Survey of Data Efficient Approaches to Formality Transfer

Advisor: Laura Dietz

Sean Smith

Fall 2020*

Abstract

Style Transfer is a problem in Natural Language Processing (NLP) that takes sequences from an input corpus and renders them in the style of a target corpus. The challenge of this process comes from creating rewrites in the desired style, while preserving content and fluency. One of the major barriers to progress in style transfer is the large data requirements needed to achieve state of the art results. This survey focuses on low cost data augmentation and adversarial techniques specifically for the sub-task of Formality Transfer. Unsupervised methods are briefly discussed, in addition to the limitations of currently available automatic evaluation metrics.

1 Introduction

Style Transfer was introduced in image processing by Gatys et al (2015) [5], in which a content image is rendered in a target style image. This problem is solved through clever manipulation of a cost function that can create impressive re-rendered images.

Formality Transfer, a specific sub-problem of style transfer, is typically treated as a Seq2Seq problem and borrows techniques from from neural machine translation. This task gained heavy interest from researchers with

*All code can be found at github.com/sms1097/formality-transfer

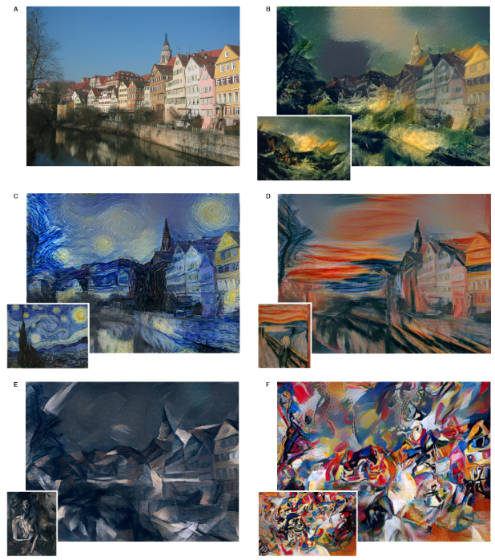


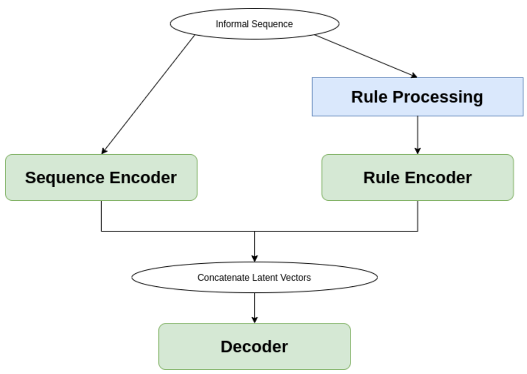
Figure 1: Example from A Neural Algorithm of Artistic Style

the creation of the GYAFC dataset form Rao and Tetreault [10]. The GYAFC dataset is a cleaned version of the yahoo L6 corpus [1]. Using the L6 corpus, Rao and Tetreault had workers translate the informal sequences from the corpus to formal rewrites. Two corpora exist, one from the Entertainment and Music category and one from the Family/Relationships category. All results included in this report come from models trained on the entertainment and music corpus.

One of the biggest barriers to this task, or in general any style transfer task, is the lack of parallel data. Machine translation models require large amounts of data to learn patterns and provide quality rewrites. As a result of this lack of parallel data, unsupervised methods and data augmentation methods are currently of great interest to researchers.

2 Supervised Models

2.1 Rules with Parallel Encoders



The approach here is similar to assisting a formality transfer model as shown in Wang et al. (2019) [12]. Their approach included using rule pre-processing on input sequences and feeding a concatenated sequence into two encoders and using hierarchal attention on GPT blocks.

This approach uses the ideas introduced in Chen et al (2018) [2] by using parallel encoders. Two sequences are fed in parallel to two different encoders and the hidden states learned by the encoders are concatenated.

For the Transformer based architecture used by Wang et al., hierarchal attention appears to be the best solution specifically for formality transfer. Chen at al. found for RNN architectures the concatenation of the hidden states gave superior results in machine translation. The concatenation approach is implemented, in addition to averaging the hidden states and encoder output.

2.2 Part of Speech Tagging with Parallel Encoders

This model follows the same paradigm as the rule based encoder, except using part of speech labels for the sequence instead of rule pre-processed sequences. A CRF was trained to detect parts of speech on a separate corpus and used to create assisted data. This data was then fed through two encoders, and their hidden spaces were concatenated. Two versions of this model were trained, one with concatenated hidden states and one with averaged hidden states.

3 Semi-Supervised Models

3.1 Formality Discrimination

Zhang et al (2020) [13] proposed formality discrimination, which augments data through a round trip translation to a pivot language. In this implementation the training data is translated to a pivot language and then translated back. This round trip translation often results in more formal sequences which can be used as additional data.

Formally, we can define our informal corpus as $\mathcal{S}$ with sequences s. We can then feed $\mathcal{S}$ through our machine translation model to the pivot language, then translate the sequence back to english; these round trip translations are denoted s'. With these re-writes a formality classifier can assign a probability of being formal to each sequence. Using the predictions we can define an additional training set $\mathcal{T}$ such that

$$\mathcal{T} = \{(s_i, s_i')|P_F(s_i') - P_F(s_i) > \sigma\}$$

where P_F is the probability of a sequence being formal. The condition for adding the data back to the data set is if the round trip translation increased formality by a percent greater than the parameter σ. For this implementation a threshold of 0.4 is used instead of 0.6 as seen in Zhang et al, as the 0.6 threshold did not increase the data set by a noteworthy result. In the end the data set was increased by approximately 8%.

3.2 Backtranslation

Backtranslation augments data through a reverse training process. This process starts by training a machine translation system to do the reverse task of converting formal sequences to informal sequences. Modifications to training were performed using techniques discussed in Edunov et al. (2018) [4].

For this task, random sampling is used of the top 10 choices, and noise is added to the decoding. Edunov et al. showed that random sampling of the synthetic data is much more effective than using a standard beam search for generating data through backtranslation. The approach used here is to sample the k most likely words form the target distribution, re-normalize the new distribution and sample once more.

Edunov et al. also showed that introducing noise in the decoding process can increase the quality of sequences generated. This technique performs a regular beam-search decoding but also introduces dropout, filler token replacement, and deleting words. This technique was what was adopted in the training of the back translation model used in this survey.

4 Unsupervised Models

4.1 Delete, Retrieve, Generate

Due to the sparsity of available data, unsupervised models are an active area of research in style transfer. One of the first prominent models for semantic transfer without parallel corpora comes from Li et al. (2018) [6]. Li et al. proposed 4 models, each building off of success from other models:

- **RetrieveOnly** returns a randomly selected output sequence x^{tgt}. This approach will always produce a fluent sequence, but will almost always fail to return a sequence with matching content.

- **TemplateBased** finds attribute words $a(x, v^{src})$ (where $v_i \in V$ are the valid style attributes, such as positive or negative) and replaces those words with attributes $a(x^{tgt}, v^{tgt})$. An underlying assumption is that all attribute words appear in the same context, which produces some grammatically incorrect sequences.

- **DeleteOnly** embeds $c(x, v^{src})$ using a RNN and appends a learned embedding for v^{tgt} and feeds this into a decoder RNN.

- **DeleteAndRetrieve** embeds $c(x, v^{src})$ and $a(x^{tgt}, v^{tgt})$, concatenates both and feeds this vector into a decoder.

Since parallel corpora are not available at training time, the loss function for **DeleteOnly** and **DeleteAndRetrieve** need to be creative. **DeleteOnly** is trained by generating new sequences using $c(x, v^{src})$ and a target sequence v^{src} in an order to attempt to create a similar sequence to the previous one. More formally, we are trying to maximize

$$L(\theta) = \sum_{(x, v^{src}) \in D} \log p(x \mid c(x, v^{src}),\ v^{src}; \theta)$$

DeleteAndRetrieve requires more of a rework. With a traditional auto-encoder approach, the model will learn to recreate the same sequence. The solution is to introduce a denoising auto-encoder which adds a random variability to each of the attribute words to create $a'(x, v^{src})$. The formal definition of the loss function becomes

$$L(\theta) = \sum_{(x,v^{src})\in D} \log p(x \mid c(x, v^{src}),\ a'(x, v^{src}); \theta)$$

While this problem works well for semantic transfer, it fails to generalize to formality transfer. All four of the models rely on the salience, which relies on n-grams. We define an attribute marker u if the salience $s(u, v)$ is greater than a threshold γ. Formally the salience is

$$s(u, v) = \frac{\text{count}(u, D_v) + \lambda}{(\sum_{v' \in V, v' \neq v} \text{count}(u, D_{v'})) + \lambda}$$

where D_v is the document of style v. The n-gram count in the formal and informal corpora are both low, and as a result this would not have been a successful implementation for the formality transfer task.

Using non-parallel corpora, content could be completely lost with no n-gram overlap. In the formal corpus the most common n-gram appeared only 178 times, while in the informal corpus it appeared 441 times; however, the most common n-gram in the informal corpus was an accidental double period, which does provide motivation to the rule based model presented earlier.

5 Results

5.1 Evaluation Metrics

The following are a brief explanation of the metrics used to assess the models.

- character n-gram F-score (chrF):

$$(1+\beta^2)\frac{CHRP \times CHRR}{\beta^2 CHRP + CHRR}$$

  where $CHRP$ is the percentage of n-grams in the predicted sequence that are in the target sequence and $CHRR$ is the percentage of character n-grams in the predicted sequence that are also in the target sequence. chrF was first introduced by Popovic (2015) [9] to produce a metric that provided better human correlated similarity evaluation than existing metrics. In studies Popovic found that chrF provided equal or higher correlation with human evaluation than BLEU.

- BiLingual Evaluation Understudy (BLEU): BLEU, first introduced by Papineni et al (2002) [8], calculates n-gram precision for sequences against the target sequence. 1-4 gram BLEU is calculated as well as individual 1-gram BLEU.

- Formality: Formality is calculated by a RNN trained to classify formal sequences on a holdout corpus from the GYAFC dataset. The formality score is the probability the model assigns to the sequence being formal.

Table 1: Transformer Based Model Results [Bold is significant ($\alpha = 0.01$)]

Model	BLEU	BLEU (1-grams)	Formality	CHRF
Transformer	10.38 ± 0.07	30.93 ± 0.56	78.13 ± 13	35.56 ± 0.89
Back-Translation	**10.72 ± 0.03**	**33.59 ± 0.23**	75.68 ± 0.9	9.79 ± 0.16
Formality-Discrimination	**11.78 ± 0.07**	**37.13 ± 0.51**	75.86 ± 1.53	26.73 ± 0.55
Rule-Assisted	10.55 ± 0.04	32.2 ± 0.28	**87.56 ± 0.43**	13.12 ± 0.39
POS-Assisted	10.48 ± 0.07	31.68 ± 0.51	**83.19 ± 0.98**	7.9 ± 0.35

* underlined model is baseline

Table 2: RNN Based Model Results [Bold is significant ($\alpha = 0.01$)]

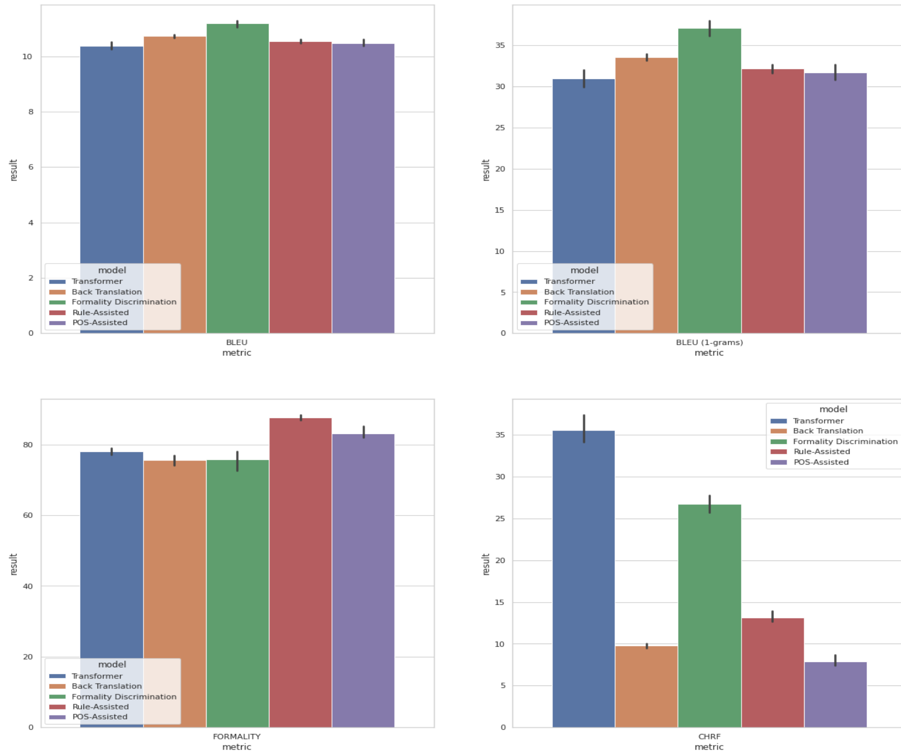
Model	BLEU	BLEU (1-grams)	Formality	CHRF
Global Attention	10.2 ± 0.07	29.51 ± 0.52	1.48 ± 0.26	16.71 ± 0.29
Bahdanau Attention	9.61 ± 0.07	24.93 ± 0.54	1.78 ± 0.32	**26.75 ± 0.34**
CRF POS (Concat)	7.04 ± 0.02	4.88 ± 0.14	**78.49 ± 0.52**	4.63 ± .07
CRF POS (Avg)	7.1 ± 0.02	5.34 ± 0.14	**80.54 ± 0.34**	4.21 ± 0.14
Rule Assist (Concat)	7.6 ± 0.1	9.35 ± 0.43	**67.21 ± 0.76**	3.95 ± 0.19
Rule Assist (Avg)	7.54 ± 0.04	8.77 ± 0.32	**69.47 ± 1.62**	4.31 ± 0.1

* underlined model is baseline

5.2 Methodology in Training

Two base architectures were used in the training of models: Transformers and Attentional RNNs. All of the RNN models trained were custom implementations of an Encoder-Decoder model using an attention mechanism. The architecture of all models included an embedding initialized with GloVe

Transformer Based Model Results

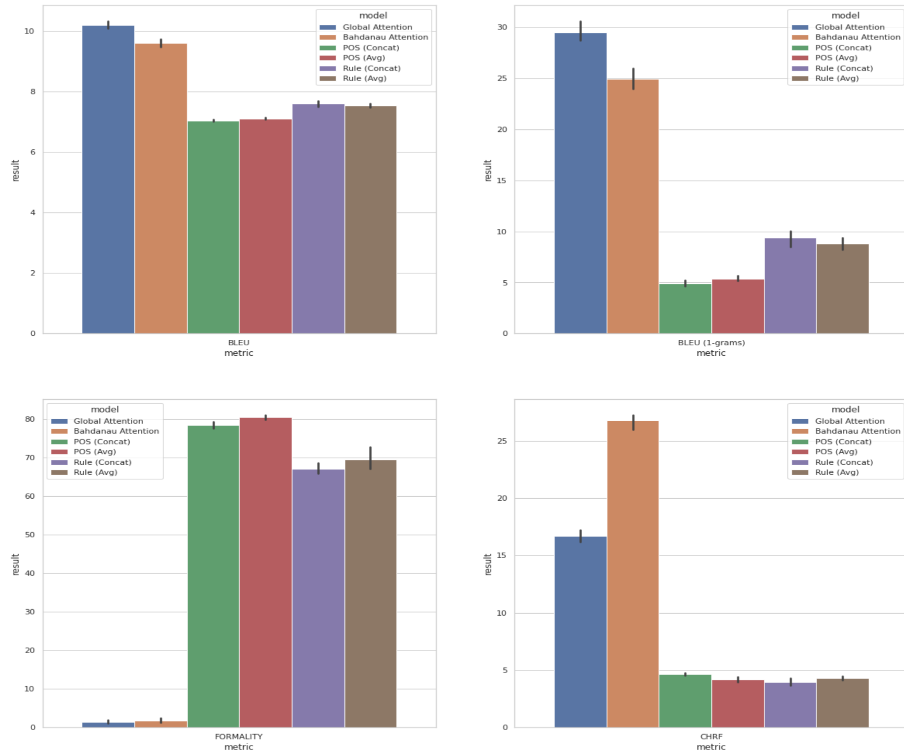


weights and single bidirectional LSTM in the encoder of 1024 units. From there, sequences were fed into a decoder with a 200 dimensional embedding and an LSTM layer, following into fully connected layers and an output. All models were optimized using Adam and Sparse Categorical Cross Entropy. All models used global attention, except for the Bahdanau attention network.

The transformer models were trained using OpenNMT-tf. While a custom implemented transformer was written as part of this survey, the computational requirements to train the model prevented practical application. The baseline, backtranslation, and formality discrimination models used the prebuilt transformer within the library. The rule-assisted and pos-assisted models used custom implementations that inherit from the base transformer class. This was done to fit the transformer with multi-column encoders. For these transformers, all hyperparameters were chosen from Vaswani et al. [11]

The tables were separated into architecture due to the different origin of the models. The transformers come form the OpenNMT-tf library, which is more carfully optimized compared to my implementations. The goal of

RNN Based Model Results



the results is to show how using the data augmentation techniques improves the power of the model. Combing the results into one table and significance testing against a baseline either naturally inflates the Transformer models or penalizes the RNN models.

All models were trained on the same training set consisting of 25,000 sequences and evaluated on a test set of 2000 sequences. Significance testing was performed by breaking the test group into 7 groups and computing metrics. A paired t-test between the baseline model and the target model was calculated and results that were significantly greater than the baseline model were recorded.

5.3 Model Results

The Global Attention RNN produced poor outputs. The Global Attention model has significant shortcomings in repeated tokens and grammatical errors. The Global Attention model also suffers from serious loss of content. Most sequences do not preserve any content whatsoever.

The baseline transformer performed well on translations and produced sequences that were mostly grammatically correct. Upon review most of the sequences retain content strongly while others lose some content. An example of this is the translation of the sequence *ray charles is the god of smooth music* was translated to *I am streaming music on the job.* While the sequence is about music, it is more specifically about the legacy of Ray Charles as a musician.

The RNN models aided by part of speech tagging had significant problems in how they learned to generate formal sequences. All three of the CRF models did score significantly better than their respective baseline model in formality, however they learned to generate sequences that are nonsensical. These models learned to produce sequences that had excessive punctuation and capitalized the beginning of sentences. For example average sequences resembled *They ,* which is completely nonsensical. It is worth noting that the CRF POS with averaged attention did test significantly better for BLEU, BLEU (1-grams), and formality compared to the CRF POS with concatenated hidden states, but it did not actually produce sequences that correlated better with human judgements.

The problems seen in the RNN POS assisted models are not observed in the trasnformer POS assisted model, but rather a new set of problems were discovered. This model learned to generate sequences that are mostly grammatically incorrect. This is largely because the model learned a limited vocabulary of 149 tokens, most of which are stop words such as *the*, *in*, and *they.* From the three part of speech models trained, it can be concluded that part of speech does not help increase the power of the model and actually appears to hinder the ability of the models.

The RNN Rule assisted models did not perform much better than the POS models. The rule assisted models learned similarly to the POS models to capitalize the beginning of sequences. They did not learn to use excessive punctuation as seen in the POS models. The sequences generated here learned to create partially grammatically correct sequences, but did not learn to finish them (i.e. *Bruce is a* or *Yes is* .) This model did appear to converge, which likely indicates that a more complex model was needed to counteract the issues seen with the model's high bias.

The Transformer based rule assisted model produces mostly grammatically correct sequences, but also loses content in translation. An example of this is the sequence *ray charles is the god of smooth music* is translated to *I believe the Green Day is playing.* The sequence is mostly correct, and the model did learn to associate Green Day with music but completely missed that the content of this sequence was about Ray Charles. Another example of this is seen in the translation of *hillary's good but mary-kate and ashley olsen not at all* to *There is a lot of girls, then they are not as well.* This did learn that the entity names were female but completely missed that the important part of the sequence is who the girls actually are.

These sequences were some of rule-assisted transofmer model's better performances. An example of the model truly underperforming is the translation of *yO ARE SO UGLY THAT WHEN YOU WHERE BORN YOU WERE PUT IN AN INCU-BATER WITH TINTED WINDOWS!* was translated to *If you do not know, but they should be able to know but they do not know who does not know.* This is an extreme example of lost content in the model, where complete content is lost. In careful examination this does appear to be the standard behavior and not the exception.

The backtranslation model achieved a statistically significant result in both BLEU scores, however it is one of the worst performing models. In an analysis of the results, there appear to be no translations that carry content. An example of this is the translation of the sequence *Best band: Motley Crue worst: Poison, Stryper(bad christian rock).* translated to *I am a Harry Potter movie but not very funny.* Clearly using the data translated during backtranslation shows that a model learns to focus on producing sequences with good grammar, but terrible transfer of content. This model may be successful if the amount of backtranslated data was limited to prevent the generated sequences from having as much influence over the model as the original sequences.

The formality discrimination model was the best performing model out of all tested. Most of the sequences maintain content and all are grammatically correct. The downside of this model is it is the most computationally expensive model to work with, since all sequences need to be translated to the pivot language and back. Then after the round trip translation all sequences need to be classified and a threshold for keeping sequences needs to be tuned.

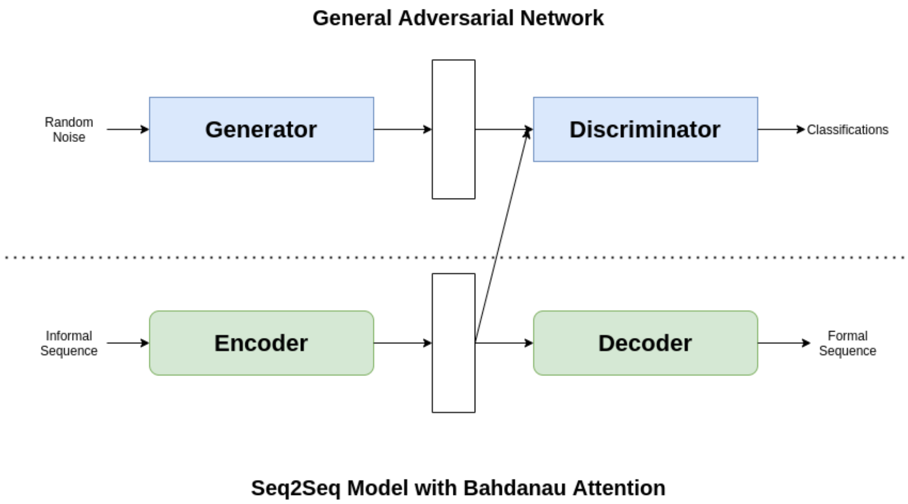
5.4 Discussion of Metrics

Individual metrics proved to be unreliable in determining success of a model. Human reviewers were asked to look at a subset of the data from the Bahdanau Attention model and the Global Attention model to assess fluency, content preservation, and formality. While the Global Attention model tested to have significantly better results compared to the Bahdanau Attention model for both BLEU metrics, these metrics did not correlate with the human reviewers.

Rather than making a determination about a model based on one metric, it appears that a good balance of scores is needed to assess if automatic evaluation of generated sequences align with human reviewers. The best balance seems to be between BLEU and chrF, and seeing a lower score in BLEU 1-grams. High BLEU 1-gram scores with low BLEU scores tended to indicate models did not learn to produce well rounded sequences. If the model can only learn 1-grams, then likely meaning is being lost. The balance between BLEU and chrF allows us to evaluate how many n-grams are matching but also how many subsets of words are matching. For example if the model translates a word with the wrong tense, chrF can see it is still attempting to learn the correct content transfer but did not achieve the correct fluency. This kind of translation mistake may not bother human reviewers but would penalize the sentence greatly in BLEU scores.

The idea of aggregating metrics is suggested in Pang (2019) [7], where a weighting scheme could be enforced on the metrics to find a combination that is correlated with human judgement. The procedure for creating such a metric would involve taking outputs from many models and randomly sampling sequences to show to reviewers. After review, a new metric could be trained on the results of human review. A limitation of this metric is that the aggregation would not necessarily translate between data sets. In order to achieve a universal metric, outputs from many models trained on different data sets would need to be collected and reviewed.

Another interesting takeaway is the sensitivity of the Formality metric. The model was trained to achieve 91% accuracy on a holdout set that also comes from the GYAFC corpus. In training, a jump from 80% to 91% was achieved simply from preventing the tokenizer from only learning lowercase representations and tokenizing the period character. This likely indicates that the biggest factors in what the neural network is learning comes from complete sentences. The classifier also learned to favor sentences as formal



when they began with a capitalized letter and ended with a period. This is similar to how the POS models got caught up in generation, where learning to capitalize the beginning character and use excessive punctuation was how they learned to produce formal output.

6 Future Work

6.1 Data Augmentation with Seq2Seq GAN

This approach is based on ideas expressed in Donahue et al (2018) [3]. A Seq2Seq model is trained using available parallel data until minimally acceptable results are achieved. A generator is then fed random noise to create tensors that are equal in size to the intermediate tensor in between the encoder and decoder. The discriminator is fed both tensors and attempts to distinguish between which sequences are real and which are fake. The generator is rewarded for fooling the discriminator and the discriminator is rewarded for detecting the generator outputs. Training is stopped once the discriminator is no longer able to distinguish if a tensor came from the generator of the Seq2Seq model.

It is important to distinguish that the GAN must be trained on the intermediate sequence instead of output sequences or tokens. When training the GAN we are essentially playing a minimax game, such that the loss of the

Discriminator is maximized while the loss of the generator is minimized. In a normal RNN architecture, the network is trained to minimize cross entropy with the target token at the step of a sequence. With this minimax algorithm the new goal for the RNN is to minimize the loss of the discriminator. When iterating through the sequence and choosing the most likely token, we were performing an operation that was not differentiable, since the loss was calculated according to how well that token matched. In order to be able to apply a loss function to the discriminator, we need a differentiable operation, and we can achieve this by learning to generate the intermediate sequence between the encoder and decoder since it is continuous.

Using the GAN we can generate useful data in two ways. First a pre-trained back translation model could be used to generate informal sequences from the formal sequences. The upside of this approach is the potential for unlimited data, however there exists potential problems with the balance of generated data and original data. The quality of the data would have to be assessed, which would require the training of a metric to ensure the backtranslated data fits the informal distribution.

The second approach for using GAN data is using similarity metrics between the informal corpus and generated sequences to pair rewrites. Jaccard similarity is a retrieval metric which computes the intersection over union for sequences, and could be used to find potential matches between the data. Another similarity metric is cosine similarity, which computes distances between vectors representing term counts. Similarly to Jaccard similarity this metric could be used to retrieve sequences from the generated sequences that are close to the informal sequences. A minimum distance could be selected based upon results.

Due to time and resource constraints this GAN approach could not be fully developed. Further experimentation needs to be done to produce a GAN that can generate adequate data. All implementations that currently exist suffer from mode collapse. Different combinations of additional noise, learning rate tuning, and label smoothing were attempted, but with no success. Likely the current approach will not prove to be adequate and a pure autoencoder would need to be trained. In experimentation, it was difficult to find an autoencoder that produced sequences of high enough quality for the generation to be worth pursuing. This approach might also require expanding the data set with the other half of the supervised corpus.

6.2 Entity Transfer Metric

As seen in many translations, an essence of the original content was transfered but important entities were dropped. This does motivate exploration of a metric that can detect the transfer of entities from one sequence to another, and harshly penalize sequences that lose entities. This could also be explored at the model level to find ways that entity transfer can be enforced in the translations.

7 Conclusion

Style transfer is a great upcoming field in NLP that has seen considerable amounts of clever research and still has great room for potential. This survey showed novel experimental ideas for style transfer stemming from machine translation. The shortcomings of current evaluation metrics was also shown. Areas for future research were presented based upon the results from implemented models, including experimental data augmentation strategies and evaluation metrics.

References

[1] Yahoo! webscope. 2007. yahoo! webscope dataset l6 corpus.

[2] M. X. Chen, O. Firat, A. Bapna, M. Johnson, W. Macherey, G. Foster, L. Jones, N. Parmar, M. Schuster, Z. Chen, Y. Wu, and M. Hughes. The best of both worlds: Combining recent advances in neural machine translation, 2018.

[3] D. Donahue and A. Rumshisky. Adversarial text generation without reinforcement learning. *CoRR*, abs/1810.06640, 2018.

[4] S. Edunov, M. Ott, M. Auli, and D. Grangier. Understanding back-translation at scale. *CoRR*, abs/1808.09381, 2018.

[5] L. A. Gatys, A. S. Ecker, and M. Bethge. A neural algorithm of artistic style. *CoRR*, abs/1508.06576, 2015.

[6] J. Li, R. Jia, H. He, and P. Liang. Delete, retrieve, generate: A simple approach to sentiment and style transfer, 2018.

[7] R. Y. Pang. The daunting task of real-world textual style transfer auto-evaluation, 2019.

[8] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. 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, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.

[9] M. Popović. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal, Sept. 2015. Association for Computational Linguistics.

[10] S. Rao and J. Tetreault. Dear sir or madam, may i introduce the gyafc dataset: Corpus, benchmarks and metrics for formality style transfer, 2018.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[12] Y. Wang, Y. Wu, L. Mou, Z. Li, and W. Chao. Harnessing pre-trained neural networks with rules for formality style transfer. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3573–3578, Hong Kong, China, Nov. 2019. Association for Computational Linguistics.

[13] Y. Zhang, T. Ge, and X. Sun. Parallel data augmentation for formality style transfer, 2020.