# 

[1. Writing Style Conversion using Neural Machine Translation:](#_q8aofk6zwrgf)

[Background/related works](#_no39uuikvj4c)

[Data Collection](#_1w2ax536dn3e)

[Simple Seq2Seq with Attention](#_31sbbgfllvdk)

[Bidirectional Seq2Seq with Attention and Fixed Embeddings](#_ybmu0sdqjio5)

[Buckets for Sentence Length](#_gumgoelfkux1)

[Embedding Strategy](#_6ba7foxjxjyi)

[Attention on Bidirectional Context](#_f3zkb2871i8)

[Decoding Strategy](#_ocf65a5x29hi)

[Optimization and Learning Rate](#_dlu5snkqxbms)

[Evaluations](#_gbxglshc2e0z)

[Human Evaluation](#_xabztuv1shi3)

[BLEU Measure](#_j05d2alnyakt)

[2. Efficient Transfer Learning Schemes for Personalized Language Modeling using Recurrent Neural Network](#_rflmp9jpa29i)

[Abstract](#_dsrcgkqwh0oq)

[3. Multimodal Transfer: A Hierarchical Deep Convolutional Neural Network for Fast Artistic Style Transfer](#_i9ihg8vqeh4p)

[Abstract](#_k0lr5os85k7v)

[Architecture](#_20t8vmqtoq2i)

[Loss Functions](#_eyb1i28vd55g)

[Our thoughts:](#_u4sklpvoqdd)

[4. From A to Z: Supervised Transfer of Style and Content Using Deep Neural Network Generators](#_sskjv814g5ae)

[Abstract](#_i1eyq4c6mvq6)

[5. Demystifying Neural Style Transfer](#_x1zsft864e5i)

[Abstract](#_48olfj72n9ch)

# 

# 

# 1. Writing Style Conversion using Neural Machine Translation:

<https://web.stanford.edu/class/cs224n/reports/2757511.pdf>

### Background/related works

Abstract: apply sequence to sequence neural machine (Seq2Seq[[1]](#footnote-0)) 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

* Scarce effort in understanding people’s writing style
* Google tried RNN, particularly seq2seq to solve syntactic parsing problem
* Seq2Seq can be applied to other monolingual for grammar parsing
* Research idea: RNN for writing style conversion

Data acquisition: Google Translator engine

* Successful in translation between 2 parallel corpora
* Encoder-decoder model is capable of preserving the meaning of the encoded text and decode it into the target language

Performance:

* embedding matrix of the source and target vocab,
* hyperparameters,
* global attention mechanism,
* other details of bidirectional Sequence to Sequence model (Seq2Seq) mimicking BLEU score

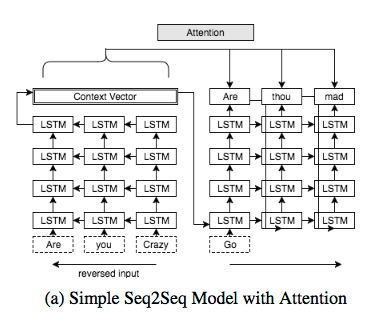
Challenge to solve style conversion problems:

1. Monolingual writing styles have no corresponding representation in colloquial style
   1. Only modern English has parallel dataset for translation from old English literature: Sparknotes
   2. Seq2seq requires a parallel corpora of both source and target languages
2. Bucketing and padding strategy to handle sentences of different length:
   1. Sentences up to length L1 will be assigned to 1 bucket
3. Training the embedding matrices within a bidirectional seq2seq model with LSTM -> increases number of parameters by millions -> hard to optimize
4. Bidirectional seq2seq captures directional info of English sentences
5. Other hyperparameters: learning rate, decay rate for learning rate, # of layers, # of cells, word embedding dimensions, etc.

### Data Collection

* We need parallel corpuses that are aligned at sentence level -> chose:
  + chose Shakespeare’s literary works ( <https://github.com/cocoxu/Shakespeare>)
    - Parallel corpuses from Sparknotes (40,000 sentences)
  + English rap lyrics (no dataset with sentence-level aligned parallel corpuses) -> scraping with 300,000 lines from 8K songs
    - Google Translate to translate rap lyrics between English <-> French
    - Using Scrappy, Selenium (<https://github.com/sjang92/nmt-style-converter/tree/master/dataCollector>)
    - Converted rap lyric files into token-id based files

### Simple Seq2Seq with Attention

* Input were passed in reverse order (better performance)
* Encoder: Up to 4 layers of unidirectional LSTM, each layer produces a context vector of dimension dim(state). Context vectors are fed to decoder LSTM cells
* Decoder: each timestep, decoder output is projected back onto the vocab space and produces softmax loss combined w the target symbol to train the parameters occurring in all past timesteps

This model performs poorly where encoder and decoder sentences had drastically different structures

### Bidirectional Seq2Seq with Attention and Fixed Embeddings

This is the solution to structural difference

* Encoder: LSTM cells are bidirectional. Each timestep t, we concatenate fwd/bwd states to create attention candidates to be used by decoder layer
* Decoder: only forward information was passed to the decoder as context vector
* Respond heavily for test example pairs while responsive for reverse order sentence pairs

### Buckets for Sentence LengthScreen Shot 2017-06-10 at 1.58.00 PM.png

Shakespeare model: (5, 10), (10, 15), (20, 25), (40, 50)

Rap: (7, 9), (9, 11), (13, 15), (40, 50)

### Embedding Strategy

* Simple Seq2Seq
  + 4 layers: each has 256 LSTM cells, input vocab size = 10K, output vocab size = 10K. Embedding size = 300
* Bidirectional Seq2Seq:
  + High-dim embedding size raised perplexity -> hinder model training
  + Training set is small -> with more parameters to train in bidirectional Seq2Seq, could not drop perplexity -> use fixed embeddings
    - training word2vec on merged dataset of input language (modern English in Shakespeare model) and target language (Shakespearean English) and using this single word2vec embedding to both the encoder and decoder works better.

### Attention on Bidirectional Context

Seq2Seq model performs well when combined with attention mechanism

* Concatenating fwd + bwd hidden states -> capture future and past info at a given timestep -> allow decoder to consider both fwd and bwd directional info of its encoder counterpart
* New parameters decreased training speed
* Performs better with attention mechanism

### Decoding Strategy

* Use scoring functions to get most likely output for each time step
  + Drawback: treating each timestep independently
  + Use beam size = 3: keep 3 hypotheses with highest scores for each step

### Optimization and Learning Rate

3 optimizers: SGD, AdaGrad, Adam -> Adam optimizer with start learning rate 0.5 and decays by 0.9 for every 1000 epochs after 4000 epochs -> low perplexity

### Evaluations

* Bilingual Evaluation Understudy (BLEU) score is not an exact measure for style similarity of 2 texts
* Therefore evaluation is done by human for semantic similarity and stylistic aspect

#### Human Evaluation

* 38 participants
* 10 questions on Shakespeare
  + Each sentence: 2 diff style conversions (simple and bidirectional Seq2Seq) -> score on if meaning is preserved and style looks like Shakespeare
  + Result: both models are not good at preserving meaning of relatively long sentences, but bidirectional generates more Shakespeare-like sentence
* 10 questions on rap:
  + Similar score on both semantic preservation and style conversion

#### BLEU Measure

* BLEU score from Python nltk library
* Ran models against 1444 sentences in randomly generated dev set
* Store results of each model and their BLEU scores with respect to the original Shakespeare sentences

# 2.

# 3. Efficient Transfer Learning Schemes for Personalized Language Modeling using Recurrent Neural Network

<https://arxiv.org/pdf/1701.03578v1.pdf>

## Abstract

Efficient transfer learning methods for training a personalized language model using current NN with short-term memory architecture:

* Model is updated to a personalized language model w small user data + limited computing resource -> useful for mobile env when data is not transferred out of the device

Not relevant to our project. We probably don’t need it

# 4. Multimodal Transfer: A Hierarchical Deep Convolutional Neural Network for Fast Artistic Style Transfer

<https://arxiv.org/pdf/1612.01895v2.pdf>

## Abstract

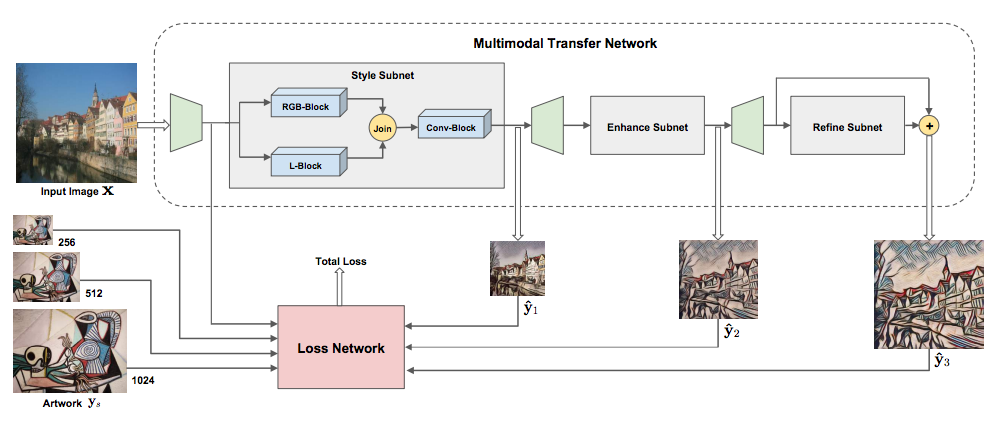
Transferring artistic styles -> photos fails to capture small, intricate textures and maintain correct texture scales of the artworks

Proposal: multimodal convolution NN that considers faithful presentation of both color and luminance channels

-> results are visually pleasing and more similar to desired artistic styles of color

**Issues:**

* Transfer run time is long because of the online interactive optimization procedure
  + Network can be trained offline with the same loss criterion -> 100 times faster
* Resolution constraint -> texture scale mismatch
  + Hierarchical network and training scheme to learn both coarse texture and fine brushwork by using multiple scales of a style image -> combine multiple models into 1 to handle larger image size
  + Use both color and luminance channels for style transfer

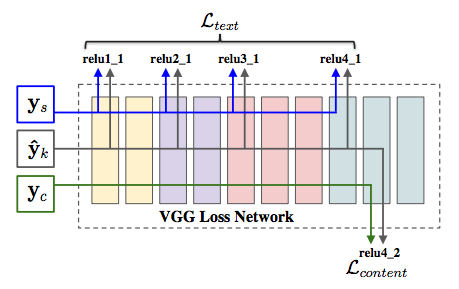


## Architecture

* Style subnet: input image is resized to 256 to capture large color and texture traits of the art work ()
* Enhance subnet: 1st output is then sampled into 512, transferred to output . It enhances the stylization strength and resized back to 1024
* Refine subnet: removes local pixelization artifacts and further refines the result ()

### Loss Functions

* Stylization loss is also derived from a pre-trained VGG-19 network optimized for object recognition



## Our thoughts:

* While this is a paralleling idea of our project, this paper is about image and visual conversion technology which is a completely different animal than ours so we may not be able to use this for our project (language style transfer)

# 5. From A to Z: Supervised Transfer of Style and Content Using Deep Neural Network Generators

<https://arxiv.org/pdf/1603.02003v1.pdf>

## Abstract

New NN architecture to solve single-image analogies: generate an entire set of stylistically similar images from just a single input image

Again, this is about image processing which is not relevant to our project

# 6. Demystifying Neural Style Transfer

<https://arxiv.org/pdf/1701.01036v1.pdf>

## Abstract

Neural Style Transfer from 1 image to another: minimize the Maximum Mean Discrepancy (MMD) with the 2nd order polynomial kernel.

* To match the feature distributions betw style images and the generated images

Again, this is not about language processing so it’s not relevant to our project

1. https://www.tensorflow.org/versions/master/tutorials/seq2seq [↑](#footnote-ref-0)