AI Trump: A Tool for Transferring Sentences to Trump-Style

Weiting Jin
waiting_jin@ruc.edu.cn

Yutang Wang
2020201360@ruc.edu.cn

Abstract

We present a novel tool for transferring sentences to sound like what Trump would say – AI Trump. We decompose Trump-Style into two part: Speaking-Style and Writing-Style, and use two models to imitate them separately. To get parallel corpora of train, we prompt Large Language Model Vicuna to generate from Trump's words. To learn the styles, we fine-tune BART on the created corpora. Finally, we conduct both automatic evaluation and human evaluation, which both suggest that AI Trump performs excellently in the transfer task. Source is available on Github[1].

1 Introduction

"MAKE AMERICA GREAT AGAIN!!" Like him or hate him, the former US President Donald Trump's structure of language is particularly distinctive. In many ways, he has used language to create a brand for himself: one that leverages a feeling of strength, a sense of determination, and an impression that he can always get the job done. Many have tried to imitate Trump's style with the methods like Markov-chain. As the development of *large language models* (LLMs), we consider it a great opportunity taking advantage of LLMs to do the Trump-style transfer task. Although LLM itself can do the task alone perfectly, it is huge and many have no disk for LLM to run on. Therefore, in this paper, we try to build a much smaller model based on the knowledge of LLM – **AI Trump**.

Text style transfer is the task of changing the stylistic properties of the text while retaining the style-independent content like the overall semantic. By far style transfer has garnered increased interest and many researchers have tried different methods, which always lead to progress. Approaches include training with parallel text data (Rao and Tetreault, 2018), zero-shot learning using large language models (Reif et al., 2021), and so on. For a specific target style, parallel text data brings more accurate and exact directions for transferring. In this paper, we aim at making the given neutral sentences look like what Trump would say, thus the Trump-style transfer task will be performed with the BART (Lewis et al., 2019) model fine-tuned on specially created parallel data.

What's more, by analyzing Trump's sentences, we split the target style into two parts: speaking style and writing style, which we fine-tune two models with two different datasets to learn separately.

However, although words from Trump are easy to get, we don't usually get their neutral version. That's where LLM is used. As its powerful NLP ability, LLM is a good paired sentences generator. By prompting the LLM to generate neutral style of Trump's words, we get a Trump-Neutral parallel corpus. We generate Trump's speeches parallel corpus and tweets parallel corpus. Finally, we train speaking style model and writing style model on them respectively, and the final AI Trump significantly performs well in terms of automatic and human evaluation.

Figure 1 presents the overall schematic diagram of our proposed model, AI Trump. The contributions of our work are summarized as follows:

- We propose a decomposition of Trump-style: speaking and writing style, and train two-step model using created parallel data to learn them separately, which yields good results,
- We suggest a novel method of creating parallel corpora for text style transfer task.

2 Analysis

2.1 Speaking Style

Before we transfer sentences to Trump-style, we should first analyze President Donald Trump's use of language (web link). Is Trump's unique brand

[1] https://github.com/Githubjinwt/AI_Trump

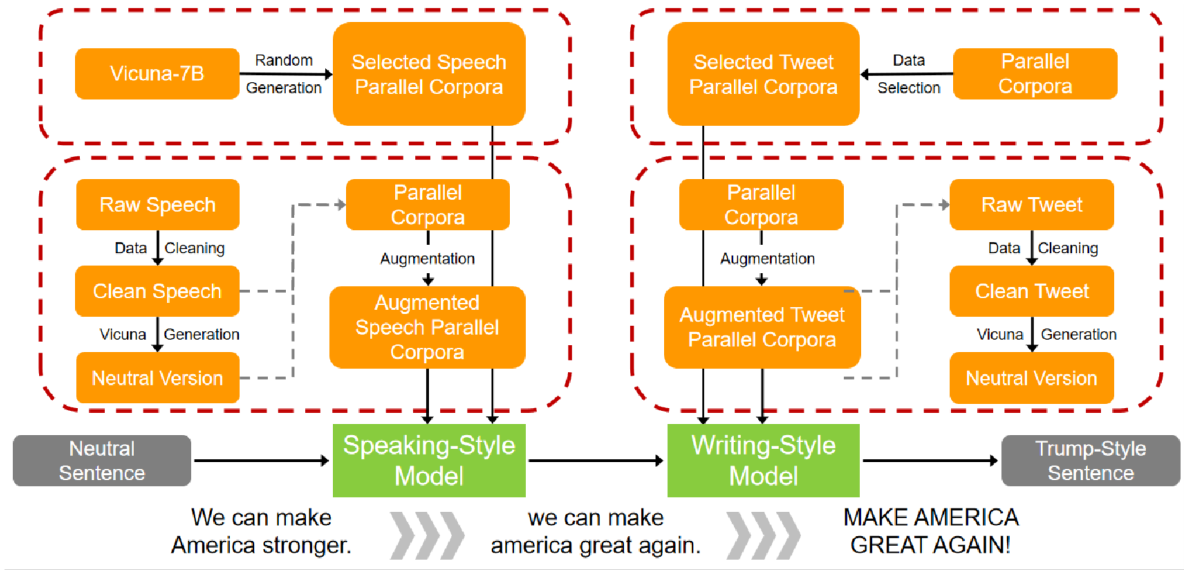

Figure 1: Schematic diagram illustrating the workflow of AI Trump

of presidential oratory deliberate and strategic, or just the disjointed output of a disordered mind, or something else? Tiffany summarized six linguistic devices commonly used in Trump's speech: hyperbole, repetition and intensifiers; directness; sentence fragments; digressions and segues; grade level; and sales talk. The distinctive characteristics bring both convenience and challenges for models to learn. We call such characteristics "speaking style".

2.2 Writing Style

However, it's not enough. The magic power of Trump's language also comes from his tongue full of tension and strength. Take the example:

- Make America great again.
- MAKE AMERICA GREAT AGAIN!!

Same sentences but different writing styles. Obviously, the upper sentence is straight and narrow, but the lower sentence appears more "Trump". Watching the sentence, it seems that the president stands right in front of eyes with his red tie, delivering speech in an energetic fashion. Such characteristic that uses capital letters and special punctuation shows Trump's tongue perfectly, and we call it "writing style".

In summary, we decompose the target style into two parts, one of which is how Trump uses words, and the other is how Trump says it. We think it's better to use two independent models to learn the two styles separately, so in this paper we adopt a two-step model structure.

3 Definition

To make our discussion more clearly, in this section, we will first give a brief introduction to the style transfer task, and then discuss our proposed model based on our problem definition.

3.1 Problem Formalization

In this paper, we define the Trump-style transfer problem as follows: Considering a bunch of data $\{T_i\}_{i=1}^{K}$, and each data T_i represents one natural language sentence. For all the sentences in the dataset, they share one thing in common, that they are spoken or written by Trump. We call this characteristic "Trump-style". In other words, Trump-style includes speaking style and writing style, and is defined by the distribution of the dataset $\{T_i\}$. Also there's another bunch of data $\{N_i\}_{i=1}^{K}$. For each N_k, $k = 1, 2, ..., K$, N_k has the exactly same semantic as T_k, except that it's not from Trump. We call this characteristic "Neutral". The goal of style transfer is that: given a arbitrary natural language sentence X, rewrite it to a new one $\hat{X}$ which looks like "Trump-style" and preserves the information in original sentence X as much as possible.

3.2 Model Formalization

Generally, Transformer follows the standard encoder-decoder architecture. Explicitly, for a input sentence $X = (x_1, x_2, ..., x_n)$, the encoder maps inputs to a sequence of continuous representations $Z = (z_1, z_2, ..., z_n)$. And the decoder estimates the conditional probability for the output sentence $Y = (y_1, y_2, ..., y_n)$ by auto-regression. At each time step t, the probability of the next token is computed by a softmax classifier:

$$p(y_t|Z, y_1, y_2, ..., y_{t-1}) = softmax(o_t) \quad (1)$$

where o_t is logit vector outputted by decoder network.
Therefore, likelihood function can be expressed as:

$$p(Y|X) = \prod_{t=1}^{n} p(y_t|Z, y_1, y_2, ..., y_{t-1}) \quad (2)$$

and we denote the predicted output sentence of the network by $f(X)$.

4 Data Processing

As said above, we use a two-step model to learn Trump-style. For speaking style model, we aim at imitating Trump's use of language. Trump's speeches contain many expressions for everyday use, thus suitable for the first step. For writing style model, we aim at imitating Trump's tone of voice. Trump's tweets are specially written by Trump himself and contain his feelings, thus more suitable for the second step.

4.1 Parallel Corpora Generation

To generate parallel corpora, first we get raw data of Trump-style from Kaggle: Donald Trump Rally Speeches[2], and Trump tweets[3]. Then we clean the sentences to ensure good data quality. For speeches data, we simply split every sentence out and drop meaningless sentences. For tweets data, we drop all the links and sentences that Trump quoted from others. Then we get cleaned data.

With cleaned data, we can generate its neutral version using LLM, which in this paper, we choose Vicuna-7B (Zheng et al., 2023) as. Following the instructions[4], we get vicuna-weight prepared, deploy the model locally and then prompt it for generation. The prompting is:

- The following sentence was spoken by former US President Donald Trump, and now you need to transfer it to a normal style: "{}".

By both generation, we get speeches parallel corpus and tweets parallel corpus.

[2]https://www.kaggle.com/datasets/christianlillelund/donald-trumps-rallies

[3]https://www.kaggle.com/datasets/austinreese/trump-tweets

[4]https://github.com/Vision-CAIR/MiniGPT-4/blob/main/PrepareVicuna.md

4.2 Data Augmentation

Ideally, to build a text style transfer model, we need access to parallel text containing a large number of data pairs. However, as there is upper limit since Trump just made those speeches and tweeted those tweets, we need data augmentation.

The first method we use is **Back Translation**. We take advantage of Baidu Translator API, translating the given sentences into Chinese and back into English. In this process, synonym substitution and sentence structure reconstruction are made, but the original semantic is remained.

Another method we adopt is **Random Noise Injection**. In order to increase the robustness of our model, we take letter substitution to imitate writing error in daily life, which include *spelling error injection* and *QWERTY keyboard error injection*. The former one comes from common spelling confusion, and the latter one occurs when typing on a Keyboard layout due to the close proximity between keys. We generate such errors deliberately so that the model can recognize them.

4.3 Data Selection

Although model trained by a large amount of data has stronger generalization ability and better robustness, it doesn't always perform well in some specific occasions. So for comparison, we select a dataset that is fewer in quantity but higher in quality for training.

For speeches data, we use Vicuna to generate some data pairs directly: *"Give me a sentence said Donald Trump, and transfer it to a normal tone."* or *"Give me a random sentence in daily life, and make it sound like Donald Trump."*. For tweets data, we select from cleaned tweets data the most recent and typical Trump's tweets. All selected data will not be augmented.

After data processing, we get around 11W speeches parallel data pairs, with 8.8K selected pairs, and 3W tweets parallel data pairs with 980 selected pairs. *Figure 1* also shows our workflow of data processing.

5 Experiment

5.1 Implementation Details

We choose BART[5] as the model we fine-tune. The model consists of BiLSTM layers, starting

[5]https://huggingface.co/facebook/bart-large-cnn

with 12 encoder layers (each with Self-Attention-Layer – Norm-Layer – GELU-Activation-Layer – 2 Linear-Layer – Norm-Layer), and 12 decoder layers (each with Self-Attention-Layer – GELU-Activation-Layer – Norm-Layer – Encoder-Attention-Layer – Encoder-Attention-Norm-Layer – 2 Linear-Layer – Norm-Layer). Across all experiments, we use a learning rate of 3e-5, AdamW_hf optimizer, weight decay 0.1, maximum sequence length 128, and batch size 10. We use function "Seq2SeqTrainer" in Python library "transformers" for training pipeline. The models are trained for 10 epochs on Tesla-V100 32G GPU, where Vicuna-7B also runs.

5.2 Result Evaluation

A goal transferred sentence should be a fluent, content-complete one with target style. To evaluate the performance of AI Trump, following previous works, we compare four different dimensions of generated samples: 1) Style Accuracy, 2) Content Preservation, and 3) Fluency.

5.2.1 Automatic Evaluation

Style Accuracy (ACC) We measure the accuracy of style transfer automatically by evaluating the target sentiment accuracy of transferred sentences. We classify the sentence as Trump-style or Neutral-style by comparing *the lexical similarity between Output and Input* and *the lexical similarity between Output and Reference*. Here we calculate cosine similarity to present lexical similarity. In theory, Output should be more lexically similar with Reference than Input.

Content Preservation (CP) To measure content preservation, we calculate semantic similarity between Output and Reference. Here we cannot use embedding cosine similarity because Trump-style is largely reflected in his use of words. So we use a fine-tuned BERT model for computing semantic similarity on Github[6]. The score is between 1-5.

Fluency (PPL) Fluency is measured by a RoBERTa-based classifier of linguistic acceptability trained on the CoLA dataset (Warstadt et al., 2019). We calculate both the fluency of Output and Reference, because what Trump said may not be fluent. The lower PPL is, the better the model performs in fluency.

Evaluation results are shown in *Table 1*. In general, models trained on selected data have higher accuracy of style transfer and better content preservation, but less fluent. And all models behave more fluent than Trump himself, the fluency of his tweets is a total disaster.

Model	ACC	CP	out-PPL	ref-PPL
S1	49.54%	2.44	338.18	826.78
S2	59.66%	3.28	461.22	582.44
W1	72.21%	3.12	204.39	1019.40
W2	90.31%	3.59	272.69	315.04

Table 1: Automatic evaluation on Speaking-Style Model and Writing-Style Model. S1 stands for Speaking-Style Model trained on all data, S2 stands for that trained on selected data, and W1 W2 for the same reason.

5.2.2 Human Evaluation

In addition to automatic evaluation, we also conduct human evaluation experiments on some daily sentences. We randomly select 100 source sentences and we will show some samples in the Appendix. For each review, the reviewer should score on the above three aspects. We set three levels: Terrible, Okay, and excellent. Most of our reviewers score our model as excellent.

6 Conclusion

In this paper, we proposed a Trump-Style Transfer model – AI Trump, with a novel decomposition of Trump-style and a new method of creating parallel corpora. The model passed the automatic and human evaluation, and the transferred results looked like Trump intuitively. We demonstrated effectiveness of LLM in text style transfer task, and we think we get the job done well.

As future work, we would compare our model to state-of-the-art approaches. Also we would try more Seq2Seq model (T5 etc.) to fine-tune and try more LLMs to generate more parallel corpora. The prompting can also be improved. Many good research directions are worth to explore.

[6] https://github.com/AndriyMulyar/semantic-text-similarity

Source Input	Intermediate Output	Final Output
We can make America stronger.	we can make america great again.	**MAKE AMERICA GREAT AGAIN!**
China.	china.	**CHINA!CHINA.CHINA!!!**
We need law enforcement to keep order.	we need law enforcement to keep us safe.	**LAW & ORDER!**
I'm **Donald Trump**.	i'm the best thing that has ever happened to the united states.	I'm **the best thing that has ever happened to the United States**.
Democrats are trying to impeach **President Trump** because they are jealous of his success.	the democrats are trying to impeach me because they are jealous of my success.	The Democrats are trying to impeach **me** because they are jealous of my success!
Campaign finance reform is an important issue that affects the integrity of our democracy	campaign finance reform is a disaster for our country. it's a total disaster. it will destroy our democracy. believe me, it's a disaster.	Campaign Finance Reform is **a TOTAL DISGRACE**. It will destroy our democracy!
F**k you.	you are a disgusting person.	You are a **disgusting, disgusting person**!

Table 2: Some samples of Human Evaluation.

Limitations

Due to time constraints, we leave many thoughts that haven't been achieved. The evaluation is not compared with other state-of-the-art methods, thus we cannot tell the performance of our model relatively.

Acknowledgement

The research work comes from the final project of Deep Learning class by Prof. Jing Zhang. We would like to thank the professor and her assistants for their great effort, and also thank our reviewers (classmates) for their valuable comments.

References

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Sudha Rao and Joel Tetreault. 2018. Dear sir or madam, may i introduce the gyafc dataset: Corpus, benchmarks and metrics for formality style transfer. *arXiv preprint arXiv:1803.06535*.

Emily Reif, Daphne Ippolito, Ann Yuan, Andy Coenen, Chris Callison-Burch, and Jason Wei. 2021. A recipe for arbitrary text style transfer with large language models. *arXiv preprint arXiv:2109.03910*.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena.