TIGTEC : Token Importance Guided TExt Counterfactuals

Anonymous ACL submission

Abstract

Counterfactual examples explain a prediction by highlighting modifications of instance that change the outcome of a classifier. This paper proposes TIGTEC, an efficient and modular method for generating sparse, plausible and diverse counterfactual explanations for textual data. TIGTEC is a text editing heuristic that targets and modifies words with high contribution using local feature importance. A new attention-based local feature importance is proposed. Counterfactual candidates are generated and assessed with a cost function integrating semantic distance, while the solution space is efficiently explored in a beam search fashion. The conducted experiments show the relevance of TIGTEC in terms of success rate, sparsity, diversity and plausibility. This method can be used in both model-specific or model-agnostic way, which makes it very convenient for generating counterfactual explanations.

1 Introduction

The high level of performance in the field of natural language processing (NLP) achieved by Transformer models (Vaswani et al., 2017) comes along with complex architectures. The domain of eXplainable Artificial Intelligence (XAI) aims at understanding and interpreting the predictions made by such complex systems (Molnar et al., 2021). One of the main categories of XAI approaches is local feature importance (Barredo Arrieta et al., 2020) that quantifies the impact of each feature on a specific outcome. Another family of XAI methods consists in explaining with counterfactual examples (see (Guidotti, 2022) for a recent survey), defined as instances close to the instance of interest but associated with another prediction.

This paper proposes a new method to generate counterfactual explanations in the case of textual data. This work presents a new method called

Figure 1: Example of *sparse*, *plausible* and *diverse* counterfactual examples generated by TIGTEC for a film genre classifier that discriminates between horror and comedy synopses. Here, the counterfactual generation goes from comedy to horror.

Token Importance Guided TExt Counterfactuals (TIGTEC). For example, given a film genre classifier and an instance of interest predicted to be a comedy synopsis, TIGTEC outputs several slightly modified instances predicted to be horror synopses (see Figure 1).

The main contributions of TIGTEC are as follows: (*i*) textual counterfactual examples are generated by masking and replacing important words using local feature importance information, (*ii*) a new model-specific local feature importance method based on attention mechanisms (Bahdanau et al., 2014) from Transformers is proposed, (*iii*) initial instance content is preserved with a cost function integrating textual semantic distance, (*iv*) the solution space is explored with a new tree search policy based on beam search that leads to diversity in the generated explanations. In this manner, TIGTEC bridges the gap between local feature importance, mask language models, sentence embedding and counterfactual explanations. TIGTEC can be applied to any classifier in the NLP framework in a model-specific or model-agnostic fashion, depending on the local feature importance method employed.

This paper is organized as follows: we first introduce some basic principles of XAI and the

related work in Section 2. The architecture of TIGTEC is defined in Section 3. Section 4 describes the performed experimental study and compare TIGTEC to a competitor. Finally Section 5 concludes this paper by discussing the results and future work.

2 Background and related work

We recall here some basic principles of XAI methods and existing counterfactual generation methods in NLP.

2.1 XAI background

Local feature importance. Let $f : \mathcal{X} \rightarrow \mathcal{Y}$ be a NLP classifier mapping an input space $\mathcal{X}$ to an output space $\mathcal{Y}$. Let $x_0 = [t_1, ..., t_{|x_0|}] \in \mathcal{X}$ be a sequence of interest with $f(x_0) = y_0$. A local feature importance (or *token importance* in NLP) operator $g : \mathcal{X} \rightarrow \mathbb{R}^{|x_0|}$ explains the prediction through a vector $[z_1, ..., z_{|x_0|}]$ where z_i is the contribution of the $i-$th token.

Two common local feature importance methods are LIME (Ribeiro et al., 2016), whose interest is limited in NLP because of its very high computation time, and SHAP (Lundberg and Lee, 2017).

Counterfactual explanation Counterfactual explanations aim to emphasize what should be different in an input instance to change the outcome of a classifier. Their interest in XAI has been established from a social science perspective (Miller, 2019). The counterfactual example generation can be formalized as a constrained optimization problem. For a given classifier f and an instance of interest x_0, a counterfactual example x^{cf} must be close to x_0 and is defined as:

$$x^{\text{cf}} = \underset{z \in \mathcal{X}}{\operatorname{argmin}}\, d(x_0, z) \text{ s.t. } f(z) \neq f(x_0) \quad (1)$$

with $d : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ a given distance operator measuring proximity. A counterfactual explanation is then the difference between the intial data point and the generated counterfactual example, $x^{\text{cf}} - x_0$.

Many additional desirable properties for counterfactual explanations have been proposed (Guidotti, 2022; Mazzine and Martens, 2021) to ensure their informative nature that we summarize in three categories. *Sparsity* measures the number of elements changed between the instance of interest and the generated counterfactual example. It is defined as the l_0 norm of $x^{\text{cf}} - x$. *Plausibility* encompasses a set of characteristics to ensure that the counterfactual explanation is not out-of-distribution (Laugel et al., 2019), while being feasible (Poyiadzi et al., 2020) and actionable. Since several instances of explanation can be more informative than a single one (Russell, 2019; Mothilal et al., 2020), *diversity* measures to what extent the counterfactual examples differ from each other.

2.2 Related work

This section presents two existing categories of methods for generating textual counterfactual examples.

Text editing heuristics. A first family of methods aims at addressing the problem introduced in Eq. 1 by slightly modifying the input text to be explained with heuristics.

Model specific methods depend structurally on the models they seek to explain. CLOSS (Fern and Pope, 2021) focuses on the embedding space of the classifier to explain. After generating counterfactual candidates through optimization in the latent space, the most valuable ones are selected according to an estimation of Shapley values. MiCE (Ross et al., 2021) iteratively masks parts of the initial text and performs span infilling using a T5 (Raffel et al., 2019) fine-tuned on the corpus of interest. This method targets tokens with high predictive power using model-specific gradient attribution metrics. While the label flipping success rate of CLOSS and MiCe are high and the counterfactual texts are *plausible*, the notion of *semantic distance* and *diversity* are not addressed. We show in Section 3 how the TIGTEC approach that we propose tackles these constraints.

Generating counterfactual examples shares similarities with generating *adversarial attacks*, aiming to incorrectly flip the prediction by minimally editing the initial text. Numerous heuristics have been proposed differing in constraints, text transformation methods and search algorithms (Morris et al., 2020). Contrary to counterfactual explanations, adversarial attacks seek to fool intentionally a model without explanatory purpose. Therefore, *plausibility* and *sparsity* are not addressed.

Text generation with large language models. A second category of methods aims at generating

counterfactual examples in NLP with large pre-trained *generative language models*. A first approach (Madaan et al., 2022) applies a Plug and Play language model (Dathathri et al., 2020) methodology to generate text under the control of the classifier to explain. It consists in learning latent space perturbations from encoder-decoder models such as BART (Lewis et al., 2020) in order to flip the outcome. Polyjuice (Wu et al., 2021) proposes to fine-tune a GPT-2 (Radford et al., 2019) model on a set of predefined tasks. It results in a generative language model capable of performing negation, quantification, insertion of tokens or sentiment flipping based on prompt engineering. Polyjuice needs to be trained in a supervised way on ground truth counterfactual examples in order to be able to generate the expected text. The use of Polyjuice to generate counterfactual examples is therefore not generalizable because counterfactual training data does not exist for all classification problems.

3 Proposed approach, TIGTEC

This section describes the global architecture of TIGTEC by detailing its four components. The main idea is to iteratively change tokens of the initial text by decreasing order of importance instance to find a compromise between proximity to the initial instance and label flipping. This way, TIGTEC belongs to the *text editing heuristics* category of counterfactual example generators in NLP.

3.1 TIGTEC overview

TIGTEC is a 4-step iterative method illustrated in Figure 2. Algorithm 1 describes the generation and evaluation steps, Algorithm 2 summarizes the whole process. TIGTEC takes as input a classifier f and a text of interest $x_0 = [t_1, ..., t_{|x_0|}]$.

Targeting. To modify the initial text to explain, tokens with highest impact on prediction are targeted given their local importance. TIGTEC implements two methods of local token importance and a random importance generator as a baseline.

Generating. High importance tokens are masked and replaced, with a fine-tuned or pretrained mask model. Various counterfactual candidates are then generated.

Evaluating. The generated candidates are evaluated by a cost function that balances the probability score of the target class and the semantic distance to the initial instance. Candidates minimizing the cost function are considered valid if they meet acceptability criteria.

Tree search policy. The lowest cost candidates are kept in memory and a new iteration begins from the most promising one. The solution space is explored in a beam search fashion until a stopping condition is reached.

As outlined in Figure 2, the counterfactual search heuristic is a tree search algorithm, in which each node corresponds to a counterfactual candidate, and each edge is a token replacement. Therefore, the root of the tree corresponds to the instance to explain, and the deeper a node is in the tree, the more it is modified.

3.2 Targeting

The first step consists in identifying the most promising tokens to be replaced in the initial instance to modify the outcome of the classifier f. We use token importance metrics to focus on impacting tokens and efficiently guide the search for counterfactual examples. In particular, we integrate the possibility of computing both model-agnostic (SHAP) and model-specific token importance metrics. We propose a new model-specific token-importance method based on the attention coefficients when the classifier f is a Transformer. Token importance is computed by focusing on the attention of the last encoder layer related to the classification token representing the context of the entire sequence. The efficiency gain of this token importance method is shown in Section 4. If the information provided by SHAP is rich, its computation time is high, whereas attention coefficients are available at no cost under a *model-specific* paradigm.

TIGTEC is also defined by its `strategy` which can take two values. The static `strategy` consists in fixing the token importance coefficients for the whole search, whereas the evolutive `strategy` recomputes token importance at each iteration. Since SHAP has a high computational cost, it is not recommended to combine it with an evolutive `strategy`.

In order to consider several counterfactual candidates at each iteration, several tokens can be targeted in parallel. The `beam_width` parameter allows to control the number of tokens of highest importance to target at each step to perform a beam search during the space exploration.

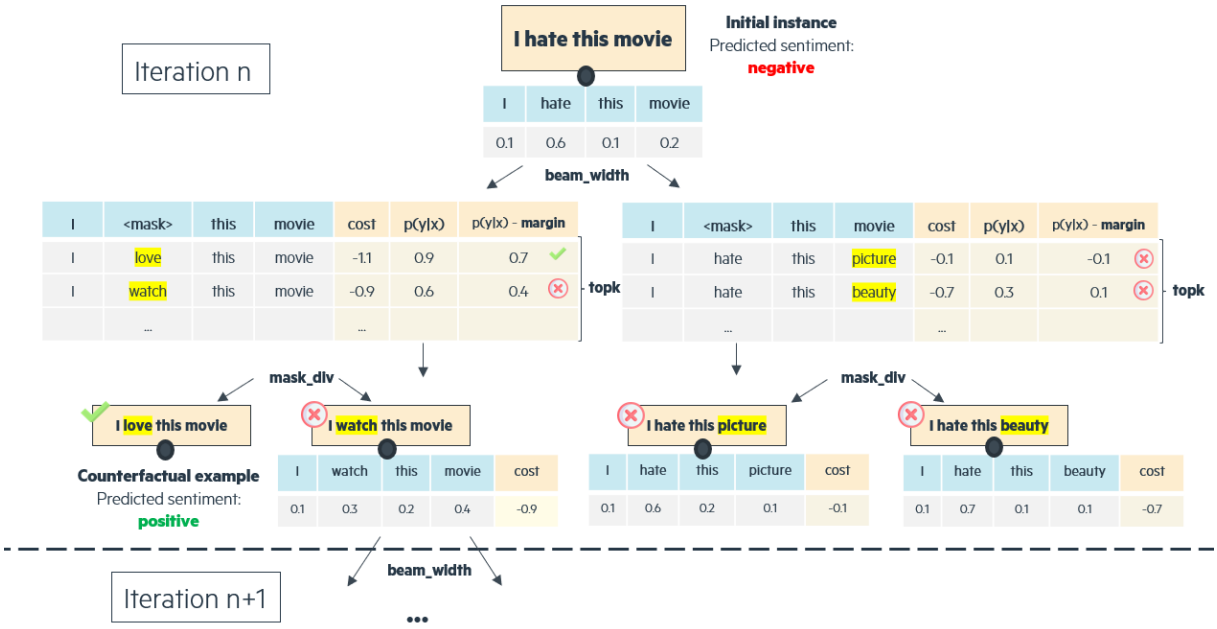

Figure 2: Illustration of the tree search policy with `beam_width` = 2, `mask_div` = 2, `strategy` = evolutive, `margin` = 0.2. At each step, the `beam_width` highest important tokens are masked and replaced. The substitution token is selected considering the cost function depending on the semantic similarity method s and the balancing parameter α. Among the `topk` candidates, only `mask_div` are considered in the tree search. A candidate is accepted if the prediction of the classifier changes and moves `margin` away from the prediction threshold. Here, "I love this movie" is accepted. Since only one counterfactual candidate was found out of two, the next iteration starts from the nodes with the lowest cost value, here "I watch this movie".

3.3 Generating

The second step of TIGTEC generates counterfactual candidates and corresponds to the first part of the mask language inference (MLI) formally described in Algorithm 1, from line 1 to 5. Once high importance tokens have been targeted in the previous step, they are masked replaced with a BERT-type mask language model denoted $\mathcal{M}$. Mask language models enable to replace tokens considering the context while keeping grammatical correctness and semantic relevance. This step ensures the plausibility of the generated text. Such models take a masked sequence $[t_1, ..., [\text{MASK}], ..., t_n]$ as an input and output a probability score distribution of all the tokens contained in the BERT-type vocabulary. The mask model can be either pretrained or fine-tuned on the text corpus on which the classifier f has been trained.

Since replacing a token with another with low plausibility can lead to out-of-distribution texts, inaccurate prediction and grammatical errors, the number of substitutions proposed by $\mathcal{M}$ is limited to `topk`. The higher `topk`, the more we consider tokens with low contextual plausibility.

3.4 Evaluating

Once the `topk` candidates are generated, we build a cost function to evaluate them. This evaluation step corresponds to Algorithm 1 line 6. The cost function has to integrate the need to flip the outcome of the classifier f and the distance to the original instance as formalized in Eq. 1. In order to ensure semantic relevance, we define a distance based on text embedding and cosine similarity measures. Finally, conditions for the acceptability of counterfactual candidates are introduced to ensure the reliability of the explanations.

Distance. The widely used Levenshtein distance and BLEU score (Papineni et al., 2002) do not integrate the notion of semantics. An alternative is to compare sentence embeddings in order to measure the similarity of representations in a latent space. Sentence embeddings have been introduced to numerically represent textual data as real-value vectors, including Sentence Transformers (Reimers and Gurevych, 2019). Such networks have been trained on large corpus of text covering various topics. This encoder is compatible with a model-agnostic approach, as it does not require any

Algorithm 1 Mask Language Inference (MLI)

Require: $x = [t_1, ..., t_n]$ an input sequence
Require: $f : \mathcal{X} \rightarrow \mathcal{Y}$ = {1,2,...,k} a classifier
Require: i the input token to be masked
Require: $\mathcal{M}$ a BERT-like mask language model
Require: s, α, `topk`, `mask_div`
Ensure: $\hat{x}$ = $[\hat{x}_{(1)}, ..., \hat{x}_{(\texttt{mask\_div})}]$
1: $t_i \leftarrow$ [MASK]
2: $x_{\text{mask}} \leftarrow [t_1, ..., [\text{MASK}], ..., t_n]$
3: $[\hat{t}_1, ..., \hat{t}_{\text{topk}}] = \mathcal{M}(x_{\text{mask}})$ the `topk` most likely tokens
4: **for** j in {1,...,`topk`} **do**
5: $\quad \hat{x}_j = x[t_i \leftarrow \hat{t}_j]$
6: $\quad$ Compute $\texttt{cost}(\hat{x}_j)$ see Eq. 4
7: **end for**
8: Retrieve in $\hat{x}$ the `mask_div` sequences with lowest cost
9: **return** $\hat{x}$

information about the classifier f.

Another text embedding approach can be used when the classifier f is a BERT-like model and when the prediction is made through the classification token. It consists in using the embedding of the classification token directly from f. This embedding is however strongly related to the task of the classifier f. Therefore, if the model has been trained for sentiment analysis, two texts with the same associated sentiment will be considered similar, regardless of the topics covered.

We derive the textual distance from the normalized scalar product of the two embeddings: $d : \mathcal{X} \times \mathcal{X} \rightarrow [0, 1]$ with:

$$d_s(x, x') = \frac{1}{2}(1 - s(x, x')) \tag{2}$$

$$s(x, x') = \frac{\langle e_x, e'_x \rangle}{||e_x||.||e'_x||} \tag{3}$$

where e_x is the embedding representation of input sequence x.

Cost. The cost function aims to integrate the objective of the counterfactual optimization problem introduced in Eq. 1. We propose to intergrate the probability score of the target class to define the cost as:

$$\texttt{cost}(x^{\text{cf}}, x_0) = -\left(p(y_{\text{target}}|x^{\text{cf}}) - \alpha d_s(x^{\text{cf}}, x_0)\right) \tag{4}$$

where y_{target} is the target class and $p(y_{\text{target}}|x^{\text{cf}})$ represents the probability score of belonging to the class y_{target} given x^{cf}, outputted by classifier f. The probability score is the information that guides the heuristic towards the target class. The α coefficient enables for a balanced approach to the need to reach the target class while remaining close to the initial point. The generated `topk` candidates are evaluated with the cost function defined above.

Acceptability criteria. A counterfactual candidate x^{cf} is accepted if two conditions are met:

$$f(x^{\text{cf}}) = y_{\text{target}} \tag{5}$$

$$p(y_{\text{target}}|x^{\text{cf}}) \geq \frac{1}{k} + \texttt{margin} \tag{6}$$

where k is the number of classes of the output space, and $\texttt{margin} \in [0, \frac{k-1}{k}]$ a regularization hyperparameter in order to ensure the certainty of the prediction of the model f. We assume then that all the counterfactual examples must reach the same target class. The closer `margin` is to its upper bound, the more polarized the classifier prediction must be in order to satisfy the acceptability criterion, and the the stronger the constraint.

3.5 Tree search policy

TIGTEC generates a set of diverse counterfactual examples. We address the diversity constraint by considering the `mask_div` candidates with the lowest cost function among the generated `topk` from Algorithm 1 and keep them in memory in a priority queue (see line 15 in Algorithm 2). Therefore, we evaluate more possibilities and aim to foster diversity in the counterfactual examples found by TIGTEC. Once these candidates are stored in memory, the iterative exploration step (Algorithm 2 from line 6 to 11) starts again, until a stopping condition is reached.

The candidate with the lowest cost is then selected from the priority queue (see line 6 in Algorithm 2) in order to apply again the targeting, generation and evaluation sequence. We call predecessor this previous candidate. Since we evaluate several possibilities in parallel through beam search, Algorithm 1 is this time applied to the `beam_width` tokens with the highest token importance within the predecessor. From this perspective, the exploration approach enables to start from a candidate that seemed less advantageous at a specific stage, but leads to better results by going deeper into the tree. A tree search example is illustrated in Figure 2.

Dataset	Method	Success rate ↑%S	Proximity ↑s	Sparsity ↓%T	Plausiblity ↓ Δ PPL	Diversity ↑div
IMDB	CLOSS	97.3	95.4	**2.3**	1.47	-
	TIGTEC-specific	**98.2**	**95.8**	4.4	**1.34**	**0.019**
	TIGTEC-agnostic	91.6	95.0	4.6	**1.33**	**0.085**
Movie genre	TIGTEC-specific	88.4	89.8	9.0	1.38	0.120

Table 1: TIGTEC evaluation on 2 datasets and comparison with CLOSS on IMDB.

4 Experimental analysis

This section presents the conducted experimental study and introduces five metrics to quantitatively assess the counterfactual examples generated by two different versions of TIGTEC and one comparable state-of-the-art competitor.

4.1 Evaluation criteria

Considering the various objectives to be achieved, we propose a 5-metric evaluation. Given an instance associated with p counterfactual examples, the evaluation metrics are aggregated on average over the generated examples, except for diversity. The same operation is performed on all the instances to be explained, and the average metrics are finally computed.

Success rate. Since TIGTEC does not guarantee to find counterfactual examples in all cases, the success rate (**%S**) is calculated.

Sparsity. As defined in Section 2, sparsity (**%T**) is measured as the l_0 norm of $x^{\text{cf}} - x_0$, normalized by the length of the sequence.

Proximity. We evaluate *ex-post* the semantic proximity between x_0 and x_{cf} with cosine similarity (**s**) between Sentence Transformer embedding. This choice is justified by the wish to remain in a general framework that does not depend on the classifier f and the task for which it has been trained.

Plausibility. One approach to evaluate text plausibility is the perplexity score (Jelinek et al., 1977). This score can be computed based on the exponential average loss of a foundation model like GPT-2. We calculate the ratio (**ΔPPL**) between the perplexity of the initial text and its counterfactual examples to compare the quality of the generated text with the original one.

Diversity. Based on the distance measure d, we define diversity (**div**) as in (Mothilal et al., 2020) where $div_d = det(K)$ with $K_{i,j} = \frac{1}{\lambda + d(x_i^{\text{cf}}, x_j^{\text{cf}})}$ and $\lambda \in \mathbb{R}$ a regularization weight set to 1.

4.2 TIGTEC agnostic or specific

Two different version of TIGTEC are assessed. The first one is model-specific with access to the corpus of interest. Attention coefficients guide the counterfactual example search and a fine-tuned mask language model is used to mask and replace important tokens. We call this version TIGTEC-specific. The second version is model-agnostic without access to the corpus of interest. SHAP is used to compute token importance and the mask language model is only pre-trained. We call this second version TIGTEC-agnostic. Since SHAP computational cost is high compared to attention, we use the static `strategy` for the *agnostic* version of TIGTEC, whereas the evolutive `strategy` is used for the *specific* one.

4.3 Datasets and competitors

We test these two versions of TIGTEC quantitatively on two DistilBERT (Sanh et al., 2020) binary classifiers. The first one performs sentiment analysis on the IMDB dataset (Maas et al., 2011) containing movie reviews. The second one is trained on movie genre classification on a dataset of horror and comedy synopses from Kaggle[1]. More information about the datasets and the performance of the classifiers are provided in Appendix A.1.

The two versions of TIGTEC are compared to CLOSS to assess their relevance. The objective of each version of TIGTEC is to generate 3 counterfactual examples associated with an initial instance, whereas CLOSS only tries to generate one. Each method is evaluated on the same 1000 texts from IMDB. The hyperparameters of TIGTEC are fixed at their optimal level as described in the next section. TIGTEC-specific

[1]https://www.kaggle.com/competitions/movie-genre-classification/overview

Hyperparameter		mean success rate $\pm\sigma$	mean proximity $\pm\sigma$
Token importance	attention	**98.2*** $\pm$ 5.2	**94.9**** $\pm$ 1.5
	random	92.5 $\pm$ 13.5	92.4 $\pm$ 3.4
Exploration strategy	evolutive	**96.6** $\pm$ 7.3	**94.8**** $\pm$ 1.4
	static	93.8 $\pm$ 13	92.6 $\pm$ 3.5

Table 2: Ablation study of attention-based token importance and evolutive exploration `strategy`. With σ the standard deviation and p as the p-value of the one-tailed t-test, $^*p < 5\%$, $^{**}p < 1\%$.

is also tested on the movie synopsis dataset from Kaggle on 474 texts. Since this is a more complex task, we relax the hyperparameters by lowering the margin to 0.05 and alpha to 0.15.

4.4 Hyperparameter setting

We optimize success rate, similarity, perplexity, diversity and computation time with the 9 hyperparameters presented in Section 3. The optimization is done on IMDB with the Optuna library which generally obtains good results in few iterations (Akiba et al., 2019). Further details about the optmization search space are in Appendix A.2.

We perform the optimization over 50 iterations, with the objective to generate 3 counterfactual examples on 20 different initial texts. The analysis of the contribution of attention-based token importance and evolutive `strategy` are presented in the next section. For the other hyperparameters, `beam_width` = 4, `mask_div` = 4, `topk` = 50, `margin` = 0.15 and $\alpha = 0.3$ and Sentence Transformer embedding seem to be reasonable. Additional analyses are presented in Figure 3 and 4.

4.5 Results

Global results. The results of the conducted experiment are presented in Table 1. Overall, TIGTEC-specific gives very good results on IMDB, succeeding in more than 98% of the time in generating counterfactual examples. The counterfactual examples are sparse, plausible and highly similar to their original instance. TIGTEC-agnostic succeeds significantly less than the *specific* version, but still has a success rate higher than 91%. Proximity, sparsity and plausibility are at the same level as the *specific* version, while the counterfactual examples are significantly more diverse. Since the movie genre classification task is more difficult (see classifiers accuracy in Table 3), the results are slightly less good. TIGTEC still manages in most cases to generate plausible counterfactual examples close to the initial instance, with more diversity compared to the sentiment analysis task.

Comparative results. The TIGTEC-specific method succeeds more often than CLOSS, while remaining on average closer to the initial instance and being more plausible. However, CLOSS generates sparser counterfactual examples than each version of TIGTEC. TIGTEC-agnostic generates more diverse and plausible counterfactual examples, with the same level of proximity. Since the objective of CLOSS is to generate one counterfactual explanation per instance to explain, it does not address the diversity constraint. Generating 3 counterfactual examples per instance rather than one being more challenging, it mechanically decreases the average results of the two versions of TIGTEC on the performance metrics other than diversity. This additional constraint makes the results of TIGTEC overall even better. Examples of counterfactual examples generated by TIGTEC-specific on the sentiment analysis and film genre classification tasks are listed in Appendix A.4.

Ablation study. We evaluate the impact of two main contributions of TIGTEC from the hyperparameter optimization. We compare the attention-based token importance to a random baseline and the evolutive exploration `strategy` to the static one through the success rate and the average similarity with the initial instance. The comparison is made with a one-tailed t-test to determine whether the mean of a first sample is lower than the mean of a second one. The results of the analysis are presented in Table 2. Attention-based token importance success rate and mean proximity are higher than with the random token importance. This difference is statistically significant with a level of associated risk at respectively 5% and 1% for success rate

and proximity. The evolutive `strategy` induces statistically significant higher proximity at a risk level of 1%.

5 Discussion

We have shown that TIGTEC can generate sparse, plausible, content-preserving and diverse counterfactual examples in an *agnostic* or *specific* fashion. Most of the other NLP counterfactual generators strongly depend on the classifier to explain or the text corpus on which it has been trained. As matter of fact, CLOSS generates counterfactual candidates by optimizing in the latent space from the classifier. MiCE uses gradient-based information from the classifier to target important tokens, while modifying the initial instance with a language model fine-tuned on the corpus of interest. Polyjuice needs to learn to generate counterfactual examples in a supervised way, which requires ground-truth counterfactual data. The adaptability of TIGTEC to any type of NLP classifier and the fact that it works in an *agnostic* way make it particularly flexible.

The use of TIGTEC is not limited to BERT-like classifiers. Our proposed framework could be adapted to any type of classifier as long as a token importance method is given as input. For other NLP classifiers such as recurrent neural networks, SHAP or gradient-based methods could be used to target token to be masked. TIGTEC could also be applied to explain machine learning models such as boosted trees by using LIME as token importance method.

6 Conclusion and future work

This paper presents TIGTEC, a reliable method for generating sparse, plausible and diverse counterfactual explanations. The architecture of TIGTEC is modular and can be adapted to any type of NLP model and to classification tasks of various difficulties. TIGTEC can cover both model-agnostic and model-specific cases, depending on the token importance method used to guide the search for counterfactual examples.

A way of improvement of TIGTEC could be to cover more types of classifiers as mentioned in the previous section. Other gradient-based token importance methods could also be integrated to TIGTEC. Furthermore, diversity is only implicitly addressed through the exploration `strategy`. We believe that diversity could be improved by transcribing it into the cost function during the evaluation step or sharpening the exploration `strategy`.

Finally, automatic evaluation of the counterfactual examples quality has its limits. The metrics introduced above provide good indications of the performance of TIGTEC, but they do not ensure human understanding. From this perspective, human-grounded experiments would be more appropriate to assess the relevance of the generated text and its explanatory quality.

Ethics Statement

Since the training data for mask language models, Sentence Transformers and classifiers can be biased, there is a risk of generating harmful counterfactual examples. One using TIGTEC to explain the predictions of one's classifier must be aware of this biases in order to stand back and analyze the produced results. On the other hand, by generating unexpected counterfactual examples, we believe that TIGTEC can be useful in detecting bias in the classifications it seeks to explain. We plan to share our code to make it accessible to everyone. We will do this once the anonymity period is finished. Finally, like any method based on deep learning, this method consumes energy, potentially emitting greenhouse gases. It must be used with caution.

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A Next-generation Hyperparameter Optimization Framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pages 2623–2631, New York, NY, USA. Association for Computing Machinery.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. 2020. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58:82–115.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski,

and Rosanne Liu. 2020. Plug and play language models: A simple approach to controlled text generation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Xiaoli Fern and Quintin Pope. 2021. Text counterfactuals via latent optimization and shapley-guided search. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5578–5593.

Riccardo Guidotti. 2022. Counterfactual explanations and how to find them: literature review and benchmarking. *Data Mining and Knowledge Discovery*.

F. Jelinek, R. L. Mercer, L. R. Bahl, and J. K. Baker. 1977. Perplexity—a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, 62(S1):S63–S63. Publisher: Acoustical Society of America.

Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, X. Renard, and Marcin Detyniecki. 2019. The dangers of post-hoc interpretability: Unjustified counterfactual explanations. In *International Joint Conference on Artificial Intelligence*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Scott M Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning Word Vectors for Sentiment Analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Nishtha Madaan, Srikanta Bedathur, and Diptikalyan Saha. 2022. Plug and Play Counterfactual Text Generation for Model Robustness. ArXiv:2206.10429 [cs].

Raphael Mazzine and David Martens. 2021. A Framework and Benchmarking Study for Counterfactual Generating Methods on Tabular Data. *arXiv:2107.04680 [cs]*. ArXiv: 2107.04680.

Tim Miller. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence*, 267:1–38.

Christoph Molnar, Gunnar König, Julia Herbinger, Timo Freiesleben, Susanne Dandl, Christian A. Scholbeck, Giuseppe Casalicchio, Moritz Grosse-Wentrup, and Bernd Bischl. 2021. General Pitfalls of Model-Agnostic Interpretation Methods for Machine Learning Models. ArXiv:2007.04131 [cs, stat].

John X. Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. In *Conference on Empirical Methods in Natural Language Processing*.

Ramaravind K. Mothilal, Amit Sharma, and Chenhao Tan. 2020. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 607–617, Barcelona Spain. ACM.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. 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, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Rafael Poyiadzi, Kacper Sokol, Raul Santos-Rodriguez, Tijl De Bie, and Peter Flach. 2020. FACE: Feasible and Actionable Counterfactual Explanations. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 344–350. ArXiv:1909.09369 [cs, stat].

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. page 24.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1).

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP/IJCNLP (1)*, pages 3980–3990. Association for Computational Linguistics.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Alexis Ross, Ana Marasović, and Matthew Peters. 2021. Explaining NLP models via minimal contrastive editing (MiCE). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3840–3852, Online. Association for Computational Linguistics.

Chris Russell. 2019. Efficient search for diverse coherent explanations. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19, page 20–28, New York, NY, USA. Association for Computing Machinery.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. ArXiv:1910.01108 [cs].

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel Weld. 2021. Polyjuice: Generating counterfactuals for explaining, evaluating, and improving models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6707–6723, Online. Association for Computational Linguistics.

A Appendix

A.1 Dataset and classifiers

Data sets of interest. TIGTEC is tested on two different data sets. The first one is used for sentiment analysis and is called IMDB. The overall data set is used to train the classifier and TIGTEC is tested on the same sub-sample than the CLOSS competitor. This sub-sample is constituted of 1000 random data points of length less than or equal to 100 words. The second dataset comes from a Kaggle competition to classify movie genres. We propose here to test TIGTEC on a binary classification task between horror and comedy movies. In particular, we test TIGTEC on texts in the Kaggle dataset on which the classifier did not fail. This is equivalent to testing TIGTEC on 474 film synopses. The average number of tokens per sequence per dataset is presented in the Table 3.

Descriptive statistics	IMDB	Movie genre
Avg. tokens	57.4	69.71
DistilBERT acc. %	90.1	88.3

Table 3: Data sets descriptive statistics and classifiers performance

Explained classifiers. Each DistilBERT in initialized as a DistilBERT base uncased from Hugging Face on PyTorch. The text preparation and tokenization step is performed via Hugging Face's DistilBERT tokenizer. The forward path is defined as getting the embedding of the classification token to perform the classification task. A dense layer is added to perform the classification and fine-tune the models. Each classifier has therefore 66 million parameters and is trained with 3 epochs, with a batch size of 12. The loss for the training is a CrossEntropyLoss, and the optimization is done using Adam with initial learning rate of $5e-5$ and a default epsilon value to $1e-8$. The performances of the classifiers are presented in Table 3.

A.2 Hyperparameter optimization search space

The hyperparameter optimization was performed on the solution space presented below:

- $g \in \{random, attention\}$, the input token importance method. Since SHAP is much more time consuming than attention, we exclude it from the optimization. However, it can still be used in our framework.
- $\mathcal{M} \in \{\mathcal{M}_{ft}, \mathcal{M}_{pt}\}$ where $\mathcal{M}_{ft}$ is a mask language model fine-tuned on the corpus in which the classifier f has been trained. $\mathcal{M}_{pt}$ is a pretrained mask language model without fine tuning phase.
- $\alpha \in [0, 1]$ the parameter balancing target probability and distance with the initial point in the cost function
- `topk` $\in \{10, 11, ..., 100\}$ the number of candidates considered during mask inference
- `beam_width` $\in \{2, 3, ..., 6\}$ the number of paths explored in parallel at each iteration
- `mask_div` $\in \{1, 2, 3, ..., 4\}$ the number of candidates kept in memory during a tree search iteration
- `strategy` $\in \{static, evolutive\}$ where *static* is the strategy consisting in computing token importance only at the beginning of the counterfactual search. The *evolutive* strategy consists in computing token importance at each iteration.
- `margin` $\in \{0.05, 0.3\}$ the probability score spread defining the acceptability threshold of a counterfactual candidate
- $s \in \{sentence\_transformer, CLS\_embedding\}$

A.3 Hyperparameter optimization result

We present here the evolution of the quality metrics over all the iterations of the hyperparameter optimization. The results are presented in two parts on categorical and numerical variables in Figure 3 and 4. A point in a graph represents an iteration during the hyperparameter optimization. The metrics are therefore calculated on average over the 20 texts of the iteration.

A.4 Counterfactual examples

Here we show some counterfactual examples related to the tasks of sentiment analysis and film genre classification. Figure 5 and 6 shows counterfactual examples for the sentiment analysis. Figure 7 and 8 show counterfactual examples for film genre classification. The higher the color intensity in red, the higher the token importance coefficient. The tokens appearing in blue in the counterfactual examples are those that have been modified.

Algorithm 2 TIGTEC: Token Importance Guided Counterfactual Text Generation

Require: $f : \mathcal{X} \to \mathcal{Y}$ a k-classes classifier
Require: $x_0 = [t_1, ..., t_n]$ an input sequence of n tokens to be explained
Require: y_{target} : target counterfactual class
Require: p : number of counterfactual examples to generate
Require: g, s, $\mathcal{M}$, α, `topk`, `beam_width`, `mask_div`, `strategy`, `margin`, *early_stop*
Ensure: $x^{\text{cf}} = [x_1^{\text{cf}}, ..., x_p^{\text{cf}}]$

`waiting_list` = $[(x_0, \texttt{cost}(x_0))]$ the priority queue of counterfactual candidates sorted by increasing `cost` (see Eq. 4)
$i \leftarrow 0$ the number of evaluated texts
$x^{\text{cf}} \leftarrow []$
Compute token importance $[z_1, ..., z_n] = g(x_0)$
while $len(x^{\text{cf}}) < p$ and $i <$ *early_stop* **do**
  `parent_node` $\leftarrow$ `waiting_list`.pop() the candidate with the lowest cost (see Eq. 4)
  $[t_{(1)}, ..., t_{(n)}] \leftarrow$ `sort`(`parent_node`) by decreasing importance order with respect to `strategy` and g
  for t in $[t_{(1)}, ..., t_{(\text{beam\_width})}]$ **do**
    $i \leftarrow i + 1$
    $[x_1, ..., x_{\text{mask\_div}}]$ = MLI(`parent_node` , f, t, $\mathcal{M}$, `topk`, `mask_div`, s α) (see Algorithm 1)
    for x in $[x_1, ..., x_{\text{mask\_div}}]$ **do**
      if $p(y_{\text{target}}|x) \geq \frac{1}{k} +$ margin **then**
        x^{cf}.append(x)
      else
        `waiting_list`.push$((x, \texttt{cost}(x)))$ keep in the waiting list rejected candidates with their `cost`
      end if
    end for
  end for
end while
return x^{cf}

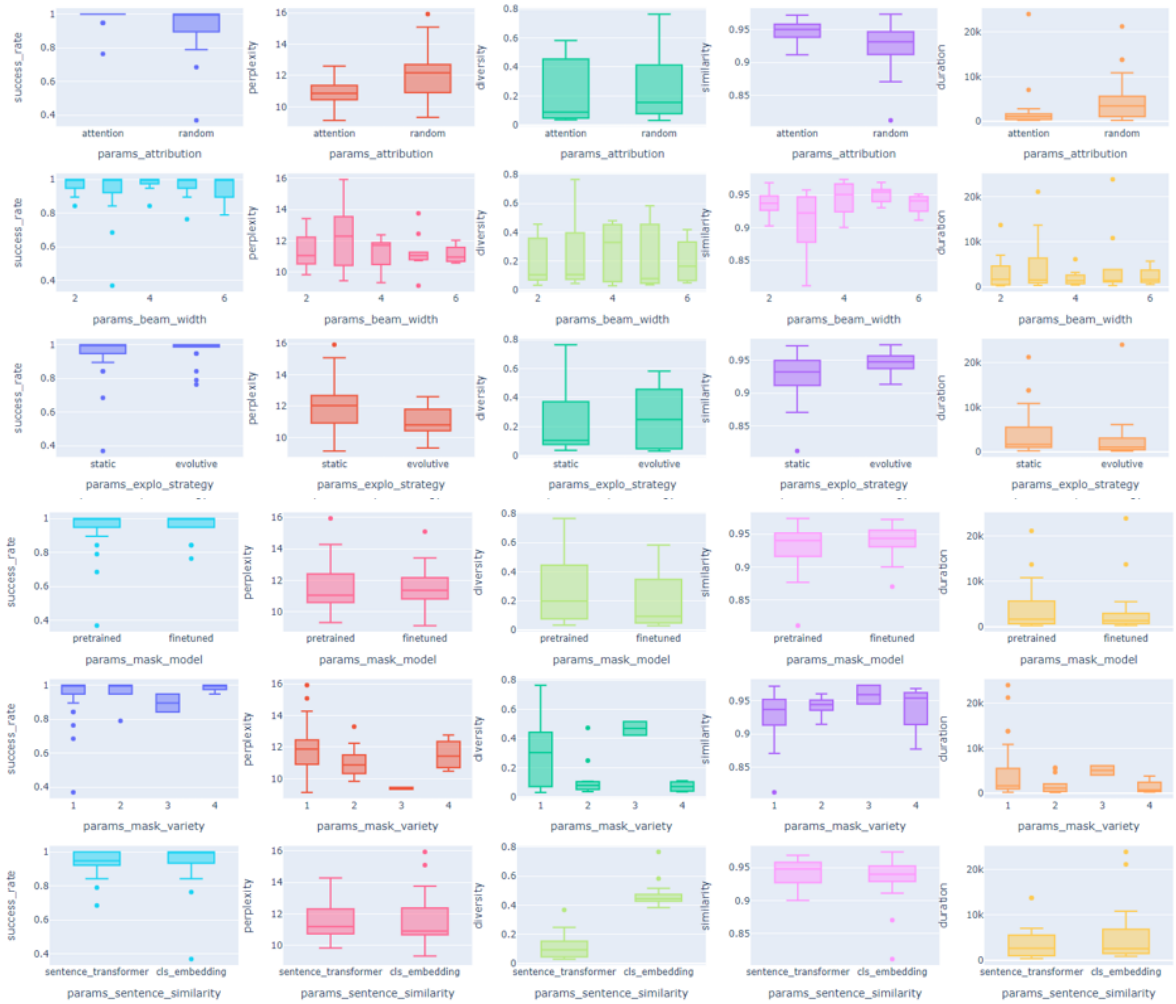

Figure 3: Categorical hyperparameter (in column) optimization according to quality metrics (in rows).

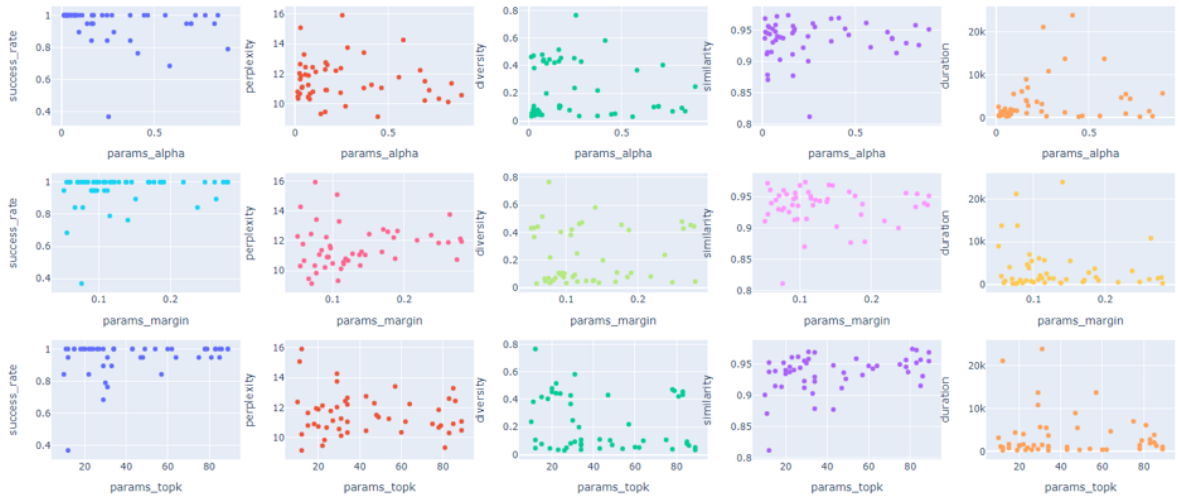

Figure 4: Numeric hyperparameter (in column) optimization according to quality metrics (in rows).

Initial instance

belushi at his most ingratiating and courtney cox before friends has a small role i often think belushi is underused in hollywood and this film role is one of his best for those of you who watch his tv show this is a very different and likable character the movie itself is not

Counterfactual examples

belushi at his most bad and courtney cox before friends has a small role i often think belushi is underused in hollywood and this film role is one of his best for those of you who watch his tv show this is a very different and worse character the movie itself is not

movie at his most bad film courtney cox before friends has a small role i often think belushi is underused in hollywood and this film role is one of his best for those of you who watcl show this is a very different and likable character the movie itself is not

movie at his most bad movie courtney cox before friends has a small role i often think belushi is underused in hollywood and this film role is one of his best for those of you who watch his tv show this is a very different and likable character the movie itself is not

Initial instance

this was a great film in every sense of the word it tackles the subject of tribadism in a society that is quite intolerant of any deviations from the norm it criticises a great many indian customs that many find oppressive such as the arranging of marriages by others the importance of status and face religious

Counterfactual examples

this was a dreadful film in every sense of the word it tackles the subject of tribadism in a society that is quite intolerant of any deviations from the norm it criticises a great many indian customs that many find oppressive such as the arranging of marriages by others the importance of status and face religious

this was a stupid film in every sense of the word it tackles the subject of tribadism in a society that is quite intolerant of any deviations from the norm it criticises a great many indiar. customs that many find oppressive such as the arranging of marriages by others the importance of status and face religious

absolutely not enough great film in every sense of the word it tackles the subject of tribadism in a society that is quite intolerant of any deviations from the norm it criticises a great many indian customs that many find oppressive such as the arranging of marriages by others the importance of status and face religious

Figure 5: Sentiment analysis TIGTEC counterfactual generation, from positive to negative.

Initial instance

to me this film is just a very very lame teen party movie with all the normal cliches and boring stereotyped characters nerds jocks popular girls sleezy guys etc but with an underlying anti drugdrinking theme if you ever have the unfortunate chance of seeing this film keep an eye out for

Counterfactual examples

to me this film is just a very very fun teen party movie with all the normal cliches and boring stereotyped characters nerds jocks popular girls sleezy guys etc but with an underlying anti drugdrinking theme if you ever have the unfortunate chance of seeing this film keep an eye out for

to me this film is just a very very realistic teen party movie with all the normal cliches and boring stereotyped characters nerds jocks popular girls sleezy guys etc but with an underlying anti drugdrinking theme if you ever have the unfortunate chance of seeing this film keep an eye out for

to me this film is just no very very lame teen party movie with all the normal cliches and boring stereotyped characters nerds jocks popular girls sleezy guys etc but with an underly drugdrinking theme if you ever have the unfortunate chance of seeing this film keep an eye out for

Initial instance

this animated inspector gadget movie is pretty lamethe story is very weakand there is little actionmost of the characters are given little to nothing to dothe movie is mildly entertaining at bestbut really doesnt go any where and is pointlessits watchable but only just and is nowhere near the calibre

Counterfactual examples

this great inspector gadget movie is pretty good story is very weakand there is little actionmost of the characters are given little to nothing to dothe movie is mildly entertaining at bestbut really doesnt go any where and is pointlessits watchable but only just and is nowhere near the calibre

this wonderful inspector gadget movie is pretty good story is very weakand there is little actionmost of the characters are given little to nothing to dothe movie is mildly entertaining at bestbut really doesnt go any where and is pointlessits watchable but only just and is nowhere near the calibre

this great inspector gadget movie is absolutely good story is very weakand there is little actionmost of the characters are given little to nothing to dothe movie is mildly entertaining at bestbut really doesnt go any where and is pointlessits watchable but only just and is nowhere near the calibre

Figure 6: Sentiment analysis TIGTEC counterfactual generation, from negative to positive.

Initial instance

casey beldon has nightmarish hallucinations of strangelooking dogs in the neighborhood and an evil child with bright blue eyes following her around while babysitting matty her neighbors son she finds him showing his infant sibling its reflection in a mirror matty attacks casey smashing the mirror on her head and tells her jumby wants to be born now she puts him to bed and leaves in shock caseys friend romy tells her of a superstition that newborns should not see their reflections in the mirror for at least a year because otherwise they will die soon caseys eyes begin to change color a doctor asks if

Counterfactual examples

casey beldon has nightmarish adventures of puppy dogs in the neighborhood and an american child with bright blue eyes following her around while babysitting matty her neighbors son she finds him showing his infant sibling its reflection in a mirror matty attacks casey smashing the mirror on her head and tells her jumby wants to be born now she puts him to bed and leaves in shock caseys friend romy tells her of a superstition that newborns should not see their reflections in the mirror for at least a year because otherwise they will die soon caseys eyes begin to change color a doctor asks if

casey beldon has nightmarish adventures of puppy dogs in the neighborhood and an eccentric child with bright blue eyes following her around while babysitting matty her neighbors son she finds him showing his infant sibling its reflection in a mirror matty attacks casey smashing the mirror on her head and tells her jumby wants to be born now she puts him to bed and leaves in shock caseys friend romy tells her of a superstition that newborns should not see their reflections in the mirror for at least a year because otherwise they will die soon caseys eyes begin to change color a doctor asks if

casey beldon has fun adventures of puppy dogs in the neighborhood and an american child with bright blue eyes following her around while babysitting matty her neighbors son she finds him showing his infant sibling its reflection in a mirror matty attacks casey smashing the mirror on her head and tells her jumby wants to be born now she puts him to bed and leaves in shock caseys friend romy tells her of a superstition that newborns should not see their reflections in the mirror for at least a year because otherwise they will die soon caseys eyes begin to change color a doctor asks if

Initial instance

in 1799 new york city police constable ichabod crane is deployed to the westchester county hamlet of sleepy hollow new york which has been plagued by a series of brutal slayings in which the victims are found decapitated peter van garrett a wealthy farmer his son dirk and the widow emily winship crane learns that locals believe the killer is the undead apparition of a headless hessian mercenary from the american revolutionary war who rides a black steed in search of his missing head crane begins his investigation remaining skeptical about the paranormal elements until he actually encounters the headless horseman who kills the town magistrate samuel phillipse boarding at the

Counterfactual examples

in 1799 new york city police constable ichabod crane is deployed to the westchester county hamlet of sleepy hollow new york which has been plagued by a series of stakes matches in which the cowboys are found decapitated peter van garrett a wealthy farmer his son dirk and the widow emily winship crane learns that locals believe the film is the haunting version of a headless hessian mercenary from the american revolutionary war who rides a black steed in search of his missing head crane begins his investigation remaining skeptical about of supernatural events until he actually encounters the unknown horseman who kills the town magistrate samuel phillipse boarding at the

in 1799 new york city police constable ichabod crane is deployed to the westchester county hamlet of sleepy hollow new york which has been plagued by a series of stakes matches in which the outlaws are found decapitated peter van garrett a wealthy farmer his son dirk and the widow emily winship crane learns that locals believe the film is the haunting version of a headless hessian mercenary from the american revolutionary war who rides a black steed in search of his missing head crane begins his investigation remaining skeptical about of supernatural events until he actually encounters the unknown horseman who kills the town magistrate samuel phillipse boarding at the

in 1799 new york city police constable ichabod crane is deployed to the westchester county hamlet of sleepy hollow new york which has been plagued by a series of stakes matches in which the cowboys are found decapitated peter van garrett a wealthy farmer his son dirk and the widow emily winship crane learns that locals believe the film is the haunting image of a headless hessian mercenary from the american revolutionary war who rides a black steed in search of his missing head crane begins his investigation remaining skeptical about of supernatural events until he actually encounters the unknown horseman who kills the town magistrate samuel phillipse boarding at the

Figure 7: Movie genre TIGTEC counterfactual generation, from horror to comedy.

Initial instance

the film is largely plotless a series of vignettes linked together by interstitial pieces featuring mr mike discussing how upsetting and odd the sequences are he introduces some of the pieces via voiceover and some open with no introduction sequences include

Counterfactual examples

the film is largely plotless a series of vignettes linked together by interstitial pieces featuring mr mike discussing how bizarre and odd the sequences are he introduces some of the pieces via voiceover and some open with no introduction sequences include

the film is largely plotless a series of vignettes linked together by interstitial pieces featuring mr mike discussing how weird and horrific the sequences are he introduces some of the pieces via voiceover and some open with no introduction sequences include

the film is largely plotless a series of vignettes linked together by interstitial pieces featuring mr mike discussing how weird and creepy the sequences are he introduces some of the pieces via voiceover and some open with no introduction sequences include

Initial instance

a group of bachelor professors including a widower have lived together for some years in a new york city residence compiling an encyclopedia of all human knowledge the youngest professor bertram potts is a grammarian who is researching modern american slang the professors are accustomed to working in relative seclusion at a leisurely pace with a prim housekeeper miss bragg keeping watch over them their impatient financial backer miss totten suddenly demands that they finish their work soon venturing out to do some independent research bertram becomes interested in the slang vocabulary of saucy nightclub performer sugarpuss oshea she is reluctant to assist him in his research until she finds a place

Counterfactual examples

a group of bachelor professors including a psychiatrist have lived together for some years in a new york city residence compiling an encyclopedia of all supernatural knowledge the youngest professor bertram potts visits a grammarian who is researching modern american texts the professors are accustomed to working in relative seclusion at a leisurely pace with a prim housekeeper miss bragg keeping watch over them their impatient financial backer miss totten suddenly demands that they finish their work soon venturing out to do some independent research bertram becomes interested in the slang vocabulary of renowned nightclub performer sugarpuss oshea she becomes reluctant to assist him in his research u she finds a place

a group of bachelor professors including a psychiatrist have lived together for some years in a new york city residence compiling an encyclopedia of all paranormal knowledge the youngest professor bertram potts visits a grammarian who is researching modern american texts the professors are accustomed to working in relative seclusion at a leisurely pace with a prim housekeeper miss bragg keeping watch over them their impatient financial backer miss totten suddenly demands that they finish their work soon venturing out to do some independent research bertram becomes interested in the slang vocabulary of renowned nightclub performer sugarpuss oshea she becomes reluctant to assist him in his research until she finds a place

a group of bachelor professors including a psychiatrist have lived together for some years in a new york city residence compiling an encyclopedia of all supernatural knowledge the youngest professor bertram potts visits a grammarian who is researching modern american texts the professors are accustomed to working in relative seclusion at a leisurely pace with a prim housekeeper miss bragg keeping watch over them their impatient financial backer miss totten suddenly demands that they finish their work soon venturing out to do some independent research bertram becomes interested in the slang vocabulary of renowned nightclub performer the oshea she becomes reluctant to assist him in his research until she finds a place

Figure 8: Movie genre TIGTEC counterfactual generation, from comedy to horror.