A Survey on Natural Language Counterfactual Generation

Anonymous ACL submission

Abstract

Natural Language Counterfactual generation aims to minimally modify a given text such that the modified text belongs to a different class. The generated counterfactuals provide insights into the reasoning behind a model's predictions by highlighting which words significantly influence the outcomes. Additionally, they can be used to detect model fairness issues or augment the training data to enhance the model's robustness. A substantial amount of research has been conducted to generate counterfactuals for various NLP tasks, employing different models and methodologies. With the rapid growth of studies in this field, a systematic review is crucial to guide future researchers and developers. To bridge this gap, this survey comprehensively overview textual counterfactual generation methods, particularly including those based on Large Language Models. We propose a new taxonomy that categorizes the generation methods into four groups and systematically summarize the metrics for evaluating the generation quality. Finally, we discuss ongoing research challenges and outline promising directions for future work.

1 Introduction

The recent success in Natural Language Processing (NLP) comes with a variety of Large Language Models (LLMs) such as GPT-3 (175B) (Brown et al., 2020), PaLM (540B) (Chowdhery et al., 2023), and GPT-4 (1.7T) (Achiam et al., 2023). These LLMs have demonstrated superior performance on various downstream tasks. However, alongside the performance, there is a rising concern about their occasionally unexpected behaviors, like hallucinations in their responses (Ji et al., 2023) and misalignment with human expectations (Vafa et al., 2024). These phenomena coincide with the long-standing issue of training deep learning models, which were known to be vulnerable to spurious correlations with artifacts, shortcuts, and biases, prevalent in the real-world training data (Geirhos et al., 2020; Hermann and Lampinen, 2020). Hence, there is a growing demand for AI transparency, particularly in high-stakes applications. This demand underscores the need for research to understand model decisions and enhance their robustness.

Counterfactual generation has emerged as effective means to probe and understand the reasoning behind a model's predictions by highlighting which part of the input influences the outcomes (Wachter et al., 2017; Miller, 2019; Kaushik et al., 2019). It makes minimal modifications to an original instance to create counterfactual examples (CFEs) that have different predicted classes from the original instance. CFEs can help detect model fairness issues against minority groups (Kusner et al., 2017; Russell et al., 2017) and enhancing model robustness and generalizability through the augmentation of training datasets (Sen et al., 2021; Wang and Culotta, 2021; Gat et al., 2024).

In NLP domain, early studies (Jung et al., 2022; Robeer et al., 2021) were inspired by traditional CFE generators for tabular data. However, due to the vast and discrete perturbation space of each word, directly applying these techniques in NLP domain becomes less effective and inefficient. Additionally, textual CFEs should adhere to lexicon and grammar rules, and follow the language context and logic (Sudhakar et al., 2019b; Wu et al., 2021; Ross et al., 2021b). Subsequent research has begun to utilize controlled text generation model conditioning on a sentence and a label (Robeer et al., 2021; Madaan et al., 2021) or replace influential words with proper ones for the target prediction (Ross et al., 2022; Zhu et al., 2023). Recently, the rise of LLMs enable users to craft sophisticated prompts to obtain desired CFEs (Chen et al., 2023; Sachdeva et al., 2024). However, these natural language counterfactual generation methods are not systematically included in surveys on tabular data (Verma et al., 2020; Stepin et al., 2021; Karimi et al., 2022; Guidotti, 2022).

As research in NLP domain rapidly grows, a systematic review is crucial to provide clear guidelines for future researchers and developers. However, comprehensive surveys on this topic are lacking. The challenge may stem from the following aspects. (1) The approach depends on the specific NLP task. Tasks like sentiment analysis, natural language inference, and story rewriting have domain-specific generation strategies. (2) Multiple modifications such as word replacements, deletions, insertions, reordering, and suggesting sarcasm (Kaushik et al., 2019; Wu et al., 2021) can lead to the same desired outcome. Different models, such as BERT (Devlin et al., 2019a) and T5 (Raffel et al., 2020), operate under different mechanisms. This leads researchers to address and formulate the generation problem from broad perspectives. (3) A broad knowledge base including deep generative modeling, causality, AI explanation, beyond NLP is required to comprehensively understand different algorithms, inevitably increasing the review burden.

In this survey, we review past research on natural language counterfactual generation and categorize these methods into four groups, as shown in Figure 1: (1) Manual generation, where a human annotator is asked to edit a limited number of words for a given text to change its label (Kaushik et al., 2019), (2) Gradient-based optimization involves fine-tuning a controlled text generation model using gradient descent, given the input sentence encoding and a desired target (Robeer et al., 2021; Yan et al., 2024), (3) Identify and then generate, a two-stage approach that pinpoints and then substitutes words to alter the labels (Malmi et al., 2020; Gilo and Markovitch, 2024; Martens and Provost, 2014), and (4) LLMs as counterfactual generators, which directly create the counterfactuals via prompting LLMs (Bhattacharjee et al., 2024; Gat et al., 2024; Sachdeva et al., 2024). We also summarize the qualitative and quantitative metrics used to evaluate the quality of the generated counterfactuals. Finally, we discuss the remaining challenges in this field and outline promising research directions, particularly in the era of LLMs.

The rest of this paper is organized as follows: Section 2 introduces the definition of CFEs and practical considerations during generation. Section 3 presents our novel taxonomy and describes each group. Section 4 summarizes the metrics used to evaluate generation quality. Section 5 discusses ongoing challenges and promising research directions. Finally, Section 6 concludes the paper.

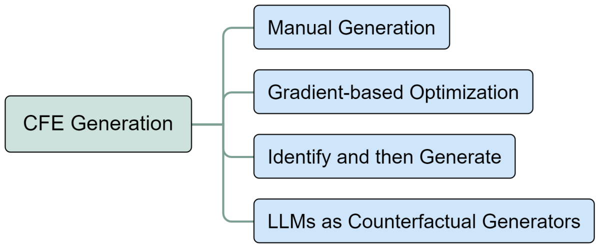

Figure 1: Overview of the proposed taxonomy for natural language counterfactual generation. Refer to Figure 5 in the Appendix for the complete taxonomy.

2 Definition of Counterfactual Example

In machine learning, a counterfactual example (CFE), was initially proposed to explain model decisions on tabular data (Wachter et al., 2017; Miller, 2019; Verma et al., 2020). CFE explains why the model predicts an instance $\boldsymbol{x}$ as the target y instead of its alternative y' by making a *minimal yet necessary* change to $\boldsymbol{x}$ to obtain the desired change in its prediction. To better declare the research scope, we clarify two related terms in Section A, Appendix.

Mathematically, given an input sentence $\boldsymbol{x} \in \mathcal{X}$ with its label $y \in \mathcal{Y}$. $\mathcal{Y}$ represents a set of finite discrete labels for a classification task; whereas for a regression task, $\mathcal{Y}$ denotes a continuous real space. A model $f : \mathcal{X} \subset \mathbb{R}^d \rightarrow \mathcal{Y}$ is employed to predict the label of $\boldsymbol{x}$: $f(\boldsymbol{x}) = y$. A counterfactual generation method $g : f \times \mathcal{X} \rightarrow \mathcal{X}$ modifies a minimal subset of the words in $\boldsymbol{x}$ to produce a counterfactual example, $\boldsymbol{c}$, that alters the model's prediction to a desired label: $f(\boldsymbol{c}) = y'$, where $y' \in \mathcal{Y}$. Hence, generating counterfactual examples can be achieved by solving the following constrained optimization problem,

$$\begin{aligned} &\arg\min_{\boldsymbol{c}} dist(\boldsymbol{x}, \boldsymbol{c}) \\ &s.t.\ f(\boldsymbol{c}) = y', \end{aligned} \tag{1}$$

where $dist(\cdot, \cdot)$ is a distance function that measures the changes required to alter the prediction. This definition describes the essence of the counterfactual problem for simple classification or regression tasks. For a detailed introduction of how counterfactuals are defined in different NLP tasks, please refer to Section B in the Appendix.

During implementation, researchers often need to consider specific constraints to guide the generation and evaluation of counterfactual examples.

Below, we outline the common desiderata.

Validity: is often defined as the distance function (Verma et al., 2020) between the model's prediction of the counterfactual, $f(\boldsymbol{c})$, and the desired label, y'. Minimizing the differences between $f(\boldsymbol{c})$ and y' encourages a higher rate of successful label flips, indicating that the generated CFE is valid.

Proximity: It is the key constraint to create "close possible worlds" with minimal modifications to achieve desired outcomes (Wachter et al., 2017). During implementation, proximity is also defined as the distance between the original $\boldsymbol{x}$ and its counterfactual $\boldsymbol{c}$. This forces the generators to preserve most of the original content while altering only the critical words.

Diversity: A diverse set of CFEs contains multiple possible revisions of a sentence to achieve the target prediction where each revision may tell a different prediction logic. A broad reasoning analysis enhances our trust in a model's prediction (Wachter et al., 2017) and allows us to fine-tune the model for greater robustness (Joshi and He, 2022). To maximize the separation of CFEs during generation, we can incorporate a pair-wise distance function to measure inter-discrepancy (Mothilal et al., 2020; Chen et al., 2021b). Alternatively, we can heuristically guide the generator to modify different spans of a sentence (Chen et al., 2023).

Fluency: measures how smooth and natural a CFE reads, akin to the plausibility in tabular CFE generation (Gilo and Markovitch, 2024). This constraint is typically quantified by metrics like perplexity, which computes the log-probability on the full sentence generated by a language model (Wu et al., 2021). Lower perplexity indicates that the text is more grammatically correct and coherent, which is important to ensure a natural language CFE is understandable.

Other constraints such as controllability (Ribeiro et al., 2020; Wu et al., 2021) and stability (Gardner et al., 2020; Geva et al., 2022) are also considered under specific NLP scenarios. However, these constraints are applied to better control or stabilize process of counterfactual generators, not the final counterfactual examples (Guidotti, 2022). Due to page limit, detailed discussion is omitted.

3 Generation Methods

In this section, we introduce 64 publications for textual counterfactual generation. The paper collection process is illustrated in Section C of Appendix. Based to methodology differences, we propose a novel taxonomy that categorize existing methods into four groups, as shown in Figure 1. Within each group, we further divide these methods into fine-grained classes or successive steps, to ensure that this paper is hierarchically organized. The full taxonomic structure is shown in Figure 5 of the Appendix.

3.1 Manual Generation

Generating fluent textual CFEs proved challenging for early neural network models. Consequently, researchers often relied on domain experts or crowdsourcers from platforms like Amazon Mechanical Turk to manually modify the sentences (Kaushik et al., 2019; Gardner et al., 2020; Yang et al., 2020; Samory et al., 2021).

Before editing, human annotators are given detailed instructions and examples. The editing principles are: (1) Minimal Edits: Make only necessary changes, such as deletion, insertion, replacement, and reordering, to minimally edit the original text using domain knowledge. (2) Fluency, Creativity, and Diversity: Ensure the edits maintain fluency and grammatical accuracy and meanwhile make diverse modifications such as changes to adjectives, entities, and events. (3) Adhere to task-specific rules. For instance, in five-sentence story rewriting, revisions must align with the initial sentence and influence the subsequent storyline (Qin et al., 2019). In question-answering (QA) tasks, counterfactual questions should be answerable based on the given context (Khashabi et al., 2020).

Multiple annotators are often employed to cross-validate the CFEs created by a single annotator (Gardner et al., 2020). Those with lower consensus are then filtered out. However, creating a high-quality CFE dataset through human labor is both time-consuming and expensive (Sen et al., 2023). For instance, Kaushik et al. (2019) reported that modifying and verifying a single CFE typically takes four to five minutes and costs approximately \$0.80. For a detailed discussion on filtering techniques, refer to Section 3.5.

3.2 Gradient-based Optimization

The constrained problem in Equation (1) can be converted to the Lagrange function below,

$$\mathcal{L} = dist(\boldsymbol{x}, \boldsymbol{c}) + \lambda_1 \cdot \ell(f(\boldsymbol{c}), y'), \qquad (2)$$

where $\ell(\cdot, \cdot)$ describes the difference between the desired target y' and current prediction $f(\boldsymbol{c})$, and

$\lambda_1 \in \mathbb{R}^+$ is the Lagrange multiplier. A larger λ_1 will encourage the CFEs to be closer to the desired prediction. Additional desired properties or constraints, such as diversity and fluency, can also be formulated using corresponding mathematical equations, which are appended after Equation (2).

Neural network models, such as, BERT (Devlin et al., 2019b), GPT-2 (Radford et al., 2019), are differentiable, and the distance function $dist(\cdot)$ typically uses L_1 or L_2 norm. We can deliberately choose differential equations or models for other constraints so that the overall loss function $\mathcal{L}$ is also differentiable. Consequently, researchers (Madaan et al., 2021; Hu and Li, 2021; Jung et al., 2022) can employ gradient descent to iteratively minimize the total loss until specific stopping conditions are met.

Following common practices in tabular CFE, this loss can be minimized for a specific sentence $\boldsymbol{x}$ (Jung et al., 2022), but it is not effective. Therefore, most textual generators aim to learn a controlled text generation model by optimizing the total loss over a collection of training samples (Madaan et al., 2021; Hu and Li, 2021; Madaan et al., 2023; Yan et al., 2024). These research have the following two characteristics:

(1) Steering controlled text generation models to generate CFEs. For example, GYC (Madaan et al., 2021) and CASPer (Madaan et al., 2023) leverage the controlled text generation framework PPLM (Dathathri et al., 2020) and generate CFEs conditioning on an input $\boldsymbol{x}$ and desired target.

(2) Selecting proper models to achieve desired properties. CounterfactualGAN (Robeer et al., 2021) uses the StarGAN (Choi et al., 2018) to ensure that the CFEs adhere to the data distribution. To avoid model bias toward spurious correlations, Hu and Li (2021) develop a causal model using variational auto-encoders (VAEs). Yan et al. (2024) disentangle content and style representations using a VAE model. They then intervene in the style variable while maintaining the content variable constant, enabling the generation of counterfactual explanations through the decoder model.

During inference, the original input and the desired target are fed into the pretrained model. As these models are trained in an end-to-end manner, one limitation is that we cannot control the generation process, such as which words should be revised. Additionally, the model is trained by minimizing the loss over a collection of samples, which may compromise the quality of CFEs for certain sentences, such as those pertaining to minority groups. Lastly, controlled text generation does not necessarily produce CFEs with minimal and diverse perturbations.

3.3 Identify and then Generate

A popular family of approaches decomposes the generation task into two steps: (1) identifying the word positions to be revised in the original text, and (2) minimally editing those positions to generate CFE candidates with target predictions, as shown in Figure 2.

3.3.1 Identification step

The simplest strategy involves either selecting random word positions (Fu et al., 2023) or revising all word positions (Fern and Pope, 2021). However, such approaches fail to discriminate between word positions that potentially contribute to valid counterfactuals and those that do not. Consequently, the subsequent generation step may produce futile results, leading to unnecessary costs. Therefore, researchers propose more deliberately designed identifiers, which are summarized as follows:

(1) Words statistics. This approach (Madaan et al., 2020; Li et al., 2018) first calculates the frequency of words or n-grams that appear in the target domain corpus using traditional term frequency (TF) and/or inverse document frequency (IDF) measures. It then marks those words or n-grams whose frequency scores exceed a specific threshold.

(2) Syntactic parser. Syntax plays a crucial role in model predictions across many tasks. For example, adjectives ('good', 'delicious') and verbs ('like', 'hate') are often considered closely linked to sentiment polarity. Subjects and objects are important for understanding the logical relationship in the NLI task. Consequently, researchers (Chen et al., 2023; Geva et al., 2022) adopt a syntactic parser to split a sentence into spans. Control codes (Ribeiro et al., 2020; Wu et al., 2021) are incorporated into parsers to produce different types of perturbations for various purposes. Additionally, Tailor (Ross et al., 2022) analyzes text syntax to extract high-level and semantic control codes, enabling flexible and meaningful perturbation strategies.

(3) Word importance. The approaches in this category identify important words that significantly contribute to the original prediction. For example, given a positive text such as "It is a fantastic moment," the word 'fantastic' would be identified as the crucial word for the positive label. Compared

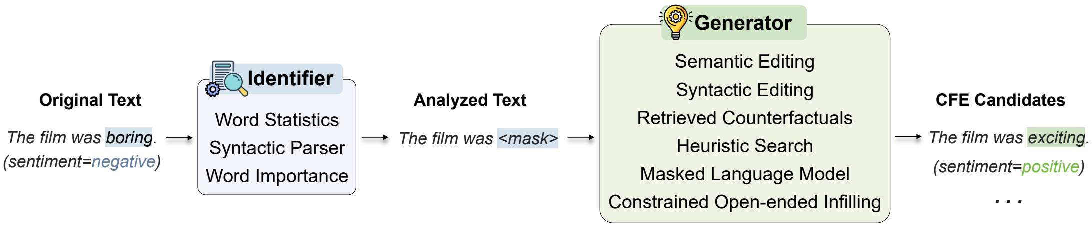

Figure 2: Demonstration of the Identify-and-then-Generate CFE generation.

to identifiers based on word statistics and syntactic parsers that only require an input sentence, the word importance-based identifier additionally necessitates a pretrained model to judge word importance via prediction differences.

Conveniently, importance scores can be readily obtained from current feature importance approaches such as gradients (Simonyan et al., 2014), integrated gradients (Sundararajan et al., 2017), LIME (Ribeiro et al., 2016), SHAP (Lundberg and Lee, 2017), and CURE (Si et al., 2023). For instance, MICE (Ross et al., 2021b) uses gradients to determine which tokens to mask; LEWIS (Reid and Zhong, 2021) identifies style-related tokens with above-average attention weights; Polyjuice (Wu et al., 2021) and AutoCAD (Wen et al., 2022) incorporate LIME and SHAP as plugins to identify mask positions; Martens and Provost (2014) identify a minimal set of words whose removal would revert the current prediction. Typically, a higher importance score indicates a greater significance to the original prediction, and such tokens are more likely to be replaced in the generation step. □

The above techniques can be combined to achieve more precise identification of editing locations. For instance, AC-MLM (Wu et al., 2019) combines word frequency and attention scores to obtain accurate locations.

For word statistics and word importance-based identifiers, each word is assigned a score. Then, we need to determine how many words should be masked. Masking too many words increases CFE's proximity, while masking too few may result in void CFEs. Empirically, recent studies often employ predefined rules, such as selecting the top-K words or spans (Malmi et al., 2020; Wen et al., 2022), choosing words whose scores exceed a certain threshold (Wu et al., 2019; Hong et al., 2023), or adaptively controlling the number of masked tokens (Reid and Zhong, 2021; Madaan et al., 2020).

3.3.2 Generation step

Once the word positions are identified, the next step is to replace the original words at these positions with appropriate replacements to achieve the target prediction. We list common generators below:

(1) Semantic editing. An intuitive solution is to substitute the important words with their corresponding semantic counterparts such as antonyms. They can be readily obtained with existing lexical databases like WordNet (Chen et al., 2021b; Wang and Culotta, 2021; Chen et al., 2021a). Alternatively, they can be searched within the dataset of the target class (Li et al., 2018; Gilo and Markovitch, 2024). This strategy is limited to tasks related to semantic understanding (Wen et al., 2022).

(2) Syntactic editing. These methods (Li et al., 2020a; Zhu et al., 2023; Longpre et al., 2021; Geva et al., 2022) leverage existing language parsers to decompose a sentence into several syntactic spans, then design customized rules to transform each span into the desired output. Examples include inserting 'not' before verbs or adjectives, swapping subjects and objects, modifying tense, substituting a word with another entry from the corpus, or tampering with factual evidence. Such approaches are primarily designed for tasks like natural language inference, named entity recognition, and fact verification, where the model predictions are sensitive to the tense, location of passive and subject, and evidence.

(3) Retrieved counterfactuals. Retrieval-based approaches (Li et al., 2018) first retrieve an open-source database using the masked original sentence. Subsequently, filtering techniques are used to keep valid and minimally revised candidates. RGF (Paranjape et al., 2022) directly generates counterfactual questions based on retrieved context and answers in QA task. Although RGF does not need to identify word positions, we categorize this method here due to its use of retrieval techniques. The major concern with this approach is that the retrieved counterfactuals may not be as similar to the original sentence as other methods.

(4) Heuristic search. These methods (Fern and Pope, 2021; Gilo and Markovitch, 2024) employ heuristic search to find appropriate replacements within a defined search space. The key contributions of these methods are the construction of the search space and the development of search strategies. Fern and Pope (2021) first identify the k potential substitutions for each word and adopt a Shapley-value guided search method. Gilo and Markovitch (2024) start from a CFE in the training dataset and leverage the weighted A^* algorithm to iteratively reduce the edit cost.

(5) Masked language models (MLMs). The identified word locations can be masked with specific tags such as '[MASK]'. An MLM can then be used to edit these tags to achieve the target prediction. For example, consider a masked sentence like "There is a [MASK] moment," with a goal to generate a negative expression, MLMs might fill in the mask with words like 'terrible' or 'dismal'.

The primary contributions of approaches in this family revolve around how they leverage and train MLMs for infilling tasks. (1) Some methods (Ribeiro et al., 2020; Chen et al., 2022; Chemmengath et al., 2022) directly leverage the pretrained MLMs to infill the blanked words. While convenient, the generated words may not always align with the desired properties, often necessitating post-hoc filtering to meet user expectations. (2) Other approaches involve finetuning MLMs on target domain data (Malmi et al., 2020; Reid and Zhong, 2021) and then used to infill the blanks. (3) A widely adopted method (Wu et al., 2019; Ross et al., 2021b; Hao et al., 2021; Calderon et al., 2022; Wen et al., 2022) involves finetuning the MLM to reconstruct sentences from the masked sentences and their original predictions. Here, the MLM learns to infill the blank in a way that is consistent with a given prediction. During inference, the MLM is provided with a masked sentence and the target prediction to produce CFEs. (4) Some researchers directly finetune an MLM to learn the controlled generation from the source domain to the target domain (Wu et al., 2021; Ross et al., 2022). However, this approach often requires a substantial amount of training data. For instance, (Wu et al., 2021) recommends collecting 10,000 instances per control code, which can be burdensome.

The primary drawback of these approaches is that MLMs focus solely on revising the masked positions, which leads to a lack of linguistic diversity in generated CFEs.

(6) Constrained open-ended infilling. This approach aims to infill the masked positions more flexibly while restricted by a label flip rate constraint, compared to MLM approaches that strictly infill the mask locations with replacements. For example, NeuroCFs (Howard et al., 2022) first identify key concepts and then use a GPT-2 model, adapted to the target prediction, to decode these concepts. DeleteAndRetrieve (Li et al., 2018) concatenates the embeddings of the masked original sentence and a retrieved sentence with the target prediction, then adopts a decoder to generate a CFE.

3.4 LLMs as Counterfactual Generators

In the past two years, LLMs have shown remarkable proficiency in synthesizing natural languages for downstream tasks (Meng et al., 2022; Ye et al., 2022; Meng et al., 2023; Yu et al., 2024). Significant research has focused on designing effective prompts to harness the advanced reasoning and understanding capabilities of these models for generating desired content, including CFEs (Dixit et al., 2022; Gat et al., 2024; Chen et al., 2023). In recent literature, two key technologies in enhancing the generation results are In-Context Learning (ICL) and Chain-of-Thought (CoT).

Introduced with GPT-3 (Brown et al., 2020), ICL improves prompts by including examples that demonstrate the expected type of reasoning or output. To generate counterfactuals for a given instance, the prompt typically consists of the task requirement and one (Sachdeva et al., 2024) or a few pairs of original and counterfactual examples as demonstrations (Dixit et al., 2022; Chen et al., 2023; Gat et al., 2024; Sachdeva et al., 2024). These in-context counterfactuals are either manually created (Chen et al., 2023; Gat et al., 2024) or retrieved from an external unlabeled corpus (Dixit et al., 2022).

CoT prompting, introduced by Wei et al. (2022), elicits the emergent reasoning capability of LLMs by incorporating a series of intermediate reasoning steps into the prompt. For example, in sentiment classification, generating counterfactuals for a positive sentence involves two steps: (1) identifying and (2) altering words that convey positive sentiment (Bhattacharjee et al., 2024; Nguyen et al., 2024; Li et al., 2024). This technique is more evident in question-answering tasks, where Sachdeva et al. (2024) demonstrate that the counterfactuals for an answer can be obtained by first generating a counterfactual question based on the factual context and then producing the corresponding answer.

3.5 Filter

Since the automatic counterfactual generators may produce degenerate counterfactuals (incoherent, illogical, or invalid) for some inputs, post-hoc filtering is typically employed to filter out these degenerate cases.

Human filtering (Zhang et al., 2019) ensures high-quality filtering but is time-consuming and labor-intensive. Therefore, researchers often use automated tools to remove undesired outputs. These automated methods include eliminating CFE candidates that are incorrectly predicted by state-of-the-art (SOTA) models (Reid and Zhong, 2021; Zhang et al., 2023; Chang et al., 2024); deleting degenerations with low fluency scores computed by language models (Li et al., 2020a; Wu et al., 2021; Ross et al., 2022; Gilo and Markovitch, 2024); and selecting human-like counterfactuals based on proximity scores (Yang et al., 2021).

3.6 Summary

Owing to length constraints, we cannot discuss all papers in each group. Instead, we focus on a few of the most pertinent studies for each point to ensure that the essential information is conveyed clearly. Complete references for each group can be found in Appendix Section D.

4 Evaluation Metrics

Validity. It measures the proportion of CFEs that achieve the desired target among all generated CFEs. Formally, the validity over N test samples is defined by,

$$Validity = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}(\hat{f}(\boldsymbol{c}_i) = y'_i), \tag{3}$$

where y'_i is the desired target of a CFE $\boldsymbol{c}_i$. The predictor $\hat{f}$ can be human annotation (Wu et al., 2021; Chen et al., 2021b), fined-tuned SOTA models (e.g., RoBERTa (Ross et al., 2021b; Wen et al., 2022; Betti et al., 2023; Balashankar et al., 2023; Gat et al., 2024), or BERT (Betti et al., 2023; Bhattacharjee et al., 2024) in sentiment analysis, and DeBERTa (Chen et al., 2023) in natural language inference), or voting with multiple models (Sachdeva et al., 2024). A higher validity is preferred.

Similarity. Similarity measures the editing effort of a CFE required during generation (Wu et al., 2021; Kaushik et al., 2019), formally defined as,

$$Similarity = \frac{1}{N}\sum_{i=1}^{N} dist(\boldsymbol{x}_i, \boldsymbol{c}_i). \tag{4}$$

For lexical and syntactic similarity evaluations, widely used methods include the word-level Levenshtein edit distance (Levenshtein et al., 1966) and the syntactic tree edit distance (Zhang and Shasha, 1989). For assessing semantic similarity, models like SBERT (Reimers and Gurevych, 2019) and the Universal Sentence Encoder (USE) (Cer et al., 2018) are commonly used. They encode both the CFE and the input text and then calculate the cosine similarity between their sentence representations.

Diversity. This score is measured as the average pairwise distance between K returned CFEs for a sentence $\boldsymbol{x}$, defined as follows,

$$Diversity = \frac{1}{\binom{K}{2}}\sum_{i=1}^{K-1}\sum_{j=i+1}^{K} dist(\boldsymbol{c}_i, \boldsymbol{c}_j). \tag{5}$$

For lexical diversity, Self-BLEU (Zhu et al., 2018) reports the average BLEU score between any two CFEs, while Distinct-n (Li et al., 2016) gauges diversity by calculating the ratio of unique n-grams to the total number of n-grams in the generated CFEs. When semantic diversity is assessed, the $dist(\cdot)$ function can be metrics like SBERT embedding similarity (Reimers and Gurevych, 2019), BERTScore (Zhang et al., 2020), semantic uncertainty (Kuhn et al., 2023).

Fluency. As fluency describes the resemblance of a CFE to human writing, a simple measurement is to ask human raters to evaluate a CFE based on cohesiveness, readability, and grammatical correctness (Robeer et al., 2021; Madaan et al., 2021). Due to the irreproducibility and high cost of human evaluation, automated fluency evaluations such as the likelihood and the perplexity score have become popular in recent studies (Ross et al., 2021b; Sha et al., 2021; Treviso et al., 2023).

(1)*Likelihood* (Salazar et al., 2020). Given a sentence of length n, we create n copies by individually masking each of the n tokens. We then use a masked language model (MLM), such as T5-based models, to compute the loss for both the original sentence and its n masked copies. The likelihood is calculated as the average ratio of the loss of each masked copy to the loss of the original sentence.

(2) *Perplexity score* (Jelinek et al., 2005). This score evaluates whether the produced CFEs are natural, realistic, and plausible. In practice, we quantified this using the powerful generative LMs (e.g., GPT-2 (Radford et al., 2019)), formally described as follows,

$$perplexity = \exp\left[-\frac{1}{n}\sum_{i=0}^{n}\log p_\theta(t_i|t_{<i})\right], \quad (6)$$

where $p_\theta(t_i|t_{<i})$ is the probability of the i-th token of a CFE $\boldsymbol{c}$, given the sequence of tokens ahead.

Model Performance. As modifications in CFEs ideally adhere to domain knowledge, we can either incorporating CFEs into training to enhance model robustness (Chen et al., 2021b; Qiu et al., 2024) or leverage CFEs as test sets to evaluate existing model's generalization (Ribeiro et al., 2020; Ross et al., 2021b). Researchers then report the classification performance, such as accuracy, F1-score, and the standard deviation of these metrics on out-of-domain datasets or counterfactual test sets

The commonly used metrics are summarized in Appendix Section E. The metrics used in each paper are also listed in Appendix Section D.

5 Challenges and Future Directions

Fair evaluation. Counterfactuals are inherently speculative, making it difficult to compare CFEs from different methods due to the absence of ground truth. This challenge arises from two main aspects: (1) Existing metrics evaluate CFEs from various, often non-comparable perspectives. For example, prioritizing higher proximity (minimal changes to the original text) typically results in lower diversity. Optimizing one metric often compromises another, making it difficult to dominate across all metrics and conclusively identify the best method. (2) Many methods use filtering techniques to discard undesired results. Direct comparisons between filtered and unfiltered CFEs may introduce bias in the evaluation process. For instance, methods employing GPT-2 to filter out grammatically incorrect or nonsensical sentences (Radford et al., 2019; Ross et al., 2022) often outperform those that do not use such filters on fluency score.

Model privacy and security. Model privacy and security are crucial considerations in the development and deployment of machine learning systems. CFEs, which reveal sensitive changes near the decision boundary, can be exploited by adversaries to extract high-fidelity surrogate models (Aïvodji et al., 2020; Wang et al., 2022), posing risks to model integrity. Future research should focus on strategies to mitigate model extraction risks while maintaining the utility of CFEs.

Counterfactual multiplicity. Multiple counterfactuals can exist with similar evaluation scores. For example, replacing 'terrible' with 'good' or 'excellent' results in similar edit distances and flip rates. While current research often centres on generating CFEs, having diverse CFEs is crucial for understanding models from various perspectives (Wachter et al., 2017), enhancing fairness detection with higher test coverage (Mothilal et al., 2020), and training robust models (Joshi and He, 2022; Qiu et al., 2024). Future work should focus on selecting diverse CFEs or incorporating diversity into the objective, possibly with a cardinality constraint.

LLM-assisted CFEs. To better leverage LLMs' understanding and reasoning capabilities, an in-depth task analysis of the counterfactual problem is the premise, which could help design clear and constructive prompts. This is particularly important when we design prompts for different NLP tasks. Prompts framed in a "What-if" scenario may outperform those framed as optimization problems. On the other hand, LLMs are not without flaws and face several challenges, including bias, fairness issues, hallucinations, and difficulty in retaining long-term context. We should consider integrating debiasing techniques and fairness constraints, developing advanced memory architectures and integrating external knowledge to mitigate these issues.

6 Conclusion

In this survey, we systematically review recent advancements, including the latest LLM-assisted generation approaches. Based on algorithmic differences, we propose a novel taxonomy that categorizes these methods into four groups, providing an in-depth comparison, discussion, and summary for each group. Additionally, we summarize the commonly used metrics to evaluate the quality of counterfactuals. Lastly, we discuss research challenges and aim to inspire future directions. With the widespread use of LLMs, model explanation, fairness, and robust training have received increasing attention. We believe this survey can serve as an easy-to-follow guideline to motivate future advances that harness these problems.

7 Limitations

While this survey provides a comprehensive overview of counterfactual generation in the NLP domain, it has several limitations. Firstly, it focuses predominantly on generating CFEs while excluding other areas like counterfactual thinking or reasoning, which are also applicable to NLP tasks. Secondly, although counterfactual generation in NLP intersects with fields like causality, linguistics, and social sciences, this survey centres on NLP-specific aspects and may not fully explore these interdisciplinary connections, potentially limiting the depth of understanding in those areas. Thirdly, the survey outlines general evaluation metrics but does not include empirical evaluations or practical experiments. The diversity and task-specific nature of counterfactual generation methods in NLP make unified evaluation challenging. However, the general automatic evaluation metrics reviewed in Section 4 could facilitate comparative experiments on different generation strategies. Lastly, although this survey mentions issues like bias and fairness, it does not delve deeply into the ethical and practical implications of deploying CFEs in real-world applications. Understanding these impacts is crucial but beyond the scope of this paper.

8 Ethics Statement

We recognize the importance of ethical considerations in our research and have adhered to responsible practices throughout this study. To ensure transparency and accountability, we have listed all the papers mentioned in our survey in Appendix D, including our qualitative classifications and annotations for public scrutiny. To address potential bias in categorizing reviewed papers, five independent researchers meticulously validated the categorizations and definitions, enhancing the reliability and accuracy of our analysis. Additionally, CFEs can significantly impact decision-making in high-stakes applications, affecting fairness and accountability. Therefore, it is crucial to use CFEs responsibly in real-world deployments. Researchers and practitioners should be cautious of unintended consequences when applying these techniques. Disclosing these ethical considerations underscores our commitment to ethical and accountable research practices.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Ulrich Aïvodji, Alexandre Bolot, and Sébastien Gambs. 2020. Model extraction from counterfactual explanations. *arXiv preprint arXiv:2009.01884*.

Akari Asai and Hannaneh Hajishirzi. 2020. Logic-guided data augmentation and regularization for consistent question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5642–5650.

Ananth Balashankar, Xuezhi Wang, Yao Qin, Ben Packer, Nithum Thain, Ed Chi, Jilin Chen, and Alex Beutel. 2023. Improving classifier robustness through active generative counterfactual data augmentation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 127–139.

Lorenzo Betti, Carlo Abrate, Francesco Bonchi, and Andreas Kaltenbrunner. 2023. Relevance-based infilling for natural language counterfactuals. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 88–98.

Amrita Bhattacharjee, Raha Moraffah, Joshua Garland, and Huan Liu. 2024. Towards llm-guided causal explainability for black-box text classifiers. *arXiv preprint arXiv:2309.13340*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Nitay Calderon, Eyal Ben-David, Amir Feder, and Roi Reichart. 2022. DoCoGen: Domain counterfactual generation for low resource domain adaptation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7727–7746, Dublin, Ireland. Association for Computational Linguistics.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium. Association for Computational Linguistics.

Mingshan Chang, Min Yang, Qingshan Jiang, and Ruifeng Xu. 2024. Counterfactual-enhanced information bottleneck for aspect-based sentiment analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38(16), pages 17736–17744.

Saneem Chemmengath, Amar Prakash Azad, Ronny Luss, and Amit Dhurandhar. 2022. Let the CAT out of the bag: Contrastive attributed explanations for text. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7190–7206, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Hao Chen, Rui Xia, and Jianfei Yu. 2021a. Reinforced counterfactual data augmentation for dual sentiment classification. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 269–278, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jiangjie Chen, Chun Gan, Sijie Cheng, Hao Zhou, Yanghua Xiao, and Lei Li. 2022. Unsupervised editing for counterfactual stories. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36(10), pages 10473–10481.

Qianglong Chen, Feng Ji, Xiangji Zeng, Feng-Lin Li, Ji Zhang, Haiqing Chen, and Yin Zhang. 2021b. Kace: Generating knowledge aware contrastive explanations for natural language inference. 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 2516–2527.

Zeming Chen, Qiyue Gao, Antoine Bosselut, Ashish Sabharwal, and Kyle Richardson. 2023. Disco: distilling counterfactuals with large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5514–5528.

Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. 2018. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8789–8797.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

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 *International Conference on Learning Representations*, pages 1–34.

Michael Denkowski and Alon Lavie. 2011. Meteor 1.3: Automatic metric for reliable optimization and evaluation of machine translation systems. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 85–91, Edinburgh, Scotland. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019a. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019b. Bert: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186.

Tanay Dixit, Bhargavi Paranjape, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. CORE: A retrieve-then-edit framework for counterfactual data generation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2964–2984, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Deqing Fu, Ameya Godbole, and Robin Jia. 2023. SCENE: Self-labeled counterfactuals for extrapolating to negative examples. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7832–7848, Singapore. Association for Computational Linguistics.

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. 2020. Evaluating models' local decision boundaries via contrast sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1307–1323, Online. Association for Computational Linguistics.

Siddhant Garg and Goutham Ramakrishnan. 2020. Bae: Bert-based adversarial examples for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6174–6181.

Yair Ori Gat, Nitay Calderon, Amir Feder, Alexander Chapanin, Amit Sharma, and Roi Reichart. 2024. Faithful explanations of black-box NLP models using LLM-generated counterfactuals. In *The Twelfth*

International Conference on Learning Representations, pages 1–34.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. 2020. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673.

Mor Geva, Tomer Wolfson, and Jonathan Berant. 2022. Break, perturb, build: Automatic perturbation of reasoning paths through question decomposition. *Transactions of the Association for Computational Linguistics*, 10:111–126.

Daniel Gilo and Shaul Markovitch. 2024. A general search-based framework for generating textual counterfactual explanations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38(16), pages 18073–18081.

Riccardo Guidotti. 2022. Counterfactual explanations and how to find them: literature review and benchmarking. *Data Mining and Knowledge Discovery*, pages 1–55.

Changying Hao, Liang Pang, Yanyan Lan, Yan Wang, Jiafeng Guo, and Xueqi Cheng. 2021. Sketch and customize: A counterfactual story generator. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35(14), pages 12955–12962.

Katherine Hermann and Andrew Lampinen. 2020. What shapes feature representations? exploring datasets, architectures, and training. *Advances in Neural Information Processing Systems*, 33:9995–10006.

Pengfei Hong, Rishabh Bhardwaj, Navonil Majumder, Somak Aditya, and Soujanya Poria. 2023. A robust information-masking approach for domain counterfactual generation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3756–3769, Toronto, Canada. Association for Computational Linguistics.

Phillip Howard, Gadi Singer, Vasudev Lal, Yejin Choi, and Swabha Swayamdipta. 2022. NeuroCounterfactuals: Beyond minimal-edit counterfactuals for richer data augmentation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5056–5072, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zhiting Hu and Li Erran Li. 2021. A causal lens for controllable text generation. *In Advances in Neural Information Processing Systems*, 34:24941–24955.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward controlled generation of text. In *International conference on machine learning*, pages 1587–1596. PMLR.

F. Jelinek, R. L. Mercer, L. R. Bahl, and J. K. Baker. 2005. Perplexity—a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, 62(S1):S63–S63.

Ziwei Ji, Tiezheng Yu, Yan Xu, Nayeon Lee, Etsuko Ishii, and Pascale Fung. 2023. Towards mitigating LLM hallucination via self reflection. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1827–1843, Singapore. Association for Computational Linguistics.

Nitish Joshi and He He. 2022. An investigation of the (in)effectiveness of counterfactually augmented data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3668–3681, Dublin, Ireland. Association for Computational Linguistics.

Hong-Gyu Jung, Sin-Han Kang, Hee-Dong Kim, Dong-Ok Won, and Seong-Whan Lee. 2022. Counterfactual explanation based on gradual construction for deep networks. *Pattern Recognition*, 132:108958.

Amir-Hossein Karimi, Gilles Barthe, Bernhard Schölkopf, and Isabel Valera. 2022. A survey of algorithmic recourse: Contrastive explanations and consequential recommendations. *ACM Comput. Surv.*, 55(5).

Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. 2019. Learning the difference that makes a difference with counterfactually-augmented data. In *International Conference on Learning Representations*, pages 1–17.

Daniel Khashabi, Tushar Khot, and Ashish Sabharwal. 2020. More bang for your buck: Natural perturbation for robust question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 163–170, Online. Association for Computational Linguistics.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*, pages 1–17.

Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. 2017. Counterfactual fairness. In *Advances in Neural Information Processing Systems*, volume 30, pages 1–11. Curran Associates, Inc.

Minwoo Lee, Seungpil Won, Juae Kim, Hwanhee Lee, Cheoneum Park, and Kyomin Jung. 2021. Crossaug: A contrastive data augmentation method for debiasing fact verification models. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 3181–3185.

Vladimir I Levenshtein et al. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union.

Chuanrong Li, Lin Shengshuo, Zeyu Liu, Xinyi Wu, Xuhui Zhou, and Shane Steinert-Threlkeld. 2020a. Linguistically-informed transformations (LIT): A method for automatically generating contrast sets. In *Proceedings of the Third BlackboxNLP Workshop* *on Analyzing and Interpreting Neural Networks for NLP*, pages 126–135, Online. Association for Computational Linguistics.

Dandan Li, Ziyu Guo, Qing Liu, Li Jin, Zequn Zhang, Kaiwen Wei, and Feng Li. 2023. Click: Integrating causal inference and commonsense knowledge incorporation for counterfactual story generation. *Electronics (2079-9292)*, 12(19).

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.

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.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020b. Bert-attack: Adversarial attack against bert using bert. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6193–6202.

Yongqi Li, Mayi Xu, Xin Miao, Shen Zhou, and Tieyun Qian. 2024. Prompting large language models for counterfactual generation: An empirical study. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 13201–13221, Torino, Italy. ELRA and ICCL.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. 2021. Entity-based knowledge conflicts in question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7052–7063, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.

Aman Madaan, Amrith Setlur, Tanmay Parekh, Barnabas Poczos, Graham Neubig, Yiming Yang, Ruslan Salakhutdinov, Alan W Black, and Shrimai Prabhumoye. 2020. Politeness transfer: A tag and generate approach. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1869–1881, Online. Association for Computational Linguistics.

Nishtha Madaan, Inkit Padhi, Naveen Panwar, and Diptikalyan Saha. 2021. Generate your counterfactuals: Towards controlled counterfactual generation for text. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35(15), pages 13516–13524.

Nishtha Madaan, Diptikalyan Saha, and Srikanta Bedathur. 2023. Counterfactual sentence generation with plug-and-play perturbation. In *2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pages 306–315. IEEE.

Eric Malmi, Aliaksei Severyn, and Sascha Rothe. 2020. Unsupervised text style transfer with padded masked language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8671–8680, Online. Association for Computational Linguistics.

David Martens and Foster Provost. 2014. Explaining data-driven document classifications. *MIS quarterly*, 38(1):73–100.

Yu Meng, Jiaxin Huang, Yu Zhang, and Jiawei Han. 2022. Generating training data with language models: Towards zero-shot language understanding. *Advances in Neural Information Processing Systems*, 35:462–477.

Yu Meng, Martin Michalski, Jiaxin Huang, Yu Zhang, Tarek Abdelzaher, and Jiawei Han. 2023. Tuning language models as training data generators for augmentation-enhanced few-shot learning. In *International Conference on Machine Learning*, pages 24457–24477. PMLR.

Xin Miao, Yongqi Li, and Tieyun Qian. 2023. Generating commonsense counterfactuals for stable relation extraction. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

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

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*, FAT* '20, page 607–617, New York, NY, USA. Association for Computing Machinery.

Van Bach Nguyen, Paul Youssef, Jörg Schlötterer, and Christin Seifert. 2024. Llms for generating and evaluating counterfactuals: A comprehensive study. *arXiv preprint arXiv:2405.00722*.

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.

Bhargavi Paranjape, Matthew Lamm, and Ian Tenney. 2022. Retrieval-guided counterfactual generation for QA. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1670–1686, Dublin, Ireland. Association for Computational Linguistics.

Lianhui Qin, Antoine Bosselut, Ari Holtzman, Chandra Bhagavatula, Elizabeth Clark, and Yejin Choi. 2019. Counterfactual story reasoning and generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5043–5053.

Xiaoqi Qiu, Yongjie Wang, Xu Guo, Zhiwei Zeng, Yue Yu, Yuhong Feng, and Chunyan Miao. 2024. Paircfr: Enhancing model training on paired counterfactually augmented data through contrastive learning. *arXiv preprint arXiv:2406.01382*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Yanou Ramon, David Martens, Foster Provost, and Theodoros Evgeniou. 2020. A comparison of instance-level counterfactual explanation algorithms for behavioral and textual data: Sedc, lime-c and shap-c. *Advances in Data Analysis and Classification*, 14:801–819.

Machel Reid and Victor Zhong. 2021. LEWIS: Levenshtein editing for unsupervised text style transfer. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3932–3944, Online. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992.

Marco Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 97–101, San Diego, California. Association for Computational Linguistics.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

Marcel Robeer, Floris Bex, and Ad Feelders. 2021. Generating realistic natural language counterfactuals. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3611–3625.

Alexis Ross, Himabindu Lakkaraju, and Osbert Bastani. 2021a. Learning models for actionable recourse. *Advances in Neural Information Processing Systems*, 34:18734–18746.

Alexis Ross, Ana Marasović, and Matthew Peters. 2021b. 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.

Alexis Ross, Tongshuang Wu, Hao Peng, Matthew Peters, and Matt Gardner. 2022. Tailor: Generating and perturbing text with semantic controls. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3194–3213, Dublin, Ireland. Association for Computational Linguistics.

Chris Russell, Matt J Kusner, Joshua Loftus, and Ricardo Silva. 2017. When worlds collide: Integrating different counterfactual assumptions in fairness. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Rachneet Sachdeva, Martin Tutek, and Iryna Gurevych. 2024. CATfOOD: Counterfactual augmented training for improving out-of-domain performance and calibration. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1876–1898, St. Julian's, Malta. Association for Computational Linguistics.

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. Masked language model scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2699–2712, Online. Association for Computational Linguistics.

Mattia Samory, Indira Sen, Julian Kohne, Fabian Flöck, and Claudia Wagner. 2021. "call me sexist, but...": Revisiting sexism detection using psychological scales and adversarial samples. In *Proceedings of the international AAAI conference on web and social media*, volume 15(1), pages 573–584.

Aalok Sathe, Salar Ather, Tuan Manh Le, Nathan Perry, and Joonsuk Park. 2020. Automated fact-checking of claims from Wikipedia. In *Proceedings of the*

Twelfth Language Resources and Evaluation Conference, pages 6874–6882, Marseille, France. European Language Resources Association.

Indira Sen, Dennis Assenmacher, Mattia Samory, Isabelle Augenstein, Wil Aalst, and Claudia Wagner. 2023. People make better edits: Measuring the efficacy of LLM-generated counterfactually augmented data for harmful language detection. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10480–10504, Singapore. Association for Computational Linguistics.

Indira Sen, Mattia Samory, Fabian Flöck, Claudia Wagner, and Isabelle Augenstein. 2021. How does counterfactually augmented data impact models for social computing constructs? In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 325–344, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Lei Sha, Patrick Hohenecker, and Thomas Lukasiewicz. 2021. Controlling text edition by changing answers of specific questions. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1288–1299, Online. Association for Computational Linguistics.

Jiasheng Si, Yingjie Zhu, and Deyu Zhou. 2023. Consistent multi-granular rationale extraction for explainable multi-hop fact verification. *arXiv preprint arXiv:2305.09400*.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*, pages 1–8.

Ilia Stepin, Jose M Alonso, Alejandro Catala, and Martín Pereira-Fariña. 2021. A survey of contrastive and counterfactual explanation generation methods for explainable artificial intelligence. *IEEE Access*, 9:11974–12001.

Akhilesh Sudhakar, Bhargav Upadhyay, and Arjun Maheswaran. 2019a. "transforming" delete, retrieve, generate approach for controlled text style transfer. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3269–3279, Hong Kong, China. Association for Computational Linguistics.

Akhilesh Sudhakar, Bhargav Upadhyay, and Arjun Maheswaran. 2019b. "transforming" delete, retrieve, generate approach for controlled text style transfer. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3269–3279.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328. PMLR.

Marcos Treviso, Alexis Ross, Nuno M. Guerreiro, and André Martins. 2023. CREST: A joint framework for rationalization and counterfactual text generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15109–15126, Toronto, Canada. Association for Computational Linguistics.

Keyon Vafa, Ashesh Rambachan, and Sendhil Mullainathan. 2024. Do large language models perform the way people expect? measuring the human generalization function. *arXiv preprint arXiv:2406.01382*.

Sahil Verma, Varich Boonsanong, Minh Hoang, Keegan E Hines, John P Dickerson, and Chirag Shah. 2020. Counterfactual explanations and algorithmic recourses for machine learning: A review. *arXiv preprint arXiv:2010.10596*.

Sandra Wachter, Brent Mittelstadt, and Chris Russell. 2017. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841.

Yongjie Wang, Hangwei Qian, and Chunyan Miao. 2022. Dualcf: Efficient model extraction attack from counterfactual explanations. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '22, page 1318–1329, New York, NY, USA. Association for Computing Machinery.

Zhao Wang and Aron Culotta. 2021. Robustness to spurious correlations in text classification via automatically generated counterfactuals. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(16):14024–14031.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Jiaxin Wen, Yeshuang Zhu, Jinchao Zhang, Jie Zhou, and Minlie Huang. 2022. Autocad: Automatically generate counterfactuals for mitigating shortcut learning. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2302–2317.

Dongming Wu, Lulu Wen, Chao Chen, and Zhaoshu Shi. 2024. A novel counterfactual data augmentation method for aspect-based sentiment analysis. In *Asian Conference on Machine Learning*, pages 1479–1493. PMLR.

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.

Xing Wu, Tao Zhang, Liangjun Zang, Jizhong Han, and Songlin Hu. 2019. Mask and infill: Applying masked language model for sentiment transfer. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5271–5277. International Joint Conferences on Artificial Intelligence Organization.

Hanqi Yan, Lingjing Kong, Lin Gui, Yuejie Chi, Eric Xing, Yulan He, and Kun Zhang. 2024. Counterfactual generation with identifiability guarantees. *Advances in Neural Information Processing Systems*, 36.

Linyi Yang, Eoin Kenny, Tin Lok James Ng, Yi Yang, Barry Smyth, and Ruihai Dong. 2020. Generating plausible counterfactual explanations for deep transformers in financial text classification. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6150–6160, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Linyi Yang, Jiazheng Li, Padraig Cunningham, Yue Zhang, Barry Smyth, and Ruihai Dong. 2021. Exploring the efficacy of automatically generated counterfactuals for sentiment analysis. 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 306–316, Online. Association for Computational Linguistics.

Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. 2022. ZeroGen: Efficient zero-shot learning via dataset generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11653–11669, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yue Yu, Yuchen Zhuang, Jieyu Zhang, Yu Meng, Alexander J Ratner, Ranjay Krishna, Jiaming Shen, and Chao Zhang. 2024. Large language model as attributed training data generator: A tale of diversity and bias. *Advances in Neural Information Processing Systems*, 36.

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

Mi Zhang, Tieyun Qian, Ting Zhang, and Xin Miao. 2023. Towards model robustness: Generating contextual counterfactuals for entities in relation extraction. In *Proceedings of the ACM Web Conference 2023*, pages 1832–1842.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*, pages 1–43.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. 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 1298–1308, Minneapolis, Minnesota. Association for Computational Linguistics.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. 2019. MoverScore: Text generation evaluating with contextualized embeddings and earth mover distance. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 563–578, Hong Kong, China. Association for Computational Linguistics.

Yuhang Zhou, Paiheng Xu, Xiaoyu Liu, Bang An, Wei Ai, and Furong Huang. 2023. Explore spurious correlations at the concept level in language models for text classification. *arXiv preprint arXiv:2311.08648*.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 1097–1100.

Yingjie Zhu, Jiasheng Si, Yibo Zhao, Haiyang Zhu, Deyu Zhou, and Yulan He. 2023. EXPLAIN, EDIT, GENERATE: Rationale-sensitive counterfactual data augmentation for multi-hop fact verification. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13377–13392, Singapore. Association for Computational Linguistics.

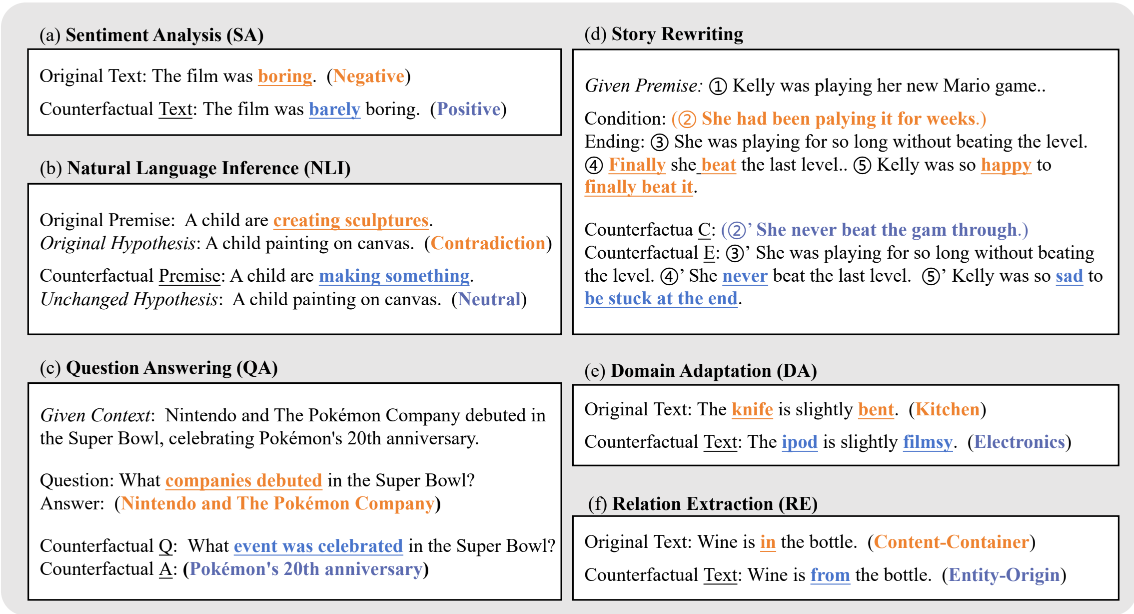

Figure 3: Use cases of counterfactuals in different NLP tasks.

A Terminology Clarification

In this section, we clarify terms related to Counterfactual Examples (CFEs) to ensure a precise review scope.

Adversarial Example v.s. CFE. Both text adversarial examples (Li et al., 2020b; Garg and Ramakrishnan, 2020) and CFEs aim to change model predictions with minimal modifications. However, adversarial examples are designed to deceive human perception, altering only the model's prediction without necessarily being human-perceivable as different. In contrast, CFEs should ideally change both human and model predictions simultaneously.

Style Transfer v.s. CFE. Style transfer (Sudhakar et al., 2019a; Hu et al., 2017) aims to revise the input sentence to achieve a target style. Unlike CFE generation, which sought for minimal perturbations, style transfer may involve complete sentence modifications to ensure the sentence conforms to the target style. However, when minimal perturbation is also required in some style transfer research, we treat both tasks the same and include these studies.

B CFE Generation in NLP Tasks

Here, we present the formulation of CFE generation across various NLP tasks. Figure 3 illustrates examples of corresponding CFE in different tasks, and Figure 4 reports the proportion of papers in each task relative to all reviewed papers.

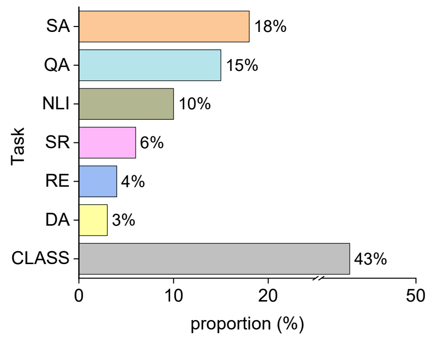

Figure 4: Proportion of papers in each task among all reviewed papers. The term 'CLASS' refers to papers applicable to general text classification tasks, including SA and NLI.

Sentiment Analysis (SA) involves determining the emotional polarity y given a text $\boldsymbol{x}$. Counterfactual generation in SA refers to minimally modify the input text $\boldsymbol{x}$ such that the new sentence $\boldsymbol{c}$ has a different prediction y', i.e., $(\boldsymbol{x}, y) \rightarrow (\boldsymbol{c}, y')$.

Natural Language Inference (NLI) is to determine whether a given hypothesis $\boldsymbol{x}_1$ can be inferred from a given premise $\boldsymbol{x}_2$, and return a logical relationship y. CFE generation in NLI aim to revise hypothesis or premise or both to change current logical relationship y to another different relationship y', i.e., $(\boldsymbol{x}_1, \boldsymbol{x}_1, y) \rightarrow (\boldsymbol{c}_1, \boldsymbol{c}_2, y')$.

Question Answering (QA) aims to automatically produce an answer $\boldsymbol{a}$ for a given question $\boldsymbol{q}$ and context $\boldsymbol{x}$. The counterfactual QA task seeks to minimally modifies either the context or the question, or both to generate counterfactual context $\boldsymbol{c}_x$

or question c_q such that (c_x, c_q, a') holds for a new answer a', i.e., $(x, q, a) \rightarrow (c_x, c_q, a')$.

Story Rewriting (SR). The example in SR task includes a 5 sentence tuple $\{s_1, s_2, s_3, s_4, s_5\}$ where s_1 is the story premise, s_2 is the initial context, and s_{3-5} are original story endings. Given a contrastive context s_2', counterfactual SR aims to minimally revise the original endings, such that the revised endings s_{3-5}' still keep narrative coherency to the new context and original premise.

Domain Adaptation (DA). Given a sentence x that belongs to the source domain d_s, counterfactual DA aims to minimally intervene the original sentence such that the edited sentence c belongs to a different target domain d_t.

Relation Extraction (RE) involves extracting the relationship r between entities in a given sentence x. In counterfactual RE, we aim to minimally revise the x such that a different relationship r' can be obtained between these entities from the revised sentence c.

C Paper Collection

This section outlines the approach we employed to collect relevant papers in this survey. We first retrieve papers from arXiv and Google Scholar with keywords "counterfactually augmented data", "counterfactual explanation", "contrast set", and "knowledge conflict", and finally we obtain over 200 publications. We then filter out papers that merely apply CFE on specific applications or generally discuss CFE, retaining approximately 40 papers as our seed references. We then applied backward and forward snowballing techniques, examining the references and citations of these seed papers to identify additional relevant studies. We carefully reviewed all identified papers, focusing on those introducing novel counterfactual generation methods, which finally form this survey.

Our research paper list is available on GitHub[1].

D Summary of CFE Generation

In this section, we summarize all collected papers for each group in Section 3. Due to the distinct characteristics of different method groups, we have organized them into four separate tables, rather than merging them into one large table. For methods within a table, we can conveniently understand a method or compare it with another. The detailed summary are shown in Table 2, Table 3, Table 4, and Table 5.

Table 1: Commonly used metrics for evaluating counterfactuals, where ↑ (↓) indicates higher (lower) values are better, and (→ 1) indicates closer to 1 is better.

Property		Metric	Trend
Validity		Flip Rate	↑
Proximity	Lexical	BLEU (Papineni et al., 2002)	↑
		ROUGE (Lin, 2004)	↑
		METEOR (Denkowski and Lavie, 2011)	↑
		Levenshtein Dist. (Levenshtein et al., 1966)	↓
		Syntax Tree Dist. (Zhang and Shasha, 1989)	↓
	Semantic	MoverScore (Zhao et al., 2019)	↑
		USE Sim. (Cer et al., 2018)	↑
		SBERT Sim. (Reimers and Gurevych, 2019)	↑
Diversity	Lexical	Self-BLEU (Zhu et al., 2018)	↓
		Distinct-n (Li et al., 2016)	↑
		Levenshtein Dist. (Levenshtein et al., 1966)	↑
	Semantic	SBERT sim. (Reimers and Gurevych, 2019)	↓
		BERTScore (Zhang et al., 2020)	↓
Fluency		Likelihood Rate (Salazar et al., 2020)	(→ 1)
		Perplexity Score (Radford et al., 2019)	↓
Model Performance		Accuracy / F1-Score	↑
		Std of accuracy / F1-score in multiple runs	↓

E Summary of Evaluation Metrics

The evaluation metrics for comparing different CFEs are summarized in Table 1. Here, we only list metrics that have been used in at least three publications.

[1] https://anonymous.4open.science/r/Awesome-CF-Generation-70FF

Table 2: Summary of CFE generation based on manual annotation.

Method	Task: Dataset	Annotators	Project Link
(Kaushik et al., 2019)	SA: IMDB; NLI: SNLI	Crowd worker	`https://github.com/acmi-lab/counterfactually-augmented-data`
(Qin et al., 2019)	SR: TIMETRAVEL	Crowd worker	`https://github.com/qkaren/Counterfactual-StoryRW`
(Khashabi et al., 2020)	QA: BOOLQ	Master worker	`https://github.com/allenai/natural-perturbations`
(Gardner et al., 2020)	SA: IMDB; NLI: PERSPECTRUM; QA: DROP,QUOREF,ROPES, MC-TACO, BOOLQ; RE: MATRES	Domain expert	`https://allennlp.org/contrast-sets`
(Sathe et al., 2020)	NLI: WIKIFACTCHECK	Crowd worker	`http://github.com/WikiFactCheck-English`
(Samory et al., 2021)	Sexism: CMSB	Crowd worker	`https://doi.org/10.7802/2251`
(Sha et al., 2021)	QA: WIKIBiOCTE	linguistics	`https://sites.google.com/view/control-text-edition/home`

Table 3: Summary of CFE generation based on gradient-based optimization. ‘MP’ means model performance. For the unique formula in validity evaluation, we list the models applied. Symbols ✗ and ✓ depict “not included” and “included” respectively. Papers are organized chronologically.

Method	Task	Solution		Evaluation				
		Objectives	Filter	Validity	Diversity	Proximity	Fluency	MP
GYC (Madaan et al., 2021)	CLASS	Val.+Pro.+ Div.	✗	XL-Net	BERTScore ↓	Syntax Dist. ↓ SBERT Sim. ↑	Human	✓
CounterfactualGAN (Robeer et al., 2021)	CLASS	Val.+Pro.	Val.	BERT	1-USE ↑	✗	Human	✗
Hu and Li (2021)	CLASS	Val.+Pro.+Flu.	✗	GPT-2 Human	Distinct-2 ↑	BLEU ↑	GPT-2 Perplexity ↓	✗
GradualCAD (Jung et al., 2022)	CLASS	Val.+Pro.	✗	✗	✗	✗	✗	✓
CASPer (Madaan et al., 2023)	CLASS	Val.+Flu.+Pro.	✗	✗	BLEU ↓	SBERT Sim. ↑	GPT-2 Perplexity ↓	✓
MATTE (Yan et al., 2024)	SA	Val.+Pro.+Flu.	✗	CNN	Diversity-2 ↑	BLEU ↑ Human	GPT-2 Perplexity ↓ Human	✓

Table 4: Summary of CGE generation based on LLM prompting. ‘MP’ represents model performance, and for the unique formula in validity evaluation, we list the models applied. Symbols ✗ and ✓ depict “not included” and “included” respectively. Papers are listed chronologically.

Method	Task	Solution		Evaluation				
		Prompting	Filter	Validity	Diversity	Proximity	Fluency	MP
CORE (Dixit et al., 2022)	CLASS	ICL	✗	Human	Self-BLEU ↓ #Perturb Type ↑	Levenshtein ↓	✗	✓
DISCO (Chen et al., 2023)	CLASS	ICL	Val.+Flu.	Human	Self-BLEU ↓ OTDD ↑	✗	✗	✓
(Zhou et al., 2023)	CLASS	ICL	✗	✗	✗	✗	✗	✓
(Sachdeva et al., 2024)	QA	ICL + CoT	Val.	FLAN-UL2 + GPT-J + GPTNeoX + LLaMA	Self-BLEU ↓ Levenshtein ↑ SBERT Sim. ↓ Semantic Equ.↓	✗	✗	✓
(Gat et al., 2024)	CLASS	ICL	Val.	✗	✗	✗	✗	✓
(Nguyen et al., 2024)	CLASS	ICL+CoT	✗	BERT	✗	Levenshtein ↓	GPT-2 Perplexity ↓	✓
(Li et al., 2024)	CLASS	CoT	Val.+Flu.	✗	✗	✗	✗	✓
(Bhattacharjee et al., 2024)	CLASS	CoT	✗	DistilBERT	✗	Levenshtein ↓ USE ↑	✗	✗

Table 5: Summary of CFE generation within Identify-and-Generate framework. "W.I." means word importance techniques, "W.S." is the word statistic techniques, and "ALL" is to leverage all words of a text. Papers are listed chronologically.

Method	Task	Solution			Evaluation				
		Identify	Generate	Filter	Validity	Diversity	Proximity	Fluency	MP
SEDC (Martens and Provost, 2014)	CLASS	W.I.	Delete	✗	SVM	✗	#Delete Word ↓	✗	✗
DeleteAndRetrieve (Li et al., 2018)	CLASS	W.S.	Retrieve Semantic Edit Open Infilling	Flu.	Bi-LSTM Human	✗	BLEU ↑ Human	Human	✗
AC-MLM (Wu et al., 2019)	SA	W.S.+W.I.	MLM Infilling	✗	Bi-LSTM Human	✗	BLEU ↑	Human	✗
PAWS (Zhang et al., 2019)	NLI	Parser	MLM Infilling	Val.	Human	✗	✗	Human	✓
Tag-and-Generate (Madaan et al., 2020)	SA	W.S.	MLM Infilling	✗	AWD-LSTM Human	✗	BLEU ↑ ROUGE ↑ METEOR ↑ Human	Human	✗
MASKER (Malmi et al., 2020)	CLASS	W.I.	MLM Infilling	✗	BERT	✗	BLEU ↑	✗	✗
LIT (Li et al., 2020a)	NLI	Parser	Syntax Edit	Flu.	Human	✗	✗	Human	✓
CheckList (Ribeiro et al., 2020)	CLASS	Parser	MLM Infilling Semantic Edit	✗	✗	✗	✗	✗	✓
REP-SCD (Yang et al., 2020)	CLASS	W.I.	MLM Infilling	✗	✗	✗	✗	Human	✓
(Ramon et al., 2020)	CLASS	W.I.	Delete	✗	SVM	✗	#Delete Word ↓	✗	✗
(Asai and Hajishirzi, 2020)	QA	Parser	Semantic Edit	Val.	✗	✗	✗	✗	✓
LEWIS (Reid and Zhong, 2021)	SA	W.I.	MLM Infilling	Val.	RoBERTa Human	✗	BLEU ↑ BERTScore ↑ Human	Human	✓
Polyjuice (Wu et al., 2021)	CLASS	Parser	MLM Infilling	Flu.	Human	Self-BLEU ↓	Levenshtein ↓ Syntax Dist. ↓	Human	✓
MiCE (Ross et al., 2021b)	CLASS	W.I.	MLM Infilling	✗	RoBERTa	✗	Levenshtein ↓	T5 Likelihood	✗
(Wang and Culotta, 2021)	SA	W.I.	Semantic Edit	✗	✗	✗	✗	✗	✓
CrossAug (Lee et al., 2021)	NLI	W.I.	Open Infilling +Syntactic Edit	✗	✗	✗	✗	✗	✓
SentimentCAD (Yang et al., 2021)	SA	W.I.	MLM Infilling	Pro.	✗	✗	✗	✗	✓
(Longpre et al., 2021)	QA	Parser	Syntactic Edit	✗	Human	✗	✗	Human	✓
SMG (Sha et al., 2021)	QA	W.I.	MLM Infilling	✗	Human	✗	BLEU ↑	KNM Perplexity ↓ Human	✗
KACE (Chen et al., 2021b)	NLI	W.I.	Semantic Edit	Val.+Pro. +Div.	Human	✗	Human	✗	✓
RCAD (Chen et al., 2021a)	SA	Parser	Semantic Edit	✗	✗	Distinct-2 ↑	✗	✗	✓
PARE (Ross et al., 2021a)	CLASS	Parser	Semantic Edit	✗	✗	✗	✗	✗	✓
CLOSS (Fern and Pope, 2021)	CLASS	ALL	Heuristic Search	✗	RoBERTa BERT	✗	BLEU ↑ Edit Fraction ↓	GPT-J Perplexity ↓	✗
Sketch-and-Customize (Hao et al., 2021)	SR	W.I.	MLM Infilling	✗	Human	✗	BLEU ↑ ROUGE-L ↑ Human	✗	✗
Tailor (Ross et al., 2022)	CLASS	Parser	MLM Infilling	Flu.	Human	Edit Fraction ↑	F1 Score ↓	GPT-2 Likelihood Human	✓
RGF (Paranjape et al., 2022)	QA	✗	Retrieved Context + Open Infilling	Val. + Pro.	T5 Human	#Edit Type ↑	Levenshtein ↓	Human	✓
BPB (Geva et al., 2022)	QA	Parser	Syntactic Edit Open Infilling	✗	Human	✗	✗	✗	✓
AutoCAD (Wen et al., 2022)	CLASS	W.I.	MLM Infilling	Val.	RoBERTa	Distinct-n ↑	✗	✗	✓
CAT (Chemmengath et al., 2022)	CLASS	W.I.	MLM Infilling	Val.+Div. +Flu.+Pro.	RoBERTa Human	✗	Levenshtein ↓ BERTScore ↑	GPT-2 Likelihood	✗
NeuroCFs (Howard et al., 2022)	SA	Parser	Open Infilling	✗	✗	Distinct-n ↑	Levenshtein ↓ BLEU-2 ↑ MoverScore ↑	GPT-J Perplexity ↓	✓
DoCoGen (Calderon et al., 2022)	DA	W.S.	MLM Infilling	Val.+Pro.	Human	✗	Human	Human	✓
EDUCAT (Chen et al., 2022)	SR	W.I.	MLM Infilling	✗	RoBERTa Human	✗	BLEU ↑ BERTScore ↑ Human	✗	✗
RACE (Zhu et al., 2023)	NLI	W.I.	Syntactic Edit +Open Infilling	Val.+Pro.	RoBERTa Human	1/BLEU ↑ Human	MoverScore ↑ Human	GPT-2 Perplexity ↓ Human	✓
RELITC (Betti et al., 2023)	CLASS	W.I.	MLM Infilling	✗	RoBERTa	✗	Levenshtein ↓ BLEU ↑ SBERT Sim ↑ Mask Fraction↓	GPT-2 Likelihood	✓
CREST (Treviso et al., 2023)	CLASS	W.I.	MLM Infilling	✗	RoBERTa Human	Self-BLEU ↓	Levenshtein ↓	GPT-2 Perplexity ↓ human	✓
CoCo (Zhang et al., 2023)	RE	Parser	Syntax Edit	Val.	PA-LSTM AGGCN R-BERT	✗	✗	✗	✓
SCENE (Fu et al., 2023)	QA	Random	MLM Infilling	Val.	✗	✗	✗	✗	✓
CCG (Miao et al., 2023)	RE	W.I.+Parser	MLM Infilling	Flu.+Val.	Human	✗	Human	*Grammarly* Tool	✓
Remask (Hong et al., 2023)	DA	W.S.+W.I.	MLM Infilling	✗	Human	✗	Human	Human	✓
CLICK (Li et al., 2023)	SR	W.I.	MLM Infilling	✗	RoBERTa	✗	BLEU ↑ BERTScore ↑	✗	✗
TCE-Search (Gilo and Markovitch, 2024)	CLASS	W.I.	Heuristic Search	Flu.	RoBERTa Human	✗	Levenshtein ↓ Syntax Dist. ↓ SBERT Sim. ↑	GPT-2 Likelihood Human	✗
(Wu et al., 2024)	SA	W.I.	MLM Infilling	Val.	✗	✗	✗	✗	✓
CEIB (Chang et al., 2024)	SA	Random	MLM Infilling	Val.	✗	✗	✗	✗	✓

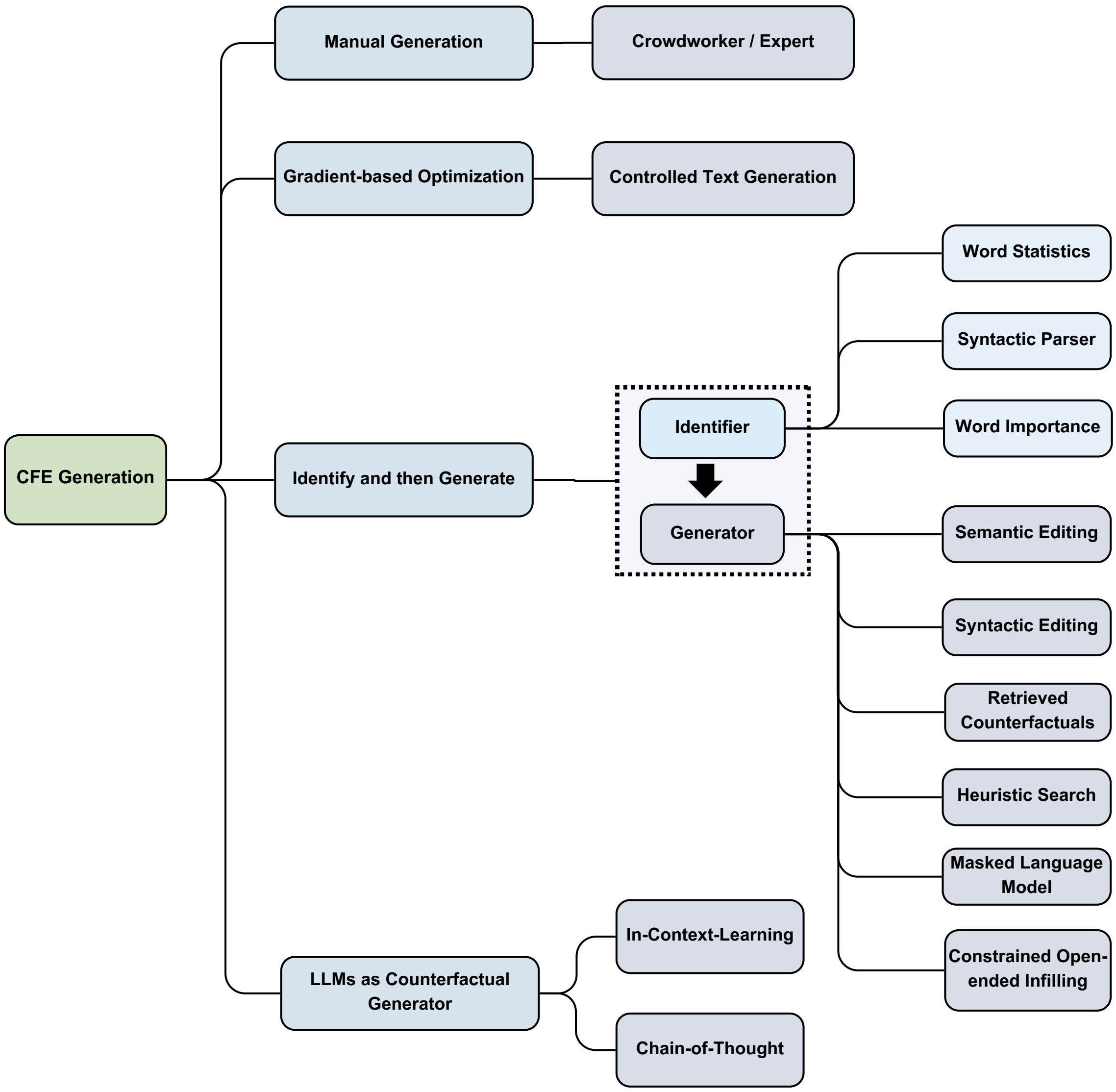

Figure 5: The complete taxonomy proposed for existing literature on natural language counterfactual generation.