms of processing time. Nevertheless, it can be easily scaled up to large sets of data using fast K-nearest neighbor methods.

3.2 Seq2Seq Transformer Neural Network

For this work, we use the Transformer Neural Network (Vaswani et al., 2017) as our seq2seq model. The Transformer consists of an Encoder-Decoder architecture, but instead of relying on recurrent neural networks such as in (Luong et al., 2015), it is based on stacked attention layers. That makes the architecture less complex and faster to be trained, and a by-product of that is that it has been consistently outperforming recurrent models in many machine translation tasks (Lakew et al., 2018).

Briefly speaking, the Transformer is based solely on attention mechanisms, not relying on recurrence and convolutions at all. Given the sequential nature of texts, positional features are encoded jointly with word embeddings. By stacking multiple attention layers in both the encoder and the decoder, combined with multi-head attention, the Transformer is able not only to achieve better results but also has a more computationally efficient architecture for training.

For this research, we make use of a publicly available implementation of the Transformer, based on the Pytorch framework for Deep Learning[3]. We have defined an architecture with the following meta parameters: 6 attention layers, 8 attention heads, word embeddings with 512 dimensions, batch size of 64, and dropout rate 0.1. This network was trained for 50 epochs with the Adam optimizer.

Based on the work described in (Lakew et al., 2018), we make use of an approach to which we refer as *shared training*. This approach consists of training a seq2seq model for multiple tasks at once, where the task is specified by an special token included in the begining of the input. In this case, since style transfer can be done from style τ_1 to τ_2 and our corpora building method takes that order into account, for converting to the other way around (from τ_2 to τ_1), we would need to invert the pairs in $\bar{X}^{1,2}$ to create the set $\bar{X}^{2,1}$ and train a second model. We shared training, we concatenate both sets $\bar{X}^{1,2}$ and $\bar{X}^{2,1}$, and include in each sample $x_i^1 \in \bar{X}^{1,2}$, a special token "from1to2". Similarly, for each $x_i^2 \in \bar{X}^{2,1}$, the token "from2to1" is included.

3.3 Evaluation Metrics

We considered the following aspects to evaluate the performance of our style transfer method:

1. *Style conversion*: if it converts the input text to the desired style;

2. *Content preservation*: if it preserves non-stylistic parts of the input sentence;

3. *Fluency*: if the method generates sentences with appropriate language fluency, i.e., grammatically, syntactically, and semantically well-formed sentences.

These aspects are implemented with the following metrics.

3.3.1 Style Transfer Accuracy (STAcc)

The STAcc metric is used to measure *style conversion* rate. Basically, it consists of computing the ratio of generated samples that have been successfully converted to the target style.

In detail, let X_{test} be the test set and $|X_{test}|$ the number of samples in that set. Also, consider that the number of correctly converted examples is represented by C, where $0 \geq C \geq |X_{test}|$, this metric can be computed as:

$$\text{STAcc} = \frac{C}{|X_{test}|}$$

[2]The smaller the distance, the higher the similarity

[3]https://github.com/jadore801120/attention-is-all-you-need-pytorch

The computation above is relatively simple, and accuracy is a well-known metric. Therefore, finding the value for STAcc is trivial once C has been found. However, finding a value for C is the main issue for the metric, since it depends on evaluating the set of generated outputs how many of them were converted successfully. That could be done either by manual inspection or by considering some automated method, such as a text classifier.

For the automated process, we take into account an approach that has been used by Shen et al. (2017) and Li et al. (2018), the TextCNN text classifier (Kim, 2014)[4]. This classifier simply takes as input a text, and provides as output the sentiment label, i.e. either positive or negative. Further details about how we train the classifier are provided in Section 4.

3.3.2 BLEU score

We consider the BLEU score to assess the similarity between ground-truth candidate sentences and the generated sentence (Papineni et al., 2002), which can present indications regarding *content preservation*.

BLEU provides a score ranging between 0 and 1, which is computed counting matching n-grams in the candidate sentence to n-grams in the generated sentence[5]. Since we are comparing a set of examples, the mean BLEU score of the samples against ground-truth candidates represents the overall score on the test set.

The ground-truth is represented by manually created references, which are provided in the datasets considered in this work.

3.3.3 Perplexity

This measure has been often used to measure the *fluency* of machine-generated text, i.e. how well-formed are the sentences generated by a given algorithm. In such case, lower perplexity means better fluency.

For this work, we use the language modeling toolkit SRILM (Stolcke, 2002), which computes the perplexity of the generated sentences in the test set, having the language model been computed from the training set, e.g. set $\bar{X}^{1,2}$.

[4]The following publicly-available implementation has been used to conduct this research: https://github.com/dennybritz/cnn-text-classification-tf

[5]We use the same BLEU evaluation used by (Li et al., 2018), available in https://github.com/lijuncen/Sentiment-and-Style-Transfer

4 Experimental Evaluation

In this section we present the experiments that have been conducted to evaluate the proposed methodology. To take advantage of the reproducibility[6] and being able to compare our results with previous works, we consider two sentiment transfer data sets, i.e., the Yelp dataset (Shen et al., 2017) and the Amazon dataset (He and McAuley, 2016), along with the publicly available results made available by (Li et al., 2018). By evaluating our trained method on the same data sets by those authors, we can directly compare our results with theirs.

4.1 Data sets

Both data sets consist of positive and negative sentences extracted from restaurant reviews and product reviews posted on Yelp and Amazon, respectively. To generate the sentiment dataset, we considered for both types of reviews that high-star reviews (i.e., rating above three) are positive and those below are negative, and final corpora contain the individual sentences of the respective reviews. It is worth mentioning that we make use of the same data used by previous works, without introducing any extra processing that could affect the results.

The Yelp dataset is slightly smaller than the Amazon one. The former is composed of a training set of 177,218 negative and 266,041 positive samples, and the validation set and the test set contain 4,000 and 1,000 samples, equally distributed in the two sentiment classes. The Amazon dataset is composed of a training set with 277,228 negative samples, and the same number of positive ones. The Amazon validation set contains 1,015 and 985, respectively negative and positive examples, while the test set has 1,000 equally-distributed samples.

4.2 Parallel Corpora Creation

We applied the proposed parallel corpora creation method (see Section 3.1) in two different scenarios:

1. *All*: With θ_{min} set to 0, and θ_{max} set to infinity, the parallel data is found and no post-filtering is applied.

2. *Filtered*: With θ_{min} set to 0.3, and θ_{max} set to 1.0, pairs that are too similar or not similar enough have been discarded.

[6]We also plan to publicly release the data sets and source codes of this paper.

In the *All* scenarios, a total of 177,218 training pairs have been created for Yelp, and 277,769 for Amazon. The *Filtered* datasets resulted in 137,616 pairs for Yelp, and 220,645 for the Amazon dataset. Some of the examples that were discarded in the *Filtered* scenario are presented in Table 1.

YELP	
I've been here twice.	I have been here twice ... .
The bartender was awesome.	The bartender was amazing!
The Arizona center is to Phoenix as the galleria is to Scottsdale.	I travel to Phoenix/Scottsdale a lot.
Was put on hold for 5+.	Customer service A+!
AMAZON	
I have wanted one of these for a while.	I have wanted one of these for a long time.
I suppose you get what you pay for.	I guess you get what you pay for.
I like driving games and i like mafia and old Chicago type of stories.	I have been a fan of Chicago cutlery for years.
Peptides signal the dermal system to produce more collagen.	No complaints from her lips to my ears.

Table 1: Some samples that were discarded in the *Filtered* scenario. For each dataset, the first two examples were found out as too similar, and the next two as too disparate.

Considering that we conduct shared training of positive to negative and negative to positive, as we mentioned in Section 3.2, the actual number of samples is doubled (354,436 and 275,232 training pairs for Yelp, and 554,456 and 441,290 for Amazon), which far exceeds the required number of 100,000 samples for training seq2seq models (Lample et al., 2018).

NEGATIVE	POSITIVE
Not even the best fried chicken in Charlotte.	The best fried chicken in Charlotte!
The food was ok.	The food was good.
Food was ok, the service was horrible.	Service was bad but the the food was good.
The macaroni salad is so bad.	The macaroni salad is good and I usually dont like macaroni salad.
They start you off with chips and salsa.	They give you chips and salsa to start.

Table 2: Examples of parallel data found on Yelp reviews

NEGATIVE	POSITIVE
If I could give no stars I would.	If i could give them more stars I would.
I would not recommend it to a friend.	I would still recommend it to a friend.
It would be better just to get a regular screen protector.	It would be nice if it came with a screen protector.
I ordered this as a present for my niece.	I ordered this as a gift for my sister.
This fits the phone well, and looks great.	Looks really nice and fits the phone well.

Table 3: Examples of parallel data found on Amazon reviews

Since it is not feasible to manually inspect the full training sets, we conducted on inspection of a subset of examples of each dataset for a qualitative analysis of the data generated. We observed that the Yelp corpus presented stronger relationship in terms of the main subject and opposite sentiments, such as the first three examples in Table 2. Remarkably, there are examples such as the third one, which presents two main subjects, i.e., food and service, with different sentiments, even though for food the change in sentiment was more subtle. As we can see, some examples may not be too aligned in terms of subjects, such as the fourth one, and samples where the sentiment may not be very clear due to the lack of proper contexts, such as the fifth one.

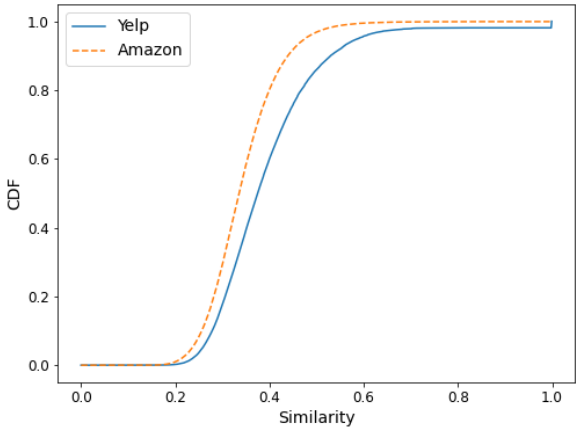


Figure 1: CDF of the similarity between the pairs on the training sets

The data on Amazon seems to be more dependent on context than Yelp data, such as the last two examples in Table 3. Even though there are pairs that are quite different in sentiment, such as the first

two examples, some pairs present also very subtle sentiment contrast, such as the third one. Such differences, compared with Yelp, might indicate that it may be harder to train the seq2seq method with this dataset. Additional evidence is presented in Figure 1, which contains a cumulative distribution function (CDF) of the similarities of the pairs in the training partitions of each data set. As can be observed, the pairs created for the Amazon data set present slightly lower similarity values, which may impact negatively the training process.

4.3 Experimental Evaluation

With the parallel training sets described in the previous sections, we have trained two versions of the seq2seq method as described in Section 3.2: Seq2Seq$_{all}$ and Seq2Seq$_{filtered}$, which use as training set the parallel corpora created with the *All* and *Filtered* scenarios, respectively.

The goal is to compare the previously-mentioned methods against five unsupervised methods: StyleEmbedding (Fu et al., 2017); CrossAligned (Shen et al., 2017); MultiDecoder (Fu et al., 2017); DeleteAndRetrive (Li et al., 2018); and RetrieveOnly (Li et al., 2018). The first three unsupervised methods are similar in the sense that an encoder is learned for representing the input sentence, then a decoder is used in different ways to generate the output sentence, with the aid of discriminator classifiers, such as a GAN. The last two consists of using markers that are style-specific, so that these markers can be replaced to transfer from one style to another. While RetriveOnly is somewhat a simpler method, which retrieves an output based on finding the target marker, DeleteAndRetrive makes use of a Recurrent Neural Network (RNN)-based decoder to generate the output sentence.

We present a quantitative evaluation using the metrics described in Section 3.3. For computing the STAcc metric, we have trained a TextCNN classifier on the training partitions of each corresponding data set, by considering the default meta-parameters provided by the implementation. This method achieved 96.1% accuracy on Yelp's test set and 79.9% on Amazon's. The classifier achieves higher accuracy on the Yelp data set, which might be another indication that the Amazon data set might be more challenging than Yelp.

Since we are comparing different methods on different metrics, it is not trivial to select a winning approach among all. We are taking into account the average ranking to make such comparison clearer, with the assumption that all metrics have the same weight. In other words, since there are three different evaluation metrics, i.e., STAcc, BLEU, and Perplexity, and seven different methods (the two proposed seq2seq and five from the literature), the methods are ranked from 1 to 7 in each metric, where 1 is the best and 7 is the worst. A final raking is then computed by considering the average ranking position of each approach across the four metrics. In that case, lower values are better.

The main results are presented in Table 4. Considering the Avg. Ranking evaluation, the proposed Seq2Seq$_{all}$ is the top performer on Yelp data, reaching an average ranking of 2.67 on both, and ranks second on Amazon, with an average ranking of 3.33. Overall, we observe that both Seq2Seq$_{all}$ and Seq2Seq$_{filtered}$ consistently present good fluency, ranking as the top performers in Perplexity for both datasets. And they tend to be balanced in terms of style conversion and content preservation. That might be a good aspect since the proposed method does not cover too much of one aspect with the penalty of hurting the other one. As observed with RetrieveOnly and StyleEmbedding, each method presents the best result in either STAcc or BLEU, but also present the worst result in the other metric.

On Amazon data, as somewhat expected from the analysis of the training sets, the seq2seq methods have not performed as well as on Yelp data. Seq2Seq$_{all}$ was the second-best on Avg. Ranking, presenting the best value for Perplexity but was ranked only third on STAcc and sixth on BLEU. Seq2Seq$_{filtered}$ performed slightly better than Seq2Seq$_{all}$ in BLEU, being the fourth-best, but was worse in STAcc and Perplexity. It is worth mentioning that CrossAligned was the top performer, reaching the best STAcc values, beating Seq2Seq$_{all}$ by 0.10 points. Nevertheless, Seq2Seq$_{all}$ presented a similar performance with the CrossAligned method in terms of BLEU score (i.e., 0.20 vs. 0.21) and better Perplexity score (i.e., 8.01 vs 17.02).

Surprisingly though, this analysis showed that filtering examples from the corpora might not result in better performance since Seq2Seq$_{filtered}$ was outperformed by Seq2Seq$_{all}$. This indicates that the method has coped well with the noise presented in the original data. But surely further investigation should be done.

To complement this analysis, we present some

YELP				
Method	**STAcc↑**	**BLEU↑**	**Perplexity↓**	**Avg. Ranking↓**
StyleEmbedding	0.10 (7)	**0.37 (1)**	93.5 (6)	4.67 (5)
CrossAligned	0.75 (5)	0.27 (4)	68.7 (3)	4.00 (4)
MultiDecoder	0.49 (6)	0.30 (3)	142.4 (7)	5.33 (6)
DeleteAndRetrieve	0.90 (4)	0.31 (2)	92.4 (5)	3.67 (3)
RetrieveOnly	**0.93 (1)**	0.13 (7)	90.4 (4)	4.00 (4)
Seq2Seq$_{all}$	0.91 (3)	0.27 (4)	**15.1 (1)**	**2.67 (1)**
Seq2Seq$_{filtered}$	0.92 (2)	0.26 (6)	16.5 (2)	3.33 (2)
AMAZON				
Method	**STAcc↑**	**BLEU↑**	**Perplexity↓**	**Avg. Ranking↓**
StyleEmbedding	0.43 (7)	0.32 (2)	72.18 (7)	5.33 (7)
CrossAligned	**0.80 (1)**	0.21 (5)	17.02 (3)	**3.00 (1)**
MultiDecoder	0.71 (2)	0.27 (3)	67.38 (6)	3.67 (3)
DeleteAndRetrieve	0.55 (6)	**0.47 (1)**	64.32 (5)	4.00 (5)
RetrieveOnly	0.67 (4)	0.17 (7)	61.96 (4)	5.00 (6)
Seq2Seq$_{all}$	0.70 (3)	0.20 (6)	**8.01 (1)**	3.33 (2)
Seq2Seq$_{filtered}$	0.65 (5)	0.23 (4)	9.72 (2)	3.67 (3)

Table 4: Main results on both datasets, wherein brackets we present the ranking position of each method for each metric, and in the *Avg. Ranking* column the average of those positions is provided. The best results are highlighted in bold.

Yelp - Positive to negative	
Input	*It's good solid food.*
StyleEmbedding	It's good solid food.
CrossAligned	It's all of pizza food.
MultiDecoder	It's good second people.
DeleteAndRetrieve	It's fake food. Indeed.
RetrieveOnly	Im ok with mistakes as things happen but to act that way was ridiculous.
Seq2Seq$_{all}$	It's not good food
Yelp - Negative to positive	
Input	*Had to returned one entree because too cold.*
StyleEmbedding	Had to returned one entree because too cold.
CrossAligned	Had to get our burgers and very nice.
MultiDecoder	Had to take however happy hour, great cold.
DeleteAndRetrieve	Had to returned one entree because it was well worth it!
RetrieveOnly	One spicy with lots of mexican oregano and one more mild one.
Seq2Seq$_{all}$	I ordered right away and my food was ready in minutes.

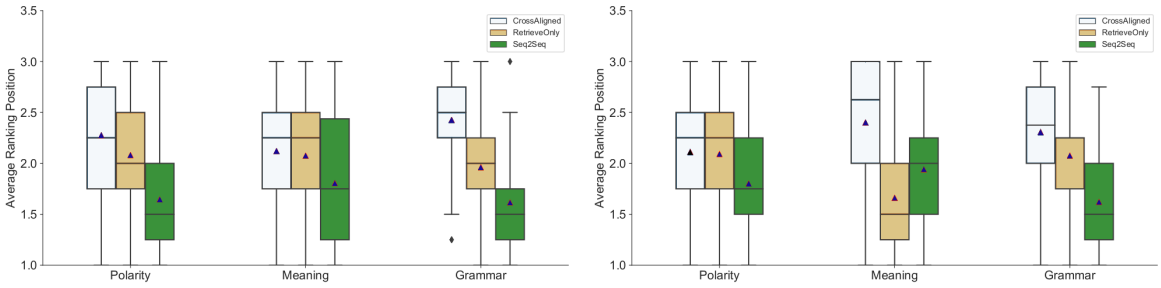
Table 5: Some selected samples of output generated by the systems, on the Yelp dataset.

generated samples from both data sets, to illustrate the performance of Seq2Seq$_{all}$ compared with the other methods. Table 5 shows that on Yelp data, the proposed method can successfully convert the sentiment (positive to negative) and maintain the non-stylistic content terms while for the negative to positive conversion the sentiment was converted with a slight change in content, but which seems to make sense in that context. The other methods, in contrast, seem not to deal well with the inputs. In the second conversion (negative to positive), DeleteAndRetrieve can change the sentiment and keep the original meaning but generating a quite awkward sentence. The examples generated using Amazon data, shown in Table 6, present similar results. Seq2Seq$_{all}$ is also able to successfully convert the sentence sentiment and preserve the original content. The other methods struggle in the task, either by not properly converting the sentiment or keeping content, or generating some awkward or ungrammatical sentences, such as CrossAligned in the first example and DeleteAndRetrieve in the second one.

4.4 Manual Evaluation

In addition to the quantitative analysis, we have also conducted a manual inspection of results presented by the Seq2Seq$_{all}$ and CrossAligned, which performed best on Amazon, and RetrieveOnly, which presented good STAcc values on both Yelp and Amazon. The main goal of the manual evaluation is to understand whether some of the results presented in the previous section with automated metrics are confirmed.

This evaluation has been conducted as follows. We asked 4 volunteers to label a random sample of the test sets either from Amazon and Yelp data sets. We asked each volunteer to rank the methods according to three criteria: if the sentiment (polarity) was the opposite from the input sentence, if the sentence had maintained the original meaning of the input sentence, and if the output sentence was



(a) Yelp (b) Amazon

Figure 2: Manual Evaluation

Amazon - Positive to negative	
Input	*Its actually a little better than the filter that came with the unit.*
StyleEmbedding	Also have a little good piece of something with it with the only.
CrossAligned	There is a little of the game, but the sound is too.
MultiDecoder	Still the toy for a very stick, this the second of this item.
DeleteAndRetrieve	Its actually worse than a little better than the filter that came with the unit.
RetrieveOnly	Make its head bigger than its body or smaller than its neck.
Seq2Seq$_{all}$	The filter is not the best filter I have ever used.
Amazon - Negative to positive	
Input	*Ridiculous! I had trouble getting it on with zero bubbles.*
StyleEmbedding	Would prefer my friend that had to follow the after one.
CrossAligned	So far I have been using it for years and now.
MultiDecoder	Beautiful I have to replace it with after using the first.
DeleteAndRetrieve	They are easy to use, I had trouble getting it on with zero bubbles.
RetrieveOnly	Had a little trouble getting bubbles out, but otherwise was worth the purchase.
Seq2Seq$_{all}$	I have had no issues with it getting bubbles.

Table 6: Some selected samples of output generated by the systems, on the Amazon dataset.

grammatically correct.

Figure 2 shows the manual evaluation results. We computed the average of the ranks given by the volunteers. The lower the value the higher ranked the method was classified by the volunteers. Note that for the Yelp data set, on average, for the three criteria the Seq2Seq method was better classified than the other two methods. For the Amazon data set, we also have the same conclusion except for the Meaning criteria, in which the RetrieveOnly method better maintained the original meaning of the sentence on average.

5 Conclusion

In this paper we proposed and evaluated an approach to create parallel data sets for training seq2seq neural networks for style transfer. We demonstrate that in the sentiment transfer use-case the seq2seq model can be a viable alternative approach to unsupervised methods, achieving the best performance in the Yelp dataset and showing a promising performance on Amazon.

In our opinion, the research presented in this paper shows that the lack of parallel data is not a definitive factor for not using seq2seq methods in text generation tasks. With proper care, a well-performing model, such as Transformer, can be applied for such cases.

However, we aware that better investigation should be conducted on several fronts. Among them, we can cite better investigation on the parallel set creation method, e.g. considering other similarity metrics. In addition, better evaluation of filtering samples should also be carried out, in special to improve the results with Amazon. Also, to better fine-tune the Seq2Seq neural network is also something that needs to be done, since that be also present a positive impact on the results, but this paper lacks a proper investigation in this specific aspect.

References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2019. An effective approach to unsupervised machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 194–203, Florence, Italy. Association for Computational Linguistics.

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, Brussels, Belgium. Association for Computational Linguistics.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2017. Style transfer in text: Exploration and evaluation. *CoRR*, abs/1711.06861.

Sayan Ghosh, Mathieu Chollet, Eugene Laksana, Louis-Philippe Morency, and Stefan Scherer. 2017. Affect-lm: A neural language model for customizable affective text generation. *CoRR*, abs/1704.06851.

Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. *CoRR*, abs/1602.01585.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *CoRR*, abs/1408.5882.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 957–966, Lille, France. PMLR.

Surafel Melaku Lakew, Mauro Cettolo, and Marcello Federico. 2018. A comparison of transformer and recurrent neural networks on multilingual neural machine translation. *CoRR*, abs/1806.06957.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Phrase-based & neural unsupervised machine translation. *CoRR*, abs/1804.07755.

Juncen Li, Robin Jia, He He, and Percy Liang. 2018. Delete, retrieve, generate: a simple approach to sentiment and style transfer. 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 1865–1874, New Orleans, Louisiana. Association for Computational Linguistics.

Fuli Luo, Peng Li, Pengcheng Yang, Jie Zhou, Yutong Tan, Baobao Chang, Zhifang Sui, and Xu Sun. 2019. Towards fine-grained text sentiment transfer. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2013–2022, Florence, Italy. Association for Computational Linguistics.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. *CoRR*, abs/1508.04025.

Makoto Morishita, Jun Suzuki, and Masaaki Nagata. 2019. Jparacrawl: A large scale web-based english-japanese parallel corpus.

An Nguyen Le, Ander Martinez, Akifumi Yoshimoto, and Yuji Matsumoto. 2017. Improving sequence to sequence neural machine translation by utilizing syntactic dependency information. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 21–29. Asian Federation of Natural Language Processing.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

Sudha Rao and Joel R. Tetreault. 2018. Dear sir or madam, may I introduce the YAFC corpus: Corpus, benchmarks and metrics for formality style transfer. *CoRR*, abs/1803.06535.

Cícero Nogueira dos Santos, Igor Melnyk, and Inkit Padhi. 2018. Fighting offensive language on social media with unsupervised text style transfer. *CoRR*, abs/1805.07685.

Holger Schwenk and Matthijs Douze. 2017. Learning joint multilingual sentence representations with neural machine translation. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 157–167, Vancouver, Canada. Association for Computational Linguistics.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi S. Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. *CoRR*, abs/1705.09655.

Andreas Stolcke. 2002. Srilm an extensible language modeling toolkit.

Sandeep Subramanian, Guillaume Lample, Eric Michael Smith, Ludovic Denoyer, Marc'Aurelio Ranzato, and Y-Lan Boureau. 2018. Multiple-attribute text style transfer. *CoRR*, abs/1811.00552.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: On the importance of pre-training compact models. *arXiv preprint arXiv:1908.08962v2*.

Jakob Uszkoreit, Jay Ponte, Ashok Popat, and Moshe Dubiner. 2010. Large scale parallel document mining for machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1101–1109, Beijing, China. Coling 2010 Organizing Committee.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Lingfei Wu, Ian En-Hsu Yen, Kun Xu, Fangli Xu, Avinash Balakrishnan, Pin-Yu Chen, Pradeep Ravikumar, and Michael J. Witbrock. 2018. Word mover's embedding: From word2vec to document embedding. *CoRR*, abs/1811.01713.

Yi Zhang, Jingjing Xu, Pengcheng Yang, and Xu Sun. 2018. Learning sentiment memories for sentiment modification without parallel data. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1103–1108, Brussels, Belgium. Association for Computational Linguistics.