Rethinking Text Attribute Transfer: A Lexical Analysis

Yao Fu[1]*, Hao Zhou[2], Jiaze Chen[2], Lei Li[2]
[1]Columbia University
[2]Bytedance AI Lab
yao.fu@columbia.edu, zhouhao.nlp@bytedance.com
teoyde@gmail.com, lileilab@bytedance.com

Abstract

Text attribute transfer is modifying certain linguistic attributes (e.g. sentiment, style, authorship, etc.) of a sentence and transforming them from one type to another. In this paper, we aim to analyze and interpret what is changed during the transfer process. We start from the observation that in many existing models and datasets, certain words within a sentence play important roles in determining the sentence attribute class. These words are referred to as *the Pivot Words*. Based on these pivot words, we propose a lexical analysis framework, *the Pivot Analysis*, to quantitatively analyze the effects of these words in text attribute classification and transfer. We apply this framework to existing datasets and models, and show that: (1) the pivot words are strong features for the classification of sentence attributes; (2) to change the attribute of a sentence, many datasets only requires to change certain pivot words; (3) consequently, many transfer models only perform the lexical-level modification, while leaving higher-level sentence structures unchanged. Our work provides an in-depth understanding of linguistic attribute transfer and further identifies the future requirements and challenges of this task[1].

1 Introduction

The task of text attribute transfer (or text style transfer [2]) is to transform certain linguistic attributes (sentiment, style, authorship, rhetorical devices, etc.) from one type to another (Ficler and Goldberg, 2017; Fu et al., 2018; Hu et al., 2017; Li et al., 2018; Shen et al., 2017). The state-of-the-art (SOTA) models have achieved inspiring transfer success rates (Zhao et al., 2018; Zhang et al., 2018; Prabhumoye et al., 2018; Yang et al., 2018). However, it is still unclear in current literature about what is transferred and what remains to be unchanged during the transfer process. To answer this question, we perform an in-depth investigation of the linguistic attribute transfer datasets and models.

	Before	After
negative to positive	service is trashy rude	service is pretty good
positive to negative	go to place for client visits with gorgeous views	go to place for client visits with terrible views

Figure 1: Examples of pivot words in sentiment transfer. Certain words are strongly correlated with the sentiment such that a transfer model only need to modify these words to accomplish the transfer task while leaving the higher level sentence structure unchanged.

Our investigation starts from a simple observation: in many transfer datasets and models, certain class-related words play very important roles in attribute transfer (Li et al., 2018; Prabhumoye et al., 2018). Figure 1 gives a sentiment transfer example from the controllable generation (CG) model (Hu et al., 2017) on the Yelp dataset. In this example, *rude* is strongly related to the *negative* sentiment and *good* is strongly related to the *positive* sentiment, thus simply substituting *rude* with *good* will transfer the sentence from negative to positive. In this work, We name these words the *pivot words* for a class. We use the term *the pivot effect* to refer the effect that certain strong words may be able to determine the class of a sentence.

Based on the observation of the pivot effect, our research questions are: (1) which words are pivot words and how do they influence the attribute class of a sentence in different datasets? (2) does the model only need to modify the pivot words to per-

[*] Work done when Yao was an intern at Bytedance AI Lab.

[1] Our code can be found at https://github.com/FranxYao/pivot_analysis

[2] Many existing works also call this task style transfer(Fu et al., 2018), our work view style as one of the linguistic attributes, and use the term style or attribute according to the context.

form the attribute transfer or it may change higher-level sentence composationality like syntax?

To answer question (1), we propose the *pivot analysis*, a series of simple yet effective text mining algorithms, to quantitatively examine the pivot effects in different datasets. The basics of the datasets we investigate are listed in Table 1. We first give the algorithm to extract pivot words (Sec 3). We statistically show the stronger the pivot effect is on a dataset, the easier for a model to transfer its sentences. To further analyze the fine-grained distributional structure of these pivot words, we propose the *precision-recall histogram* to show to what extent the datasets may be influenced by their pivot words (Sec 4.2).

To answer question (2) and discover what is changed during the transfer process, we use the pivot words to analyze the transfer results of two SOTA models: the Controllable Generation (CG) model(Hu et al., 2017) and the Cross Alignment (CA) model (Shen et al., 2017). We show that although equipped with sophisticated modeling techniques, in many datasets, these models tend to change only a few words and most of these modified words are pivot words. When we mask out the modified words (to eliminate the lexical changes) and compare the Levenshtein string edit distance (Levenshtein, 1966) of the sentence stems before and after the transfer, we find out many of the sentence stems are the same (the distance of the masked sentences equals to 0). This means that in transfer, the model only modifies few pivot words while leaving the syntactical structure of the sentence unchanged (Sec 5).

To sum up, we show that: (1) in many datasets, words are important features in classification and transfer. But still, certain hard cases require a higher level of understanding of the sentence structures. (2) SOTA models tend to perform the transfer at the lexical level, the syntax of a sentence is generally unchanged. The understanding and modification of higher-level sentence compositionality (syntax trees and dependency graphs) is still a challenging problem.

2 Background

Inspired by the image style transfer task (Gatys et al., 2016; Zhu et al., 2017), the goal of text attribute(style) transfer is to transfer the stylistic attributes of the sentence from one class to another while maintaining the content of the sentence unchanged (Fu et al., 2018; Ficler and Goldberg, 2017; Hu et al., 2017). Because of the lack of parallel datasets, most models focus on the unpaired transfer. Although plenty of sophisticated techniques are used in this task, such as adversarial learning (Zhao et al., 2018; Chen et al., 2018), latent representations (Li and Mandt, 2018; Dai et al., 2019; Liu et al., 2019), and reinforcement learning (Luo et al., 2019; Gong et al., 2019; Xu et al., 2018), there is little discussion about what is changed and what remains unchanged.

Because of the lack of transparency and interpretability, there is some retrospection on this topic. Such as the definition of text style (Tikhonov and Yamshchikov, 2018), and the evaluation metrics (Li et al., 2018; Mir et al., 2019). Our proposed pivot analysis aligns with these works and provides a new tool to probe the transfer datasets and models. The de facto metrics is to use a pretrained classifier to classify if the transferred sentence is in the target class. So our pivot analysis starts from the classification task and mines the words with strong predictive performance.

While many previous works focus on one-to-one transfer, many recent works extend this task to one-to-many transfer (Logeswaran et al., 2018; Liao et al., 2018; Subramanian et al., 2019). For simplicity, we focus on the one-to-one setting. But it is also easy to extend the pivot analysis into one-to-many transfer settings.

3 Pivot Words Discovery

To study the factors influencing attribute transfer, we start from mining words strongly correlated with the attribute class i.e. pivot words. Algorithm 1 shows the procedure of mining pivot words. This algorithm is based on a simple intuition: if one single word is strong enough to determine the sentence attributes, then when we use the existence of this word to classify the attribute, we should achieve very high precision. Consider two extreme examples: when a word only exists in one class, it should achieve 100% classification precision. When a word exists evenly in two different classes, its precision is 50%. The reason we use precision instead of recall or accuracy is that only precision reveals the influence of a single word: suppose the word “awesome” only exists in 100 positive sentences, and the whole dataset size is 100K. In this case, “awesome” will have low recall and accuracy, but high precision. This algorithm

	Yelp	Amazon	Caption	Paper	Gender	Politics	Reddit	Twitter
Source	Hu et al. (2017)	Li et al. (2018)	Li et al. (2018)	Fu et al. (2018)	Prabhumoye et al. (2018)	Prabhumoye et al. (2018)	dos Santos et al. (2018)	dos Santos et al. (2018)
Class	Positive Negative	Positive Negative	Romantic Humorous	Academic Journalism	Male Female	Democratic Republican	Polite Impolite	Polite Impolite
Size(train/ dev/test)	444K/ 63K/126K	554K/ 2K/1K	12K/ -/1K	392K/ 20K/20K	2M/ 4K/534K	537K/ 4K/56K	10M/ 19K/47K	3M/ 18K/18K

Table 1: The text attribute transfer datasets we investigate.

Yelp	**Positive**	**Negative**
pivot words	great, perfection, local, nice, good, thanks, ambiance, incredible, amazing, fair	sadly, kidding, never, sucks, disappointed, terrible, slow, frustrated, overpriced, waste
sentences w. pivots	the owner was super nice and welcoming thanks for always satisfying ! great prices , outstanding food , quick and polite service .	the service sucks , management is terrible . we completely wasted an hour of our time and left . unfortunately we left over 3/4 of our food in the trash .
sentences w/o. pivots	now it 's our weekly treat . first time going today and i got the new york pastrami . i will be back !	do yourself a favor and just stay away . this place is large enough for everything . i am not exaggerating .
Amazon	**Positive**	**Negative**
pivot words	great, easy, attractive, nicely, compact, wonderful, love, fancy, happy, fits	worst, disapointed, horrible, refused, poor, waste, uncomfortable, worse, insult, dissapointed
sentences w. pivots	nice pan ! this pizza pan is a great size it works perfectly ! i m very happy with the price it is easy to use and fits the iphone great	that s what makes this game so damn frustrating and boring incredibly worthless and complex drm that ruins the point the taste was bitter and sharp , not smooth and had a yucky fake taste
sentences w/o. pivots	it really just sits lightly over your ear give it a chance , you won t be disappointed i hastily ordered her another one and it arrived on time	this kettle looks pretty but doesn t turn off once it boils no instructions are included , although it s possible to google on my scale , it weighs over num_num ounces with batteries
Gender	**Female**	**Male**
pivot words	love, smoothie, massage, lattes, yoga, spa, chocolate, pillows, grateful, gorgeous	builds, value, wife, failure, respectable, roaster, competition, vehicle, beers, workmanship
sentences w. pivots	my favorite cupcake is the strawberry shortcake best of luck ,love this hidden gem ! hits the spot every time !	their drink selection is simple with many local brews and mass produced american beers great stadium the best setting for a ballpark anywhere in the us
sentences w/o. pivots	we had a great experience and will go back i strongly recommend this animal hospital for your pets	the managers seemed to be running a nice place the desserts are another shining feature of this place

Figure 2: The pivot words and sentence examples in three example datasets. The vocabulary of pivot words is large so we only list typical words. Sentences without pivot words are intuitively harder to classify and transfer.

calculates the precision for each word-class pair, and choose pivot words with a predefined threshold p_0.

For simplicity, we only consider binary classification in Algorithm 1, but one could easily extend it to multi-class settings. Also, we only consider unigrams(words), while it is also straightforward to extend it to ngrams. In practice, we find the unigram version performs quite good, as is shown in Table 2. As for the parameters in the algorithm, the precision threshold p_0 controls the confidence of a word to be a pivot, and the occurrence threshold f_0 prevents overfitting. We tune these parameters based on the classification performance on the validation set. Specifically, to get better classification performance, f_0 and p_0 should be lower to allow more vote (e.g. $f_0 \leq 10, p_0 \in [0.5, 0.7]$). To get more confidence and filter out stronger pivot

Algorithm 1 Pivot Words Discovery

Input: The vocabulary $\mathcal{V}$, the sentences $\mathcal{S}$ and the labels $\mathcal{Y}$, the frequency threshold f_0, the precision threshold p_0
Output: The pivot words Ω_y for each class $y \in \{0, 1\}$. The word-class precision matrix $p(x, y)$

```
procedure PIVOT WORDS DISCOVERY
    Balance the dataset by down-sampling the majority class.
    for each sentence $s$, each class $y$, and each word $x$ in the vocabulary $\mathcal{V}$ with frequency higher than $f_0$ do
        Consider the class of $s$ is $y$ or $1 - y$
        Use the existence of $x$ to classify:
        if $x$ is in $s$ then
            Classify $s$ to be $y$
        else
            Classify $s$ to be $1 - y$
    Calculate the classification precision $p(x, y)$ of word $x$ for label $y$ over all sentences $\mathcal{S}$.
    if $p(x, y) > p_0$ then
        $x$ is a pivot word for class $y$ i.e. $x \in \Omega_y$
    return $\Omega_y, p(x, y)$
```

Algorithm 2 The Pivot Classifier

```
Input: sentence $s$, the pivot words $\Omega_y$ for class $y \in \{0, 1\}$
Output: The class $y(s)$of sentence $s$
procedure PIVOT CLASSIFICATION
    View $s$ as bag of words
    For each $y \in \{0, 1\}$, calculate $s_y = ||s \cap \Omega_y||$
    Predict the class of $s$ to be $y(s) = \text{argmax}_y\{s_y\}$. Break tie randomly.
    return $y(s)$
```

words, f_0 and p_0 should be higher (e.g. $f_0 \geq 100, p_0 \geq 0.7$).

Figure 2 shows the mined pivot words in different datasets. For sentences that contain pivot words, it is clear that these words are strong features for classification. Intuitively, to transfer the class of these sentences, one could directly modify these words. But there are also cases that contain no pivot words, e.g. *i will be back* in the Yelp dataset. To modify the sentiment of these sentences, a model needs to understand a broader context and common sense. In general, the existence of pivot words gives us a method to understand in attribute transfer, what cases are easier and what cases are more difficult.

The intuition that the existence of single words is enough to determine the linguistic attribute does not necessarily hold on all datasets. But empirically, we find out many transfer datasets tend to contain strong pivot words (Figure 5). One could compare our pivot analysis with other methods that mine the word importance, such as the weights of a logistic classifier, or more sophisticated Bayesian methods like the log-odds ratio informative Dirichlet prior (Monroe et al., 2008). Our method is more straightforward and interpretable. We further develop this method as a simple yet strong classification baseline to indicate the transfer difficulty of different datasets and use the pivot words as a tool to analyze, interpret, and visualize the text attribute transfer models.

4 Analysing Datasets with Pivot Analysis

In this section, we use the pivot words to analyze the transfer datasets. We first reveal the mechanisms of how pivot words affect classification and transfer by using the pivot words as the classification boundary. Then we use the precision-recall histogram to demonstrate the distributional structure of the pivot words in different portions of the datasets.

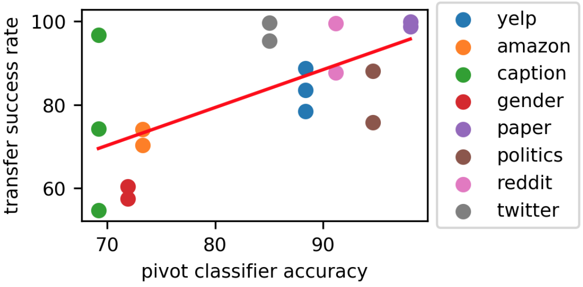

Figure 3: Pivot classification accuracy v.s. transfer success rate (correlation = 0.64, p-value = 0.003). The stronger the pivot effect is, the easier to transfer.

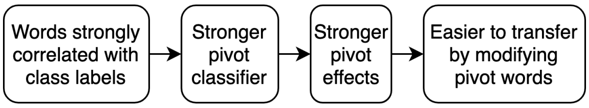
Words strongly correlated with class labels
Stronger pivot classifier
Stronger pivot effects
Easier to transfer by modifying pivot words

Figure 4: The mechanism of the pivot effect on classification and transfer.

4.1 The Pivot Classifier

Algorithm 2 gives a simple method to classify a sentence based on the pivot words output from Algorithm 1. This is essentially a voting based classifier. This classifier holds strong independence assumption that the label of a sentence is only related to the bag of words, but ignore the word orders. This is to say, the decision boundary only stays at the lexical level, and does not go to the syntax level. Then it counts the pivot words of different classes contained by the sentence and predicts the label to be one of the largest pivot words overlap. Intuitively, this algorithm classifies a sentence only based on the existence of strong attribute-related words.

The pivot classifier is a simple yet strong classification baseline, as is shown in Table 2. We use it to study different datasets and compare it with (1) a logistic classifier, (2) a SOTA CNN classifier (Kim, 2014). We have balanced the test sets so the random baseline is 50%. This voting based classifier achieves comparable performance with the two models in 4 datasets (Amazon, Gender, Paper, Politics), and only loses small margins in 2 datasets (Yelp, Caption). Although the independence assumption from our pivot classifier does not necessarily hold for all datasets, empirically it performs very well. This means that these pivot words are a meaningful approximation of the true decision boundary.

If the decision boundary of a linguistic attribute stays at the lexical level, then one could cross the

Validation	Yelp	Amazon	Caption	Gender	Paper	Politics	Reddit	Twitter
Pivot	88.00	75.85	-	72.02	97.82	98.32	90.00	85.25
Logistic	91.83	76.75	-	72.77	98.39	99.82	98.05	98.20
CNN	92.87	77.93	-	74.20	98.36	98.85	99.45	99.55
Test	Yelp	Amazon	Caption	Gender	Paper	Politics	Reddit	Twitter
Pivot	88.35	73.30	69.20	71.91	98.07	94.60	91.15	85.05
Logistic	91.97	73.80	75.20	72.91	98.67	96.83	98.05	98.05
CNN	92.96	75.80	76.10	74.29	98.66	87.91	99.65	99.45

Table 2: Classification accuracy. The voting based pivot classifier is a strong classification baseline compared with the state of art CNN classifier, indicating that in many datasets, words are strong features for class labels.

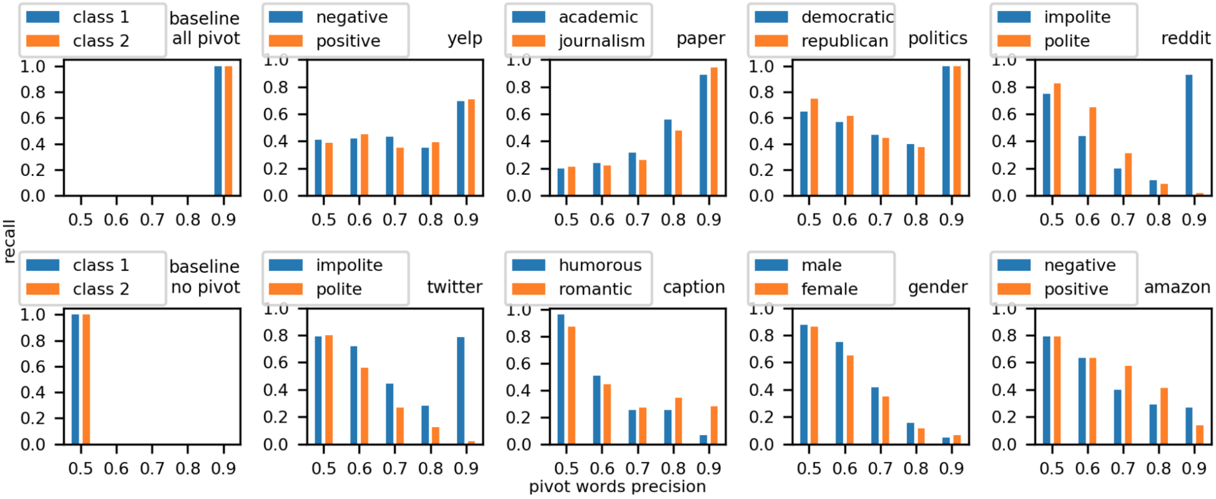

Figure 5: The precision-recall histogram. The high right bars in Yelp, Paper, Politics, Reddit, and Twitter datasets reveal the existence of strong pivot words, Each bar at location (x, y) should be interpreted as: if use pivot words with precision x to classify the sentence, the recall will be y. The higher the right bars are, the more sentences can be classified by words accurately, the stronger the pivot effect is, the easier to transfer. The baseline cases where the dataset is full of/ has no pivot words are show on the left.

boundary by simply substituting the pivot words of one class to another, thus achieving text class transfer. Intuitively, the more pivot words a dataset contains, the stronger the pivot effect is, the easier for the pivot classifier to classify, and the easier to transfer the attribute. This intuition is demonstrated in Figure 3. The pivot effect (shown by pivot classification accuracy) and the transfer difficulty (shown by the transfer success rate reported from previous models) has a strong positive correlation and is statistically significant. This mechanism is demonstrated in Figure 4. The stronger the pivot effect is, the easier to transfer.

4.2 The Precision-Recall Histogram

Now we go one step further to reveal how the pivot effect distributes in different portions of the datasets. We propose a new tool, *the precision-recall* histogram based on the results from Algorithm 1 and 2. As is shown in Algorithm 3, essentially, this algorithm use pivot words with different level of confidence (precision) to classify the dataset, and output the recall. For better visualization, we set the precision interval gap to be 0.1, but it is also possible to use smaller or larger gaps. It is also important to balance the dataset in Algorithm 1 to make the baseline precision 0.5.

The histogram for all datasets gives a fine-grained illustration of the pivot effect (Figure 5). We first look at the two baseline cases: a dataset with no pivot words, and a dataset full of pivots. If a dataset is full of pivots, i.e. the vocabulary of the two classes have no overlap, then all words should have precision 1.0 and they should achieve 1.0 recall, so the right-most bars are the highest. If a dataset has no pivot words, i.e. all words are distributed evenly in two classes, then all words have precision 0.5 and they should achieve 1.0 recall, so the left-most bars are the highest. The higher the right bars are, the stronger the pivot effect is.

The histograms of the datasets are somewhere between the two baseline cases. Generally, we

Algorithm 3 The Precision-Recall Histogram

Input: The sentences $\mathcal{S}$, the labels $\mathcal{Y}$, the pivot words for each class $\Omega_y, y \in \mathcal{Y}$, the precision matrix $p(x, y), x \in \mathcal{V}, y \in \mathcal{Y}$

Output: The precision-recall histogram

```
procedure THE PRECISION-RECALL HISTOGRAM
    for The precision range pair $(p_i, p_{i+1}) \in [(0.5, 0.6), (0.6, 0.7)...(0.9, 1.0)]$ do
        For each class $y$, gather all pivot words of the precision in the given range: $\Omega_y^{(i)} = x : p(x, y) \in [p_i, p_{i+1}]$
        Use $\Omega_y^{(i)}$ to form a pivot classifier and classify the dataset $\mathcal{S}$. Calculate the recall $r_i$.
        Store $(p_i, r_i)$
    return The list of $(p_i, r_i)$
```

see two different shape distributions. In the Yelp, Paper, Politics, Reddit, and Twitter datasets, the right-most bars are the highest, meaning that in these datasets, strong pivot words exist in a large portion of the dataset. These are close to the all-pivot baseline. Specially, we see that in the Reddit and Twitter dataset, the pivot effect only exists in the *impolite* class, while in other datasets, the pivot effect exists in both classes. Note that this phenomenon cannot be discovered simply from the overall classification accuracy. After manual inspection, we find out since the attribute of these two datasets is politeness, the pivot words for the impolite class are the common swearwords in English. These words dominate the impolite sentences.

In the Caption, Gender, and Amazon dataset, we see a decreasing height from left to right, indicating a weaker pivot effect. Highest bars exist in the 0.5 precision bars, meaning that for each class, most of them can be classified by 0.5 precision (= random guessing). This is close to the no-pivot baseline. The high-precision words still exist, but they cannot dominate the whole class. In conclusion, the precision-recall histograms give a structural examination for each class. The existence of pivots and the determination power of pivots differ from class to class, and from datasets to datasets.

5 Analysing Transfer Models with Pivot Analysis

In this section, we aim to analyze what is changed and what remains in linguistic attribute transfer systems. We perform our experiments from two perspectives: the lexical structures, and the syntactical structures. For the lexical structures, we show what words are modified by the transfer model. For the syntactical structures, we mask out the modified pivot words and compare the resulting sentence stems.

	Yelp	Amazon	Gender
CG - # modified	1.66	0.56	0.79
- percentage	18%	4%	5%
CA - # modified	1.61	3.54	5.60
- percentage	18%	23%	33%
sentence length	8.89	14.82	17.01

Table 3: Average number of modified words and their percentage in the sentence length. The transfer models tend to modify only a few attribute-related words.

	Yelp	Amazon	Gender
CG	91.25	94.77	94.17
CA	72.33	74.04	56.09

Table 4: Percentage of modified words that are pivot words. A large portion of the modified words are pivots.

We use the two most common SOTA models, the Controllable Generation (CG) model from Hu et al. (2017), and the Cross Aligned Autoencoder (CA) model from Shen et al. (2017). The CG model uses a conditional VAE with style-discriminator and trained with a wake-sleep algorithm. The CA model uses a cross-alignment mechanism to guide the transfer process. These are two strong models in many datasets compared to many other models. We direct the readers to the original papers for more details.

We test the models on three datasets: Yelp, Amazon, and Gender. The Yelp dataset is the most widely used benchmark in the text style transfer task. As is shown in the previous sections, it exists strong pivot effects. There are many sentiment words in this dataset. For the Amazon and the Gender dataset, there is less pivot effect. So our experiments give a minimum cover of different types of datasets. We use the released implementation for our experiments [3]. All hyper-parameters are followed by their official instructions. Both models are trained until the simultaneous convergence of the reconstruction loss and the adversarial loss. We refer the readers to the implementation repositories for more details.

[3] The CG model: https://github.com/asyml/texar/tree/master/examples/text_style_transfer
the CA model: https://github.com/shentianxiao/language-style-transfer

CG	0	1	2	3	4	5	6	>6
Yelp	74.65	5.05	10.68	5.71	1.84	0.72	0.66	0.69
Amazon	94.20	0.00	0.90	4.00	0.80	0.10	0.00	0.00
Gender	90.96	0.04	6.60	0.45	0.89	0.43	0.16	0.48
CA	0	1	2	3	4	5	6	>6
Yelp	41.30	1.98	13.63	10.01	8.76	7.49	6.17	10.66
Amazon	37.60	1.85	9.95	9.25	6.15	6.15	4.65	24.40
Gender	37.89	0.27	2.27	3.36	1.60	1.40	2.03	51.18

Table 5: Masked edit distance percentage distributions. For the CG model, in most of the cases($> 74\%$), the masked edit distance is 0, meaning that only few words are changed while the sentence structures are exactly the same. For the CA model, still a large portion of the sentence structures are unchanged ($> 37\%$)

Yelp	
Before	After
but not worth _num_ bucks .	but consistently worth incredible bucks
i 'm the wrong person to ask .	i 'm adds fantastic person to ask .
food is acceptable , service is terrible	food is acceptable , service is nice .
Amazon	
Before	After
this product is a ripoff on consumers	this product is a reasonable on consumers
this is a worthless item	this is a sturdy item
for me this was a disappointing product	for me this was a great product .
Gender	
Before	After
my boyfriend and i chose this restaurant	my buddy and i chose this restaurant
really do love the animals	really do love the milkshakes
you get to wait in a recreation room	you get to wait in a bathing room

Figure 6: The transfer cases. Many of the transfered words are pivot words. The model tend to transfer only a few words while leaving the higher level sentence structure unchanged.

	Before	After
original	service is trashy rude	service is pretty good
masked	service is [mask] [mask]	service is [mask] [mask]

Figure 7: An example of the masked sentences. Edit distance = 0 after masking.

	Yelp	Amazon	Gender
CG - distance	0.65	0.17	0.26
- percentage	7.3%	1.4%	1.5%
CA - distance	2.63	4.56	9.95
- percentage	29%	31%	58%
sentence length	8.89	14.82	17.01

Table 6: Edit distance after masking out the pivot words. In the CG model, only words are modified, while the higher-level sentence structures remain to be the same. For the CA model, it tries to modify more sentence structures.

5.1 Lexical Structures

We show that the two models tend to modify only a few words in a given sentence, and a large portion of these words are pivot words. The results are shown in Table 3 and 4. On the Yelp dataset, the CG model and the CA model only modify 1.66 and 1.61 words on average. The portion of pivot words is 91% and 72% respectively. This means on this dataset, both two models focus on word substitutions to change the sentence style. On the Amazon and the Gender dataset, the models take different transfer strategies. For the CG model, it concentrates on fewer words to modify (0.56 on Amazon and 0.79 on Gender). For the CA model, it tends to modify more words (3.54 on Amazon and 5.60 on Gender). Still, both models tend to modify the pivot words for class transfer. In general, a small portion of the sentences are modified ($< 30\%$ approximately), and a large portion of the modified words are pivots ($> 60\%$ approximately).

5.2 Syntactic Structures

If we eliminate the lexical differences by masking out the modified words, what is changed in the resulting sentence stems? We use the Levenshtein string edit distance (Levenshtein, 1966) to measure the distances of the masked sentences as an approximation to the distances of syntactic structures. Figure 7 gives an example of masked sentences. One could also consider more sophisticated metrics to measure the syntactic distances with parsing trees (Shen et al., 2018; Zhang and Shasha, 1989). Here we use the string edit distance for simplicity. In practice, it is informative enough to demonstrate the change of sentence structures.

Table 6 shows the edit distances after masking the pivot words. We see clear differences between the two models. For the CG model, it barely changes the sentence structures (0.1+ distances). This indicates that it takes the strategy to focus more on the substitution of pivot words. For the CA model, it takes the strategy that not only to modify the words, but also a portion of the sentence structures. We see a moderate percentage of the sentence structure modified on the Yelp and Amazon dataset (about 30%), and a large syntactic modification (58%) on the Gender dataset. Compared with the CG model, the CA model tries to modify the sentences more radically.

To show a fine-grained distribution of the distances among different cases, we list the distribution statistics in Table 5. We see that for the CG model, most of the cases $> 74\%$) sentence stems are unchanged. For the CA model, although its average edit distance is larger, in a large portion of the cases ($> 37\%$), the distance is still 0. In conclusion, both models tend to retain the sentence structures in a large portion of the datasets.

5.3 Qualitative Analysis

Now we examine the transfer cases qualitatively in Figure 6. These are cases from the CG model on the three datasets. The pivot words are highlighted. When the model tries to change the class of a sentence, it first identifies the pivot words, then substitutes them with the pivots from another class. If we mask out the highlighted pivot words, the resulting sentence stems are the same, indicating that the syntactic structures remain unchanged. Although this is not all the case, the models tend to focus on words in a large portion of the datasets.

6 Discussion

Implications: Our pivot classifier reveals that to a certain extent, in many transfer datasets, the decision boundary stays at the lexical level. Consequently, to cross the boundary and transfer the text class, many instances in the dataset only requires to modify certain pivot words. But still, there are cases with no pivot words. The decision boundary in such cases is higher than the word level. To transfer these cases, the model needs a deeper understanding of the sentence structures, which may include syntax, semantics, and common sense (Figure 2).

Considerations: In our experiments, we find out the two models are both quite unstable during training. The balance between the reconstruction loss and the adversarial loss will significantly influence the convergence point. Our pivot analysis framework requires the model to converge to a meaningful local optimum with reasonable content preservation and transfer strength at the same time(Fu et al., 2018). For our pivot algorithms, it is important to balance the datasets (both training and testing) for a reasonable precision baseline(0.5). Our algorithm is mostly sensitive to the precision threshold p_0 i.e. the confidence of how *pivot* a word is. We tune this parameter based on the development set performance.

Limitations: All of our pivot algorithms stays at *the lexical level*. These algorithms hold strong Independence assumption that the class of a sentence is independent of the order of words. So this method may not be able to capture certain linguistical phenomenons, such as anastrophe [4]. One could also consider an extreme example where the pivot analysis *does not work*: suppose we have a corpus of sentences, we label all of them to be 0, then we *reverse all sentences*, and label the reversed sentences to be 1. In this dataset, both classes share the same vocabulary, and the precision of any word will be 0.5. This is an example where only the order determines the class. Further, in our work, we only consider lexical changes, and do not consider other issues with regard to more rigorous definition of linguistic style(Tikhonov and Yamshchikov, 2018), the evaluation metrics (Mir et al., 2019), and the causality in text classification(Wood-Doughty et al., 2018). These topics will be the future directions.

7 Conclusion

In this work, we present *the Pivot Analysis*, a lexical analysis framework for the examination and inspection of text style transfer datasets and models. This analysis framework consists of three text mining algorithms, *pivot words discovery, the pivot classifier, and the precision-recall histograms*. With these algorithms, we reveal what are the important words that influence the class

[4] To change the order of certain words

of a sentence, how these words are distributed in a dataset, the mechanisms through which these words interact with a transfer model, and how the models perform the transfer. Our method serves as a probe for the transparency and the interpretability of the datasets and the transfer models. We show that a large portion of the transfer cases stays at the lexical level, while the syntactic structures are unchanged.

Since our methods stay at the lexical level, it has its own limitations in understanding higher-level sentence compositionality. These limitations are also shared by the SOTA transfer models: to understand the syntax and semantics (i.e. the structures of the sentence), and the common sense (i.e. the background and implications of the surface words). These limitations are also directions for future challenges. In the future, we need to use better inductive bias and use more powerful models towards higher-level sentence compositionality.

Acknowledgments

We thank the reviewers for their informative reviews. We thank Yansong Feng, Bingfeng Luo, and Zhenxin Fu for the helpful discussions. This work is supported by the China Scholarship Council.

References

Liqun Chen, Shuyang Dai, Chenyang Tao, Haichao Zhang, Zhe Gan, Dinghan Shen, Yizhe Zhang, Guoyin Wang, Ruiyi Zhang, and Lawrence Carin. 2018. Adversarial text generation via feature-mover's distance. In *NeurIPS*.

Ning Dai, Jianze Liang, Xipeng Qiu, and Xuanjing Huang. 2019. Style transformer: Unpaired text style transfer without disentangled latent representation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5997–6007, Florence, Italy. Association for Computational Linguistics.

Jessica Ficler and Yoav Goldberg. 2017. Controlling linguistic style aspects in neural language generation. In *Proceedings of the Workshop on Stylistic Variation*, pages 94–104, Copenhagen, Denmark. Association for Computational Linguistics.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style transfer in text: Exploration and evaluation. In *AAAI*.

Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. 2016. Image style transfer using convolutional neural networks. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2414–2423.

Hongyu Gong, Suma Bhat, Lingfei Wu, Jinjun Xiong, and Wen mei W. Hwu. 2019. Reinforcement learning based text style transfer without parallel training corpus. In *NAACL-HLT*.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2017. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1587–1596, International Convention Centre, Sydney, Australia. PMLR.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *EMNLP*.

Vladimir I Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet physics doklady*, 10(8):707–710.

Juncen Li, Robin Jia, Hua He, and Percy S. Liang. 2018. Delete, retrieve, generate: A simple approach to sentiment and style transfer. In *NAACL-HLT*.

Yingzhen Li and Stephan Mandt. 2018. Disentangled sequential autoencoder. In *ICML*.

Yi Liao, Lidong Bing, Piji Li, Shuming Shi, Wai Lam, and Tong Zhang. 2018. Quase: Sequence editing under quantifiable guidance. In *EMNLP*.

Dayiheng Liu, Jie Fu, Yidan Zhang, Chris Pal, and Jiancheng Lv. 2019. Revision in continuous space: Fine-grained control of text style transfer. *ArXiv*, abs/1905.12304.

Lajanugen Logeswaran, Honglak Lee, and Samy Bengio. 2018. Content preserving text generation with attribute controls. In *Advances in Neural Information Processing Systems*, pages 5103–5113.

Fuli Luo, Peng Li, Jie Zhou, Pengcheng Yang, Baobao Chang, Zhifang Sui, and Xu Rui Sun. 2019. A dual reinforcement learning framework for unsupervised text style transfer. In *IJCAI*.

Remi Mir, Bjarke Felbo, Nick Obradovich, and Iyad Rahwan. 2019. Evaluating style transfer for text. In *NAACL-HLT*.

Burt L. Monroe, Michael Colaresi, and Kevin M. Quinn. 2008. Fightin' words: Lexical feature selection and evaluation for identifying the content of political conflict. *Political Analysis*, 16(4):372–403.

Shrimai Prabhumoye, Yulia Tsvetkov, Ruslan R. Salakhutdinov, and Alan W. Black. 2018. Style transfer through back-translation. In *ACL*.

Cícero Nogueira dos Santos, Igor Melnyk, and Inkit Padhi. 2018. Fighting offensive language on social media with unsupervised text style transfer. In *ACL*.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi S. Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *NIPS*.

Yikang Shen, Zhouhan Lin, Athul Paul Jacob, Alessandro Sordoni, Aaron Courville, and Yoshua Bengio. 2018. Straight to the tree: Constituency parsing with neural syntactic distance. In *ACL*.

Sandeep Subramanian, Guillaume Lample, Eric Michael Smith, Ludovic Denoyer, Marc'Aurelio Ranzato, and Y-Lan Boureau. 2019. Multiple-attribute text style transfer. In *ICLR*.

Alexey Tikhonov and Ivan P. Yamshchikov. 2018. What is wrong with style transfer for texts? *ArXiv*, abs/1808.04365.

Zach Wood-Doughty, Ilya Shpitser, and Mark Dredze. 2018. Challenges of using text classifiers for causal inference. *arXiv preprint arXiv:1810.00956*.

Jingjing Xu, Xu Sun, Qi Zeng, Xuancheng Ren, Xiaodong Zhang, Houfeng Wang, and Wenjie Li. 2018. Unpaired sentiment-to-sentiment translation: A cycled reinforcement learning approach. In *ACL*.

Zichao Yang, Zhiting Hu, Chris Dyer, Eric P. Xing, and Taylor Berg-Kirkpatrick. 2018. Unsupervised text style transfer using language models as discriminators. In *NeurIPS*.

Kaizhong Zhang and Dennis Shasha. 1989. Simple fast algorithms for the editing distance between trees and related problems. *SIAM journal on computing*, 18(6):1245–1262.

Ye Zhang, Nan Ding, and Radu Soricut. 2018. Shaped: Shared-private encoder-decoder for text style adaptation. In *NAACL-HLT*.

Junbo Jake Zhao, Yoon Kim, Kelly Zhang, Alexander M. Rush, and Yann LeCun. 2018. Adversarially regularized autoencoders. In *ICML*.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2242–2251.