GAN-based Unsupervised Clickbait Style Transfer

Mehul Agarwal
Department of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
mehula@andrew.cmu.edu

Sayani Kundu
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213
sayanik@andrew.cmu.edu

Abstract

Style transfer involves taking content in a certain domain and transforming it into another related domain. With the advent of Generative Adversarial Networks (GANs) and models such as CycleGAN, this area has seen a lot of progress in image/video style transfer. Recently, CycleGAN techniques have also permeated text-based domains. In our project, we aim to use CycleGAN based models,in particular StyleTransformer, to convert between factual news titles (non-clickbait) and viral hyperbolic and sensational headlines (clickbait). However textual domains have their own set of challenges and heavily depend on the dataset and task at hand. We incorporate semantic and syntactic features of clickbaity text into the existing model to aid in transforming the style of a text while preserving legibility and meaning.

1 Introduction

The internet today is oversaturated with information all competing for more attention. This phenomenon has led to the rise of "clickbait": use of hyperbolic language to bait users to click and engage with otherwise bland content. Examples of clickbait include Buzzfeed article titles such as "30 Products I Don't Understand How You've Lived Your Whole Life Without" or "Yikes, Yikes, Yikes, These 15 People Had A Bad Week." For our project, we therefore chose the generation of clickbait titles from non-clickbait titles and vice versa. Our Clickbait dataset[1] consists of non-clickbait titles from community-verified WikiNews article headlines, and clickbait titles from popular domains such as 'BuzzFeed', 'Upworthy' and 'ViralStories'. We aim to do this via a Cycle-GAN based style transfer model. CycleGAN based models add a unique cycle consistency loss to a normal Generator-Discriminator model of a traditional GAN. Recently, there have been advances in text style transfer such as positive-negative style transfer that can make our goal possible.

2 Background

2.1 CycleGAN

CycleGAN[2] was one of the pioneer works in unsupervised image to image translation. Given two datasets X and Y, CycleGAN would use GAN (Generative Adversarial Network) based methods to transform an image from domain X to domain Y. They did this by training a mapping $G : X \rightarrow Y$ adversarially (using the standard Generator-Discriminator co-training in GANs) such that any image $G(x_i)$ is indistinguishable from any $y_i \in Y$. They then trained a mapping $F : Y \rightarrow X$ adversarially such that any image $F(y_i)$ is indistinguishable from any $x_i \in X$. Finally, they introduced a novel cycle consistency loss such that $F(G(x_i)) \approx x_i \in X$ (and vice-versa). This technique has had wide success in the field of computer vision and the ideas of cycle consistency have been adapted to other

10617 Class Project (2021).

domains such as text. The models we discuss in Methods in fact use some sort of cycle consistency to model their style transfer.

2.2 Style Transfer in Text domains

Even before the advent of CycleGAN, there has been a lot of work in the field of style transfer in text. There is a GitHub repository that lists a couple of papers in this field. However, owing to the tough nature of the task, most successful papers deal with parallel data (supervised). There hasn't been much progress in non parallel data (unsupervised) domain-independent text style transfer. However, there have been varying rates of success in specific domains using non parallel data.

There are a few papers on formal to informal style transfer such as Formality Style Transfer with Hybrid Textual Annotations[3]. This includes positive to negative sentiment transfer in text such as Style Transformer[4] and Dual-Generator Network for Text Style Transfer (DGST)[5].

2.3 Clickbait Detection Dataset

Using ML-based methods for identifying clickbait isn't an unexplored domain. The paper that we get our dataset from use it to classify clickbait and non-clickbait data[1].This paper boasts a 93% accuracy in detecting clickbait. Hence there is an inherent difference between clickbaity and non-clickbaity headlines. However, as far as we know, there hasn't been much work in converting between the two. We aim to use the text style transfer models we introduce above to help in our domain of clickbait to non-clickbait text style transfer.

2.4 DGST

The DGST[5] is a rather unique architecture because it does not have a discriminator. It is able to transfer the style of text with the help of two generators. Each generator or transferrer is a BiLSTM network and converts a text from one style to the other(Transferrer 1 - Style A to Style B, Transferrer 2 - Style B to Style A). The model is trained in a similar manner as cycleGAN[2] by transferring to another style and again back-transferring to the original style. In order to transfer a sentence to a target style while preserving the style-independent information, they have formulated two sets of training objectives: one set ensures that the generated sentences is preserved as much as possible and the other set transfers the input text to the target style.

2.5 StyleTransformer

A transformer consists of an encoder followed by a decoder. Given an input sentence embedding and the style embedding to which we want to transfer the sentence, a style transformer[4] learns how to 1) convert the sentence into the same style as its original style (sentence reconstruction) by passing through the transformer 2) convert the sentence into a different style and then convert the new sentence back to the original style by passing through the transformer twice 3) Have a separate encoder(discriminator) that tries to distinguish between the original sentence, output of 1 and the intermediate output of 2. by minimizing the loss for these three steps the style transformer is able to learn the different sets of text based on different styles.

2.6 Preliminary Results

Our main goal is to enable style transfer for the Clickbait dataset. We run the abovementioned models on this dataset and in order to compare how the model performs. We wanted to see how the models performed on our dataset when it was originally trained and performed well on the Yelp dataset for sentiment transfer.

The Clickbait dataset contains approximately 16000 headlines of each type(clickbait and non-clickbait). We use 1000 headlines for either class as a test set and the rest to learn the model. Table 1 shows the self-Bleu scores(to show content preservation) when trained on each of the models. From our qualitative results we observed that the models were not able to show any significant changes in our data. In the rest of the report we discuss the modifications we introduced in the Transformer model that could potentially help transform our data better and a thorough analysis of the results.

Table 1: Clickbait dataset performance on StyleTransformer and DGST

Model	Epochs run	self-BLEU
StyleTransformer	475	38.44
DGST	150	4.96

3 Related Work

We now discuss the two works that helped us introduce modifications to the original model.

3.1 Clickbait Classification

As mentioned in the previous section, the dataset that we have used for our task was introduced by the paper "Stop Clickbait: Detecting and Preventing Clickbaits in Online News Media" [1] for a classification task. The authors have given meaningful insights about the key contrasts between clickbait and non-clickbait data by doing extensive data analysis. By looking at the sentence structure and linguistic details they conclude that there are some prominent differences between the texts of the two classes. Leveraging this, a set of features were chosen that would help in classifying such texts. The texts were first parsed using Stanford's CoreNLP tool[6] to obtain annotations for POS-tags and the dependency tree.

With these the authors defined 14 features including length of tokens (longer tokens on an average in non-clickbait headlines), presence of word contractions, hyperboles and determiners(significantly greater presence in clickbait headlines), and n-gram based features extracted from Parts of speech and dependencies in the text (more proper nouns in non-clickbait texts, more adverbs, personal and possessive pronouns in clickbait texts). These features were shown to successfully classify the dataset giving an accuracy of 93%. Motivated by this, we tried incorporating them into our style-transfer model to see if they can indeed help in transformation as well.

3.2 Grammar Error Correction

Grammar Error Correction (GEC) can be thought of as another style transfer from the domain of text with bad grammar to good grammar. This includes fixing punctuation, spelling, word choice and other grammatical errors. For example, `"I've like turtles"` corrects to `"I like turtles"`

The use cases for GEC range from bad grammar highlighting in messaging platforms (such as WhatsApp, Telegram, Messenger, etc.) to assisted writing from humans (such as Grammarly). Though it is a generative model itself, it's recently being used to post-process machine generated text to ensure that the final output is human-understandable.

There have been significant improvements in GEC models since the past fifteen years. Recently, with advances in Transfer Learning, the state of the art GEC is achieved with transformers. Many current GEC models use T5: the Text-To-Text Transfer Transformer by Google[7] that has great accuracy with a $F_{0.5}$ score of 68.87.

4 Methods

In our background work, we had explored two models DGST and Style-Transformer to see some preliminary results on how they perform with the Clickbait dataset. Since Style-Transformer is the state-of-the-art for style transfer and contains a discriminator unlike DGST, we implement our ideas and work with the former in our final experiments and analysis.

4.1 Clickbait Features in Discriminator

The features introduced by the authors of the Clickbait dataset have helped in distinguishing between the two classes very well in their original work. Since the current style transformer was not too successful in the clickbait style transfer, we wanted to see if training a more complex discriminator that uses the clickbait feature would help the generator produce better transformations. With this

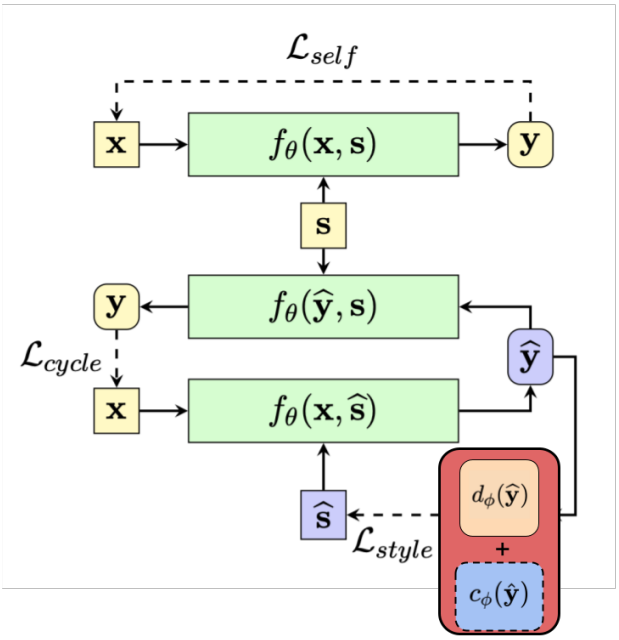

Figure 1: Training process of Clickbait Style Transformer by adding the new clickbait classifier in the Discriminator

idea in mind, we propose a modified Clickbait Style-Transofrmer with two parallel classifiers being trained in the Discriminator of the model, as shown in Figure 1. The first one, d_ϕ, is the original 3-class classifier which predicts if a class is clickbait, non-clickbait or fake (i.e. produced by the generator). The second one, c_ϕ, introduced by us, is a linear classifier that converts the given sentence into 14 features describing the linguistic and syntactic structure of the sentence. It then trains the classifier to predict if it is a clickbait or a non-clickbait sentence.

The process of combining these two classifiers into a single softmax output from the discriminator is not trivial. This mainly because the number of output classes is different from the two. To combine their outputs, we have introduced a hyperparameter, λ, which is the clickbait weightage of classifier c_ϕ. For the fake class prediction, there is no contribution from our clickbait classifier, hence we directly take the logits or hidden layer output from d_ϕ. For the other 2 classes, we take a weighted sum over the logits from the two classifers and pass those to the softmax layer. Mathematically, let $[h_{11}, h_{12}, h_{13}]^T$ be the output from d_ϕ and $[h_{21}, h_{22}]^T$ be the output from c_ϕ. Considering h_{13} is the prediction node for the text being fake, our final output probabilities from the Discriminator can be given by,

$$output = softmax([(1-\lambda)h_{11} + \lambda h_{21}, (1-\lambda)h_{12} + \lambda h_{22}, h_{13}]) \tag{1}$$

Intuitively, we do not want the new classifier to affect the discriminator's output for the fake prediction, but we want the model to have some extra information when it is training the generator for the same-style transformation, which can be provided by the structural information by c_ϕ.

4.2 Pre-trained Grammar Error Correction

For our clickbait style transfer, since the style transformer model is unsupervised, the outputs from non-clickbait headlines to clickbaity headlines and vice-versa in our baselines often included bad grammar and transformed outputs often didn't make sense. Hence, we decided to pass all style transformed outputs (clickbait to non-clickbait and non-clickbait to clickbait) through a pre-trained GEC model.

For this, we used Gramformer[1] that uses Google's T-5 Transformer to generate correct grammar. We chose Gramformer since it is recent, has its own Python package and is available on Hugging Face's model distribution network. After importing gramformer and its dependencies, at each `eval_step` (called every couple of epochs), the transformed outputs were passed to gramformer's `correct` function. Gramformer has a built in quality estimator (QE), and its `correct` function generates top N candidates, scores, ranks and returns the top ranked result.

We output this as well as the transformed text and compare their respective self-bleu and accuracy metrics in the results.

5 Results

In text generation tasks, it is not straightforward to evaluate how well a model is performing. One of the most popular metrics used is the BLEU score [8]. It calculates the quality of the text being generated and helps us understand how much of the content is being preserved after the transformation. Additionally, in order to verify if the style is being transformed, we calculate the accuracy of our outputs being predicted as the transferred style. For this we train a supervised classifier using fastText [9].

For our experiments we have trained the original Style Transformer (ST) and our Clickbait Style Transformer(CST) to see if we get any significant improvement in the generation. We have also used the GEC model at the end for better refinement of the sentences. The results are shown in Table 2.

Since the CST model has to extract structural features of every intermediate text being generated during training, it takes much longer to train. For our experiments we have given equal weightage to both the discriminator classifiers and kept λ as 0.5. We observe that the accuracy of CST is lesser than the original model for the same epoch. The lower accuracy indicates that the new model takes longer to converge. From observations we see that at higher epochs CST starts transforming the texts to an extent to which it can be classified as the other class, but at the expense of losing content and quality of the texts. BLEU scores seem to remain similar to ST even at later epochs from our observations. This indicates that the meaningfulness of the sentence is preserved equally for both the models.

We have demonstrated some qualitative results generated by the models. From these results we can see that the model focuses more on changing words in the sentence to make it sound more like texts from the opposite class. For example in the last output, CST turns the sentence into a question asking who defeated Australia in the World Cup. Though grammatically incorrect, it does make it "clickbait"-y. Detailed analysis of the results have been done in the next section.

Table 2: Clickbait dataset performance on StyleTransformer and Clickbait-Style-Tarnsformer after 500 iterations

Model	Accuracy	self-BLEU
StyleTransformer[ST]	79.95	37.54
StyleTransformer with GEC	62.05	39.19
Clickbait StyleTransformer[CST]	73.15	40.13
Clickbait StyleTransformer with GEC	53.3	41.43

6 Discussion and Analysis

6.1 Effectiveness of the Model

The transformations reflect that our model learnt some important distinctions between our domains of clickbait and non-clickbait data.

There seems to be more than a surface level understanding of what constitutes clickbait and non-clickbait. For example, the input `"Wikinews interviews Australian basketball player Tina"` gets converted to

[1] `https://github.com/PrithivirajDamodaran/Gramformer#usecases-for-gramformer`

Table 3: Output examples for Clickbait from the Two models

Model		Clickbait to Non-clickbait
	Input	People Try More Bizarre Food Combinations That Oddly Work
ST	Output	Court More Bizarre Food Combinations Cash Oddly Work
	Output with GEC	More Bizarre Food Combinations Cash Oddly Work.
CST	Output	Ceremony Try More Bizarre Food Combinations That Oddly Work
	Output with GEC	Try More Bizarre Food Combinations That Oddly Work.
	Input	We Need To Talk About How Cute Brown Is
ST	Output	In Need in Talk to How Cute Brown Is
	Output with GEC	Talk in Need about How Cute Brown Is?
CST	Output	Founder as and Talk About How Cute Brown Is
	Output with GEC	Founder and Talk About How Cute Brown Is?
		Non-clickbait to Clickbait
	Input	Chinese Makes a Name for Herself on L.P.G.A. Tour
ST	Output	Chinese Makes Should Name For Herself You L.P.G.A. Tour
	Output with GEC	Chinese Should Name Me For Herself L.P.G.A. Tour
CST	Output	BuzzFeed Makes For Name Actually Herself Or L.P.G.A. Tour
	Output with GEC	BuzzFeed Makes For Name Actually Herself Or L.P.G.A. Tour
	Input	England defeats Australia and wins 2010 Twenty20 Cricket World Cup
ST	Output	England Ode In Are Ultimate 2010 Twenty20 Cricket World Cup
	Output with GEC	England Ode In Are Ultimate 2010 Twenty20 Cricket World Cup
CST	Output	Which defeats Australia To Will 2010 Twenty20 Cricket World Cup
	Output with GEC	Which will defeat Australia In the 2010 Twenty20 Cricket World Cup?

`"BuzzFeed Crossword: Be basketball player Tina"`, which is great, since in real life, Wikinews is a to-the-point news source compared to Buzzfeed, which is a site known to have internet lingo and lot of clickbait.

This is true for other terms such as `Chief` $\rightarrow$ `Perfect` (non-clickbait to clickbait), `Reports` $\rightarrow$ `Recipes` (non-clickbait to clickbait), `MPs` $\rightarrow$ `Princesses` (non-clickbait to clickbait), and `People` $\rightarrow$ `U.S.` (clickbait to non-clickbait), `Released` $\rightarrow$ `Accused` (clickbait to non-clickbait), `try` $\rightarrow$ `announce` (clickbait to non-clickbait).

In general, looking at the results, it is apparent that non-clickbait text converted to clickbait has a more informal tone (adding `You, We, They,` etc.) such as `"Car crashes into house in UK, seriously injuring man"` converting to `"How Men Are Housed in UK, seriously injured?"` and subsequently clickbait text converted to non-clickbait gets rid of these informal language: `"Which TV Character Alter Are You"` gets converted to `"TV Door Alter at on"`.

Similarly, converting from non-clickbait text to clickbait makes a sentence more general, such as `"Deaths in Philippines ferry accidents"` converting to `"Deaths In Travel Accidents!"` and converting from clickbait text to non-clickbait removes hyperbole, such as `"Do You Know The Names Of These "Scandal" Characters"` converting to `"to the Trials to Names for U.S. "Scandals""`.

6.2 Limitations of the Model

Context often gets destroyed/changed during the style transfer. While converting from non-clickbait to clickbait removes specific context such as replacing `Somali pirates` with `Guys`. This may be justified in some contexts, however, while converting from clickbait to non-clickbait text, our model adds new (mostly wrong) context: `"This Artist Turns Colors Into Delicious Smoothies"` converts to `"Obama battle Turns Colors Into China Smoothies."`.

This is perhaps because the difference in the domains of clickbait and non-clickbait text is large and with very specific information such as names, cities, countries, occupations and events, which often get lost/changed during transformation. Moreover, this is an

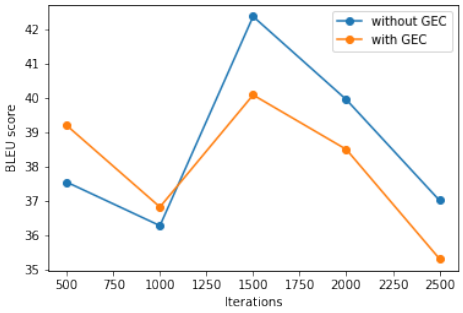

Figure 2: self-BLEU score vs Epochs

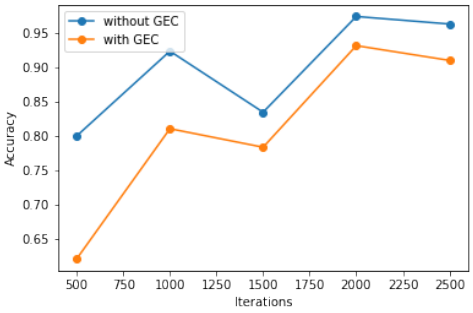

Figure 3: Accuracy vs Epochs

unsupervised model, hence, many sentences often lacks syntactic and semantic meaning when transformed. For example, `"Which "House" Should You Be On"` gets converted to `"Rodriguez "House an the Australian from""`.

The style transformer model, upon which our model is based on, was trained and made specifically for sentiment transfer, changing very specific negative words to positive words in Yelp Reviews. However, here, we are doing a more deep transfer of context than just individual word sentiment.

Compared to the Yelp Dataset with more than $200,000$ unique sentences with a unique vocabulary of $9000+$ words, our clickbait dataset has only $32,000$ sentences with a unique vocabulary of $10,000+$ words. This sentence-vocabulary ratio is definitely bound to have an impact on performance.

6.3 *self*-Bleu Score Analysis over Epochs

Initially, the *self*-bleu score starts high since there's little change in output. As the number of epochs increased, we initially see that the *self*-bleu score increases, implying good performance, but as the epochs increase even more, the scores decrease since our model seems to transform more into the domain its converted to, losing original sentence context. This has been plotted in Figure 2

6.4 Accuracy Analysis over Epochs

As seen in Figure 3, accuracy seems to have mostly an upward trend, since with each epoch, the model seems to transform more into the domain its converted to. In the final 2000+ epochs, we see a small dip, possibly indicating over-fitting.

The GEC model seems to fare worse than the standard style transformer model over large epochs for both plots. This is discussed in future subsections of analysis.

We observed similar trends in accuracy and BLEU score for both ST and CST, however CST takes longer to train. Thus, both plots have been shown for the ST model (which was easier to train for larger epochs).

6.5 Analysis of adding Classifier Features (CST)

As seen from the examples in Table 3, there isn't much difference between the results of the two models. We observed that CST converges at later epochs, owing to which the BLEU score is higher than ST and accuracy is lower. At higher epochs, the model introduces keywords into the sentence that would lead the classifier to predict it to be of the other class. In doing so, however, the sentence becomes difficult to understand and loses context.

However, comparing over similar epochs, it can be argued that the CST Model offers better "clickbaity" titles. For example, the standard style transformer converts `"Russian Report Says Moscow Will Halt Missile"` to `"19 Times Says Moscow Will Halt Missile"`, which doesn't make much sense. However, our CST model converts the same sentence `"Russian Report Says Moscow Will Halt Missile"`

to `"What Report Says Moscow Will Halt Missile"`, which seems to be more "clickbaity", while also making grammatical sense.

6.6 Analysis of adding Grammar Error Correction (GEC) over Epochs

As evident in the plots of epochs vs *self*-bleu scores and epochs vs accuracy, the accuracy and *self*-bleu scores of the model with pretrained GEC are lower than just the standard style transformer model.

6.6.1 Reasons for lower *self*-bleu scores over larger epochs

The GEC model is run over the results of the standard style transformer. In doing so, not only does it correct grammar, but it adds punctuation to the output, something that is missing even in the initial input (the dataset looks to be scraped off punctuation). Hence, adding these tokens automatically worsens the *self*-bleu scores, especially over later epochs.

Moreover, the style transformer when run on any input, doesn't add in any new words from outside the vocabulary we provide to it. However, the pretrained GEC model comes from Google's T-5 transformer, that has been trained on a much larger corpus. Hence, in order to make the sentences grammatically correct, it adds new words potentially outside the vocabulary of the style transformer. This in turn makes our output more different from our input, leading to a lower bleu score.

6.6.2 Reasons for lower accuracy over larger epochs

Accuracy is bound to suffer due to the same addition of punctuation, since both clickbait and non-clickbait sentences in our dataset are scraped off their punctuation. In real life, however, it is much more likely that clickbait text has more punctuation, implying hyperbole with ? and !, hence we kept the punctuations provided by the GEC model, even though it leaves us with lower accuracy.

Similarly, the GEC model tried to correct grammar and make word substitutions wherever possible. These substitutions (especially those out of style transformer vocabulary) result in lower accuracy.

Overall, however, compared to *self*-bleu scores, we see that accuracy isn't that much lower compared to the original style transformer. The peak of standard transformer has about 97% accuracy (incidentally higher than the 93% accuracy of the classifier from the original clickbait classification paper) and the peak of the GEC model is about 92.5%

Overall, it is apparent that the GEC model makes our generated sentences more readable and meaningful to the human eye. For example: the GEC model output `"Budget Ideas To Could Claim Landmarks!"` makes more sense and is arguably more "clickbaity" than the standard transformer output `"Budget Ideas To Could Claim Landmarks"`. Hence, it is possible that without our vocabulary and punctuation hiccups, we can see better accuracy for the GEC model. Hence, it is still a valuable addition to solve the problem of clickbait style transfer.

6.7 Conclusion

Overall, there's a lot still to be done to thoroughly and properly perform style transfer on text. While the cycleGAN architecture is very powerful in a computer vision setting, there need to be significant changes and alterations done to do the same for text. Models like Style Transformer are a good start, but they perform great only in specific domains (such as sentiment transfer in movie and restaurant reviews), our findings show that adapting it to different domains is difficult.

Especially, in the domain of clickbait and non-clickbait data, our model only manages to showcase the potential of style transformation, with a few solid examples. There need to be more models to try upon this domain, with better and larger datasets, especially for unsupervised generation. While it is easy to spot clickbait, it is still hard to distinguish in qualitative terms how successfully a text has been made more "clickbaity." Hence, better metrics need to be constructed to measure the success of a style transformation.

However, given that this is a very new domain (to our knowledge, there hasn't been any other paper attempting to do the same), it is likely that there's huge potential for quick progress. This is a great initial foray into this topic, and we hope to monitor and contribute to the progress in this space.

References

[1] Abhijnan Chakraborty, Bhargavi Paranjape, Sourya Kakarla, and Niloy Ganguly. Stop clickbait: Detecting and preventing clickbaits in online news media. In *2016 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM)*, pages 9–16. IEEE, 2016.

[2] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

[3] Ruochen Xu, Tao Ge, and Furu Wei. Formality style transfer with hybrid textual annotations. *arXiv preprint arXiv:1903.06353*, 2019.

[4] Ning Dai, Jianze Liang, Xipeng Qiu, and Xuanjing Huang. Style transformer: Unpaired text style transfer without disentangled latent representation. *arXiv preprint arXiv:1905.05621*, 2019.

[5] Xiao Li, Guanyi Chen, Chenghua Lin, and Ruizhe Li. Dgst: a dual-generator network for text style transfer. *arXiv preprint arXiv:2010.14557*, 2020.

[6] Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60, 2014.

[7] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.

[8] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[9] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.