Text Style Transfer

Zhuoming Li[1] and Gao Han[2]

[1]zli342@stanford.edu, Stanford University
[2]gh352@stanford.edu, Stanford University

https://youtu.be/2huzgVpYXDQ

Stanford
Computer Science

Abstract & Motivation

Text style transfer is a technique to rewrite sentences from one style to a different style while at the same time preserving the semantic contents. The main challenge of this project is how well we can preserve the content meaning after the style transfer process due to the lack of parallel data that connects the source and target styles. In this project, we use GANs to apply style transfer onto textual contents, trained on tweets scraped from Obama and Trump's twitter account. Our style transfer generator is able to rewrite tweets from Trump to ones that has Obama's writing style (based on his tweets) and vise versa, and the generated contents achieves 80% of accuracy based on a classifier we trained to distinguish tweets from Obama and from Trump.

In many situations, for a robust language generation system, flexible control over its expression is necessary. Contents may need to be expressed in different ways. In the case of anonymization, writings should be converted to a style that's neutral and common. Another example is to help with content understanding - for instance politicians often prefer languages of vagueness and exaggerations, and with style transfer, it's possible to strip away the hyperbole.

Stanford University

Dataset & Preprocessing

Since we plan to do text style transfer for Barack Obama and Donald Trump, we obtained their data by scraping tweets from their Twitter accounts @BarackObama and @realDonaldTrump. We use the twitter API to scrape all tweets with a scraper here. Specifically, the data for Trump is from 2017/1/1 to 2018/12/31 (6730 tweets), and the data for Obama is from 2012/1/1 to 2013/12/31 (5275 tweets). The date range is chosen based on their incumbency, because for other dates the data is sparse and the style can be different. For example, prior to 2016, Trump seldomly uses "make america great again", and after 2017 Obama only sends a few tweets in a month and is mostly about his family resulting in less content match).

Since tweets are somewhat different from normal writings and contains twitter specific marks, we need the following processing before we use the data for training.

1. Mask all targets (@), hashtags (#), and numbers to decrease OOV count.
2. Insert spaces to punctuation so that they will be treated as words. This is very important as many punctuation has sentiment and can represent one's writing style (such as "!" at the end of the sentence, which is a very strong signal).
3. Break each tweet into several examples, each containing exactly one sentence. We observed that many tweets are fairly long (more than 3 sentences and 20 words), and long ones will be especially hard to train or to get good results. In addition, chopping tweets into several examples can increase the size of our training set as well.
4. Remove irrelevant contents. Though tweets from Trump's account are always written by himself (at least it looks like so), many tweets from Obama's account are drafted by his staff and are normally in the format of "President Obama says…", which adds noise to the training set. Since the content inside these tweets are still useful data, we removed the prefix and kept the content for these tweets.

Representation & Architecture

The model consists of an encoder E, a generator G and two discriminators D1, D2 as illustrated in figure below:

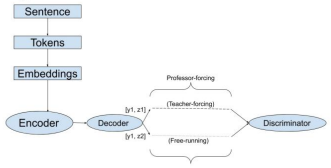

Figure 1: model overview

As part of the problem setup, we assume our focus is on two different styles y1 and y2. Sentences x1 and x2 from Tweets of each style are first broken down into tokens, which then are transformed into embeddings.

Encoder takes word embeddings as input and outputs the corresponding latent representation z. After having obtained latent z, we unroll decoder (a.k.a generator) on latent representation z to generate two sequences of intermediary states: one using teacher-forcing and the other using free-running approach that utilizes the Softmax output of the RNN cell from previous timestamp.

As the last step, we apply discriminator to differentiate between the two sequences, which aims to align the latent space during generation to ensure the transferred x2 (now of style y1) matches the population of the original x1.

To ensure reconstruction of the original content, minimize the expectation of P(x1 | y1, z1) and P(x2 | y2, z2):

$$\mathcal{L}_{rec}(\theta_E, \theta_G) = \mathbb{E}_{x_1 \sim X_1}[-\log p_G(x_1|y_1, E(x_1, y_1))] + \mathbb{E}_{x_2 \sim X_2}[-\log p_G(x_2|y_2, E(x_2, y_2))]$$

The adversarial cost is then defined as:

$$\mathcal{L}_{adv_1} = -\frac{1}{k}\sum_{i=1}^{k} \log D_1(h_1^{(i)}) - \frac{1}{k}\sum_{i=1}^{k} \log(1 - D_1(\tilde{h}_2^{(i)}))$$

To put everything together, we have the overall objective:

$$\mathcal{L}_{rec} - \lambda(\mathcal{L}_{adv_1} + \mathcal{L}_{adv_2})$$

Results & Discussion

With co-trained embedding, we obtained the following loss graph and reduced total loss from 90 to around 69.

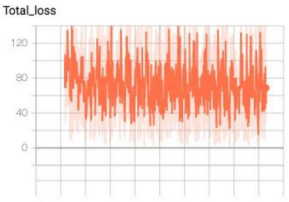

Figure 2: total loss vs. time

After training, we were able to generate sentences using beam search with width of 8 and produced the following table of examples (input is the (label 0 for Obama and 1 for Trump, original sentence) pair fed into the model; reconstructed is the model output with the same style as input, and transferred is the output with the opposite style):

Input	Reconstructed	Transferred
1 make america great again	thank you !	add your name:
0 tell congress to get the american	we want to get the american people .	i will be a great job !
0 put it up for a vote .	it's time for the american people .	the u . s .
1 we must maintain a strong southern border .	we will be a great job .	we want to do the american people .

The results are still far from ideal, but they have captured the style of each corpus to a certain degree. For instance, *tell congress to get the american* is referring to encourage congress to get more american people to vote and the reconstructed sentence was able to keep this formation and expanded it further with *american people*. The transferred counterpart is also able to embody the flamboyant Trump style of *i will be a great job !*.

Results & Discussion (cont.)

We further experimented with pre-trained Google News Word2Vec embeddings and obtained the following examples with the same set of inputs as before:

Input	Reconstructed	Transferred
1 make america great again	thank you to vote !	add your name:
0 tell congress to get the american	add your name to make your voice to get up .	thank you to see the house of our country .
0 put it up for a vote .	it's time to see the <unk> .	in the world .
1 we must maintain a strong southern border .	we will be a great job .	we need to make our country .

Generally speaking, with pre-trained embedding, the model is able to generate longer and more sensible sentences. For instance, *make america great again* is reconstructed into *thank you to vote!*, which includes *to vote!* with an exclamation point that is known to be a consistent theme of Trump's style.

Discussion & Future Works

If more time is permitted, we would like to experiment with sentiment tokenizer and increase dataset size by finding more literature written by Trump and Obama. BERT embedding is another idea that we would like to try that can potentially help model to better learn the underlying distribution.

References

[1] R. B. T. J. Tianxiao Shen, Tao Lei, "Style transfer from non-parallel text by cross-alignment," 2017.
[2] C. D. E. P. X. T. B.-K. Zichao Yang, Zhiting Hu, "Unsupervised text style transfer using language models as discriminators," 2018.
[3] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vander-plas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duch-esnay, "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research , vol. 12, pp. 2825–2830, 2011.