

Analyzing Sentiment Polarity Reduction in News Presentation through Contextual Perturbation and Large Language Models

Alapan Kuila, Somnath Jena, Sudeshna Sarkar, Partha Pratim Chakrabarti
IIT Kharagpur, India

{alapan.cse, somnathjena.2011}@gmail.com; {sudeshna, ppchak}@cse.iitkgp.ac.in

Abstract

In today’s media landscape, where news outlets play a pivotal role in shaping public opinion, it is imperative to address the issue of sentiment manipulation within news text. News writers often inject their own biases and emotional language, which can distort the objectivity of reporting. This paper introduces a novel approach to tackle this problem by reducing the polarity of latent sentiments in news content. Drawing inspiration from adversarial attack-based sentence perturbation techniques and a prompt-based method using ChatGPT, we employ transformation constraints to modify sentences while preserving their core semantics. Using three perturbation methods—replacement, insertion, and deletion—coupled with a context-aware masked language model, we aim to maximize the desired sentiment score for targeted news aspects through a beam search algorithm. Our experiments and human evaluations demonstrate the effectiveness of these two models in achieving reduced sentiment polarity with minimal modifications while maintaining textual similarity, fluency, and grammatical correctness. Comparative analysis confirms the competitive performance of the adversarial attack-based perturbation methods and prompt-based methods, offering a promising solution to foster more objective news reporting and combat emotional language bias in the media.

1 Introduction

News media plays a crucial role in building public opinion on different socio-political issues and events by providing information regarding relevant facts and events. In an ideal case, the news outlets should provide the readers with correct and objective information without any bias or slant. However, news writers intentionally or unintentionally insert their own prejudice or perspectives that reflect how they view reality and what they assume to be the truth in the news broadcasts. While

Input Sentence	Rephrased Sentence
The action of the current government hurt the soul of India.	The action of the current government impacted the spirit of India.
He alleged that the bill ignored Indian Muslims.	He claimed that the bill overlooked Indian Muslims.
Demonetisation and GST will boost the economy.	Demonetisation and GST may improve the economy.

Table 1: Instances of rewriting the sentences to lessen the polarity of implicit sentiment preserving the meaning.

writing news reports, they often use emotionally charged words like subjective adjectives and sensational verbs in news text to manipulate readers’ perceptions. These emotion triggering words induce sentiment bias in news reporting, which affects objective journalism. It is crucial to lessen the intensity of the hidden sentiment in order to provide balanced news reporting and more objective news stories. Hence, we need to rephrase the sentences while preserving the semantics to reduce the polarity of the latent sentiments.

Consider the instances of news sentences shown in the Table 1. The first sentence expresses a strong negative sentiment towards *government action*. By replacing the terms like ‘hurt’ and ‘soul’ with ‘impacted’ and ‘spirit’ respectively, we can convey the same information with less intensity compared to the original sentence. The second sentence has a negative assessment regarding *the bill*. In this case, the words like ‘alleged’ and ‘ignored’ induce extreme negativity. We toned down the polarity of the sentence by replacing those words with ‘claimed’ and ‘overlooked’. The third sentence describes *Demonetization* and *GST*, two Indian government policies with positive sentiment. We can lessen the strength of the positive polarity by using the terms ‘may’ and ‘improve’ instead of ‘will’ and ‘boost’. Overall, the problem revolves around rewriting the sentences such that the intensity of the sentiment toward a specific aspect is reduced and is repre-

sented neutrally as much as possible, preserving the semantic meaning.

There exist some works which address a similar type of problem of rewriting the sentence while focusing on neutralizing gender bias, age bias, and demographic bias. (He et al., 2021) propose a two-step framework for neutralizing sentences via rewriting. In the first step, they identify the parts of the input sentence that reveals the target attribute (age/ gender/ origin bias) and mask those words. In the second step, they regenerate the complete sentence by unmasking the sentence such that the output sentence does not reveal the target attribute. During regeneration, they use a gradient-based inference method to make the output sentence attribute-neutral. Their work is inspired by PPLM (Dathathri et al., 2019) that designs a gradient-based inference mechanism for controlled text generation from transformer-based language model. Though this method can regenerate fluent sentences that are neutral to the target attributes, it requires a fine-tuning step that may be computationally intensive and resource-demanding.

(Pryzant et al., 2019) try to neutralize the subjective bias in the sentence by suggesting edits that would make the sentence more neutral. They also provide a parallel corpus of biased and corresponding unbiased sentence pairs from Wikipedia edits. For the neutralization task, they propose a pair of sequence-to-sequence learning algorithms. However, the scope of this work is restricted to single-word modifications.

Some authors (Lample et al., 2018b; He et al., 2020) employ unsupervised style transfer methods to address the task of sentiment transfer where sentences are paraphrased such that it induces a different sentiment while preserving the original content. These works employ several techniques like VAEs (Prabhumoye et al., 2018; Shen et al., 2017), adversarial network learning (Fu et al., 2017), sequence to sequence learning (Krishna et al., 2020) for unsupervised style transfer for manipulating sentiment. However, all these approaches require explicit sentiment labels or parallel corpus containing positive-negative or biased-unbiased sentences. In real-life scenarios, accumulating domain-specific parallel corpus or explicit sentiment-labeled sentences is a cumbersome task. Besides, previous works modified the sentences to change the overall sentiment of the text e.g., to make the sentence more positive, more neutral, or more politically slanted. But the sentences in news

articles may contain more than one news aspect. For example, consider the following sentence

“The government’s successful rollout of testing and vaccination campaigns has been a significant step forward in combating the pandemic, but the persistent issues of healthcare inequality and insufficient support for healthcare professionals require sustained attention and action.”

This sentence presents *government action* positively, whereas the overall *healthcare situation* is depicted negatively. Take a look at another sentence,

“However, the government’s handling of the protests, including the imposition of internet shutdowns and arrests of protest leaders, has been viewed as suppressive and lacking in efforts to engage in constructive dialogue with the farmers.”

In this sentence, *government action* is depicted negatively but mentions *farmers’ protest* neutrally. These sentences contain more than one aspect with different sentiments. Nevertheless, the existing works do not address the problem of transferring the sentiment of one aspect, keeping the sentiment of another aspect unchanged (Jin et al., 2022).

In this paper, our primary objective is to address the novel problem of sentence rewriting, focusing on the modification of sentiment intensity pertaining to the news aspect within the sentence, all while preserving its implicit meaning. To achieve this, we employ two distinct approaches for sentence modification: the first is an adversarial attack-based perturbation method, and the second leverages prompt engineering with a large language model, ChatGPT. Our versatile approach adheres to a set of transformation constraints, facilitating the effective modification of the input news text. The proposed adversarial attack based transformation method involves perturbing the input sentence by masking a portion of the text and replacing the mask with an alternative using a context-aware, pre-trained masked language model. Within our experimentation, we explore three types of perturbations: 1) **replace**, involving the substitution of a token with a new one; 2) **insert**, entailing the addition of a new token; and 3) **delete**, which involves the removal of a token. To support these transformations, we employ a beam search method that navigates through a set of candidate perturbations to attain the desired objective. In our second approach, we harness the capabilities of a large language model, ChatGPT, by employing effective prompt engineer-

ing. This strategic combination allows us to proficiently rewrite sentences, achieving the desired sentiment modification while carefully preserving the essential semantic content. This approach not only benefits from the power of a large language model but also leverages the precision of prompt engineering, enhancing our ability to navigate the complexities of sentiment reduction in news content.

Our main contributions are threefold:

- Our work address the novel problem of sentence rewriting, focusing on the modification of sentiment intensity pertaining to the news aspect within the sentence, all while preserving its implicit meaning.
- To achieve this, we employ two distinct approaches for sentence modification: the first is an adversarial attack-based perturbation method, and the second leverages prompt engineering with a large language model, ChatGPT. These two methodologies offer a robust and flexible means to achieve our goal of neutralizing sentiment in news text.
- Our extensive sentence rewriting efforts have resulted in the creation of a valuable parallel corpus¹, encompassing original sentences and their corresponding neutral sentiment counterparts, which represents a significant contribution to the research community.

2 Methodology

2.1 Overview and Task Definition

We propose a method to mitigate the sentiment polarity in the news by rewriting the sentences. Our principle goal is to neutralize the news aspect-specific sentiment polarity as a way to support objective journalism. Therefore, we focus on maximizing the neutral probability of the targeted news aspects in the sentences. We also demonstrate the efficacy of our method to perturb the sentence for maximizing the positive and negative sentiment as well, which reflects its robustness. Table 2 demonstrates an example of sample modifications applied on an input sentence for maximizing three target sentiments.

We will now present a formal problem definition. A news article contains a sentence $S =$

¹<https://github.com/alapanju/NewsPolarityReduction>

$w_1w_2\dots w_n$ of n words that depicts a news aspect a_t . The task is to produce a modified sentence S' such that $P_s(S', a_t)$ is maximized. Here, $P_s(S, a)$ is the probability that sentence S induces sentiment s towards the news-aspect a . More specifically, $P(\cdot, \cdot) \in [0, 1]$ and $s \in \{positive, negative, neutral\}$. The number of textual modifications to generate S' from S should be minimal such that it appears visually similar to S maintaining grammaticality and overall coherence. However, in order to modify the sentence, we perform three types of textual transformation operations: *replace*, *insert* and *delete*. The perturbation process maintains the following conditions: 1) preserves the semantic meaning as much as possible, 2) does not harm fluency and grammatical correctness, and 3) satisfies certain transformation constraints. Note that the probability of each sentiment tag for an aspect in a sentence can be estimated by any pretrained or finetuned classification model, which takes a sentence and a news-aspect as input and estimates the aspect-specific probability score as output.

In the upcoming section, we will comprehensively explain the range of *transformation* operations applied and the *constraints* enforced during contextualized perturbation in sentences. Furthermore, we will illustrate how the proposed system *search* through different transformations to acquire the resulting transformed output (Figure 1).

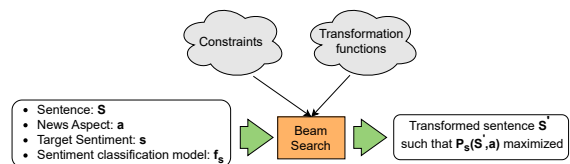


Figure 1: Illustration of Sentiment Polarity Reduction via Contextual Perturbation.

2.2 Contextualized Perturbations

Textual operations that are applied to the sentences are referenced by $transform()$ function in algorithm 1. $transform(S, i)$ denotes the transformation(s) applied on the sentence S at the word index i . We have incorporated 3 types of transformations:

- **replace(S,i):** replace word at index i in sentence S by an alternative word. (e.g. changing “argued” to “alleged” in “He alleged that the CAA ignores Indian Muslims.”). This replacement is achieved by replacing w_i with

Input Sentence	Original Sentiment	Target Sentiment	Modified sentence	Change in sentiment prediction
The decision to declare Rs 500 and Rs 1000 notes invalid from this midnight is a welcome move. (Topic: Demonetization, Aspect: note ban)	positive (0.98)	positive	The announcement to make Rs 500 and Rs 1000 notes invalid from this midnight is a brave move.	0.01
		neutral	The decision to declare Rs 500 and Rs 1000 denominations invalid from this midnight is a significant move.	0.59
		negative	The action to declare Rs 500 and Rs 1000 notes invalid from this midnight is a drastic move.	0.91

Table 2: Example output of sentence perturbation method for various target sentiments. Yellow words show the modifications done. Here the aspect was *note ban* (Topic: demonetization)

a [MASK] token and predicting a probable token at that position using BERT (Devlin et al., 2018).

- **insert(S,i)**: insert word before the word at index i in sentence S by some word.(e.g. changing “*It hurts ...*” to “*It deeply hurts ...*”. This is achieved by inserting [MASK] token before w_i and predicting a probable token at that position using BERT.
- **delete(S,i)**: delete word at index i in sentence S . (e.g. changing “*... is a welcome step.*” to “*... is a step.*”.

$transform(S, i)$ may compose of any combination of the above three transformations. e.g. $transform(S, i) = \{replace(S, i), insert(S, i)\}$ is a possible transformation that allows only replacements or insertions at position i in S . However, *insert* and *delete* operations helps in generating output sentences of varied length. Since these transformations employ a masked LM that uses context information for predicting the [MASK], we call the transformations contextualized perturbations.

2.3 Constraints

We apply a number of constraints in order to ensure that the perturbed sentences are effective in achieving the intended goal of sentiment manipulation while meeting the predefined criteria mentioned in section 2.1. The list of all possible constraints that a transformed sentence should satisfy is as follows:

- **RepeatWordModification**: already modified words are not re-modified.
- **StopwordModification**: prevent the modification of stopwords.

- **MaxModificationRate**: allow only modification of a maximum percentage of words in the input sentence, we have performed our experiments fixing this percentage as 10%.
- **BERTScore**: allow transformations that have at least a minimum cosine similarity between the sentence embeddings obtained using a BERT((Devlin et al., 2018)) model. It is used to maintain the semantics between the input and output sentences (Zhang et al., 2019).
- **Entailment**: allow only transformations which generate sentences that entail the input sentence by at least a certain threshold. Entailment is measured using a pretrained MNLI-based RoBERTa model (Conneau et al., 2019). This constraint guarantees that the transformation operations do not alter the core semantic meaning, a crucial requirement within the news domain.

2.4 Searching through transformations

We employ **Beam Search** algorithm to apply transformations and search for the text that maximizes the goal function. We maintain a beam of k current best transformed sentences, then apply the *transform* function at each word index of each of the k sentences and produces new candidate sentences for the next beam. The new beam is constructed by selecting the top k sentences from the candidate sentences. These sentences are sorted based on the higher score of the targeted aspect level sentiment. For the task of identifying the sentiment score related to the given aspect, we leverage an NLI-based RoBERTa model for aspect-specific sentiment classification (Seoh et al., 2021) system that has been fine-tuned in our lab. More specifically, $f_s(S, H_a) \in [0, 1]$ indicates the probability that the sentence S has a sentiment s towards the aspect

a that is mentioned in the hypothesis H_a , where $s \in \{positive, negative, neutral\}$.

Algorithm 1 briefly summarizes our algorithm used to perform style transfer.

Algorithm 1: Sentiment Transfer via Contextual Perturbation

inputs : S : input sentence;
 H_a : hypothesis with aspect a ;
 s : target sentiment;
 f : NLI based sentiment classification model;
 k : Beam width;
 C : list of constraints;
output : S' : modified sentence satisfying C with $f_s(S', H_a)$ maximized
initialization : $B \leftarrow [S]$; /*current beam*/
 $f_{max} \leftarrow 0$; /*initialize maximum score for target s */
while *True* **do**
 $N \leftarrow []$; /*candidates for new beam*/
 for $i \leftarrow 1$ **to** $len(B)$ **do**
 for $j \leftarrow 1$ **to** $len(S^i)$ **do**
 $S^{ij} \leftarrow transform(S^i, j)$;
 /*transform the i^{th} sentence in B , S^i , considering word at index j */
 if S^{ij} satisfies C and $f_s(S^{ij}, H_a) > f_{max}$ **then**
 | $N.append(S^{ij})$;
 end
 end
 end
 if N is empty **then**
 | return best solution from B ;
 end
 $sort(N)$ in decreasing order of f ;
 $B \leftarrow N[1 : k]$;
 $f_{max} \leftarrow f_s(N[1], H_a)$
end

3 Experiments

3.1 Dataset

For the purpose of bias estimation, we need to extract the aspects present in the news stories and classify the inherent polarity based on the story

representation using our NLI based classification models. We also need the aspect labels for each sentence for the purpose of performing style transfer on news text.

We use GDELT² (Global Database of Events, Language, and Tone) to locate news articles pertaining to specific subjects. GDELT serves as a publicly accessible repository of global events and activities, complete with links to news articles from diverse newspapers covering significant worldwide occurrences and topics. Our process involves the aggregation of pertinent news URLs to extract news stories associated with India. Subsequently, we employ a bag-of-words approach, implementing semi-supervised LDA (Wang et al., 2012), to identify news articles relevant to the four topics mentioned in Table 3. To achieve this, we manually annotate approximately 200 documents covering these four subjects, yielding 1353 annotated sentences. This annotation process is executed using a web-based annotation tool (Mullick et al., 2021).

The sentences are annotated with labels specific to both topic-related aspects and aspect-specific sentiments. Each topic boasts a unique set of aspects, with sentiment labels categorized into three types: 1) positive, 2) negative, and 3) neutral. The topics and topic relevant aspects are described in the Appendix A.1.

We conduct a comprehensive analysis of the news articles, highlighting significant aspects associated with each news topic. Table 3 offers statistical insights into the extracted data for each topic, while Table 4 outlines the selected aspects and the corresponding number of annotated sentences for each topic.

News Topic	Articles count	Time range
Agriculture Act	9000	18th Nov, 2019 - 30th Nov, 2021
Citizenship Amendment Bill (CAB)	589	9th Sept, 2019 - 25th April, 2020
Demonetization	3912	8th Sept, 2016 - 30th Nov, 2021
COVID-19 pandemic	25781	30th Mar, 2020 - 1st Mar, 2022

Table 3: Topicwise statistics of extracted data

3.2 Evaluation Metrics

Evaluating our proposed sentiment transfer model presents a challenge due to the absence of a gold standard or benchmark dataset. However inspired from (Mir et al., 2019) we evaluate the performance based on four factors: 1) change in neutrality, 2) fluency, 3) content preservation and 4) Levenshtein distance.

²<https://www.gdeltproject.org/>

Topic	Aspects	No of annotated sentences
Agriculture Act	farm laws farmer protest government action international involvement	495
CAB	Citizenship Amendment Bill government action protest National Register of Citizens	353
Demonetization	note ban money digitization control black money government action effect on public life	253
COVID-19 pandemic	healthcare situation Lockdown testing and vaccination government action effect on public life	252

Table 4: Topicwise aspects and annotated sentences statistics

- **Change in neutrality:** We evaluate the performance of our style transfer framework using the change in the prediction of target sentiment probability. Henceforth we have used the term *neutrality* to denote the neutral probability prediction. In most of the experiments, we have used the change in *neutrality* as the performance metric. When we used *positive* and *negative* as target sentiments, we measured change in *positivity* and *negativity* as the performance metric.
- **Fluency:** The fluency of the sentences is measured by *perplexity*. We employ GPT2 (Radford et al., 2019) to calculate the perplexity of the sentence.
- **Content Preservation:** We measure the correctness of the transformed sentences by measuring to what extent the output sentence entails the original input sentences using a pre-trained RoBERTa MNLi model (Conneau et al., 2019).
- **Levenshtein distance:** We also measure the similarity between the input and output sentences using 1 - Normalized Levenshtein distance (Yujian and Bo, 2007) as the metric.

3.3 Result Analysis

3.3.1 Comparison of Transformation methods

In Table 5, we evaluate the performance of our adversarial attack-based sentiment transfer framework using different transformation methods. Notably, higher beam sizes consistently improve performance across all methods.

Transformation	Beam width	Average entailment	Average Levenshtein similarity	Average neutrality change
Replace	1	0.55	0.86	0.26
Replace	2	0.57	0.85	0.28
Replace	3	0.58	0.85	0.29
Delete	1	0.61	0.69	0.24
Delete	2	0.58	0.67	0.25
Delete	3	0.57	0.66	0.26
Insert	1	0.47	0.89	0.10
Insert	2	0.46	0.89	0.11
Insert	3	0.45	0.90	0.12
Replace+Insert	1	0.51	0.88	0.26
Replace+Insert	2	0.55	0.88	0.29
Replace+Insert	3	0.53	0.87	0.30
Replace+Delete	3	0.58	0.86	0.29

Table 5: Performance comparison for various search configurations for style transfer - BERTScore threshold was set to 0.95 and maximum 10% word perturbations were allowed. Goal was to maximize the probability of neutral sentiment

We observe that using the *insertion* method alone results in limited neutrality change and lower average entailment scores compared to other methods. Specifically, *insertion* achieves an average neutrality change of around 0.1, while other methods surpass 0.2. Both *deletion* and *replacement* methods perform similarly in terms of neutrality change, although *deletion* exhibits a decrease in textual similarity.

Interestingly, combining *replacement* and *insertion* as transformation methods does not significantly improve neutrality change compared to using *replacement* alone. For instance, *replacement* with a beam size of 3 results in a neutrality change of 0.28, while *replacement* + *insertion* with a beam width of 3 achieves a similar value of 0.29. However, it is important to note that using *replacement* + *insertion* doubles the time required for style transfer compared to using *replacement* exclusively.

In summary, our findings suggest that relying solely on the *replacement* method for text transformation can achieve satisfactory neutrality change with favorable entailment values, as compared to other transformation combinations, given that *deletions* and *insertions* may remove or introduce additional information into the sentences.

3.3.2 Effect of beam size in Performance

Figure 2 shows a histogram of the number of examples with their neutrality values corresponding to each 0.1-sized bin varying from 0 to 1 on the x-axis. The blue bar represents the majority of pairs with neutrality scores between 0 and 0.1, mainly comprising sentences with positive or negative sentiments. The other bars display histograms for three

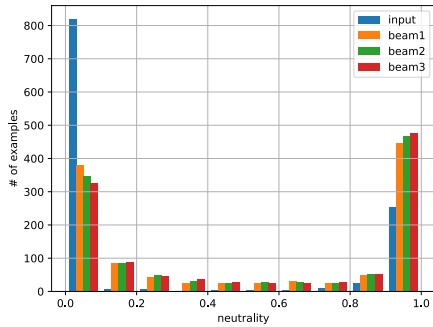


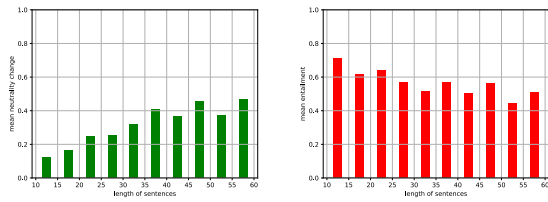
Figure 2: Number of examples for corresponding neutrality change for various beam widths. Goal: to maximize output neutrality, Transformation: only replacement, min BERTScore: 0.95.

settings, each with different beam widths (1, 2, and 3). In all three settings, the target sentiment was "neutral," and the transformation method was word replacement with a minimum BERTScore constraint of 0.95. It's noteworthy that as the beam width increases, the number of examples with high neutrality scores (0.9-1.0) also rises. This trend confirms that a larger beam width leads to better style transfer performance, reducing the chances of suboptimal results.

3.3.3 Effect of length of the sentence on Entailment

Figure 3(a) shows the mean neutrality change for various lengths of sentences. The lengths of the sentences are grouped into bin sizes of 5 in the plot. Out of the 1353 examples, 1070 sentences had lengths between 10 to 60, hence the x-axis varies in that range. The transformation method used in the experiment was word-replacement with the minimum BERTScore constraint of 0.95. Search was done using a beam width of 3.

We observe that as the length of the sentence increases, the neutrality change increases. This is mainly because in longer sentences, we can perform modifications to more number of words. But the higher neutrality change in longer sentences comes at a cost of decrease in entailment as well, as evident from figure 3(b). Since more modifications are possible and we use a pretrained model for measuring entailment and in longer sentences it has to capture larger context, this may lead to loss in entailment with the input sentence. Detailed discussions on the variations in model performance under different experimental setups can be found in the appendix section.



(a) x-axis: length of sentence,(b) x-axis: length of sentence, y-axis: mean neutrality change,y-axis: mean entailment

Figure 3: Variation of neutrality change and entailment with length of sentences. Goal: to maximize output neutrality, Transformation: only replacement.

3.3.4 Comparison with ChatGPT

We try to perform the task of sentiment transfer using ChatGPT ³ based on large language model (LLM) GPT-3.5. We use the ChatGPT API to use its pretrained model using the following prompt as input:

Given Input sentence: "<sentence>", Aspect of the sentence: "<aspect>", Aspect description: "<aspect description>", Modify the input sentence minimally and try not to add the aspect description in the output sentence, while preserving its meaning such that the sentiment towards the aspect "<aspect>" is as neutral as possible. Please return only the output sentence.

where < ... > indicates a placeholder. Note we have also added an aspect description in the prompt, since ChatGPT has never seen these aspects before, but expect it to perform reasonably since it is an LLM trained on huge variety of texts.

Style Transfer method	Average entailment	Average Levenshtein similarity	Average neutrality change
ChatGPT	0.90	0.52	0.13
Our method with beam width 3, using only replacement transformation and minimum BERTScore 0.95	0.58	0.85	0.29

Table 6: Comparison between ChatGPT and our method for sentiment transfer

From table 6, we can observe that in terms of neutrality change our method outperforms ChatGPT, but on the other hand the mean entailment of the output sentences generated by ChatGPT is much higher (0.9) as compared to our method (0.58). Also, the Levenshtein similarity is less in case of ChatGPT (0.52), since it changes the structure of the whole input sentence, instead of just modifying a few words. Better prompt design can

³<https://chat.openai.com/>

be used to leverage the power of ChatGPT in style transfer in a finer way.

3.3.5 Human Evaluation

We also perform human evaluation of the outputs generated by our sentiment transformation framework. For this assessment, we randomly selected 30 sentences from the evaluation set and their rephrased counterparts produced by our proposed methods. These sentence pairs were evaluated by a panel of three raters, which included a Ph.D. scholar specializing in English literature and humanities, as well as two Ph.D. students with a background in computer science. The raters were instructed to score each rephrased sentence based on three criteria: content preservation, neutrality, and fluency, using a scale from 1 to 5 (with 5 indicating the highest score). The criteria encompassed:

- **Content Preservation:** Assessing the extent to which the rephrased sentence preserved the original meaning and context.
- **Neutrality:** Evaluating the effectiveness of the rephrased sentence in reducing sentiment intensity, aligning it with the goal of neutrality.
- **Fluency:** Focusing on the grammatical correctness and overall coherence of the rephrased sentence.

The averaged findings across raters for different transformation settings are reported in Table 7. Across all three criteria, the mean Pearson correlation coefficient for the scores assigned by the three evaluators exceeds 0.67, affirming substantial inter-evaluator agreement.

Transformation Model	Content Preservation	Neutrality	Fluency
Only Replace	3.21	3.53	3.9
Only Delete	2.81	3.11	2.98
Only Insert	2.53	2.89	3.17
ChatGPT output	3.98	4.03	4.4

Table 7: Human Assessment of Results: Evaluating Outputs from Adversarial Attack-Based Transformations (Beam Size 3) and ChatGPT Models, Using Three Criteria: content, neutrality (sentiment) and fluency

4 Related Works

Sentiment transfer is a growing area of research in natural language processing (NLP) and sentiment

analysis. The task of sentiment transfer involves altering the sentiment of a given text while preserving its content and meaning. Several research papers have explored different approaches and techniques for sentiment transfer (Li et al., 2018; Xu et al., 2018; Lample et al., 2018a; Krishna et al., 2020). The existing approaches for transferring sentiment revolve around back-translation (Lample et al., 2018b; Xu et al., 2019), Variational Auto-encoder (Duan et al., 2020), encoder-decoder with discriminator (Majumder et al., 2021; Romanov et al., 2018), pretraining (Zhou et al., 2021) etc. However, there are few works on the fine-grained transfer of sentiments where the objective is to revise the sentence to change the intensity of the sentiment. The work by (Liao et al., 2018) involves utilizing a model based on Variational Autoencoder (VAE) and training it with pseudo-parallel data to enable sentence editing for fine-grained sentiment transfer. (Luo et al., 2019) design a cycle reinforcement learning algorithm for manipulating the fine-grained sentiment intensity of the output sentence. (Liu et al., 2021b) exploit attribute-aware word embeddings for the identification of political bias in news articles and propose a probabilistic algorithm for depolarizing the text. (Liu et al., 2021a) present a Reinforcement learning framework for mitigating political bias in news text. All these works manipulate the sentiment or sentiment polarity of the overall sentences. A few works change the aspect-specific sentiment labels in sentences (Narayanan Sundararaman et al., 2020). In contrast we aim to rewrite the sentence to manipulate the sentiment intensity of the targeted aspect present in the sentence.

5 Conclusion

Our research aims to reduce sentiment polarity in news reporting using advanced techniques like adversarial attack-based perturbations and large language models such as ChatGPT with prompt engineering, striving for more balanced and impartial news narratives. Looking forward, we are exploring avenues for improving efficiency in news sentiment mitigation, including automating the identification of news aspects within sentences. We also aim to address implicit sentiment and expand our techniques to cover a wider range of languages and domains, particularly in low-resource language contexts, with the ultimate goal of advancing news reporting towards greater equity and impartiality.

References

- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [Unsupervised cross-lingual representation learning at scale](#). *CoRR*, abs/1911.02116.
- Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2019. Plug and play language models: A simple approach to controlled text generation. *ArXiv*, abs/1912.02164.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Yu Duan, Canwen Xu, Jiaxin Pei, Jialong Han, and Chenliang Li. 2020. [Pre-train and plug-in: Flexible conditional text generation with variational auto-encoders](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 253–262, Online. Association for Computational Linguistics.
- Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2017. Style transfer in text: Exploration and evaluation. In *AAAI Conference on Artificial Intelligence*.
- Junxian He, Xinyi Wang, Graham Neubig, and Taylor Berg-Kirkpatrick. 2020. A probabilistic formulation of unsupervised text style transfer. *ArXiv*, abs/2002.03912.
- Zexue He, Bodhisattwa Prasad Majumder, and Julian McAuley. 2021. Detect and perturb: Neutral rewriting of biased and sensitive text via gradient-based decoding. *ArXiv*, abs/2109.11708.
- Di Jin, Zhijing Jin, Zhiting Hu, Olga Vechtomova, and Rada Mihalcea. 2022. [Deep learning for text style transfer: A survey](#). *Computational Linguistics*, 48(1):155–205.
- Kalpesh Krishna, John Wieting, and Mohit Iyyer. 2020. [Reformulating unsupervised style transfer as paraphrase generation](#). *ArXiv*, abs/2010.05700.
- Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc’Aurelio Ranzato. 2018a. [Phrase-based & neural unsupervised machine translation](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5039–5049, Brussels, Belgium. Association for Computational Linguistics.
- Guillaume Lample, Sandeep Subramanian, Eric Michael Smith, Ludovic Denoyer, Marc’Aurelio Ranzato, and Y-Lan Boureau. 2018b. Multiple-attribute text rewriting. In *International Conference on Learning Representations*.
- Juncen Li, Robin Jia, He He, and Percy Liang. 2018. [Delete, retrieve, generate: a simple approach to sentiment and style transfer](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1865–1874, New Orleans, Louisiana. Association for Computational Linguistics.
- Yi Liao, Lidong Bing, Piji Li, Shuming Shi, Wai Lam, and T. Zhang. 2018. Quase: Sequence editing under quantifiable guidance. In *Conference on Empirical Methods in Natural Language Processing*.
- Ruibo Liu, Chenyan Jia, Jason Wei, Guangxuan Xu, Lili Wang, and Soroush Vosoughi. 2021a. Mitigating political bias in language models through reinforced calibration. In *AAAI Conference on Artificial Intelligence*.
- Ruibo Liu, Lili Wang, Chenyan Jia, and Soroush Vosoughi. 2021b. Political depolarization of news articles using attribute-aware word embeddings. In *International Conference on Web and Social Media*.
- Fuli Luo, Peng Li, Pengcheng Yang, Jie Zhou, Yutong Tan, Baobao Chang, Zhifang Sui, and Xu Sun. 2019. Towards fine-grained text sentiment transfer. In *Annual Meeting of the Association for Computational Linguistics*.
- Bodhisattwa Prasad Majumder, Taylor Berg-Kirkpatrick, Julian McAuley, and Harsh Jhamtani. 2021. Unsupervised enrichment of person-grounded dialog with background stories. *ArXiv*, abs/2106.08364.
- Remi Mir, Bjarke Felbo, Nick Obradovich, and Iyad Rahwan. 2019. [Evaluating style transfer for text](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 495–504, Minneapolis, Minnesota. Association for Computational Linguistics.
- Ankan Mullick, Animesh Bera, and Tapas Nayak. 2021. [Rte: A tool for annotating relation triplets from text](#).
- Mukuntha Narayanan Sundararaman, Zishan Ahmad, Asif Ekbal, and Pushpak Bhattacharyya. 2020. [Unsupervised aspect-level sentiment controllable style transfer](#). In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 303–312, Suzhou, China. Association for Computational Linguistics.
- Shrimai Prabhumoye, Yulia Tsvetkov, Ruslan Salakhutdinov, and Alan W Black. 2018. [Style transfer through back-translation](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 866–876, Melbourne, Australia. Association for Computational Linguistics.

- Reid Pryzant, Richard Diehl Martinez, Nathan Dass, Sadao Kurohashi, Dan Jurafsky, and Diyi Yang. 2019. Automatically neutralizing subjective bias in text. *ArXiv*, abs/1911.09709.
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.
- Alexey Romanov, Anna Rumshisky, Anna Rogers, and David Donahue. 2018. Adversarial decomposition of text representation. *ArXiv*, abs/1808.09042.
- Ronald Seoh, Ian Birle, Mrinal Tak, Haw-Shiuan Chang, B. Pinette, and Alfred Hough. 2021. [Open aspect target sentiment classification with natural language prompts](#). *ArXiv*, abs/2109.03685.
- Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17*, page 6833–6844, Red Hook, NY, USA. Curran Associates Inc.
- Di Wang, Marcus Thint, and Ahmad Al-Rubaie. 2012. [Semi-supervised latent dirichlet allocation and its application for document classification](#). In *2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, volume 3, pages 306–310.
- 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. *ArXiv*, abs/1805.05181.
- Qionгкаi Xu, Lizhen Qu, Chenchen Xu, and Ran Cui. 2019. Privacy-aware text rewriting. In *International Conference on Natural Language Generation*.
- Li Yujian and Liu Bo. 2007. A normalized levenshtein distance metric. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):1091–1095.
- Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2019. [Bertscore: Evaluating text generation with bert](#). *ArXiv*, abs/1904.09675.
- Wangchunshu Zhou, Tao Ge, Ke Xu, and Furu Wei. 2021. Improving sequence-to-sequence pre-training via sequence span rewriting. In *Conference on Empirical Methods in Natural Language Processing*.

A Appendix

A.1 Topic specific Aspect Descriptions

- Agriculture Act ⁴:
 - government action: The action made by government, PM, Agriculture minister, and the police. These actions include planning for the meeting, requesting a meeting, police firing, lathi-charge, arrest made by police etc.
 - International involvement: Protest against farm law outside India (UK, USA, Canada, etc), discussion on government action in foreign countries.
 - Farm law: reports on farm laws or farm bills or Agriculture Acts, its long term effect on farmers, economy.
 - farmer protest: Reports depicting any type of farmer protest against the Farm Bills such as rally, road blockage, tractor rally, destruction of public property, etc.
- Demonetization ⁵:
 - effect on public life: Information related to problems faced by the public, money shortage, unemployment (demonetization's effect on public life).
 - government action: The action made by government, Prime minister, Finance Minister E.g. meeting, discussion on Note Ban.
 - control black money: black money reduction or recovery after demonetization.
 - note ban: note ban and effects of demonetization.
 - money digitization: digitization of currency
- CAB ⁶:
 - government action: The action made by government, PM, HM, police, Ruling Party. These actions include discussion in Parliament, police firing, lathi-charge, arrest against protesters etc.

⁴https://en.wikipedia.org/wiki/2020_Indian_agriculture_acts

⁵https://en.wikipedia.org/wiki/2016_Indian_banknote_demonetisation

⁶[https://en.wikipedia.org/wiki/Citizenship_\(Amendment\)_Act,_2019](https://en.wikipedia.org/wiki/Citizenship_(Amendment)_Act,_2019)

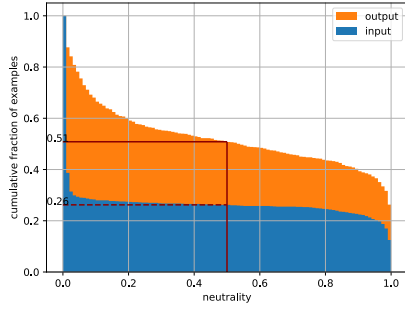
- citizenship amendment bill: reports on citizenship amendment bill (CAB) and its effects.
 - protest: protest against the bill. Eg. protest in Assam, shaheen bagh protest, student protest in the university.
 - National Register of Citizens (NRC): National Register of Citizens (NRC)
- COVID-19 Control ⁷:
 - effect on public life: Displacement of migrant workers, effect on students, jobless and unemployment.
 - healthcare situation: Shortage of healthcare, scarcity of oxygen, hospital beds, health personnel infected, etc.
 - Lockdown: Imposing lockdown in the whole country and states.
 - testing and vaccination: Testing of covid virus transmission and vaccine production and distribution.
 - government action: Relief and welfare initiatives made by the government, meetings held by the government, actions taken by the police, etc.

A.2 Effect of beam size in Performance

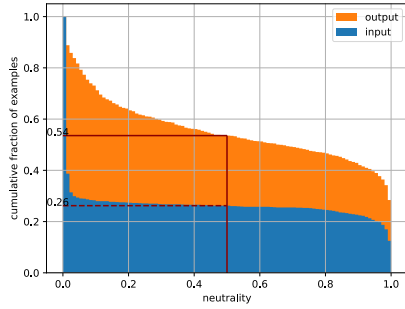
Figures 4(a) and (b) show the cumulative fraction of examples with their neutrality values as predicted by our NLI model greater than certain value. Here the neutrality values are binned into 100 bins of size 0.01 each. Assuming that an instance is classified as neutral if the probability of neutral classification is more than the sum of probabilities of positive and negative classification, we choose a marker at neutrality 0.5 to detect the fraction of examples that are classified as neutral using our NLI model. In the case when beam width was taken to be 1, 51% of the examples have a neutrality score > 0.5 , while in case of beam width 3, the percentage improved to 54%. Thus, there was a 3% increase in the number of transformed sentences that were classified as neutral.

Figure 5 shows that increasing the beam size also leads to a better entailment of the transformed sentences with the input sentences. In case of beam width 3, 61% of the examples have an entailment of > 0.5 with the input sentences which is 5% more than in the case of beam width 1.

⁷https://en.wikipedia.org/wiki/COVID-19_pandemic_in_India



(a) Beam width = 1



(b) Beam width = 3

Figure 4: Cumulative fraction of examples vs neutrality. Goal: to maximize output neutrality, Transformation: only replacement, min BERTScore:0.95

A.3 Entailment variation with neutrality of input sentences

Figure 6(a) and (b) show how the entailment varies with the neutrality change of the input sentences. We see that the mean entailment has a very small variation with neutrality change. It implies that higher neutrality change does not come at a cost of loss in entailment of the modified sentence. The heatmap shows that in most of the cases the output sentence had a high entailment with the original one.

A.4 Effect of BERTScore (Cosine similarity) as a constraint

Minimum Cosine Similarity	Average entailment	Average Levenshtein similarity	Average neutrality change
0.85	0.39	0.85	0.39
0.9	0.45	0.84	0.37
0.95	0.58	0.85	0.27

Table 8: Performance comparison for different values of minimum BERTScore (cosine similarity). In all 3 cases, maximum 10% word perturbations were allowed. Search had a beam width of 3 and only transformation used was replacement.

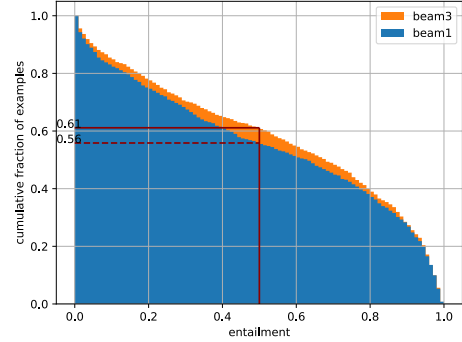
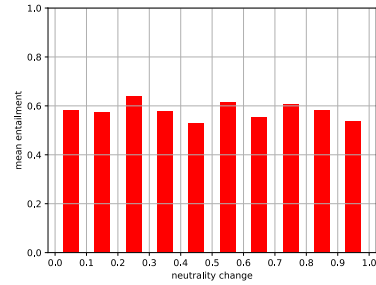
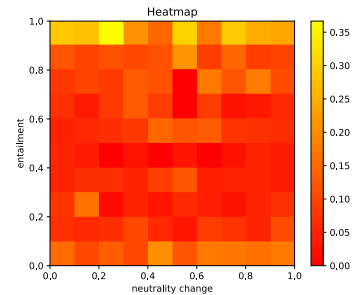


Figure 5: Cumulative fraction of examples vs entailment. Goal: to maximize output neutrality, Transformation: only replacement, min BERTScore:0.95



(a) Mean entailment vs neutrality change



(b) Heatmap of entailment vs neutrality change

Figure 6: Variation of entailment with neutrality change. Goal: to maximize output neutrality, Transformation: only replacement, min BERTScore:0.95, Beam width:3

Table 8 compares the performance of style transfer under the variation of the minimum BERTScore (cosine similarity) constraint. We observe that though decreasing the minimum cosine similarity constraint gives better neutrality change in the output, it comes at a cost of loss in entailment between the input and output sentences. This difference in average entailment is large between minimum BERTScore of 0.85 and 0.95. The levenshtein similarity almost remain unvaried.

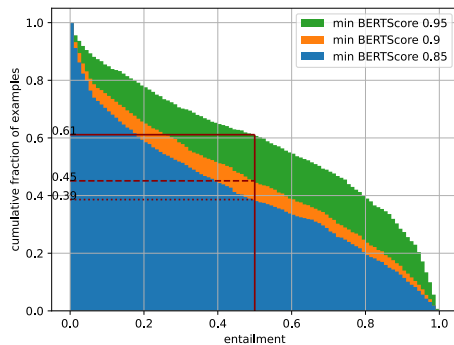


Figure 7: Cumulative fraction of examples vs entailment. Goal: to maximize output neutrality, Transformation: only replacement, Beam width:3

Figure 7 supports the results in table 8. It shows the cumulative fraction of examples with their entailment greater than a certain value. Here the entailment values are binned into 100 bins of size 0.01 each. While 61% of the transformed sentences have more than 0.5 entailment with the original text in case of 0.95 minimum BERTScore, only 39% of the output sentences have more than 0.5 entailment in case of 0.85 minimum BERTScore. This leads to the conclusion that BERTScore is an important constraint for preserving the semantics of the input sentence in the transformed sentence.

Table 9 shows how just changing the minimum BERTScore from 0.95 to 0.9 can change the meaning of the transformed sentence drastically. With minimum BERTScore of 0.9, the modified sentence tells that the decision was to declare Rs 500 and Rs 1000 notes “valid”, which implies that the semantics of the original sentence is not preserved in the output.

A.5 Effect of Entailment as a constraint

Table 10 shows how the style transfer task performs if we add an additional constraint that the output sentence should have a minimum entailment with the input sentence. The transformation method

used was replacement only with constraints of minimum 0.95 BERTScore and minimum 0.3 or 0.4 entailment. The search was done using a beam width of 3. As compared to the results in first three rows of table 5, we can observe that the average entailment shows an overall improvement and so does the Levenshtein similarity. But the gain in entailment leads to a decrease in the average neutrality change. Since the search is now more constrained, it is difficult to obtain more neutral sentences.

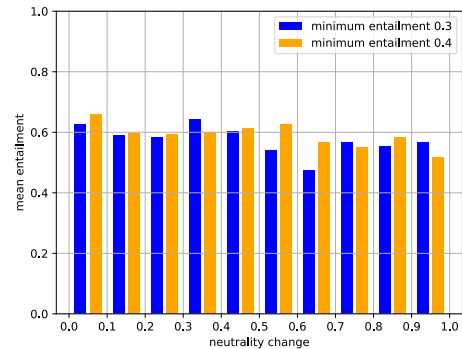


Figure 8: Mean entailment vs neutrality change. Goal: to maximize output neutrality, Transformation: only replacement, min BERTScore:0.95, Beam width:3

In figure 6(a), we saw how the entailment did not follow any increasing or decreasing pattern with increasing neutrality change. But from figure 8, we can see there is slight decrease in mean entailment as the neutrality change increases, though the decrease is not steady. The sentences with neutrality change of [0.9-1.0] have the least mean entailment. This reflects that adding minimum entailment requirement as a constraint, leads to better search of neutral sentences.

A.6 Style transfer for positive and negative sentiments

As we can observe from table 11, using our proposed style transfer technique, we can obtain improved performance in transferring the style to positive or negative sentiment while maintaining comparable entailment and similarity values as in the case of neutrality change.

In figure 4(b), we observed a change of 29% in neutral classification of sentences. In case of positive classification, we observe a change of 39% (figure 9(a)) and in case of negative classification, we observe a change of 43% (figure 9(b)). This suggests doing style transfer to increase negativity in a sentence from our dataset for a particular aspect

Input sentence	Original sentiment	Minimum BERTScore	Modified sentence	Neutrality change	Entailment with input
The decision to declare Rs 500 and Rs 1000 notes invalid from this midnight is a welcome move.	positive	0.9	The decision to declare Rs 500 and Rs 1000 notes valid from this midnight is a significant decision .	0.80	0.38
		0.95	The decision to declare Rs 500 and Rs 1000 denominations invalid from this midnight is a significant move.	0.58	0.88

Table 9: An example showing the effect of minimum BERTScore constraint on the transformed sentence

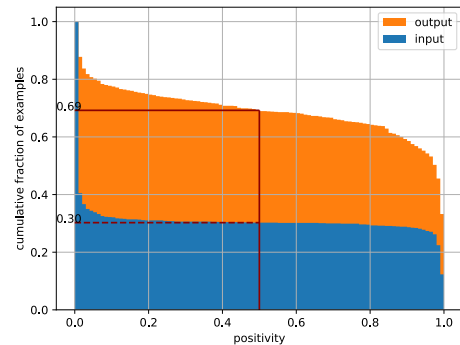
Minimum entailment	Average entailment	Average Levenshtein similarity	Average neutrality change
0.3	0.61	0.86	0.20
0.4	0.63	0.87	0.18

Table 10: Performance of style transfer adding minimum entailment as constraint. Goal was to maximize output neutrality.

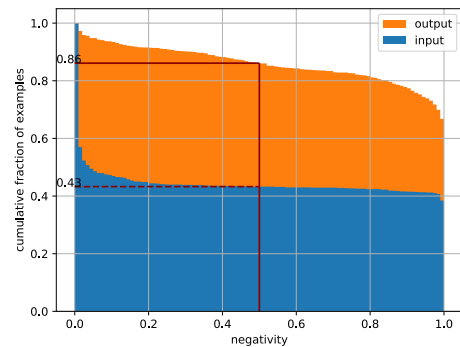
Target sentiment	Average entailment	Average Levenshtein similarity	Average sentiment change
positive	0.54	0.86	0.38
neutral	0.58	0.85	0.29
negative	0.53	0.86	0.42

Table 11: Performance comparison for different targets for style transfer. BERTScore threshold was set to 0.95 and maximum 10% word perturbations were allowed. Search had a beam width of 3 and only transformation used was replacement.

was easier as compared to increasing positivity or neutrality. Note that the input examples have a higher proportion of negative sentences as well.



(a) Target sentiment: Positive



(b) Target sentiment: Negative

Figure 9: Cumulative fraction of examples vs polarity. Transformation: only replacement, min BERTScore:0.95, Beam width:3

"""The code is based on adversarial attack based contextual perturbation in sentence for targeted sentiment modification"""

```
import os
import sys
import pandas as pd
from tqdm import tqdm
from transformers import AutoTokenizer, AutoModelForSequenceClassification,
AutoModelForCausalLM
import torch
from textattack.models.wrappers import HuggingFaceModelWrapper
from textattack.datasets import Dataset
from textattack import Attack, Attacker, AttackArgs

# from textattack.constraints.grammaticality import PartOfSpeech
from textattack.constraints.pre_transformation import (
    InputColumnModification,
    MaxModificationRate,
    RepeatModification,
    StopwordModification,
)
from textattack.constraints.overlap import MaxWordsPerturbed
# from textattack.constraints.semantics.sentence_encoders import UniversalSentenceEncoder
from modified_textattack.constraints import BERT, Entailment

# from textattack.goal_functions import TargetedClassification
from modified_textattack.goal_functions import ModifiedTargetedClassification

# from textattack.search_methods import GeneticAlgorithm
from modified_textattack.search_methods import ModifiedBeamSearch

from textattack.transformations import WordSwapMaskedLM, WordInsertionMaskedLM,
WordMergeMaskedLM, WordDeletion
from textattack.transformations import CompositeTransformation

random_seed = 1

data_folder = 'datasets'
output_folder = sys.argv[1]
gpu_id = int(sys.argv[2])
use_swap = sys.argv[3]
use_swap = True if use_swap=="1" else False
use_ins = sys.argv[4]
use_ins = True if use_ins=="1" else False
use_del = sys.argv[5]
use_del = True if use_del=="1" else False
use_ent = sys.argv[6]
use_ent = True if use_ent=="1" else False
ent_rate = float(sys.argv[7])
target_class = int(sys.argv[8])
```

```

beam_size = int(sys.argv[9])

torch.cuda.set_device(gpu_id)
device = 'cuda' if torch.cuda.is_available() else 'cpu'
torch.cuda.empty_cache()

sent_model_folder = '/home/new_folder/media_bias/temp5/sentiment_900'#the path is wrong.
"new_folder should be replaced with coder's name"

model = AutoModelForSequenceClassification.from_pretrained(sent_model_folder,
local_files_only=True).to(device)

tokenizer = AutoTokenizer.from_pretrained("gchhablani/bert-base-cased-finetuned-mnli")

if not os.path.isdir(output_folder):
    os.mkdir(output_folder)

train_file = os.path.join(data_folder, "all_sentiment_train.csv")
dev_file = os.path.join(data_folder, "all_sentiment_dev.csv")
train_output_file = os.path.join(output_folder, "all_sentiment_train.csv")
dev_output_file = os.path.join(output_folder, "all_sentiment_dev.csv")

output_columns = ["input", "aspect", "output"]

def style_transfer(input_file, output_file):
    output_data = pd.DataFrame(columns= output_columns)
    if os.path.exists(output_file):
        output_data = pd.read_csv(output_file)

    df = pd.read_csv(input_file)
    rows = len(output_data)

    model_wrapper = HuggingFaceModelWrapper(model, tokenizer)

    shared_masked_lm = AutoModelForCausalLM.from_pretrained("distilroberta-base")
    shared_tokenizer = AutoTokenizer.from_pretrained("distilroberta-base")

    composite_list = []
    if use_swap:
        composite_list.append(WordSwapMaskedLM(
            method="bae",
            masked_language_model=shared_masked_lm,
            tokenizer=shared_tokenizer,
            max_candidates=10,
            min_confidence=5e-4,
        ))
    if use_ins:
        composite_list.append(WordInsertionMaskedLM(

```



```

        masked_language_model=shared_masked_lm,
        tokenizer=shared_tokenizer,
        max_candidates=10,
        min_confidence=5e-4,
    ))
if use_del:
    composite_list.append(WordDeletion())

transformation = CompositeTransformation(composite_list)

constraints = [RepeatModification(), StopwordModification()]
constraints.append(InputColumnModification(['sentence', 'hypothesis'], ['hypothesis']))
# constraints.append(UniversalSentenceEncoder(
#     threshold=0.9,
#     metric="cosine",
#     compare_against_original=True,
#     # window_size=15,
#     # skip_text_shorter_than_window=True,
# ))
# constraints.append(PartOfSpeech(allow_verb_noun_swap=False))
constraints.append(MaxModificationRate(max_rate=0.1, min_threshold=0))
# constraints.append(MaxWordsPerturbed(max_num_words=1))
constraints.append(BERT(model_name="stsb-distilbert-base", threshold=0.95, metric="cosine"))
if use_ent:
    constraints.append(Entailment(threshold=ent_rate))

goal_function = ModifiedTargetedClassification(model_wrapper, target_class=target_class,
maximizable=True)

search_method = ModifiedBeamSearch(beam_width=beam_size)

attack = Attack(goal_function, constraints, transformation, search_method)
attack_args = AttackArgs(num_examples = -1, random_seed = random_seed, disable_stdout=True,
silent=True)

for index, row in tqdm(df.iterrows()):
    if index<rows:
        continue

    torch.cuda.empty_cache()

    data = [((row['sentence'])[9:], row['hypothesis']), row['label']]

    dataset = Dataset(data, input_columns=("sentence", "hypothesis"),
label_names=['positive', 'neutral', 'negative'])

    attacker = Attacker(attack, dataset, attack_args)
    attack_results = attacker.attack_dataset()

```

```
for attack_result in attack_results:
    sentence = attack_result.original_result.attacked_text.text.split("\n")[0]
    aspect = attack_result.original_result.attacked_text.text.split("\n")[1][:-20]
    output = attack_result.perturbed_result.attacked_text.text.split("\n")[0]

    output_row = pd.Series([sentence, aspect, output], index=output_columns)
    output_data = output_data.append(output_row, ignore_index=True)
    output_data.to_csv(output_file, index=False)

print(f'Done {index+1} out of {len(df)}')
```

```
style_transfer(train_file, train_output_file)
```

```
style_transfer(dev_file, dev_output_file)
```