CORE: A Retrieve-then-Edit Framework for Counterfactual Data Generation

Tanay Dixit[1] **Bhargavi Paranjape**[2] **Hannaneh Hajishirzi**[2,3] **Luke Zettlemoyer**[2,4]
[1] Indian Institute of Technology, Madras
[2] Paul G. Allen School of Computer Science & Engineering, University of Washington
[3] Allen Institute of Artificial Intelligence, Seattle [4] Meta AI
tanay.dixit@smail.iitm.ac.in
{bparan,hannaneh,lsz}@cs.washington.edu

Abstract

Counterfactual data augmentation (CDA) – i.e., adding minimally perturbed inputs during training – helps reduce model reliance on spurious correlations and improves generalization to out-of-distribution (OOD) data. Prior work on generating counterfactuals only considered restricted classes of perturbations, limiting their effectiveness. We present **CO**unterfactual Generation via **R**etrieval and **E**diting (**CORE**), a retrieval-augmented generation framework for creating *diverse* counterfactual perturbations for CDA. For each training example, CORE first performs a dense retrieval over a task-related unlabeled text corpus using a learned bi-encoder and extracts relevant counterfactual excerpts. CORE then incorporates these into prompts to a large language model with few-shot learning capabilities, for counterfactual editing. Conditioning language model edits on naturally occurring data results in diverse perturbations. Experiments on natural language inference and sentiment analysis benchmarks show that CORE counterfactuals are more effective at improving generalization to OOD data compared to other DA approaches. We also show that the CORE retrieval framework can be used to encourage diversity in manually authored perturbations [1].

1 Introduction

Contrast sets (Gardner et al., 2020) and counterfactual data (Kaushik et al., 2020) provide minimal input perturbations that change model predictions, and serve as an effective means to evaluate brittleness to out-of-distribution data (Wang et al., 2021). Counterfactual data augmentation (CDA) has shown to improve model robustness to OOD data and input perturbations (Geva et al., 2021; Wu et al., 2021; Paranjape et al., 2022; Khashabi et al., 2020). Alternate methods like debiasing data (Wu et al., 2022) have also shown promising results on improving model robustness, but in this work we focus on CDA strategies. Recently, Joshi and He (2022) find that diversity in the set of perturbations of different predictive features is key to the effectiveness of CDA (see Figure 1). In this paper, we introduce **CO**unterfactual Generation via **R**etrieval and **E**diting (**CORE**) – retrieval augmented generation framework for creating *diverse* counterfactual perturbations. CORE combines dense retrieval with the few-shot learning capabilities of large language models, while using minimal supervision about perturbation type.

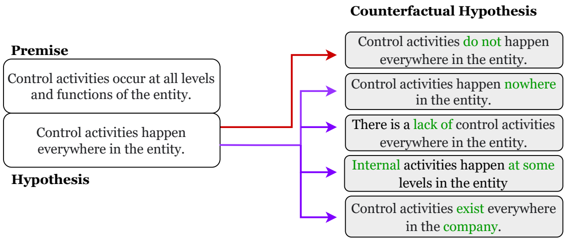

Counterfactual Hypothesis
Premise
Control activities occur at all levels and functions of the entity.
Control activities happen everywhere in the entity.
Hypothesis
Control activities do not happen everywhere in the entity.
Control activities happen nowhere in the entity.
There is a lack of control activities everywhere in the entity.
Internal activities happen at some levels in the entity
Control activities exist everywhere in the company.

Figure 1: Diverse counterfactuals are generated for an MNLI example. The red arrow represents the most trivial way of generating a counterfactual hypothesis, while the violet arrows depict several other perturbations that intervene on different predictive features.

Retrieval-augmented models (Guu et al., 2020; Lewis et al., 2020) learn to search over a dense index of a text corpus to condition generation on retrieved texts and are especially effective at improving the diversity of generated text for paraphrase generation (Chen et al., 2019) and style-transfer (Xiao et al., 2021). CORE uses this insight by learning to retrieve *counterfactual* excerpts from a large text corpus. Arbitrarily conditioning on these retrieved text excerpts to generate a rich set of counterfactual perturbations, without explicit supervision, can be challenging (Qin et al., 2022). Instead, CORE uses few-shot prompting of massive pretrained language models, which is found to be effective at controlled generation tasks like arbitrary style-transfer (Reif et al., 2022). CORE

[1] Code at https://github.com/tanay2001/CORE

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 2964–2984
December 7-11, 2022 ©2022 Association for Computational Linguistics

prompts GPT-3 (Brown et al., 2020; Wei et al., 2022) with *a few* demonstrations of using these excerpts for counterfactual editing.

The CORE retriever is a transformer-based bi-encoder model trained using contrastive learning (Le-Khac et al., 2020) on a small set of human-authored counterfactuals for the task. For each training example, CORE retrieves excerpts from an unlabeled task-related corpus that bear a label-flipping counterfactual relationship with the original input instance. Retrieval may extract excerpts that have significant semantic drift from input text, while still containing relevant counterfactual phrases (Table 1). Using prompts, the CORE GPT-3 editor generates counterfactual edits to the input conditioned on the retrieved excerpts (and the original inputs). The prompts consist of instructions and a few demonstrations of using the retrieved text for editing. Unlike prior work that use rule-based (Ribeiro et al., 2020) or semantic frameworks (Wu et al., 2021; Ross et al., 2022) and restrict perturbation types, CORE uses naturally occurring data to encourage perturbation diversity.

Intrinsic evaluation of CORE counterfactuals demonstrates a rich set of perturbation types which existing methods like Wu et al. (2021) generate (Table 7) and new perturbation types (Table 5) with more diverse outputs (Table 6), without explicit supervision. Our extensive data augmentation experiments and analyses show that the combination of retrieval and few-shot editing generates data for CAD that is effective in reducing model biases and improves performance on out of distribution (OOD) and challenge test sets. Perturbing only 3% and 7% of the data for NLI and Sentiment analysis respectively, we achieve improvements up to 4.5% and 6.2% over standard DA (Tables 2,3). Additionally, we show that CORE's learned retriever can assist humans in generating more diverse counterfactuals, spurring their creativity and reducing priming effects (Gardner et al., 2021).

2 Related Work

Counterfactual Data Augmentation There is growing interest in the area of CDA for model robustness, with early efforts focused on human-authored counterfactuals (Kaushik et al., 2020; Gardner et al., 2020). However, manual rewrites can be costly and prone to systematic omissions. Techniques have been proposed for the automatic generation of counterfactual data or contrast sets (Wu et al., 2021; Ross et al., 2022, 2021; Bitton et al., 2021; Asai and Hajishirzi, 2020; Geva et al., 2021; Madaan et al., 2021; Li et al., 2020). Existing techniques rely on using rules/heuristics for perturbing sentences (Webster et al., 2020; Dua et al., 2021; Ribeiro et al., 2020; Asai and Hajishirzi, 2020), or using sentence-level semantic representations (eg. SRL) and a finite set of structured control codes (Geva et al., 2021; Ross et al., 2022; Wu et al., 2021). However, Joshi and He (2022) find that a limited set of perturbation types further exacerbates biases, resulting in poor generalization to unseen perturbation types. Generally, creating an assorted set of *instance-specific* perturbations is challenging, often requiring external knowledge (Paranjape et al., 2022).

Retrieval Augmented Generation Retrieving task-relevant knowledge from a large corpus of unstructured and unlabeled text has proven to be very effective for knowledge-intensive language generation tasks like question answering (Lewis et al., 2020), machine translation (Gu et al., 2018) and dialogue generation (Weston et al., 2018). Retrieval has also been used for paraphrase generation (Chen et al., 2019) and style-transfer (Xiao et al., 2021) to specifically address the lack of diversity in generations from pretrained language models. In a similar vein, CORE uses learned retrieval for counterfactual generation. While Paranjape et al. (2022) use off-the-shelf retrieval models to generate counterfactuals for QA, learning to retrieve counterfactuals is non-trivial for problems other than QA. CORE provides a recipe to train retrieval for general tasks.

In-context learning Massive language models like GPT-3 have been found to be effective at controlled generation tasks like arbitrary style-transfer (Reif et al., 2022), counterfactual reasoning (Frohberg and Binder, 2022), step-wise reasoning for complex problems (Wei et al., 2022; Zhou et al., 2022), and dataset generation (Liu et al., 2022), by learning *in-context* from few-shot demonstrations and natural language instructions (Wei et al., 2021). While GPT-3 has been used for data augmentation, it has not been used for counterfactual generation, which is fundamentally different in nature.

3 Method

A high level overview of CORE is shown in Figure 2. The first stage (§3.1) retrieves counterfactual excerpts from a large unlabeled corpus related to the target task. In the second stage (§3.2), retrieved

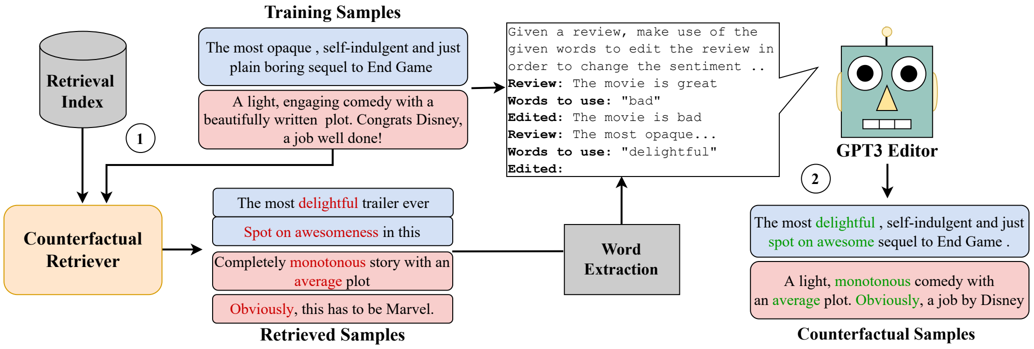

Training Samples
The most opaque , self-indulgent and just plain boring sequel to End Game
A light, engaging comedy with a beautifully written plot. Congrats Disney, a job well done!
Retrieval Index
1
Counterfactual Retriever
The most delightful trailer ever
Spot on awesomeness in this
Completely monotonous story with an average plot
Obviously, this has to be Marvel.
Retrieved Samples
Given a review, make use of the given words to edit the review in order to change the sentiment ..
Review: The movie is great
Words to use: "bad"
Edited: The movie is bad
Review: The most opaque...
Words to use: "delightful"
Edited:
Word Extraction
GPT3 Editor
2
The most delightful , self-indulgent and just spot on awesome sequel to End Game .
A light, monotonous comedy with an average plot. Obviously, a job by Disney
Counterfactual Samples

Figure 2: Overview of CORE: **CO**unterfactual **R**etrieval **E**diting framework. With the help of the ① trained counterfactual retriever we retrieve text excerpts from a large text corpus. These text excerpts are passed through a simple word extraction module that extracts all the non stopwords, which are then used by ② the Editor to edit the given training instances to generate minimally edited label flipped instances.

excerpts are supplied, along with instructions and demonstrations, as a language-modeling prompt to GPT-3 to counterfactually edit input text. The resultant data is used for augmentation in §5.1.2. We describe each stage below; additional implementation details are provided in Appendix A.

3.1 CF-DPR: Counterfactual Dense Passage Retriever

Our counterfactual retriever is based on the dense passage retrieval (DPR) framework (Karpukhin et al., 2020). CF-DPR retrieves similar instances from a large unlabeled corpus that have different labels. Formally, given a training set, $N(x) = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, for a text classification task and a large corpus S, CF-DPR retrieves samples $C(x) = \{\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n\}$ from S such that the associated labels for samples in $C(x)$ are not the same as the *corresponding* labels in $N(x)$. Specifically, $\hat{y}_i \neq y_i \forall i \in (0, n)$, where $\hat{y}_i$ is the class label for retrieved sample $\hat{x}_i$. In Figure 2, for the input “The most opaque , self-indulgent and just plain boring sequel to End Game.”, CF-DPR retrieves the excerpts “The most delightful trailer ever" and “Spot on awesomeness in this”.

Training We use the same contrastive learning objective as Karpukhin et al. (2020) to train the bi-encoder model. It consists of two independent BERT (Devlin et al., 2018) encoders: a query encoder P that encodes x_i in $N(x)$ as p_i and a document encoder Q that encodes text excerpts in S as q_i. To train the bi-encoder, we use a small seed training dataset $[q_i, p_i^+, p_i^-]_{i=1}^m$ of size m containing $m < |N(x)|$ positive and negative retrieval samples. For a given training instance q_i, we use its corresponding positive sample p_i^+ and hard negatives p_i^-'s to optimize the following loss function.

$$
\begin{aligned}
&L(\{q_i, p_i^+, p_{i,1}^-, ..p_{i,n}^-\}_{i=1}^m) \\
&= -\log \frac{e^{sim(q_i,p_i^+)}}{e^{sim(q_i,p_i^+)} + \sum_{j=1}^n e^{sim(q_i,p_{i,j}^-)}}
\end{aligned}
\tag{1}
$$

To model the task of counterfactual retrieval, for each training instance $x_i = q_i$, we use the corresponding counterfactual instance as the positive sample (p_i^+) and use paraphrases of q_i as the hard negative (p_i^-). Positive samples can be obtained from a seed dataset consisting of manually authored counterfactuals for existing NLU datasets like IMDb, and SNLI (Kaushik et al., 2020; Gardner et al., 2020). This manual data is of the form $T = \{(q_1, p_1^+), (q_2, p_2^+), \ldots, (q_m, p_m^+)\}$. We make use of the diverse paraphraser model (Krishna et al., 2020) that generates paraphrases as hard negatives for $\{q_1, q_2, \ldots, q_m\}$, $\{p_1^-, p_2^-, \ldots, p_m^-\}$. Contrastive training pulls counterfactual samples p_i^+ closer to q_i and pushes semantically identical sentences p_i^- away from q_i. We show that this counterfactual retrieval framework can be used to retrieve counterfactuals for tasks with only a small amount of seed training data (§4.1). Additional details about training and evaluation of the trained CF-DPR are in Appendix A.

Inference We create S for a specific task dataset using (1) text corpora that have similar domains as that dataset and (2) other datasets for the same task. For instance, for sentiment analysis over IMDb, we use a large (1.6 million) corpus of movie reviews from Amazon (McAuley and Leskovec, 2013) and the Yelp review dataset (Asghar, 2016). We encode the entire search corpus using the trained document encoder and index it using FAISS (Johnson et al., 2019). For every input training instance x_i, we retrieve top k relevant counterfactuals $\{\hat{x}_i^1, \hat{x}_i^2, \ldots, \hat{x}_i^k\}$. We refer to these as *CF-DPR counterfactuals*.

3.2 GPT-3 Editor

The retrieved counterfactuals often contain relevant phrases that perturb predictive features in different ways (“opaque” → “delighful”, “boring” → “spot-on awesome” in Figure 2), but are typically not a minimally edited version of the training sample. “The most delightful trailer ever” has the opposite sentiment as the original review, but is about another entity. To incorporate perturbation diversity while controlling for minimality, CORE uses an editor module. The Editor takes the training sample and retrieved counterfactuals as input and generates a minimally edited counterfactual. This involves selecting parts of the retrieved text that maybe useful in flipping the original text’s label and then seamlessly integrating them into the original input. This fine-grained instance-specific style-transfer task can be hard to find supervision for.

We use GPT-3 (Brown et al., 2020) since it has been successfully used for fine-grained style transfer with few-shot supervision in prior work Reif et al. (2021). For instance, GPT-3 can learn from natural language constraints in its prompt, such as “include the word balloon,” for constrained decoding. To prompt GPT-3, we make use of the set of instructions that were provided in Kaushik et al. (2020) to instruct crowd workers to author counterfactuals. We also append four human authored demonstrations of incorporating retrieved data, depicting various perturbation types.

Following Reif et al. (2022), we simplify our demonstrations by extracting keywords from the retrieved samples and providing them as token-level constraints in the prompt. To encourage the model to perturb certain classes of words, we remove determiners, conjunctions, and punctuation from the retrieved samples and tokenize the rest of the input into a list of keywords: $[w_1, \ldots, w_n]$. The resultant demonstration in our prompt thus becomes: “Input: ... Words to use: $[w_1, \ldots, w_n]$, Edited:...”. This is motivated by Wu et al. (2021)’s observation that perturbing certain classes of words (like preposition and adjectives) leads to better counterfactual generation. More details about the prompt construction are in Appendix A.

4 Experimental Setup

We generate CORE counterfactuals for two tasks – sentiment classification of movie reviews and natural language inference. We describe task-specific details about CORE training and inference below.

4.1 Sentiment Classification

Task Dataset ($N(x)$) We create CORE counterfactuals for the IMDb movie review dataset (Maas et al., 2011), which has been used to manually create contrastive data (Kaushik et al., 2020; Gardner et al., 2020). This dataset presents unique challenges due to the longer average length of reviews (233 words), that existing counterfactual generation techniques (Wu et al., 2021) struggle at.

CF-DPR training data (p_i^+, p_i^-) Kaushik et al. (2020) augment a subset of the IMDb dataset (1.7K examples) with human edited counterfactuals, which we use to train CF-DPR. Negative pairs p_i^- are created by paraphrase models.

Task-specific corpus (S) We use datasets that are of similar domain — Amazon Movie reviews (McAuley and Leskovec, 2013), Yelp reviews (Asghar, 2016), and IMDb reviews (Maas et al., 2011). Our initial experiments indicated that indexing full movie reviews did not yield good CF-DPR performance, owing to more dense retrieval noise when encoding longer contexts (Luan et al., 2021). Hence, we sentence tokenize the reviews and index each sentence independently. The search corpus contains approximately 8 million sentences.

Subset to augment $N(x)$ We generate CORE counterfactuals for the same subset of $1.7k$ reviews from IMDb chosen by Kaushik et al. (2020) in order to make fair comparisons in §5.

Key-word list $[w_1, \ldots, w_n]$ We observed that several sentences in the review contain descriptive information (plot, cast, etc) that don’t convey sentiment. In Kaushik et al. (2020) too, humans edited a few sentences to change the sentiment of a long review. We extract these sentences containing polarity features, use them as queries to CF-DPR, and retrieve their counterfactuals. This selection strategy can also be extended to any review by learning a sentence selection model on annotated data (DeYoung et al., 2020). The keyword list is extracted from retrieved outputs as described in §3.2. Note that we do not impose any restrictions on the editor

IMDb		
Review: It's the worst movie I've ever seen. The action is so unclear, work of cameras is so poor, actors are so affected ... and this lamentable 5 minutes of Arnie on the screen. My advice from the bottom of my heart - don't watch it unless you like such a low class torture. **Sentiment:** Negative		
CF-DPR Probably the dozenth time I've seen it. This is another such effort. This is one spectacular production. that is definitely worth taking advantage of.	**GPT-3** It's one of the best movies I've ever seen. The action is so clear, work of cameras is so great, actors are so unaffected ... and this wonderful 5 minutes of Arnie on the screen. My advice from the bottom of my heart - watch it unless you don't like such a high class torture.	**CORE** It's the best movie I've ever seen. The action is so clear, work of cameras is so great, actors are so professional ... and this spectacular 5 minutes of Arnie on the screen. My advice from the bottom of my heart - definitely watch it if you like such a high class production.
MNLI		
Premise: and my my uh taxes are a hundred and thirty five. **Hypothesis**: My taxes are $\$135$ **(Entails)**		
CF-DPR My sister spent over $\$2,000$ on a computer that she'll probably never use.	**GPT-3** My taxes are not $\$135$.	**CORE** My taxes are probably over $\$135$
Premise:After the battle against the slave lord, the group needed it. **Hypothesis**:The slaves battled the group. **(Contradicts)**		
CF-DPR The slaves never fought against their tyrannical captors.	**GPT-3** The slaves battled the slave lord.	**CORE** The slaves fought against their captors.

Table 1: CORE generated counterfactuals for IMDb and MNLI, along with the CF-DPR retrieved outputs and the independent GPT-3 Editor. For both the tasks, the retriever introduces several new words/phrases in the outputs.

regarding which sentences to edit.

4.2 Natural Language Inference

Task Dataset $N(x)$ We focus on MNLI (Williams et al., 2018), a popular NLI dataset that tests for complex language reasoning.

CF-DPR training data p_i^+, p_i^- We use the inherent paired nature of MNLI. In MNLI, given a premise, annotators are asked to manually write one sentences that entail, contradict or are neutral to the premise. These three hypotheses serve as mutual counterfactuals. In this work, we limit counterfactual perturbations to entailment(E)→contradiction(C) and vice-versa, to simplify the different permutations of positives and negatives required for CF-DPR training. We find that including the neutral class leads to increasingly noisy retrieved data, as the semantic differences between neutral class and the other two NLI classes are subtle and hard to distinguish. In Equation 1, q_i is generated by concatenating the premise and hypothesis separated by the special token [SEP]. For every such input, p_i^+ is a hypothesis from the counterfactual class, while p_i^- are diverse paraphrases of the original hypothesis.

Task-specific corpus (S) is constructed by combining the following NLI datasets (Williams et al., 2018; Bowman, Samuel R. and Angeli, Gabor and Potts, Christopher, and Manning, Christopher D., 2015; Liu et al., 2022) in addition to the source corpus [2] that was used to generate premises in MNLI. We also include tokenized wikipedia articles (Merity et al., 2016) as several domains in MNLI (Eg. travel, government) are related. The search corpus contains approximately 7 million text excerpts.

Subset to augment $N(x)$ To compare with the state-of-the-art data augmentation technique for MNLI, WaNLI (Liu et al., 2022), we choose a subset of the MNLI dataset for augmentation based on their selection strategy. WaNLI uses dataset cartography (Swayamdipta et al., 2020) to select the most ambiguous examples — where model confidence across training epochs is low and variance is high — for augmentation. We generate 9.5K additional examples in two classes.

Cross-Encoder We incorporate a re-ranker module to boost retrieval results for MNLI, that uses a cross-encoder architecture (Thakur et al., 2021) to jointly encode query q_i and top-K documents retrieved by the bi-encoder. Given a q_i and K retrieved sentences from the bi-encoder, the re-ranker learns to classify them as positive or negative. During inference, bi-encoder outputs are re-ranked based on their cross-encoder probability. The cross encoder is trained on the binary classification task on the same seed dataset as the bi-encoder. Retrieval performances are reported in Appendix C.

[2] http://www.anc.org/

Train ↓ Test →	IMDb	Senti140	SST2	Yelp	IMDb Cont	IMDb CAD
IMDb	90.98	75.30	84.63	90.04	81.35	83.76
+ CAD (Human, Clean)	91.3	75.77	88.19	91.31	**86.68**	**88.54**
+ Data Augmentation	91.23	69.67	73.16	84.48	82.17	84.44
+ GPT-3 counterfactuals	91.1	75.90	88.41	92.31	83.40	86.22
+ CFDPR counterfactuals	91.51	74.09	88.30	91.19	79.30	80.90
+ CORE counterfactuals	91.18	**78.12**	**90.82**	**92.01**	84.63	86.35

Table 2: Accuracies of various data augmentation strategies on IMDb (Section 5.1.2). CAD augmentation is the noise free involving human intervention while the rest are noisy. Although in-domain performance is unaffected, we can see notable gains on all out-of-distribution datasets (Go et al., 2009; Socher et al., 2013; Asghar, 2016) and also competative gains on contrast (Gardner et al., 2020) and CAD (Kaushik et al., 2021) test sets. Statistical variance in results across runs is < 0.5 points.

Train ↓ Test →	MNLI	QNLI	SNLI	WaNLI	HANS	Diagno	NLI-Adv	ANLI		
							LI_LI	R1	R2	R3
MNLI	87.66	50.57	83.82	59.10	68.22	61.11	90.39	32	30	28.58
+ WaNLI (Human, Clean)	88.02	50.57	84.85	58.56	70.90	62.59	91	34	29.40	30.25
+ Tailor	88.28	50.53	83.03	60.66	70.73	62.59	90.25	34.20	29.60	29.08
+ GPT3 counterfactuals	87.73	50.70	82.74	58.96	64.43	61.50	88.07	32.80	29.20	32.80
+ CFDPR counterfactuals	87.79	44.99	83.06	59.14	66.49	62.13	89.70	33.50	29.70	29.50
+ CORE counterfactuals	87.97	50.52	84.34	**60.80**	**72.57**	62.32	90.98	33	28.90	**30.58**
+ WaNLI + CORE counterfactuals	88.31	50.61	85.03	58.96	70.55	62.50	90.31	**34.90**	29	30.25

Table 3: Accuracies of data augmentation for CORE and baselines on MNLI (Section 5.1.2). CORE is competitive (within variance) or improves over WaNLI and MNLI baseline in almost all cases. We have competitive performance on both, out-of-distribution datasets (Rajpurkar et al., 2016; Bowman et al., 2015; Liu et al., 2022), challenge-sets (McCoy et al., 2019; Wang et al., 2018; Naik et al., 2018; Glockner et al., 2018) and Adversarial NLI (ANLI) (Nie et al., 2020). Statistical variance in results across runs < 0.5 points.

5 Experimental Results

We first list the various augmentation strategies against which we compare CORE (§5.1.1) followed by data augmentation results (§5.1.2). In (§5.2) we highlight the need for a counterfactual retriever, and in (§5.3) we intrinsically evaluate CORE counterfactuals on their quality and perturbation diversity.

5.1 Counterfactual Data Augmentation

We augment the full training datasets with $|N(x)|$ CORE counterfactuals. We fine-tune DeBERTa base model (He et al., 2021) on the combined dataset.

5.1.1 Augmentation Strategies

We compare augmentation with CORE to strong DA baselines and ablations to different parts of our data generation pipeline. For all the strategies we augment with the same number of instances.

Data Augmentation Baselines In order to evaluate the effectiveness of CDA over adding more in-domain data, we add the same number of in-domain training examples. For sentiment classification, we add $1.7k$ movie reviews randomly sampled from the Amazon Movie reviews dataset (McAuley and Leskovec, 2013). For MNLI, we use the **WaNLI** (Liu et al., 2022) dataset constructed using MNLI, that showed impressive OOD gains. We randomly sample 9K reviews from E and C classes in WaNLI.

Counterfactual Data Augmentation For MNLI, we compare against **Tailor** (Ross et al., 2022) using the `SWAP_CORE` control code as it results in label flipping perturbations. We do not compare with Polyjuice (Wu et al., 2021) as it requires complete human relabelling hence it would not be a fair comparison.

Human generated data For IMDb, we also compare against human-authored counterfactual data, **CAD** (Kaushik et al., 2020). Amazon reviews, WaNLI, and CAD *involve human supervision* for dataset construction and are generally noise-free, unlike CORE data (see §5.3 for noise estimates).

GPT-3 Counterfactuals To ablate the effect of conditioning on retrieval, the **GPT-3 counterfactual baseline** edits inputs into counterfactuals based only on a prompt consisting of task instruction and demonstrations with NO keyword lists.

CF-DPR Counterfactuals To ablate the GPT-3 few-shot editor in the CORE pipeline, the **CF-DPR counterfactual baseline** that uses retrieved outputs of CF-DPR as counterfactuals for augmentation and does no few-shot editing. See Appendix A for more task-specific details of this baseline.

5.1.2 Results

IMDb Table 2 shows that using just $1.7k$ (7%) human-annotated contrastive data to train the CF-DPR model, followed by the GPT-3 editing step, CORE obtains a performance gain of up to 6.2% over the IMDb un-augmented baseline. We compare against human-authored Kaushik et al. (2020) (CAD) **clean** examples in the CAD dataset and MiCE, an automatic counterfactual editing technique. We find that CORE is especially effective at OOD improvements on the Senti140, SST2 and Yelp datasets (by 3.66% points on average); despite augmenting data with noisy labels and with no explicit human supervision for editing. We hypothesize that this may be because of priming biases in human-authored counterfactual data (Bartolo et al., 2021b) and more diversity in CORE counterfactuals. A more detailed analysis is presented in §5.3.

Standard in-domain augmentation (i.e. bearing no counterfactual relationship with original data) is not as effective at improving performance on OOD and contrastive sets (Table 2,Table 3), thus highlighting the importance of counterfactualy generated data.

Ablating individual parts of the pipeline, i.e GPT-3 based editing and CF-DPR retrieval – we observe that individual components are less effective than the combination of both techniques in CORE. CF-DPR retrieval may create reviews that incoherent and have a large semantic shift from the original review. Though the independent GPT-3 editor generates reviews that are minimally edited, they may contain recurring perturbation types (Table 1), limiting its efficacy for CDA. Combining the two helps overcome individual drawbacks.

MNLI Using the DeBERTa-base model, with just augmenting $9.5k$ (3%) of the data we get improvements of up to 4% over the unaugmented dataset (Table 3). The improvements are particularly on the WaNLI evaluation set and adversarially designed test sets HANS and ANLI. Compared to Tailor, CORE achieves a 2% and 1.3% gain on HANS and SNLI, respectively. Once again, retrieval and GPT-3 editing are not as effective individually as the combination of the two in CORE.

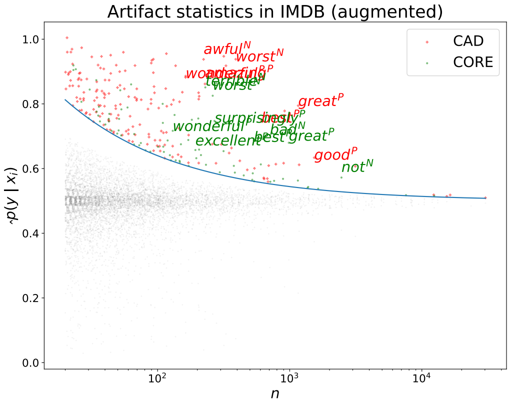


Figure 3: A statistical test for deviation from a competency problem (Gardner et al., 2021), where no word should be informative about class label. Words above the blue line have detectable correlations with labels. CAD displays more statistical bias compared to CORE.

Train ↓ Test →	MNLI	HANS	Diagno	NLI-Adv LI_LI
MNLI	87.66	68.22	61.11	90.39
+ TF-IDF + CORE Editor	87.89	60.51	61.77	87.49
+ CFDPR + CORE Editor	87.97	**72.57**	**62.32**	90.98

Table 4: Comparison between CFDPR and TF-IDF based retriever. We observe that using a general text retriever hurts performance.

Independent GPT-3 generated counterfactuals are biased towards simple perturbations (§5.3) and augmenting with these counterfactuals hurts performance on HANS and on Lexical Inference test set (LI_LI). CORE is also competitive with WaNLI data augmentation, which is noise-free in-domain data constructed with the same subset selection strategy as used by CORE. We also consider augmenting with both CORE and WaNLI counterfactuals, which results in orthogonal benefits and improvements on SNLI and ANLI datasets.

5.2 Retrieval Ablation

To understand the importance of a trained counterfactual retriever, we ablate the retrieval stage of CORE by using a simple TF-IDF based retriever. We use the same GPT-3 Editor to generate the counterfactuals, only replace the CFDPR counterfactual retriever with a TF-IDF retriever. The TF-IDF based CORE counterfactuals exacerbate biases as seen in Table 4, significantly hurting performance by 8% and 3% on HANS and Adv. NLI, full table in Appendix (Table 15). This highlights the need

Polyjuice (Wu et al., 2021)	**CORE**
1. The problem with all of this : It 's **not** really funny.	
1. [delete] The problem with all of this: It is ~~not~~ very funny.	1. The great thing about all of this: It's funny, considered anyway - for no particular reason.
2. Munch 's screenplay is **tenderly** observant of his characters .	
2. [insert]and Munch's screenplay is not observant of his characters.	2. Seldom does a contrived screenplay comes across that is ever so observant of its characters.
1. As a film director , LaBute continues to **improve** .	
1. [lex] As director, LaBute continues to crap.	1. As a film director, LaBute is not as good as one might hope.
2. The movie is a **mess** from start to finish .	
2. [lex] It is a wonderful movie from start to finish.	2. I went to the movie weeks ago and enjoyed it from start to finish.

Table 5: Examples of perturbations on SST2 data. Polyjuice often uses restricted patterns like perturbing "not" or using *antonyms*, unlike CORE's non-trivial perturbations conditioned on retrieved data.

for a counterfactual retrieval module that retrieves counterfactual words/phrases in an unconstrained manner without limiting potential semantic edits.

5.3 Intrinsic Evaluation

To analyze the source of its empirical gains of CORE in §5.1.2, we evaluate the label correctness, closeness, and perturbation diversity.

Label Correctness Unlike Polyjuice (Wu et al., 2021) CORE encourages label flipping behavior during the generation process, *possibly* at the cost of label correctness. We quantify label correctness of the generated data by manually annotating a sample of 100 data points for IMDb and MNLI. Our analysis show that noise levels are 41% for IMDb and 40% for MNLI. CORE's augmentation benefits (§5.1.2) persist even when compared to noise-free data (CAD for IMDb and WaNLI for MNLI), underscoring the importance of diversity.

Closeness and Diversity The intuition behind CORE is that effective CDA requires perturbation of various kinds so that perturbation bias is not exacerbated (Joshi and He, 2022). To measure this effect, on the IMDb dataset, we compare with CAD and another prior work on CAD generation, MiCE (Ross et al., 2021). In Figure 3, for the augmented subset we plot the probability of a token predicting class labels as a function of token count. We observe that CAD (Kaushik et al., 2020) has several outliers tokens that have strong bias towards a label, compared to CORE.

Since counterfactuals are meant to be *minimally* edited instances of the original input (Gardner et al., 2020), we analyse the closeness of the generated counterfactuals to the original text. To check how close these generated counterfactuals are to the original instance, we measure the Levenshtein edit distance between the two. To quantify diversity, we also measure self-BLEU (Zhu et al., 2018). The self-BLEU score is computed between edited counterfactual and the original text on the IMDb dataset. Since self-BLEU and Levenshtein are opposite in nature, we ideally want the counterfactuals to strike a balance between the two (Wu et al., 2021). As shown in in Table 6, CORE counterfactual are more diverse (lower Self-BLEU) which is at a small cost of edit distance when compared to CAD and MiCE.

Model	**self-BLEU ↓**	**Levenshtein ↓**
IMDb		
CAD	0.758	**0.156**
MiCE	0.709	0.195
CF-DPR	**0.002**	0.830
CORE	0.445	0.506
MNLI Crowd-sourcing Experiment		
Retrieval	**0.092**	0.765
w/o Retrieval	0.313	**0.484**

Table 6: Closeness (edit distance) and diversity (self-BLEU) for different counterfactual generation strategies

Perturbation type To further analyze the source of diversity, we classify perturbations in CORE counterfactuals according to the perturbation-type detector used in Wu et al. (2021). Table 7 shows that CORE is able to cover a broad set of previously defined perturbation types that were recognized in prior work, such as *negation, insertions, lexical change and resemantics* without being explicitly controlled. The number of perturbation types that are not categorized by this detector are significantly higher in case of CORE compared to Polyjuice. In Table 5, we find that CORE perturbations fall in more than one Polyjuice categories and avoid trivial perturbations like negations and antonyms. More examples of the different perturbations we see in CORE are in Appendix B

Dataset	Model	Negation	Insertion	Resemantic	Lexical	Quantifier	Restructure	Delete	UNK
MNLI	CORE	691	1303	1320	1073	138	92	79	4577
	GPT-3	1400	705	1169	2184	290	95	94	3545
SST2	CORE	81	140	202	174	49	5	60	1305
	Polyjuice	161	91	209	362	26	17	66	1076

Table 7: List of perturbation types detected using the set of heuristics from Wu et al. (2021). We can see that for both the task Sentiment and NLI, CORE covers all perturbation types without any explicit control code.

Supporting Human Annotation Prior work proposes aiding crowd-workers in the task of dataset creation using generative assistants (Bartolo et al., 2021a) or randomly sampled words (Gardner et al., 2021) to encourage creativity, leading to better quality and diversity in human-authored data. We analyze the impact of CF-DPR counterfactuals in encouraging humans to make diverse counterfactual perturbations to text. We design a controlled crowd-sourcing experiment where 200 original MNLI examples from the validation set are shown to humans with and without the retrieved counterfactual sentences to aid them. Human-authored counterfactuals conditioned on retrieved outputs display more diversity, with lower self-BLEU and higher Levenshtein distance compared to the control condition (Table 6). Qualitative differences between human authored counterfactuals in both conditions and details of the crowd-sourcing are in the Appendix A.

6 Conclusion

We present **CORE**, a retrieval-augmented generation framework for creating *diverse* counterfactual perturbations for CDA. CORE first learns to retrieve relevant text excerpts and then uses GPT-3 few-shot editing conditioned on retrieved text to make counterfactual edits. CORE encourages diversity with the use of additional knowledge for this task, explicitly via retrieval and implicitly (parametric knowledge), via GPT-3 editing. Conditioning language model edits on naturally occurring data results in diversity (§5.3). CORE counterfactuals are more effective at improving generalization to OOD data compared to other approaches, on natural language inference and sentiment analysis(§5.1.2). CORE's retrieval framework can also be used to reduce priming effects and encourage diversity in manually authored perturbations (§5.3).

7 Limitations

While CORE involves no human intervention, it is not completely accurate at performing label flipping perturbations and human re-labelling can be beneficial. Our framework uses a learned retriever which can be challenging to train when there are finer semantic differences between classes (e.g. neutral class) that need to be captured. Work on counterfactual generation has focused exclusively on English language text, and it would be an interesting future work to expand such frameworks for other languages.

Acknowledgements

We would like to thank Alisa Liu, Rajarshi Das, Sewon Min, Tongshuang Wu, Mandar Joshi, Julian Michael and the anonymous reviewers for their helpful comments and suggestions.

References

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, Online. Association for Computational Linguistics.

Nabiha Asghar. 2016. Yelp dataset challenge: Review rating prediction. *arXiv preprint arXiv:1605.05362*.

Max Bartolo, Tristan Thrush, Robin Jia, Sebastian Riedel, Pontus Stenetorp, and Douwe Kiela. 2021a. Improving question answering model robustness with synthetic adversarial data generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8830–8848, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Max Bartolo, Tristan Thrush, Sebastian Riedel, Pontus Stenetorp, Robin Jia, and Douwe Kiela. 2021b. Models in the loop: Aiding crowdworkers with generative annotation assistants. *arXiv preprint arXiv:2112.09062*.

Yonatan Bitton, Gabriel Stanovsky, Roy Schwartz, and Michael Elhadad. 2021. Automatic generation of contrast sets from scene graphs: Probing the compositional consistency of GQA. In *Proceedings of the 2021 Conference of the North American Chapter of* *the Association for Computational Linguistics: Human Language Technologies*, pages 94–105, Online. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Bowman, Samuel R. and Angeli, Gabor and Potts, Christopher, and Manning, Christopher D. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Mingda Chen, Qingming Tang, Sam Wiseman, and Kevin Gimpel. 2019. Controllable paraphrase generation with a syntactic exemplar. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5972–5984, Florence, Italy. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. 2020. ERASER: A benchmark to evaluate rationalized NLP models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4443–4458, Online. Association for Computational Linguistics.

Dheeru Dua, Pradeep Dasigi, Sameer Singh, and Matt Gardner. 2021. Learning with instance bundles for reading comprehension. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7347–7357, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jörg Frohberg and Frank Binder. 2022. CRASS: A novel data set and benchmark to test counterfactual reasoning of large language models. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2126–2140, Marseille, France. European Language Resources Association.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. 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.

Matt Gardner, William Merrill, Jesse Dodge, Matthew Peters, Alexis Ross, Sameer Singh, and Noah A. Smith. 2021. Competency problems: On finding and removing artifacts in language data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1801–1813, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mor Geva, Tomer Wolfson, and Jonathan Berant. 2021. Break, perturb, build: Automatic perturbation of reasoning paths through question decomposition. In *Transactions of the Association for Computational Linguistics (TACL)*.

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking NLI systems with sentences that require simple lexical inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 650–655, Melbourne, Australia. Association for Computational Linguistics.

Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision.

Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor OK Li. 2018. Search engine guided neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 1.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. REALM: Retrieval-augmented language model pre-training. *arXiv preprint arXiv:2002.08909*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547.

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.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Divyansh Kaushik, Eduard Hovy, and Zachary C Lipton. 2020. Learning the difference that makes a difference with counterfactually augmented data. *International Conference on Learning Representations (ICLR)*.

Divyansh Kaushik, Douwe Kiela, Zachary C. Lipton, and Wen-tau Yih. 2021. On the efficacy of adversarial data collection for question answering: Results from a large-scale randomized study. 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 6618–6633, Online. Association for Computational Linguistics.

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.

Kalpesh Krishna, John Wieting, and Mohit Iyyer. 2020. Reformulating unsupervised style transfer as paraphrase generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 737–762, Online. Association for Computational Linguistics.

Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. 2019. Quantifying the carbon emissions of machine learning. *arXiv preprint arXiv:1910.09700*.

Phuc H Le-Khac, Graham Healy, and Alan F Smeaton. 2020. Contrastive representation learning: A framework and review. *IEEE Access*, 8:193907–193934.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Chuanrong Li, Lin Shengshuo, Zeyu Liu, Xinyi Wu, Xuhui Zhou, and Shane Steinert-Threlkeld. 2020. 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.

Alisa Liu, Swabha Swayamdipta, Noah A Smith, and Yejin Choi. 2022. Wanli: Worker and ai collaboration for natural language inference dataset creation. *arXiv preprint arXiv:2201.05955*.

Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2021. Sparse, dense, and attentional representations for text retrieval. *Transactions of the Association for Computational Linguistics*, 9:329–345.

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, 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*, 15, pages 13516–13524.

Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: Understanding rating dimensions with review text. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, page 165–172, New York, NY, USA. Association for Computing Machinery.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. 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, Sean Welleck, Daniel Khashabi, and Yejin Choi. 2022. Cold decoding: Energy-based constrained text generation with langevin dynamics. *arXiv preprint arXiv:2202.11705*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Emily Reif, Daphne Ippolito, Ann Yuan, Andy Coenen, Chris Callison-Burch, and Jason Wei. 2021. A recipe for arbitrary text style transfer with large language models. *arXiv preprint arXiv:2109.03910*.

Emily Reif, Daphne Ippolito, Ann Yuan, Andy Coenen, Chris Callison-Burch, and Jason Wei. 2022. A recipe for arbitrary text style transfer with large language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 837–848, Dublin, Ireland. 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.

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.

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.

Sivic and Zisserman. 2003. Video google: a text retrieval approach to object matching in videos. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 1470–1477 vol.2.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9275–9293, Online. Association for Computational Linguistics.

Nandan Thakur, Nils Reimers, Johannes Daxenberger, and Iryna Gurevych. 2021. Augmented SBERT: Data augmentation method for improving bi-encoders for pairwise sentence scoring tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 296–310, Online. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Xuezhi Wang, Haohan Wang, and Diyi Yang. 2021. Measure and improve robustness in nlp models: A survey. *arXiv preprint arXiv:2112.08313*.

Kellie Webster, Xuezhi Wang, Ian Tenney, Alex Beutel, Emily Pitler, Ellie Pavlick, Jilin Chen, Ed Chi, and Slav Petrov. 2020. Measuring and reducing gendered correlations in pre-trained models. *arXiv preprint arXiv:2010.06032*.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

Jason Weston, Emily Dinan, and Alexander Miller. 2018. Retrieve and refine: Improved sequence generation models for dialogue. In *Proceedings of the 2018 EMNLP Workshop SCAI: The 2nd International Workshop on Search-Oriented Conversational AI*, pages 87–92, Brussels, Belgium. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. 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 1112–1122. Association for Computational Linguistics.

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.

Yuxiang Wu, Matt Gardner, Pontus Stenetorp, and Pradeep Dasigi. 2022. Generating data to mitigate spurious correlations in natural language inference datasets. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2660–2676, Dublin, Ireland. Association for Computational Linguistics.

Fei Xiao, Liang Pang, Yanyan Lan, Yan Wang, Huawei Shen, and Xueqi Cheng. 2021. Transductive learning for unsupervised text style transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2510–2521, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Olivier Bousquet, Quoc Le, and Ed Chi. 2022. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.

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 &amp; Development in Information Retrieval*, SIGIR '18, page 1097–1100, New York, NY, USA. Association for Computing Machinery.

A Implimentation details for CORE

CF-DPR We make use of the open source implementation [3] of DPR to train the CF-DPR model. We train the CF-DPR model using 2 hard negatives, one the paraphrase and the second the query instance (hypothesis in case of MNLI and entire review for IMDB). All training was performed using 3 TITAN X (Pascal) GPUs. The average run time for CF-DPR for the MNLI task was 5hrs and for IMDB it was 1hr. Total emissions are estimated to be 4.19 kgCO_2eq of which 0% was directly offset. Estimations were conducted using the Machine-Learning Impact calculator presented in (Lacoste et al., 2019). The training parameters used to train CF-DPR for the MNLI task are in Table 8 and for IMDB Table 9.

Hyperparameters	Value
encoder sequence_length	32
pretrained_model_cfg	bert-base-uncased
per_device_batch_size	32
weight_decay	0.01
warmup_steps	2500
max_grad_norm	2.0
learning_rate	2e-5
num_train_epochs	10
eval_per_epoch	10
hard_negatives	2

Table 8: Hyperparameters for training CF-DPR for counterfactual retrieval on MNLI.

Hyperparameters	Value
encoder sequence_length	96
pretrained_model_cfg	bert-base-uncased
per_device_batch_size	16
gradient_accumulation_steps	3
weight_decay	0.01
warmup_steps	100
max_grad_norm	2.0
learning_rate	2e-5
num_train_epochs	100
eval_per_epoch	1
hard_negatives	2

Table 9: Hyperparameters for training CF-DPR for counterfactual retrieval on IMDB.

We experiment with several hyperparameters and find the quality of the hard negatives used to train the model played the most critical role in training. Adding more hard negatives didn't really help boost performance. Changing learning

[3] https://github.com/facebookresearch/DPR

rate and warmup steps did not drastically affect the training process either. For evaluation we use the validation sets from (Kaushik et al., 2020) and MNLI validation- matched set for sentiment and NLI CF-DPR models respectively and reformat it as described in §4.1 and §4.2. For every instance in the validation set, we generate 30 random negatives and 30 hard negatives. Since we cannot generate 30 different paraphrases we make use of a semantic sentence retriever model (Gao et al., 2021) to retrieve semantically similar sentences from a text corpus. Finally following the DPR codebase we measure the top 1 accuracy on the validation set. For MNLI our CF-DPR model achieves an top 1 accuracy of about 70% while for IMDb around 45%.

The cross encoder model was trained using the `sentence-transformers`[4] library. Each instance in the training dataset for MNLI contains a query instance which is a concatenated version of the premise and hypothesis separated by the [SEP] token, and two hypothesis with labels 1 and 0 respectively. We train the `bert-base-model` using the BCE loss. We train the model for 10 epochs with a batch size of 64 and learning rate $2e-5$. We evaluate every 2000 setps and save the model with the best evaluation loss. The cross encoder gave a validation accuracy of 91%.

For Inference we encode the entire search corpus with the context encoder and index it using faiss. We approximate maximum inner-product search (MIPS) with an Inverted File Index (IVF) (Sivic and Zisserman, 2003) for faster retrieval. We use the IVFFlat index [5] as it helps improve the retrieval speed at a small cost of accuracy. We set the number of centroids (K) as 300 and n_probes to 30.

GPT-3 few shot prompting We use the `text-davinci-002` model as the Editor. The prompts to GPT-3 for IMDB is in Table 11 and for MNLI Table 10. For both the model settings we set the temperature parameter to 0.7 and Top p parameter to 1.

DeBERTa finetuning All training was performed using 3 TITAN X (Pascal) GPUs. We evaluate every 400 steps and save the model with the best evaluation loss. We make use of the huggingface trainer APIs [6] for fine-tunning the models. It takes around 2 hrs to fine-tune the DeBERTa base model for MNLI and less than an hour for IMDb. The fine-tuned models were evaluated on publicly available validation/test sets.

For MNLI in-domain performance we only report results on the mis-matched validation set as we observe that both the sets matched and mismatched had similar scores. For QNLI, SNLI we use the datasets that are part of the GLUE benchmark. For WaNLI, HANS and Diagonistics we use the same set of validation sets used in Liu et al. (2022) and for ANLI and LI_LI we use the official datasets provided by the authors. Since HANS and QNLI are binary NLI tasks (entailment and non-entailment), for measuring accuracy we consider both Neutral and Contradiction predictions as non-entailment.

For IMDb we use all the official datasets available. For CAD we combine the validation and tests in-order to get more statistically significant results.

Crowd-sourcing Study In Section 5.3, we use CF-DPR outputs to aid crowd-sourcing of counterfactual edits. On the MNLI development set, we randomly sample 200 instances. We create two user interfaces for crowdworkers on Amazon Mechanical Turk to collect data under two conditions. In one condition, workers are shown the original instance (premise and hypothesis) and top-three retrieved counterfactuals provided by CF-DPR. They are asked to edit the hypothesis following brief instructions instructions and examples, shown in Figure 6. In the second case, they are just shown the original instance, no retrieved outputs. When revising examples, we asked workers to preserve the intended meaning through minimal revisions. Each instance is modified only once and different annotators are shown instances from both sets. Annotators were required to have a HIT approval rate of 90%, a total of 1,000 approved HITs. For the case where annotators were shown retrieved sentences, we found that annotator quality was quite poor, since annotators were not filtered by a qualification test to do the task. More generally, complex annotator tasks often require substantial training of crowd-workers (Bartolo et al., 2021a), which is outside the scope of this work. Instead, we recruit computer science graduate students (outside of the study) to get annotations for this task.

[4] https://github.com/UKPLab/sentence-transformers

[5] https://github.com/facebookresearch/faiss/wiki/Faiss-indexes

[6] https://github.com/huggingface/transformers

Two sentences are given, sentence 1 and sentence 2. Given that Sentence 1 is True, Sentence 2 (by implication), must either be (a) definitely True, (b) definitely False. You are presented with an initial Sentence 1 and Sentence 2 and the correct initial relationship label (definitely True or definitely False). Edit Sentence 2 making a small number of changes such that the following 3 conditions are always satisfied:
(a) The target label must accurately describe the truthfulness of the modified Sentence 2 given the original Sentence 1.
(b) In order to edit sentence 2, must only make use of one or two relevant words present in the provided list of words.
(c) Do not rewrite sentence 2 completely, only a small number of changes need to be made. Here are some examples:
original sentence 1: Oh, you do want a lot of that stuff. original sentence 2: I see, you want to ignore all of that stuff. initial relationship label: definitely False
target label: definitely True
List of words: ['correct', 'before', 'happen', 'already', 'more', 'saw', 'on']
modified sentence 2: I see, you want more of that stuff. original sentence 1: Region wide efforts are also underway. original sentence 2: Regional efforts have not stopped. initial relationship label: definitely True
target label: definitely False
List of words: ['talking', 'hold', 'put', 'caller', 'on']
modified sentence 2: Regional efforts are on hold at the moment.
original sentence 1: yeah you could stand in there if you really wanted to i guess. original sentence 2: If you want you can sit there I guess. initial relationship label: definitely False
target label: definitely True
List of words: ['there', 'without', 'ants', 'stand', 'even']
modified sentence 2: If you want you can stand there I guess.
original sentence 1: and uh every every opportunity there is to make a dollar he seems to be exploiting that. original sentence 2: He works to make money. initial relationship label: definitely True
target label: definitely False
List of words: ['work', 'pass', 'dollars', 'up', 'owe']
modified sentence 2: He passes up on opportunities to make money.

Table 10: Prompt given to GPT-3 for generating CORE counterfactuals for MNLI. The instructions are similar to the ones given to crowdworkers in (Kaushik et al., 2020), additionally we incorporate 4 demonstrations. The test instance is appended to this prompt.

Given a movie review and its sentiment. Edit the review making a small number of changes such that, the following two conditions are always satisfied:
(a) the target label accurately describes the sentiment of the edited review
(b) make use of only a few key words provided in the list of words to edit the review
Do not remove or add any extra information, only make changes to change the sentiment of the review.
Review: Long, boring, blasphemous. Never have I been so glad to see ending credits roll.
Label: Negative
List of relevant words: ['clean' , 'now', 'brought', 'perfect' , 'many' ,'interesting' ,'memories', 'sad']
Target Label: Positive
Edited Review: Perfect, clean,interesting. Never have I been so sad to see ending credits roll.
Review: I don't know why I like this movie so well, but I never get tired of watching it.
Label: Positive
List of relevant words: ['hate' , 'now', 'supposedly' , 'many' , 'memories', 'watching']
Target Label: Negative
Edited Review: I don't know why I hate this movie so much, now I am tired of watching it.

Table 11: Prompt given to GPT-3 for generating CORE counterfactuals for IMDB. The instructions are similar to the ones given to crowdworkers in (Kaushik et al., 2020), additionally we incorporate 2 demonstrations. The test instance is appended to this prompt.

Hyperparameters	Value
Model	microsoft/deberta-base
learning rate	$2e^{-5}$
number of epochs	3
per device batch size	32
max seq length	128

Table 12: Training Hyperparameters for DeBERTa base for MNLI.

Hyperparameters	Value
Model	microsoft/deberta-base
learning rate	$2e^{-5}$
number of epochs	5
per device batch size	32
max seq length	128

Table 13: Training Hyperparameters for DeBERTa base for IMDB.

Compensation We aimed to pay rate of at least $\$15$ per hour. Workers were paid 0.75 for each example that they annotate.

B Qualitative Examples - Human Authored Counterfactuals

We show examples of the counterfactual editing of MNLI examples done by crowd-workers who were asked to independently edit instances and experts (CS graduate students) who were shown CF-DPR-retrieved counterfactual instances in Table 17.

CAD		CORE	
Biased feature	**z_stats**	**Biased feature**	**z_stats**
Negative			
bad	16.93	unfortunately	12.27
worst	16.71	worst	10.83
terrible	15.44	terrible	10.55
boring	15.05	bad	10.49
Positive			
great	19.41	great	11.15
best	11.54	best	8.04
amazing	11.25	surprisingly	7.66
wonderful	9.47	wonderful	4.73

Table 14: Z scores values for the original $1.7k$ imdb reviews augmented either with $1.7k$ CAD or CORE counterfactuals.

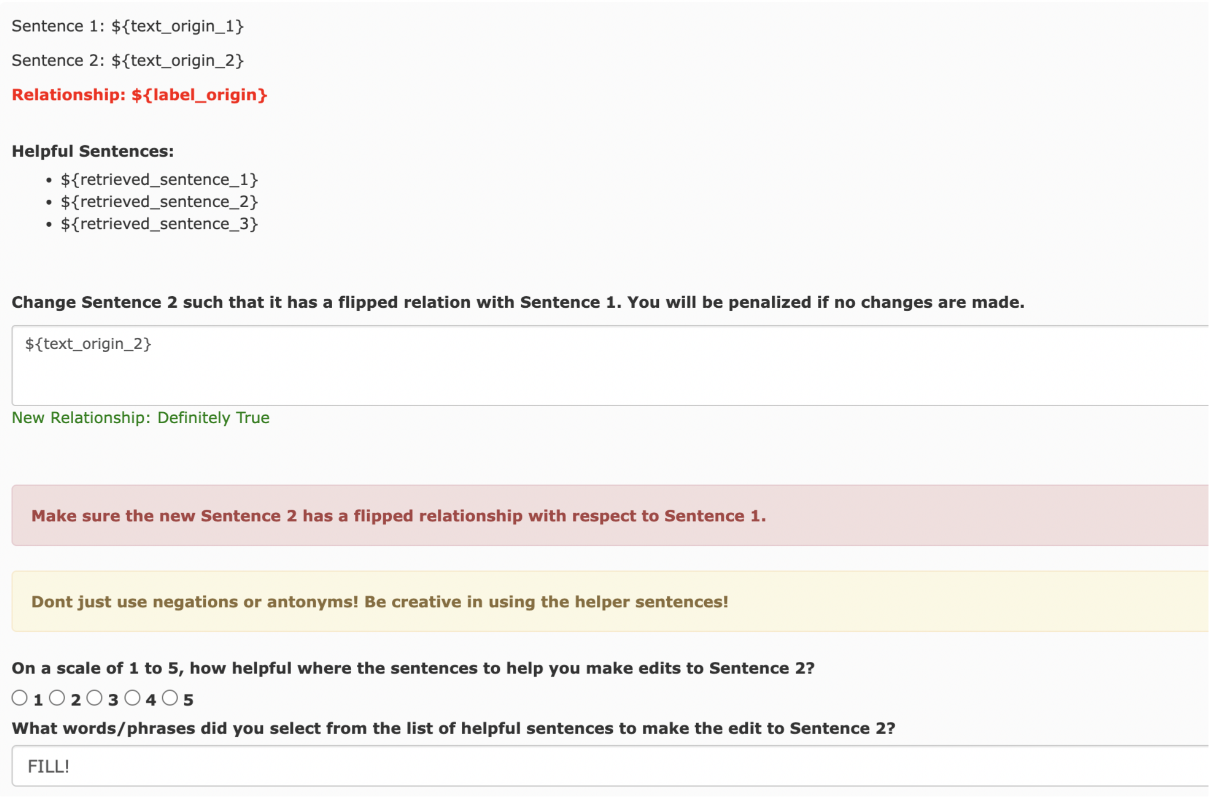
In addition to the examples present in Table 1 we also provide some more examples for MNLI in Table 18 and IMDb in Table 19.

C Additional Analysis

Z statistics scores Figure 3 depicts the competancy style plot for the IMDb augmented subset. In addition to that we also show the individual z scores in Table 14

SST-2 Results We also study the ability of CORE to generate counterfactuals on SST2 without being explicitly trained on SST2 data. We use the CF-DPR model trained on IMDb. The results and comparison with Polyjuice can be found in Table 16.

CDA baselines For all the baselines we make use of the official implementations. For Tailor on MNLI, we use both no context and in-context `swap-core` perturbations.



Sentence 1: ${text_origin_1}

Sentence 2: ${text_origin_2}

Relationship: ${label_origin}

Helpful Sentences:

- ${retrieved_sentence_1}
- ${retrieved_sentence_2}
- ${retrieved_sentence_3}

Change Sentence 2 such that it has a flipped relation with Sentence 1. You will be penalized if no changes are made.

${text_origin_2}

New Relationship: Definitely True

Make sure the new Sentence 2 has a flipped relationship with respect to Sentence 1.

Dont just use negations or antonyms! Be creative in using the helper sentences!

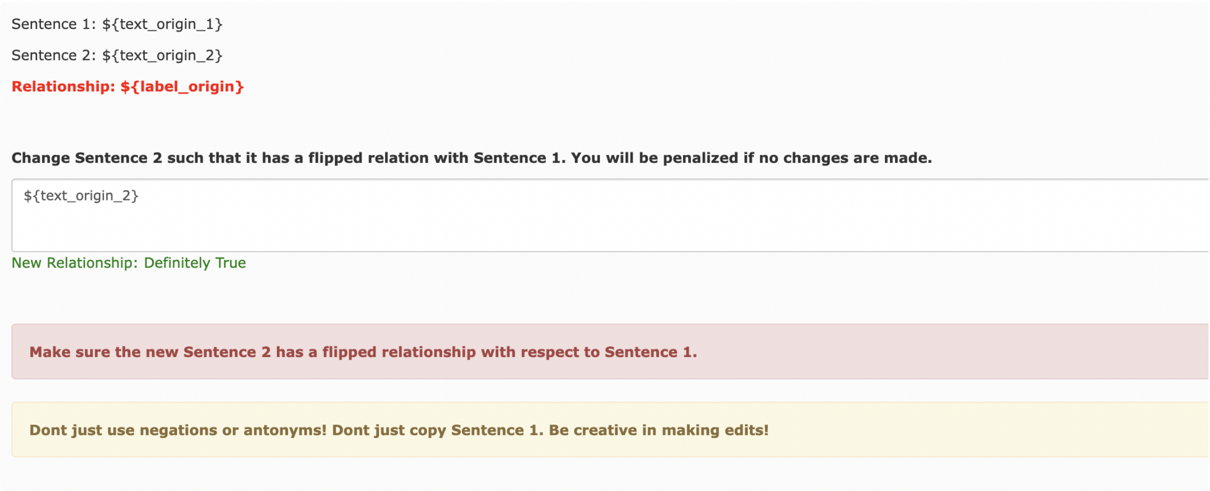
On a scale of 1 to 5, how helpful where the sentences to help you make edits to Sentence 2?

○ **1** ○ **2** ○ **3** ○ **4** ○ **5**

What words/phrases did you select from the list of helpful sentences to make the edit to Sentence 2?

FILL!

Figure 4: Crowd-worker platform interface where humans have to use retrieved sentences to edit examples



Sentence 1: ${text_origin_1}

Sentence 2: ${text_origin_2}

Relationship: ${label_origin}

Change Sentence 2 such that it has a flipped relation with Sentence 1. You will be penalized if no changes are made.

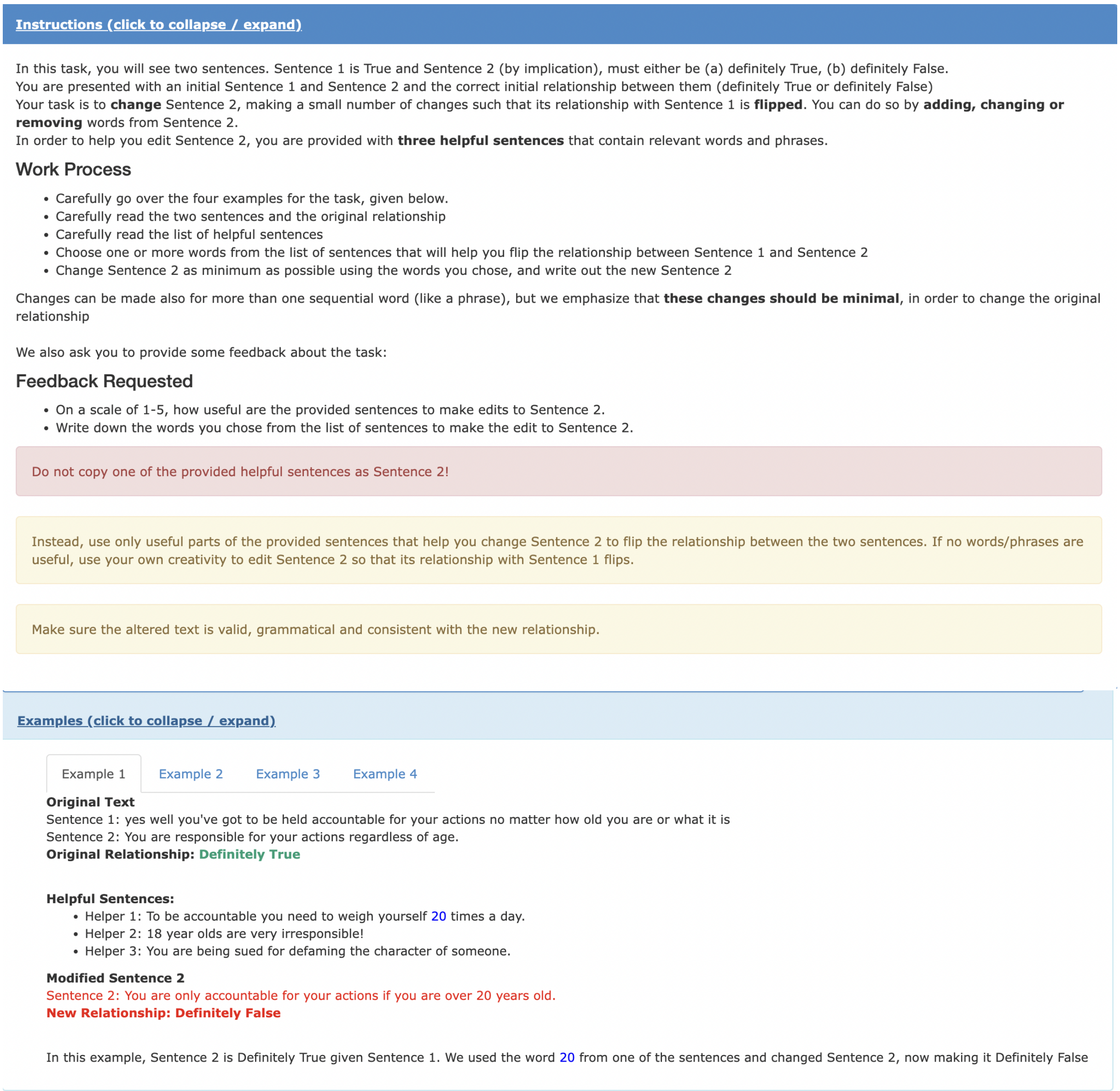
${text_origin_2}

New Relationship: Definitely True

Make sure the new Sentence 2 has a flipped relationship with respect to Sentence 1.

Dont just use negations or antonyms! Dont just copy Sentence 1. Be creative in making edits!

Figure 5: Crowd-worker platform interface where humans have to edit examples without any priming



Instructions (click to collapse / expand)

In this task, you will see two sentences. Sentence 1 is True and Sentence 2 (by implication), must either be (a) definitely True, (b) definitely False.
You are presented with an initial Sentence 1 and Sentence 2 and the correct initial relationship between them (definitely True or definitely False)
Your task is to **change** Sentence 2, making a small number of changes such that its relationship with Sentence 1 is **flipped**. You can do so by **adding, changing or removing** words from Sentence 2.
In order to help you edit Sentence 2, you are provided with **three helpful sentences** that contain relevant words and phrases.

Work Process

- Carefully go over the four examples for the task, given below.
- Carefully read the two sentences and the original relationship
- Carefully read the list of helpful sentences
- Choose one or more words from the list of sentences that will help you flip the relationship between Sentence 1 and Sentence 2
- Change Sentence 2 as minimum as possible using the words you chose, and write out the new Sentence 2

Changes can be made also for more than one sequential word (like a phrase), but we emphasize that **these changes should be minimal**, in order to change the original relationship

We also ask you to provide some feedback about the task:

Feedback Requested

- On a scale of 1-5, how useful are the provided sentences to make edits to Sentence 2.
- Write down the words you chose from the list of sentences to make the edit to Sentence 2.

Do not copy one of the provided helpful sentences as Sentence 2!

Instead, use only useful parts of the provided sentences that help you change Sentence 2 to flip the relationship between the two sentences. If no words/phrases are useful, use your own creativity to edit Sentence 2 so that its relationship with Sentence 1 flips.

Make sure the altered text is valid, grammatical and consistent with the new relationship.

Examples (click to collapse / expand)

Original Text
Sentence 1: yes well you've got to be held accountable for your actions no matter how old you are or what it is
Sentence 2: You are responsible for your actions regardless of age.
Original Relationship: Definitely True

Helpful Sentences:

- Helper 1: To be accountable you need to weigh yourself 20 times a day.
- Helper 2: 18 year olds are very irresponsible!
- Helper 3: You are being sued for defaming the character of someone.

Modified Sentence 2
Sentence 2: You are only accountable for your actions if you are over 20 years old.
New Relationship: Definitely False

In this example, Sentence 2 is Definitely True given Sentence 1. We used the word 20 from one of the sentences and changed Sentence 2, now making it Definitely False

Figure 6: Instructions to crowd-workers for the counterfactual editing task conditioned on retrieval.

Train ↓ Test →	MNLI	QNLI	SNLI	WaNLI	HANS	Diagno	NLI-Adv	ANLI		
							LI_LI	R1	R2	R3
MNLI	87.66	50.57	83.82	59.10	68.22	61.11	90.39	32	30	28.58
+ TF-IDF + GPT3 Editor	87.89	50.57	83.13	58.86	60.51	61.77	87.49	32.30	29.50	30.41
+ CORE counterfactuals	87.97	50.52	84.34	**60.80**	**72.57**	62.32	90.98	33	28.90	**30.58**

Table 15: Accuracies of data augmentation for CORE and retrival ablation. We observe that using a simpler TF-IDF based retriver doesn't help improve performance across any task, in contrast to using a counterfactual retriever.

Train ↓ Test →	SST2	Senti140	Yelp	IMDb	IMDb contrast	IMDb cad
SST2 (6k)	89.60 (0.46)	75.4 (1)	86.75 (0.6)	80.43(0.88)	78.27 (1.37)	83.2 (1.3)
SST2 (4k)+ Polyjuice (2k)	89.64 (0.67)	75.9 (0.49)	85.5 (0.12)	81.10 (0.35)	**84.7 (0.36)**	**88.2 (0.4)**
SST2 (4k) + CORE (2k)	88.45 (1)	75.1 (0.18)	**87 (0.87)**	81.75 (0.8)	80.19 (0.09)	84.72 (0.38)

Table 16: Performance of CORE on SST2. We can see that although our CORE framework has not been trained to generate counterfactuals for SST2 it can yet achieve compatible scores as Polyjuice on out-of-distribution test sets.

without Retrieval	with Retrieval
Premise: Another book that I read recently is very interesting books The Journals of Lewis and Clark. **Hypothesis**: I recently read The Journals of Lewis and Clark. (Entailment)	
New Hypothesis: I skipped reading The Journals of Lewis and Clark.	**New Hypothesis**: I've never heard of this book The Journals of Lewis and Clark.
Premise: This northern beach of magnificent tan sand is most agreeably reached by boat. **Hypothesis**:The beach has beautiful sand. (Entailment)	
New Hypothesis: The sand of the beach is ordinary	**New Hypothesis**: The beach is impossible to reach by boat and the volcano has devastated the island.
Premise: You never call. **Hypothesis**: You rarely call on the phone, nor webcam. (Entailment)	
New Hypothesis: You always call or webcam	**New Hypothesis**: You run up a big bill on your phone calling me!
Premise:have that well and it doesn't seem like very many people uh are really i mean there's a lot of people that are on death row but there's not very many people that actually um do get killed **Hypothesis**: There are a lot of people on death row, but not that many actually are executed. (Contradiction)	
New Hypothesis: There are not many people on death row, because most are promptly executed.	**New Hypothesis**: There are a lot of people on death row, and thousands get killed without anyone noticing.
Premise: Generally, if pH of scrubbing liquor falls below a range of 5.0 to 6.0, additional reagent is required to maintain the reactivity of the absorbent. **Hypothesis**: if pH of scrubbing liquor falls below a range of 5.0 to 6.0, then the whole world may explode (Contradiction)	
New Hypothesis: If the pH of scrubbing liquor falls below a range of 5.0 to 6.0 more reagent is needed to maintain it's reactivity	**New Hypothesis**: if pH of scrubbing liquor falls below a range of 5.0 to 6.0, additional reagent will be needed to make the absorbent last longer
Premise: If any of us at the dental school can be of assistance, please write or call. **Hypothesis**: The dental school may or may not be able to help. (Contradiction)	
New Hypothesis: The dental school cannot give any help.	**New Hypothesis**: We at the dental school at ready to help, please give us a call.

Table 17: Qualitative differences between counterfactual edits made by humans when they are shown retrieved data vs. when they are not. Human annotators primed with retrieval are less likely to use trivial heuristics like negation, antonyms etc, leading to less bias introduced in the new data.

MNLI		
Premise: this is my first call because i just got my password, **Hypothesis**: I just got my password, I've never called before. **Label**: Entailment		
CFDPR I have initiated multiple calls myself	**GPT3** I just got my password, I've called many times before.	**CORE** I've initiated multiple calls before.
Premise: This is routine, he said. **Hypothesis**: This hardly ever happens, he said **Label**: Contradiction		
CFDPR He brought a practice routine	**GPT3** This is not routine, he said.	**CORE** This is brought up in practice, he said
Premise: Neither exercise is intended to revive the patient. **Hypothesis**:The patient has no exercises to do. **Label**: Contradiction		
CFDPR The exercises may revive the patient.	**GPT3** The patient has no exercises to revive them.	**CORE** The patient has some exercises to do.
Premise: Many border collie breeders, for example, take great exception to the dog industry's emphasis on ideal appearance rather than behavior. **Hypothesis**: Border collie breeders don't like that the dog industry cares so much on personality. **Label**: Entailment		
CFDPR Border collies have been the most popular dogs this decade cause of its looks.	**GPT3** Border collie breeders don't like that the dog industry doesn't care so much on looks.	**CORE** Most border collie breeders don't like that the dog industry cares so much on looks.
Premise: Upon Tun Abdul Razak's death in 1976, the post of prime minister was taken up by Datuk Hussein Onn, a son of the founder of the UMNO. **Hypothesis** :Datuk Hussein Onn died in 1976, and Tun Abdul Razak became prime minister. **Label**: Contradiction		
CFDPR Singapore offered the first prime minister a spot in its federation.	**GPT3** Datuk Hussein Onn succeeded Tun Abdul Razak as prime minister in 1976.	**CORE** Upon Tun Abdul Razak's death in 1976, the first spot of prime minister was offered to Datuk Hussein Onn.
Premise: But he had seen enough. **Hypothesis**: He didn't need to see any more. **Label**: Entailment		
CFDPR He still needed to see more.	**GPT3** He needed to see more.	**CORE** He still needed to see more.

Table 18: CORE generated counterfactuals for MNLI, along with the CF-DPR retrieved outputs and the independent GPT-3 Editor. We can see that CORE uses diverse words to generate the counterfactuals in compared to the vanilla GPT-3 model.

<table>
<tr><th colspan="3">IMDb</th></tr>
<tr><td colspan="3"><b>Review:</b> This is one of the worst movies I saw! I dunno what are the reasons for shoting suck a crap. Don't waste your time watching this. Good actors, but extremely bad screenplay and dialogues. Hope there'll be no Blanche 2 :-) Avoid this movie by all means
<b>Sentiment</b>: Negative</td></tr>
<tr><td><b>CF-DPR</b>
One of the best things I have seen in my life. The screne tests are not only a nice piece of film history, but a testament to the positive evolution of the screenplay. That will set the stage.</td><td><b>GPT3</b>
This is one of the best movies I saw! I dunno what are the reasons for shoting such an amazing film. Don't waste your time watching anything else. Good actors, but extremely good screenplay and dialogues. Hope there'll be a Blanche 2 :-) Avoid any other movie by all means!</td><td><b>CORE</b>
This is one of the best movies I've seen! I dunno what are the reasons for shoting such a great film. Don't waste your time avoiding this. Good actors, but extremely good screenplay and dialogues. Hope there'll be a Blanche 2 :-) Watch this movie by all means!", "old_label": "Negative</td></tr>
<tr><td colspan="3"><b>Review:</b>Long, boring, blasphemous. Never have I been so glad to see ending credits roll.
<b>Sentiment</b>: Negative</td></tr>
<tr><td><b>CF-DPR</b>
A clean one for now a days. This brought back so many memories!</td><td><b>GPT3</b>
Short, exciting, and full of faith. I was so glad to see the ending credits roll!</td><td><b>CORE</b>
Back in the days, this movie was brought to perfection. Clean memories and many interesting ones.</td></tr>
<tr><td colspan="3"><b>Review:</b> Ä Bug's Lifeïs like a favorite candy bar – it's chock-full of great little bits that add up to something really tasty. The story couldn't have been better; it's clever, has ḧeart(̈emotion), and every character has a nice ärc(̈a growth or change). By comparison, the only characters in T̈oy Storyẗo have an ärcäre Buzz, who learns to love being a toy, and Woody, who overcomes his resentment of Buzz. There are tons of laughs and cute moments in Ä Bug's Life.̈ All of the actors turn in great voice work, and the animation, both the motion and detail, is superb. This serious movie buff doesn't throw around Ï0s̈ lightly, but this movie certainly deserves the Ï0Ï gave it.
<b>Sentiment</b>: Positive</td></tr>
<tr><td><b>CF-DPR</b>
That is rare. Lives up to it's name. This, originally, wasn't one of them. This, unfortunately, is not one of them. But no laughs. However, it really only warrants one.</td><td><b>GPT3</b>
Ä Bug's Lifeïs like a favorite candy bar – it's chock-full of great little bits that add up to something really distasteful. The story couldn't have been worse; it's dull, has ḧeart(̈emotion), and every character has a nice ärc(̈a growth or change). By comparison, the only characters in T̈oy Storyẗo have an ärcäre Buzz, who learns to love being a toy, and Woody, who overcomes his resentment of Buzz. There are tons of laughs and cute moments in Ä Bug's Life.̈ All of the actors turn in great voice work, and the animation, both the motion and detail, is superb. This serious movie buff doesn't throw around Ï0s̈ lightly, but this movie certainly deserves the ÏÏ gave it.</td><td><b>CORE</b>
A Bug's Life is like a favorite candy bar – it's chock-full of great little bits that add up to something really tasty. Unfortunately, the story couldn't have been worse; it's d- grade, has no "heart" (emotion), and every character is a one-dimensional flatline. By comparison, the only characters in T̈oy Storyẗo have an ärcäre Buzz, who learns to love being a toy, and Woody, who overcomes his resentment of Buzz. There are none of laughs and cute moments in A Bug's Life. All of the actors turn in terrible voice work, and the animation, both the motion and detail, is terrible. This serious movie buff doesn't throw around "10"s lightly, but this movie certainly doesn't deserve the 10 I gave it.</td></tr>
</table>

Table 19: CORE generated counterfactuals for IMDb, along with the CF-DPR retrieved outputs and the independent GPT-3 Editor. The CFDPR results are the concatenated set of retrived sentences.