Writing Style Conversion using Neural Machine Translation

Se Won Jang(swjang) Jesik Min(jesikmin) Mark Kwon(hjkwon93)

Department of Computer Science, Stanford University

Abstract

Writing style is an important indicator of a writer's persona. In the age of intelligent chatbots, writing style conversion can enable intimate human-AI interaction, allowing us to bridge the inherent gap between AI agents and human beings. In this paper, we apply sequence to sequence neural machine translation model with global attention mechanism to two writing style conversion tasks, mostly focusing on Shakespearean style conversion task, to explore its capabilities and limitations. In order to acquire parallel corpora of two unique writing styles, we suggest a new method of data acquisition, which leverages the Google Translator engine when only given a single corpus of target writing style. Another decision problem that is crucial to the performance of this model is the embedding matrix of the source and target vocabulary, hyperparameters, global attention mechanism, and many other details of bidirectional Sequence to Sequence model. The bidirectional Seq2Seq model we suggest here outperformed the previous models[1] for style mimicking in BLEU score by greater than 25%. In addition, through human evaluated metrics, we could observe that our bidirectional Seq2Seq model performed better than our simple attentive Seq2Seq model in preserving original meaning and imitating target style.

Keywords: neural machine translation; recurrent neural network; seq2seq; writing style conversion

1 Introduction

With the wide adoption of personalized AI assistants such as Amazon's Alexa and Microsoft's Cortana, embedding human-like intimacy in AI interaction has become a new field of importance. Since this is a relatively recent phenemenon, we have not yet seen many attempts to tackle this problem by utilizing neural networks. Past works tend to utilize predefined English language structure and approach the problem from an algorithmic point of view.

Google Translator's successful adoption of sequence to sequence neural machine transation model have suggested its effectiveness in translation tasks between two parallel corpora. The encoder-decoder model is capable of preserving the meaning of the encoded text and decode it into the target language. Therefore we have chosen to explore its applicability in writing style conversion tasks.

However, it is not easy to adopt seq2seq models to solve writing style conversion problems. There are some important challenges we need to tackle before applying the model to writing conversion tasks.

First, unlike many human language model pairs that have a great number of parallel language datasets, monolingual writing styles often have no corresponding representation in colloquial style. There are not many parallel datasets for monolingual writing styles, that pair up a specific writing style to its colloquial counterpart. One of the very few, and possibly the only parallel dataset that is readily available to the public is the modern English translation of some old English literature, such as Sparknotes' modern translation of Shakespeare's works. Since seq2seq requires a parallel corpora of both source and target languages, the unavailability of colloquial, general counterparts for target writing styles poses a great challenge to our task.

Second, we applied bucketing and padding strategy to handle sentences of different length. Instead of giving a strict heuristic that a sentence of length of L_1 in style A is converted into a sentence of length L_2 in style B, we assign a bucket to each sentence so that a sentence of up to L_1 length in Style can possibly be a sentence of any length within the bucket, namely L_1 to L_2. We also introduce a special symbol, padding, to implement this strategy.

Third, we discovered that training the embedding matrices within a bidirectional sequence-to-sequence model with LSTM cell and global attention increases the number of parameters drastically by order of millions. This large number of parameters are hard to optimize when we are given relatively small amount of corpora (rap lyrics of about 30MB compared to standard English-French corpora of 3GB), and therefore, it becomes implausible to train our model if we try to train our word embedding matrices along with other parameters. We discuss how we created fixed embeddings that we want to feed in for writing style conversion task in **3. Approaches** section.

Fourth, we utilized bidirectional Seq2Seq model to capture directional information of English sentences. While a simple Seq2Seq model encodes one input sequence into a forward direction context vector, we also built a bidirectional Seq2Seq model to take account of the fact that a word in a sentence is highly related with the words before and after itself.

Fifth, we explored a lot of possible options for hyperparameters including learning rate, decay rate for the learning rate, number of layers, number of cells, word embedding dimensions and so on. We figured out slightly different hyperparameters for Shakespeare and English rap lyric model.

2 Background / Related Works

While there have been many breakthroughs in natural language processing, there was a scarce effort in understanding people's writing style on our knowledge. In 2012, Wei Xu et al[1] suggested a computational model that replicates the writing style of William Shakespeare. However, their approach did not take advantage of modern machine learning or neural net based model and solely relied on phrase-based machine translation by building a "phrase table".

At the same time, recurrent neural network and its variant - Seq2Seq model - has drawn attention of many studies. In particular, there was a study by Google researchers that tried to solve syntactic parsing problem using Seq2Seq model. The study showed that Seq2Seq model with attention elicit "state-of-the-art results on the most widely used syntactic constituency parsing dataset"[2]. This study implies that while Seq2Seq model was initially devised for machine translation between two languages, it can be applied to other monolingual tasks such as grammar parsing.

Recently, a group of researchers have combined those two different ideas and suggested using recurrent neural networks for writing style conversion. The basic idea proposed by the two scholars is to investigate "neural network approaches to learn and incorporate stylistic features in the process of language generation"[3]. However, the work is merely a research proposal and how they will apply RNN to style conversion is not clear. In this study, we explore how Seq2Seq model can be applied to solve writing style conversion task, how to generate a sufficient number of parallel corpus, and how we can change the original Seq2Seq model so that it performs better in style conversion.

3 Approaches

3.1 Data Collection

In order to train a neural machine translator for writing conversion tasks, we need parallel corpuses that are aligned at sentence level. After exploring several different options such as UN documents and ex-president Obama's speech scripts, we ended up choosing two different clean and obtainable datasets - Shakespeare's literary works and English rap lyrics. Regarding Shakespeare data, we used sentence-aligned parallel copora of 17 Shakespeare's works, collected by Professor Wei Xu (https://github.com/cocoxu/Shakespeare). The data were parallel corpuses collected from Sparknotes' No Fear Shakespeare series, total up to 40,000 sentences. However, when it comes to English rap lyrics, as mentioned in the introduction, there wasn't any dataset that had a sentence-level aligned parallel corpuses.

After scraping rap lyrics, which were around 300,000 lines from over 8,000 songs, to acquire the parallel version of the raps, we queried Google Translate Engine to translate rap lyrics to French, and back to English. By doing so, we could acquire a drier, more colloquial text that preserves the meaning and is stripped of the rap-like styles. We chose French as an intermediate language, because of the syntactic similarity that it has with English. Data collection, style-stripping, and parallel corpus generation were fully automated by using Scrapy and Selenium opensource python frameworks.(https://github.com/sjang92/nmt-style-converter/tree/master/dataCollector) Then, we preprocessed the collected data to construct token id files and converted the rap lyric files into token id based files (see Figure 1).

(i) 1$\rightarrow$ __PAD, 2$\rightarrow$ __GO, 3$\rightarrow$__EOS, 4$\rightarrow$ __UNK, $\cdots$, 1018$\rightarrow$ shylock, 1019$\rightarrow$ deliver, $\cdots$

$\Downarrow$

(ii) 7 29 436 11 265 9 659 6 136 6 138 149 5
38 36 228 20 446 4 1614 4 1104 30 132 4 10 41 132 14 1531
207 4 24 14 88 5

Figure 1: Preprocessing Each Sentence into Token-id Based Representation

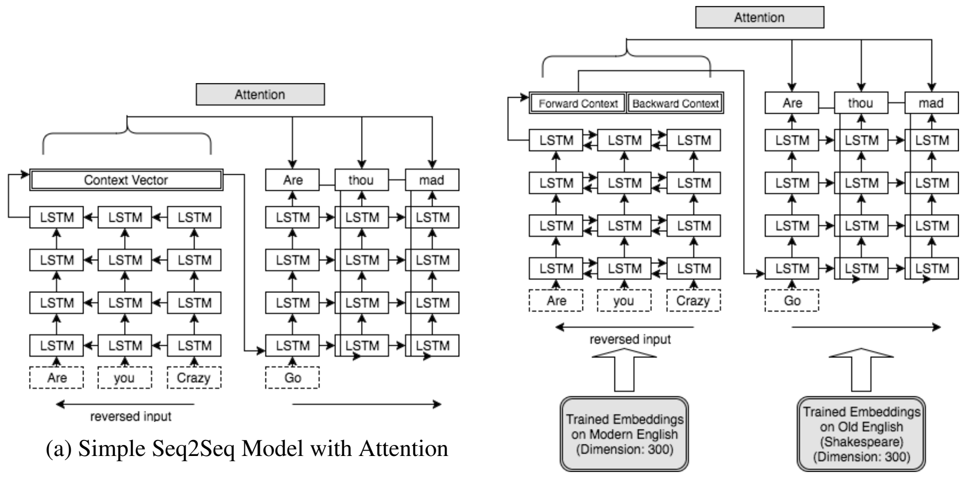
3.2 Architecture

We hypothesized that neural machine translation model with encoder and decoder, or also widely known as "Seq2Seq" model will work well with the writing style conversion style problem. While the original, simple Seq2Seq model[4] is usually trained upon corpora of two different languages, we aimed to train the model and variants of the model with parallel corpus of monolingual datasets. After many rounds of trial and error, we have set our focus on two different models in this paper: simple Seq2Seq model with attention and bidirectional Seq2Seq model with attention and fixed embeddings. We briefly introduce two models in sections below (**3.2.1** and **3.2.2**) and details are discussed in the following sections.

3.2.1 Simple Seq2Seq with Attention

The first model we implemented was a simple sequence-to-sequence model with global attention. As described in Figure 2a,it has up to 4 layers of unidirectional LSTM for its encoder, which produces per layer a context vector of dimension $dim(state)$. The encoder input symbols were passed in reverse order since it is found to perform better than feeding the input symbols in forward order. Once the context vectors are produced, they are fed as the initial state for the decoder LSTM cells. At each timestep, the decoder output is projected back onto the target vocabulary space, and produces softmax loss combined with the target symbol. This is used to train the parameters occurring in all past timesteps.

While this model turned out to perform moderately in writing conversion tasks, (section 4.1) we have found that it performs rather poorly for test examples where the encoder and decoder sentences had drastically different structures. This was a prominent phenomenon in Shakespearean style translation tasks, since a lot of the training pairs had different sentence structures.



(a) Simple Seq2Seq Model with Attention

(b) Bidirectional Seq2Seq Model with Attention and Fixed Pre-trained Embeddings

Figure 2: Our Two Main Models: Simple Seq2Seq and Bidirectional Seq2Seq

3.2.2 Bidirectional Seq2Seq with Attention and Fixed Embeddings

In order to account for the structural difference in encoder and decoder training example pairs, we decided to utilize bidirectional LSTM for our encoder. As described in Figure 2b, encoder LSTM cells build up one bidirectional rnn per layer, which produces both forward and backward states. At each timestep t, we concatenate the forward and backward states to create attention candidates to be used later during decoding. Our attention mechanism is explained with detail in section 3.2.5. Since the attention mechanism depends on the concatenated state vectors, at each decoder timestep the decoder is able to capture future and past sentence information of the encoder symbols in the sentence. In order to give some heavier weight on the forward information of our encoder symbols, only the forward state was passed to the decoder as the context vector. This allowed our model to respond heavily for test example pairs with the same sentence order, while being able to stay responsive for reverse order sentence pairs.

3.2.3 Buckets for Sentence Length

Our translation model takes advantage of bucketing strategy. We use "buckets" to handle situations where one sentence in one language set is converted into a sentence with different length in the other set. Consider the Shakespeare parallel corpora as an example. After running some analysis, we found that the sentence written in modern English tends to be less wordy or more concise compared to the same one written in Shakespearean English. In other words, when translating modern English to Shakespearean English, we will have modern version sentences of different lengths, supposedly L_1 for input and Shakespearean sentences of longer lengths L_2 for output. Hence, for Shakespeare model, we assigned four buckets: $(5, 10), (10, 15), (20, 25), (40, 50)$. This implies that a modern English sentence with five tokens are matched with a Shakespearean English sentence with ten tokens so that if a 5-words-long sentence is fed into encoder as an input, we expect 10-words-long sentence as a decoder output. For rap lyrics, we have done analysis and assigned buckets: $(7, 9), (9, 11), (13, 15), (40, 50)$.

We also introduce a special "__PAD" symbols for flexible bucketing. Using bucketing, we do not have to take account for every possible pair of L_1 and L_2. Instead, we can simply put __PAD symbols at the end of the output if "EOS" appears before the bucket size.

3.2.4 Embedding Strategy

In simple Seq2Seq model[4], Sutskever et al trained 1000 dimensional word embeddings. Following that, in our simple Seq2Seq model, we implemented 4 layers of 256 LSTM cells for each layer with an input vocabulary size of $10,000$ and output vocabulary size of $10,000$. The embedding size was 300.

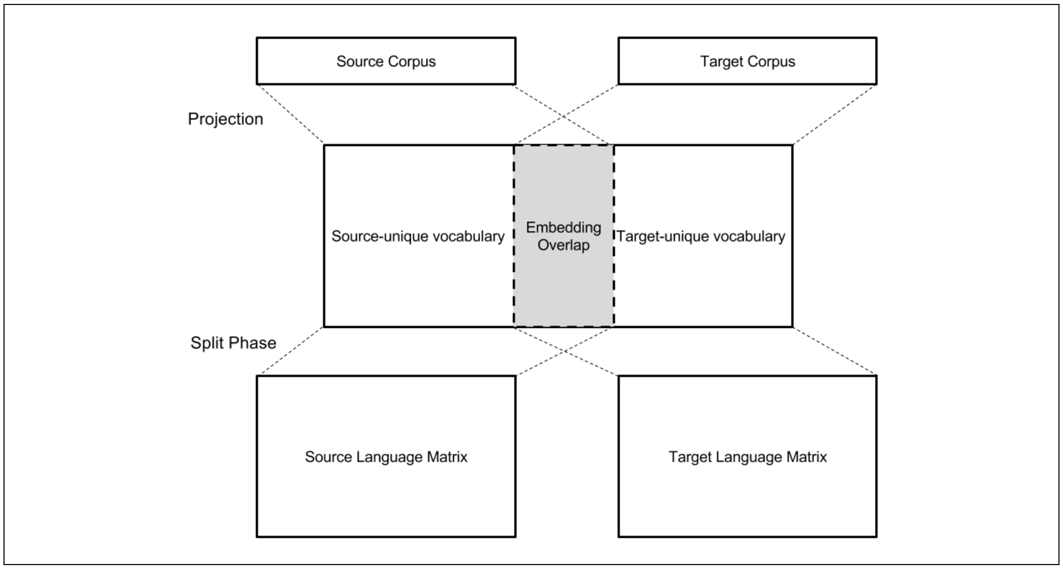


Figure 3: Embedding projection and split

However, when it comes to the attentive bidirectional Seq2Seq model, high-dimensional embedding size raised perplexity, hindering the model training. The training sets were inherently small compared to parallel corpora usually used for neural machine translation (about 9MB for Shakespeare and about 30MB for rap compared to few GBs that are usually used for NMT), and therefore, as we introduce more parameters to train in bidirectional Seq2Seq, we could not really drop the perplexity of the model to meaningful value. For this reason, we tried to use fixed embeddings for both encoder and decoder. We have tried many different choices. We first tried Google's pretrained word embeddings ([9]) and tried using different word embeddings for the encoder and decoder. To be more specific, we trained word vectors for the input language (modern English in the Shakespeare model) and the output language (Shakespearean English) separately. Both choices worked much better than using trainable embeddings, but we concluded that training word2vec on merged dataset of input language and target language and using this single word2vec embedding to both the encoder and decoder works better. The insight behind this is the following: while word choices are crucial in defining writing style, many words such as "I", "and", "the", and "but" perform the same roles and functions in two different styles. In this sense, if we train word embeddings on the merged data set of two different writing styles, we are not only able to capture meanings of words in different writing styles, but also meanings and functionalit ies of common words that coexist in two different styles.

3.2.5 Attention on Bidirectional Context

Sequence-to-sequence model has been shown to perform very well when combined with attention mechanisms. We adopted the attention mechanism used in[2]. The mechanism is as follows :

$$u_i^t = v^T tanh(W_1' h_i + W_2' d_t) a_i^t = softmax(u_i^t)$$
$$d_t' = \Sigma_{i=1}^{TA} a_i^t h_i$$

where W_1', W_2' are parameters to be learned. The encoder hidden states, one for a pair of Bidirectional LSTM cell in the encoder, are denoted $(h1, ..., hTA)$. However, the dimension of each h_i is double the dimension of the hiddens states that google used. This is because unlike google's model which had an uni-directionall RNN for its encoder, our model utilized bidirectional LSTM for our encoder and thus had double the dimension for top-layer hidden states. By concatenating the forward and backward hidden states, we were able to capture the future and past information at a given timestep. This allowed our decoder to consider both the forward and backward directional information of its encoder counterpart. Although introducing new parameters decreased the speed of the training process, we have found that our model performs much better with the attention mechanism.

3.2.6 Decoding Strategy

In our simple Seq2Seq model and bidirectional Seq2Seq model, we utilize scoring functions to get the most likely output for each time step. The drawback is that we are treating each time step independently when we are deciding the final output. It is true that recurrent structure of the model already captures time dependency in some sense, but we thought that using beam search when choosing final decode outputs might help our results. This decoding method is used by Google's up-to-date neural machine translator[6]. For our model, we used beam size of 3 as it is known that "using fewer (4 or 2) has only slight negative effects on BLEU scores"[6]. Hence, we keep three hypotheses that score the highest for each step and prune all other possibilities.

3.2.7 Optimization and Learning Rate

We have tested on three different optimizers: stochastic gradient descent, AdaGrad optimizer, and Adam optimizer. After having tested three different optimizers with several different initial learning rate varying from 0.001 to 0.5 and decaying rate of 0.9, we trained our models with Adam optimizer with starting learning rate of 0.5 and decays it by 0.9 for every 1000 epochs after 4000 epochs. We could reach substantially low perplexity, 1.24 and 1.98 for the first and second bucket of bidirectional Shakespeare model respectively. This is a large improvement from the simple Seq2Seq model that reported 3.28 and 7.65 for each same bucket.

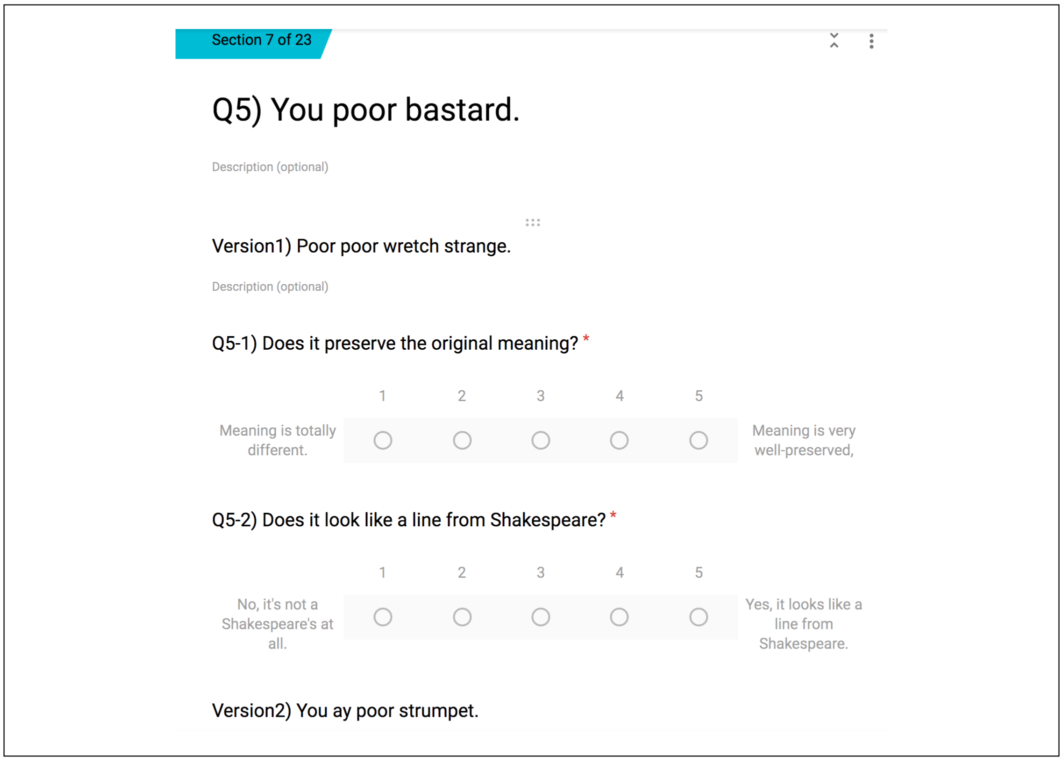
4 Evaluations

While "Bilingual Evaluation Understudy" or BLEU[8] score is typically used to measure the quality of machine-translated texts between two different languages, it seems reasonable to assume that it could also be useful for measuring stylistic alternations[1] as we have a very well-polished parallel corpora for our Shakespeare dataset.That is, if our models convert given texts into different texts by not only preserving semantics but also mimicking styles well, the results from our model will elicit high BLEU score with respect to the desired results.

However, BLEU is not a exact measure evaluating style similarity of two texts. For this reason, we also executed an evaluation based on human judgments, particularly on semantic similarity and stylistic aspect.

4.1 Human Evaluation

We have conducted survey on twenty cherry-picked examples from our dev set - 10 questions were on Shakespeare and 10 questions were on rap. For Shakespeare conversion examples, we gave two different versions of style conversion for each sentence, one from simple Seq2Seq and one from bidirectional Seq2Seq. Then, participants were asked to score on whether meaning is preserved and style does look like Shakespeare's for each version. The scale was integer value from 1 to 5 (see Figure 4). We collected data from 38 participants.



Section 7 of 23

Q5) You poor bastard.

Description (optional)

Version1) Poor poor wretch strange.

Description (optional)

Q5-1) Does it preserve the original meaning? *

	1	2	3	4	5	
Meaning is totally different.	○	○	○	○	○	Meaning is very well-preserved,

Q5-2) Does it look like a line from Shakespeare? *

	1	2	3	4	5	
No, it's not a Shakespeare's at all.	○	○	○	○	○	Yes, it looks like a line from Shakespeare.

Version2) You ay poor strumpet.

Figure 4: Actual Survey Questionnaire Example - Shakespeare

Here are some more examples of sentences asked on the survey. We could observe that there is a evident trend that people thought the results from bidirectional model preserved original meaning of the text better and converted given sentences into more Shakespearean-like sentences. Also, people thought that both models were not good at preserving meaning of relatively long sentences, but they assessed that the bidirectional model still generated a more Shakespearean-like sentence (see Table 1 and Table 2).

Input	Simple Seq2Seq	Meaning Score (out of 5)	Style Score (out of 5)
Have you killed Tybalt and are you crazy	Hast you have lost mad	3.11	3.17
You poor bastard	Poor poor wretch strange.	3.17	2.67
I can explain everything	I can have all all.	2.44	2.44
my unhappy son comes home in order to hide from the light	My son comes to the home comes to the son , come.	2.28	2.17

Table 1: Examples of Outputs of Our Simple Seq2Seq Model

Input	Bidirectional Seq2Seq	Meaning Score (out of 5)	Style Score (out of 5)
Have you killed Tybalt and are you crazy	Hast thou kill Tybalt and art thou murder	4.00	4.11
You poor bastard	O poor strumpet	4.00	4.22
I can explain everything	I can confess	3.56	3.61
my unhappy son comes home in order to hide from the light	My poor master come in home to fetch from the light	2.61	3.39

Table 2: Examples of Outputs of Our Bidirectional Seq2Seq Model

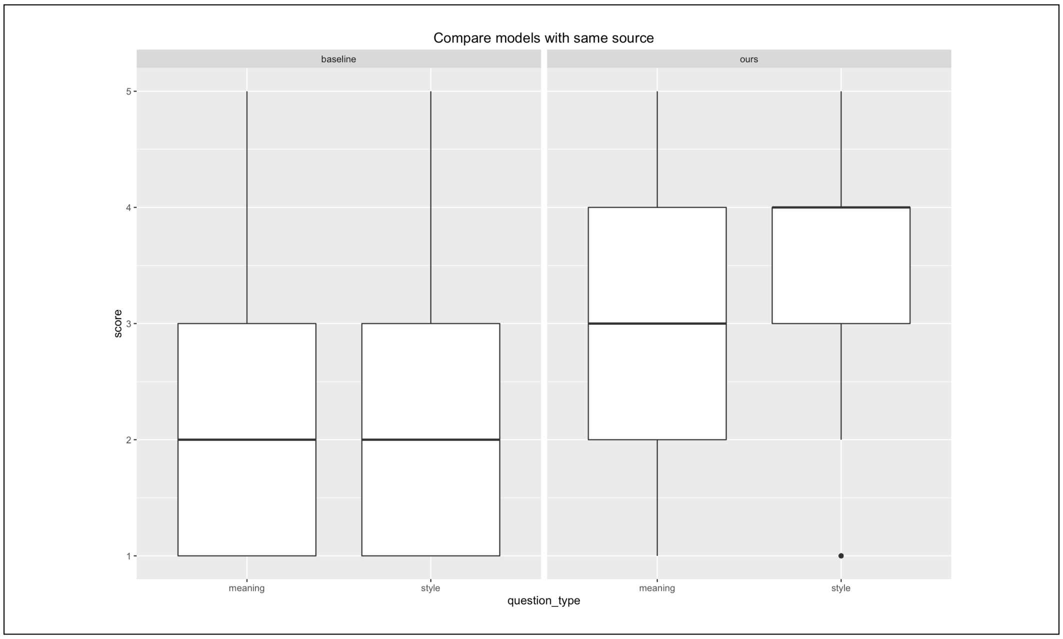


Figure 5: Box Plots Comparing Model Performance with Baseline

In order to verify whether our bidirectional model for writing style conversion is adaptable to different style conversion task (rap lyrics in our case) and scalable with different number of layers(or parameters), we tested on results on rap dataset as well. One of the examples was *“What is up”* to *“Whats happenin.”* As shown in Figure 6, the participants reported a similar score on both semantic preservation and style conversion compared to the Shakespeare task.

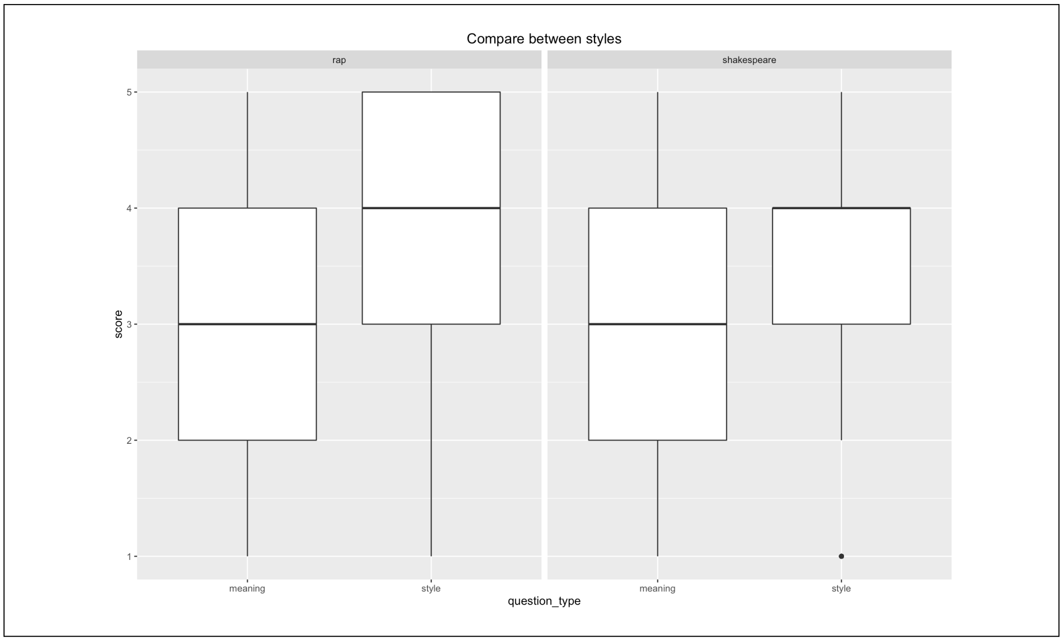


Figure 6: Box Plots Comparing Model Train from Different Corpuses

4.2 BLEU Measure

As in Wei Xu et al's previous paper on writing style conversion[1], we have computed BLEU scores of our models besides the human evaluation metrics. We utilized a widely-used, standard BLEU score[8] scheme implemented in Python nltk library. We ran our model against 1444 sentences in our randomly generated dev set, stored the results of each model, and computed BLEU scores of the results with respect to the original Shakespeare sentences. The third and fourth column in the Table 3 are arithmetic means of 1444 BLEU scores of our models. The first two columns are BLEU scores reported by Wei Xu et al in paper[1]. Since the paper only contains plots not exact numerical values, we put the range instead of numerical values for their BLEU scores. Still, we can easily observe that our models have significantly improved the BLEU scores. It is a impressive result that even our simple Seq2Seq model with global attentions outperformed all the previous models and in addition, our bidirectional sequence to sequence model with global attention and fixed pretrained embeddings for encoders and decoders outscored 16plays_16LM model (Wei Xu et al's model based on phrase dictionary) by greater than 25%.

	16plays_16LM[1]	Dictionary Model[1]	Simple Seq2Seq Model with Global Attention	Bidirectional Seq2Seq Model with Global Attention and Separately Pretrained Encoder/Decoder Embeddings
BLEU Score	< 0.3	< 0.2	0.4208	0.5677

Table 3: BLEU Scores of Two Baseline Models Discussed in Wei Xu's Paper[1] and Our Models

5 Conclusion

We explored whether the state-of-art neural machine translation models and techniques can be applied to writing style conversion problem. Through many experiments on network structures (simple sequence-to-sequence model vs. deep bidirectional model), cells (LSTM vs. GRUs), embedding methods (train them on-line vs. pre-train them with source and target vs. simply use Google word2vec or Glove), and hyperparameters, we concluded that 2-layered (or 4 layers for rap) bidirectional LSTM sequence-to-sequence model with global attention and word embeddings that are trained on source and target dataset separately performed best. When it comes to BLEU measure, one of the standard metrics for translation tasks, our model (and the simple Seq2Seq model with attention as well) outperformed previous models mentioned in Wei Xu et al[1], which were not based on modern machine learning techniques. Our human evaluation results also showed that our model did impressive work on converting a given sentence into a sentence with target style. However, there was also some limitations. The human evaluation result showed that our model was not equally good at preserving the original meaning of text. In addition, when it comes to rap, it was hard for our model to capture some domain specific features such as rhyme. Hence, for future applications, we believe that we can use this sequence-to-sequence model along with some carefully picked, hand-engineered features that are important in target domain.

Acknowledgments

We thank Professor Chris Manning for being our mentor and all other teaching staffs for their assistance with this study and for their comments that greatly improved the paper.

References

[1] Wei Xu et al. Paraphrasing for Style, *Proceedings of COLING 2012: Technical Papers*, pp. 28992914, COLING 2012, Mumbai.

[2] Oriol Vinyals et al. Grammar as a Foreign Language, *Proceedings of Neural Information Processing Systems 2014.*

[3] Stylistic Transfer in Natural Language Generation Systems Using Recurrent Neural Networks, *Proceedings of EMNLP 2016 Workshop on Uphill Battles in Language Processing: Scaling Early Achievements to Robust Methods*, pp. 4347, Austin, TX.

[4] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to Sequence Learning with Neural Networks, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014*, pp. 31043112, Montreal, Quebec, Canada.

[5] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. Effective Approaches to Attention-based Neural Machine Translation, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1412-1421, Lisbon, Portugal. Association for Computational Linguistics.

[6] Yonghui Wu et al. Googles Neural Machine Translation System: Bridging the Gap between Human and Machine Translation, *ArXiv e-prints, 2016*

[7] Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate, *Proceedings of International Conference on Learning Representations 2015*

[8] Papineni, Kishore, Salim Roukos, Todd Ward, and WeiJing Zhu. BLEU: A Method for Automatic Evaluation of Machine Translation, *Association of Computational Linguistics, 2002*

[9] ”Word2vec.” Google Code Archive - Long-term storage for Google Code Project Hosting. N.p., n.d. Web. 21 Mar. 2017.