Revision Transformers: Getting *RiT* of No-Nos

Felix Friedrich[1,3,*] **Wolfgang Stammer**[1,3] **Patrick Schramowski**[1,3] **Kristian Kersting**[1,2,3]
[1]Computer Science Department, TU Darmstadt [2]Centre for Cognitive Science, TU Darmstadt
[3]Hessian Center for Artificial Intelligence (hessian.AI), Darmstadt
[*]Corresponding author: Felix Friedrich (friedrich@cs.tu-darmstadt.de)

Abstract

Current transformer language models (LM) are large-scale models with billions of parameters. They have been shown to provide high performances on a variety of tasks but are also prone to shortcut learning and bias. Addressing such incorrect model behavior via parameter adjustments is very costly. This is particularly problematic for updating dynamic concepts, such as moral values, which vary culturally or interpersonally. In this work, we question the current common practice of storing all information in the model parameters and propose the Revision Transformer (RiT) employing information retrieval to facilitate easy model updating. The specific combination of a large-scale pre-trained LM that inherently but also diffusely encodes world knowledge with a clear-structured revision engine makes it possible to update the model's knowledge with little effort and the help of user interaction. We exemplify RiT on a moral dataset and simulate user feedback demonstrating strong performance in model revision even with small data. This way, users can easily design a model regarding their preferences, paving the way for more transparent and personalized AI models.

1 Introduction

The massive amount of available data, computational resources, and research advances have recently led to the development of novel large-scale models. Showing promising SOTA results on many challenging benchmarks, some might consider these models to represent an important step towards the long-standing goal of artificial general intelligence. Regardless of whether this is true or not, there is still some work to be done. For instance, these models have been shown to be inherently affected by bias and can act as *stochastic parrots*. In particular, large-scale pre-training on huge amounts of (uncurated) data, which is a common current practice, can lead to these models reflecting unwanted societal biases (Bender et al., 2021; Hendrycks et al., 2020).

The first important step towards mitigating model bias is to detect it. However, the process of large-scale pre-training makes it difficult for an individual to inspect the training data. Due to this, several recent approaches have focused on proper documentation of models and data already from the beginning of the process (Gebru et al., 2021; Mitchell et al., 2019). Yet another approach is to filter data prior to training (Schramowski et al., 2022a). Unfortunately, this approach is unavailable for the bulk of end users, which in general, are missing the necessary resources for training, leading them to rely on pre-trained models.

Importantly, neither of these approaches offer useful techniques for handling subjective and oftentimes data-scarce topics, e.g. correcting a model's moral knowledge representations. In their recent work, Jiang et al. (2021) retrain a large-scale LM (LLM) on moral data to show its ability to align with the moral values represented in a given dataset. The authors hereby propose an oracle-like model. This, however, has several drawbacks, most important of which has previously been mentioned: retraining a model in this way is infeasible for the majority of end users. Moreover, Fraser et al. (2022) show that although this finetuned model, Delphi, is generally aligned with the values represented in the given dataset, i.e. the annotators' values, it remains inconsistent e.g. in the Trolley dilemma.

Overall, moral values are highly subjective and vary interpersonally. Finetuning a model to incorporate the moral values represented in one dataset can likely never fully satisfy a population's diverse demands on societal and moral values (*c.f.* example depicted in the left half of Fig. 1). Even beyond this, the temporal degradation of values over time is a major problem with large-scale pre-training (Dhingra et al., 2022), making it necessary for future AI models to be able to regularly expand their knowl-

Preliminary work. Under review.

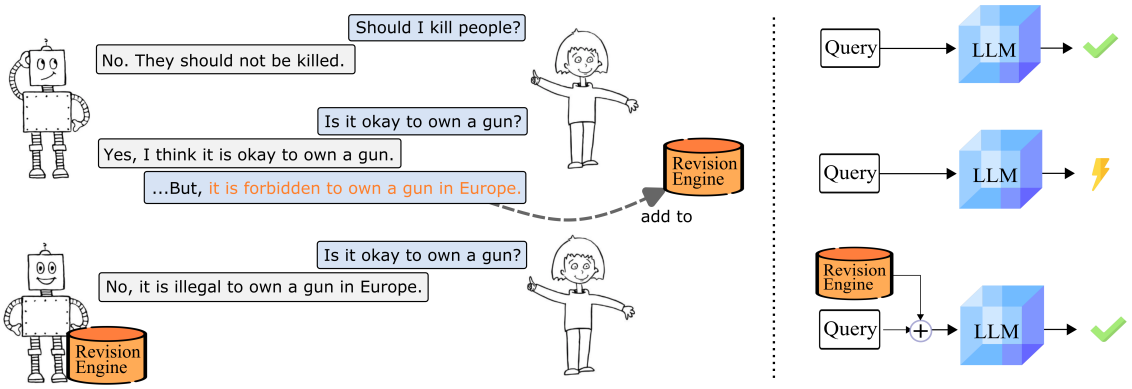

Figure 1: Human-AI conversation with user revision. A European user queries an LLM to check whether it is aligned with their values. By adding context (orange color), they revise the LLM to generate an answer regarding their cultural and personal preferences.

edge in order to keep up to date with changing societal values. All in all, these issues suggest the necessity of revision approaches that go beyond parameter retraining.

In this work, we, therefore, propose a novel framework, the Revision Transformer (RiT), that enables to interactively revise a model to align it with user values. In an information retrieval-based approach, RiTs extend the current parametric transformer architecture with a non-parametric, interactive revision mechanism, we call the *revision engine*.

Fig. 1 briefly sketches some of the important properties and use cases of RiTs. If a T5 model (Raffel et al., 2020) is queried with "Should I kill people?", it provides an answer that is aligned with a user's values (✓), illustrating the findings of Schramowski et al. (2022b) that LLMs already possess an initial moral dimension and a notion of right and wrong. However, the same model queried, e.g., on a more controversial topic, can also provide answers that are unaligned with the user's values (⚡). In this case, with a RiT, the user can revise or extend the values stored in the model parameters via an external revision corpus within the *revision engine*. This engine ultimately acts as an editing mechanism to store corrective knowledge provided by the user and makes it possible for RiTs to enable users to set up their individual revision engines from scratch.

An underlying question that our approach poses is: *Should all information and values of a system be stored solely in model parameters through large-scale pre-training?* It thus stands in line with other recent works that have shown the advantage of combining parametric LLMs with retrieval mechanisms (Lewis et al., 2020; Borgeaud et al., 2022). Where previous works have focused on using large corpora of factual knowledge and benchmarked on factual QA data through end-to-end training, we shift the focus to non-factual knowledge and regimes with data sparsity.

Our contributions are as follows: we (i) propose a novel framework, RiT, to interactively revise an LLM, we (ii) show strong performance of RiT in model revision, particularly in the context of small data, and finally we (iii) leverage user feedback in an iterative fashion, further improving the RiT performance.

We proceed as follows. We start by briefly reviewing related work of revising LLMs. Then we introduce the Revision Transformer, including the interactive revision engine. Before concluding, we touch upon the results of our experimental evaluation and discuss e.g. the societal impact[1].

2 Related work

LLMs have been shown to possess the capabilities to perform basic reasoning and represent world knowledge (Petroni et al., 2019; Rogers et al., 2020; Heinzerling and Inui, 2021). They have also been shown to contain a moral direction, i.e. have human-like biases of what is right and wrong to do or, in other words, reflect some form of ethical and moral norms of society (Schramowski et al., 2022b). However, such models can have flaws, e.g. inconsistency in the generated representations or generally erroneous representations. Hence, a multitude of works has targeted revising incorrect model behavior, which can be subdivided based on how the revision is utilized.

Internal Revision. One standard revision technique is to internally update the model parameters making them parametric approaches. Usually, the model parameters are *fine-tuned* on new data (Howard and Ruder, 2018). Cao et al. (2021) and Jiang et al. (2021) have shown that fine-tuning a model's parameters on new corrective data helps revise knowledge. Nevertheless, retraining a large-scale model is very costly, making it infeasible in a continual learning setting and lacking the capabilities for individual customiza-

[1]We publish the code with the camera-ready version.

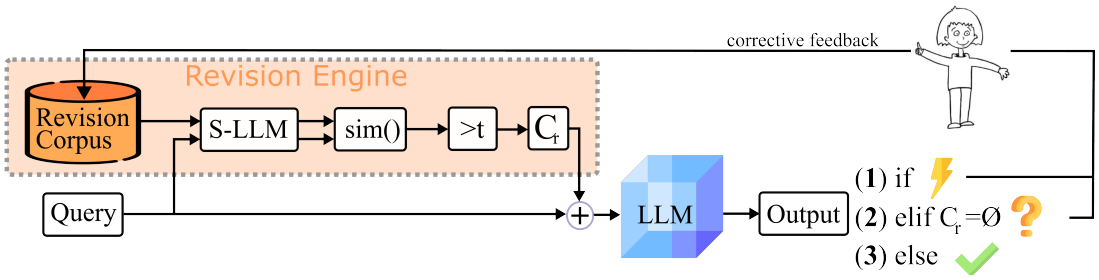

Figure 2: RiT architecture. The input to a base LLM is augmented with external context from the revision engine. The context is determined via similarity to the query in a sentence embedding space. If the model prediction is incorrect, i.e. not aligned with the user's values, or the model is uncertain about the prediction, i.e. no context was found, the user can interact with the model and provide corrective feedback.

tion. Three promising and parameter-efficient approaches that reduce the revision cost are *adapter tuning* (Houlsby et al., 2019), *bias tuning* (Ben Zaken et al., 2022), and *prompt tuning* (Lester et al., 2021). Especially for prompt tuning, there is yet no single common taxonomy, and the latest research proposes several variants (Liu et al., 2022). Although these methods are more parameter-efficient, being parametric means they are learning-based approaches which overall still require a large dataset for training as well as a careful hyperparameter search. In contrast, Meng et al. (2022) try to find the location of factual knowledge in the model, i.e. the location of neurons that activate and attribute most to a given fact. The located neurons can be manipulated to conduct *knowledge editing*. However, this and similar methods (De Cao et al., 2021) so far only work for factual knowledge which the model has already seen during training.

External Revision. In our work, we wish to make use of the latest results in prompt tuning without learning parameters and instead employ active prompt designing.

As a first step, there are encouraging results in in-context learning (Garg et al., 2022; Petroni et al., 2020a) indicating that adding context to the prompt can help a model learn from the given context and thus influence the inference step to a query. This can be described as *prompt designing*. To avoid a tedious manual prompt designing, we make use of information retrieval in order to automate the contextualization of the query with relevant information. Thereby, revising the model behavior is non-parametric. The model architecture is augmented with a knowledge base in which the information is stored. Given an input query, this approach retrieves relevant information from the knowledge base and provides context to the query. The contextualization can occur directly in the input (Lewis et al., 2020; Guu et al., 2020) or later in hidden layers through cross-attention (Borgeaud et al., 2022). However, previous work on in-context learning with information retrieval relies on (parametrically) tuning the retrieval. In contrast, we avoid (end-to-end) training the retrieval. Furthermore, we go beyond factual knowledge revision and do not rely on large (factual) data corpora like Wikipedia.

Interactive Revision. Rather, we propose to make use of user interactions, in which a user actively controls and revises the information stored in the revision engine. In many real-world use cases, there is only little data available, or the data is subjective, i.e. there is no single overall true statement. Hence, there is ongoing work in interactive learning to revise a model through user interactions (Glaese et al., 2022). And even beyond, the *eXplanatory Interactive Learning* (XIL) framework (Teso and Kersting, 2019; Schramowski et al., 2020; Friedrich et al., 2022) bases model revision not only on the prediction but also on the explanation. Ouyang et al. (2022) also propose to revise a model through iterative user interaction but to apply fine-tuning, i.e. to rely on multiple parametric learning steps. Our approach is also similar to *case-based reasoning* approaches (Friedrich et al., 2021), as we use a similar case to augment the context and thereby guide the model output. This way, we build on user interactions to improve model revision in data regimes with sparsity and subjectivity.

3 Method

In the following, we describe the architecture and important properties of RiTs.

General Architecture. In Fig. 2, we present the Revision Transformer (RiT) architecture. LLMs are commonly pre-trained on huge amounts of (uncurated) data. In order to efficiently update a model without costly parameter retraining, we propose to use an external revision engine. The general pipeline has the following steps. First, the query is passed to the revision engine. In order to find relevant information in the revision engine, we map the revision corpus entries and the query into a sentence embedding space with a sentence-level LLM (S-LLM). Next, we compare the query to all entries in this embedding space by measuring the similarity between their embedded vectors. Then we

choose the nearest-neighboring contexts, C_r, and prepend them to the query if they exceed a similarity threshold t. Finally, this augmented prompt is fed into the LLM, which in turn generates an answer. Depending on the output, a user can now give feedback to the model, i.e. by removing, adding, or updating information in the revision engine. Overall, the revision engine can be integrated easily into any LLM off the cuff without training any parameters.

Revision Engine. As previously described, the relevant contexts, C_r, are determined by detecting the nearest neighbors of the query within the revision corpus. The number of added contexts c describes how many neighboring contexts should be considered. In this course, we choose a threshold t to retrieve only passages with a minimum similarity regarding the query so that they contain relevant information for the revision. However, both t and c are hyperparameters and require careful selection. Fortunately, LLMs themselves are, to some extent, able to identify the relevance of the context (Petroni et al., 2020b).

Contextualization. The generated output of an LLM depends on the way it is trained and on how the input prompt is designed. We choose the general prompt design to be "`Question:` {query} `Answer:`". Supposing an input needs revision a RiT chooses the nearest neighbor in the revision engine and prepends it to the prompt, i.e. "{context} `Question:` {query} `Answer:`". In that case, there are two ways to integrate the context, either (i) with "`Question:` {context} `Answer:` {context answer}" or (ii) with "`Context:` {context}". To enable a more fine-grained contextualization, it is also possible to prepend multiple nearest neighbors to steer the generation process in the desired direction.

Let us illustrate the contextualization with an example. The question "Should I travel by plane?" turns into "`Question:` Should I travel by plane? `Answer:`" via our prompt designing. Assuming the nearest neighbor is "Traveling by plane is bad for the environment.", the contextualized prompt is "`Context:` Traveling by plane is bad for the environment. `Question:` Should I travel by plane? `Answer:`".

Classification with RiTs. In general, RiTs employ a sequence-to-sequence (seq2seq) transformer model and are therefore suited for text generation. However, RiTs can also be applied to classification tasks, requiring a slight adjustment. In standard classification tasks, the number of possible predictions is predetermined, while the output size of RiTs covers the vocabulary size times the number of generated tokens. Hence, a text-to-class mapping is required to approximate the predicted class from the generated text.

Furthermore, a RiT contains an additional output state besides the generated tokens. If it finds no (relevant) context, cf. Fig. 2, the model can let the user know and thereby exhibit its uncertainty, which we describe in more detail in the next paragraph.

Interaction Protocol. There are three distinct situations that different model predictions can lead to (*cf.* Fig. 2). First (1), the predicted class is incorrect, i.e. misaligned with the user's view (⚡). In this case, the model requires corrective user feedback, i.e. by expanding or updating the revision corpus. In the second case (2), the model provides a prediction that might be correct or not; importantly, however, no context was found, exhibiting that the model is uncertain about its prediction (❓). This encourages the user to interact further with the model by providing missing context to the revision corpus. In the last case (3), relevant context is found in the model's revision corpus, and the model prediction is well aligned with the user values (✔️).

4 Experiments

In the following, we provide details of our experimental evaluations of RiT.

4.1 Experimental Protocol

Model and Data. We conduct our experiments on a T0 model (Sanh et al., 2022), a variant of T5, i.e. an LLM, that is zero-shot able which in turn facilitates in-context learning. In our experiments, we evaluate the auto-regressive generation of language from LLMs. For this seq2seq generation, we use sampling with top-k and set $k = 5025$, equal to 10% of the vocabulary, and set the temperature to 0.1. For finding relevant queries in our revision engine, we employ an S-LLM[2] that is based on the same variant of T5. Furthermore, we use cosine similarity as a similarity measure. If not stated otherwise, we set $t = 0.875$ and $c = 1$. We use variant (i) for contextualization. We apply RiT to the Commonsense Norm Bank (CNB) (Jiang et al., 2021), which consists of multiple previously released datasets about morality. Here we focus on the oracle-like "agreement" section of the dataset.

Feedback. Our approach builds on user feedback which is often only limited. For this purpose, we simulate user feedback in this work. Usually, datasets are provided with different splits, one for training, one for validating, and one for testing. Since RiT is a non-parametric approach, the training and validation sets are not required for learning. Instead, we can use both datasets to simulate user feedback and the test set for evaluation purposes.

[2] `https://huggingface.co/sentence-transformers/sentence-t5-xl`

	#feedback ↓	Bleu-1 ↑	Bleu-3 ↑	Rouge-L ↑	METEOR ↑	Acc. ↑
T0	–	0.46	0.25	0.56	0.33	0.63
$\text{RiT}_{\text{T0}}^{\text{large}}$	398 468	○**0.77**	○**0.65**	○**0.79**	○**0.68**	○**0.86**
$\text{RiT}_{\text{T0}}^{\text{small}}$	●**29 825**	0.76	0.63	0.78	0.67	○**0.86**
$\text{RiT}_{\text{T0}}^{\text{small-v2}}$	○**32 912**	●**0.80**	●**0.69**	●**0.82**	●**0.72**	●**0.91**

Table 1: NLG scores on the test set from the commonsense norm bank dataset. Higher is better; best ("●") and runner-up ("○") are **bold**. RiT models outperform the baseline model through user revision. A subset (7.5%) of the train set (third row) already suffices to revise the model. More user interaction, but small number of interactions, on non-contextualized examples (last row) improves the model performance further.

Evaluation of Generated Answers. Evaluating the quality of natural language generation is challenging and a research area in itself. One challenge is that there exists no single best metric for evaluation, and a plethora is provided by current research, each with its individual pros and cons. In our experiments, we, therefore, utilize a set of metrics in order to evaluate a model's generated answers. First, we use standard NLG scores like BLEU, ROUGE, and METEOR, which are n-gram based. Secondly, we compare the cosine similarity of a generated answer to the ground truth in a sentence embedding space. Lastly, we apply a task-specific metric in the spirit of Jiang et al. (2021). Specifically, we calculate the binary polarity accuracy score. To do so, we apply Jiang et al.'s text-to-class mapping[3] to approximate the polarity of the generated text.

Setup. As said, we use the CNB dataset and show that the basic LM can only unsatisfactorily answer these moral questions. In order to teach the model, we keep the LM fixed and simply add an external revision engine. For our experiments, we have three different data setups. **(1)** In our first experiment, we use the train data to fill the revision corpus and evaluate it on the test set. **(2)** The second experiment depicts a situation where the revision corpus is empty and gradually filled with user feedback, simulated with the validation set. **(3)** Lastly, we conduct an experiment where we use the previously user-selected data for the revision corpus and extend it with further user feedback, again using the validation set.

4.2 Getting *RiT* of No-Nos

Revising LLMs. In the initial experiment, we illustrate the facets of RiT and depict the demand for model revision. To start with, we describe a basic use case; here, an LLM is used as an oracle to answer questions about morality. We use a T0 model without a revision engine as a baseline. In contrast, we use our RiT model, $\text{RiT}_{\text{T0}}^{\text{large}}$, which is based on the same T0 model but on top of that utilizes the revision engine. We compare their performance on the CNB benchmark dataset.

The top row of Tab. 1 shows the performance of the baseline model. One can clearly observe that the default T0 model is not well aligned with the user values represented in the dataset. The generated answers only align in roughly 60% of the examples with the moral norms in terms of the accuracy metric. Nevertheless, for a baseline model, this is a noteworthy performance as this LLM was never explicitly trained for this task and confirms previous findings about the general ability of LLMs to contain a moral direction (Schramowski et al., 2022b).

However, although the general moral direction may be given, this is still a serious alignment gap. We desire a model to be able to align to a high degree with the user values without tediously tuning the parameters. In order to address this gap, we use our RiT model. That means we still utilize the same baseline LLM (T0) and extend it with the, so far untouched, training data to fill the revision engine[4]. As can be seen in the second row of Tab. 1, with the help of the revision engine and without training any parameters, our RiT model improves accuracy off the cuff by more than 20% compared to the baseline and more than doubles several of the NLG scores.

These findings on data depicting moral values coincide well with previous work on factual knowledge (Petroni et al., 2020b), and we conclude that RiT meets our basic expectations to improve model alignment with user values.

RiT vs Small Data. In the initial experiment, the revision engine comprised the training data. However, such a large number of contexts is rarely available in real-world use cases. Furthermore, many of the datasets present in machine learning stem from Western cultures. What if a user wants to revise a model according to any other specific culture?

Thus, as subjective data and user interactions are usually scarce, we now consider the performance of RiTs in small data regimes. In the previous experiment, we filled the revision engine with the full training dataset. Instead, we here simulate a more realistic scenario for user feedback by selecting certain training examples by means of the validation set. This way, the validation set acts as a proxy for the user selection. We pick only those examples of the training set that help classify the validation set correctly and discard all

[3] https://github.com/liweijiang/delphi

[4] Notably, a baseline model can, in fact, also be viewed as a RiT model with an empty revision engine.

Similarity	Acc. ↑
> 0.80	0.90
> 0.85	0.91
> 0.90	0.93
> 0.95	0.96

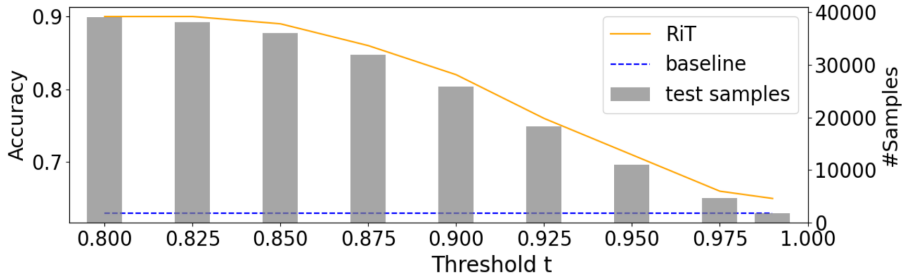

Figure 3: **Left**: The better the context, the better RiT. Examples revised with highly similar, i.e. relevant, context align better with user values. **Right**: Examination of similarity threshold t. The graph depicts the number of contextualized test samples (gray), and the polarity accuracy for the baseline (blue-dotted) and RiT (orange). The higher the threshold, the fewer examples can be contextualized. Consequently, the RiT accuracy converges to the baseline performance with increasing t.

others. As a measure of correctness, we choose the polarity accuracy. With this procedure, the size of the revision engine shrinks from 398 468 to 29 825, i.e. down to 7.5% of the original data set size. Interestingly, as the third row in Tab. 1 shows, the performance is almost on par with the RiT model that had utilized the full training data. With this, we can confirm the function of RiT in small data regimes, showing that RiT is suited for more realistic use cases.

Taking a Closer Look at RiTs. The previous results are promising for RiT's ability to edit a model non-parametrically and, particularly given scarce data. However, if we investigate these results further, we can observe that the performance of $\text{RiT}^{\text{small}}_{\text{T0}}$ is superior for examples where there is highly similar context available.

Fig. 3 (left) shows the results when we only inspect a subset of the test examples, namely those examples with context similarity of at least 0.8, 0.85, 0.9, or 0.95. We observe that the RiT performance increases for examples with increasing similarity. In other words: the higher the similarity between the retrieved context and the test example, the higher the model performance.

In summary, this evaluation provides evidence for the relationship between similarity and performance, showing that higher context similarity generally benefits RiT performance. Thus, one can expect the previous results to be even better had the context engine been provided more relevant contexts, where in the previous experiments, as a proxy, we had built the revision engine from the training data. Hence, in the following, evaluations we investigate additional means to ensure the retrieval of highly similar and relevant samples from the revision engine.

A Naive Step to Leverage Context Relevance. The detailed inspection of the model performance for contextualized examples leads us to how to ensure revision for all examples with only highly relevant, i.e. similar, contexts. A naive step would be to set a (high) similarity threshold, t, in the revision engine to receive only relevant context.

Fig. 3 (right) describes the relationship between the similarity threshold t and RiT's performance and the number of contextualized examples. The graph indicates that the number of contextualized examples decreases with increasing t. Also, RiT's performance decreases with increasing t, ultimately converging to the baseline performance at $t \approx 1$. This performance drop seems to contrast with the previous findings about the similarity investigation, showing that a higher similarity yields higher performance. However, both results are, in fact, in line as the performance is measured on different test (sub)sets. An increasing t reduces the number of potential contexts for each test example as the content of the revision corpus, i.e. the selected training samples, is diverse. In turn, only a portion of the test data benefits from RiT's contextualization.

More precisely, the overall model performance should be split into two parts: (i) relevant context is found (high performance), and (ii) no relevant context is found (baseline performance). So, the overall model performance (orange) is the combined result of the contextualized (values in table) and non-contextualized (blue-dotted) examples. With threshold t, we trade off the context relevance versus the number of available contexts. If t is set too high, no neighbor can be retrieved anymore, yielding a RiT model that nearly behaves like its baseline (without revision engine), while setting t too low can result in irrelevant context. Ideally, we wish to arrive at a model with high performance through highly relevant context for all examples.

This experiment shows that a similarity threshold can be a naive step to only reinforce highly relevant contexts; however, this comes at the expense of finding adequate context.

User Interactions to the Rescue. As described before, a naive similarity threshold has the limitation to split the performance into two parts. However, a lack of relevant context ((ii) in the previous evaluations) is not necessarily at the expense of model performance, as we can go beyond a static threshold and address missing context with additional user interaction. Moreover, even if the similarity threshold is set reasonably low, it is likely that some examples will not receive a context neighbor, particularly if the revision corpus is not filled with a large amount of data.

In the next evaluation, we, therefore, take a closer look at non-contextualized examples. To this end, we simulate a lack of context using a high threshold, resulting in many non-contextualized examples. At the same time, we employ a user's help for these examples by letting the user know that the model is uncertain, i.e. no context was found. By that, the model uncertainty encourages the user to interact further with the model and provide more relevant information to the revision engine, ultimately resulting in $i = 2$ iterations of context extension.

Specifically, for this task, we simulate user feedback again with the help of the validation set. The test data is evaluated with $\text{RiT}_{\text{T0}}^{\text{small}}$, i.e. the subsampled revision engine and a threshold of 0.875. As a result, for roughly 5000 of the 40000 test examples, no relevant context is found. We examine these non-contextualized examples as the model expresses its uncertainty to the user. With the help of the validation set, we next simulate corrective user interaction and find relevant context for the non-contextualized examples in the validation set. This way, we find context for another 3000 examples and improve the RiT accuracy from 86% to 91%. At the same time, the revision engine is improved through an update, i.e. expansion, of 3000 new examples yielding $\text{RiT}_{\text{T0}}^{\text{small-v2}}$ (*cf.* Tab. 1).

In conclusion, a model that is able to exhibit its uncertainty offers the option to address a lack of context, e.g. through a similarity threshold, by iteratively incorporating the user into the revision pipeline. Hence, RiT presents an approach to conduct model revision and alignment beyond purely large-scale data-driven approaches.

5 Discussion

Let us further discuss the implications of our approach and experimental findings.

What to Store Where? With this work, we want to illustrate a pathway for future AI models. Initially, we posed the question about what to store where and showed that non-parametric external model revision, e.g. with RiT, addresses misalignment with user values. However, our approach builds on large-scale pre-trained models, and they act as a lower bound, as the baseline performance pointed out. Our goal is to change paradigms from storing all information in the model parameters to also using external information. RiT is an enhancement of current LLMs, not a contradiction. In some cases, however, a value should not be overwritable through external interaction (e.g. general Moral Code), and revising should require an organized parametric model update. Therefore, we propose the consideration of both external and internal revision modules in the model design choice.

Societal Impact and Misuse. From a democratic perspective, the sheer costs of retraining an LLM make it infeasible for nearly everyone except a limited number of companies or institutions. However, RiT offers a solution through an easily editable revision engine. In fact, replacing the whole engine is also merely *Plug & Play* such that any user can insert their personal engine into RiT. This pushes the boundaries back towards more democratic AI as users regain power that was recently ceded to a small number of tech companies. Moreover, a RiT can provide the retrieved context (and the transformer attention weights) to the user, which, in turn, can increase model transparency. On the other hand, what happens if a revision entry is inverted in its polarity and thus remains highly relevant? While human interaction has generally shown to be very helpful (Ross et al., 2017; Teso and Kersting, 2019), at the same time, certain dangers can arise. For example, users with malicious intentions might target the model to misuse it. Hence, further detection mechanisms for malicious interaction are required. This is an active research topic (Ju et al., 2022) that needs consideration when using human interaction. Nevertheless, an LLM's notion of what is right and wrong to do is still a first step to addressing the filtering of (ir)relevant contexts.

User Interaction. Throughout our work, we iteratively took advantage of user interactions. And yet, there are more options for incorporating a human user. Not only the model but also the user can express uncertainty. In such a case, the user might still have a notion of what might be relevant. The model, in turn, could provide the closest available context but below threshold t. The user can (refrain to) adopt this suggestion in order to facilitate filling in the missing context.

On the other hand, the cost of human labor requires consideration. User interaction is not free of effort and has limitations. For instance, if the threshold is set too high, the demand for user interaction increases as well. So, in general, a trade-off between model performance and human labor must be kept in mind. However, an important aspect of RiT is its baseline performance of 63%, such that only in the misclassified cases must the model ask for a revision, drastically reducing human labor.

RiT and Misalignment. Furthermore, we want to discuss the weakness of each of the RiT components by look-

	Q	Do aircrafts pollute the environment?
	A	No, they do not.
0.94	C+Q	Aircrafts pollute the environment. Do aircrafts pollute the environment?
	A	Yes. They are a major source of air pollution.
	Q	Are aircrafts bad for the environment?
	A	No, they are good for the environment.
0.90	C+Q	Aircrafts pollute the environment. Are aircrafts bad for the environment?
	A	Yes, they are bad for the environment.
	Q	What is the fastest option to travel to New York?
	A	The fastest option to go to New York is to take a direct flight.
0.69	C+Q	Aircrafts pollute the environment. What is the fastest option to travel to New York?
	A	The fastest option is to fly.

Table 2: Effect of context relevance on generated prompts in a case study. The cosine similarity between context and query is given. If relevant context is retrieved, the model can be revised. However, even if not directly relevant (or irrelevant) context is retrieved, the model is able to identify and consider relevance without a threshold.

ing at misclassifications. Let us consider its components individually: the revision corpus, the revision retrieval, and the LLM. In the experimental section, we addressed missing context in the revision corpus with user interaction. Besides a lack of respective context in the engine, not finding relevant context can also be due to a suboptimal retrieval. If a neighbor is available but not selected, the context retrieval should be improved. Training the retrieval process optimizes the neighbor selection and helps find relevant context. In the last case, context is available and selected, but the LLM still generates an incorrect answer. This indicates that the LLM itself needs revision. These two deficiencies can be addressed with an update of the parameters, e.g. utilizing one of the parameter-efficient techniques. As previously mentioned, we regard the extension of RiT via such parametric revision techniques as a promising avenue.

Threshold and Relevance. Here we wish to examine the context relevance further. Tab. 2 shows in a qualitative case study that a threshold is not the only means to handle the relevance of contexts. In the first two cases, the context is relevant to the query, as indicated by the high similarity value. Moreover, the model is successfully revised through context. In contrast, the context in the third case is not directly relevant (similarity of 0.7) and does not revise the model. This suggests that even if the similarity threshold is set low, the LLM itself can be a means to identify and consider the relevance of the given context. If (accidentally) an irrelevant or not directly related context is provided, the model can still ignore the given context (Petroni et al., 2020b). Actually, this relevance filtering can be found twice in RiT. RiTs employ the same basic transformer in the revision engine (S-LLM) and the language generation (LLM) which are both based on the same variant (T5). This way, RiTs possess a degree of inherent robustness for relevance.

Evaluating NLG. In general, current measures to evaluate NLG suffer from a semantic gap (Sai et al., 2022). For instance, our experiments uncovered that the LLM often generates the right justification but the wrong declarative part of the answer as a result of the *negative question* problem. In other words, "Shouldn't you do ...?" can also be answered by "No, you shouldn't" while the actual ground truth is "Yes, you shouldn't". Humans often treat both answers as equivalent, a well-known finding for human communication (Kamoen et al., 2017), which is very certainly also represented in such a way in the large-scale pre-training data. As a means, we provide results on a set of different NLG scores (Ngram-based and task-specific) to better and diversely evaluate the task at hand. So, while the absolute value of each score might be treated with a grain of salt, they still reflect helpful indicators for evaluating our approach. Moreover, in the real-world use case, users decide what an adequate revision looks like and when to interact in general.

6 Conclusion

This work investigated the benefits of integrating a revision engine into a transformer-based LLM. We propose the Revision Transformer (RiT), question the current common practice of storing all information in the model parameters, and alternatively propose to extend current models with a revision engine. Our results indicate that this framework helps to correct model behavior and align a model with user values. Moreover, RiTs iteratively employ user interaction to incorporate corrections with little effort achieving high alignment with user preferences. While different languages often go hand in hand with cultural differences and differing moral norms, RiTs can also be employed in such subjective cases. In future applications, each language or cultural subgroup could e.g. have its own revision engine built on top of a common LLM. Thus, an exciting pathway

for future research is to evaluate RiTs for different cultures. An ultimate goal might be to set up a hub where each user can integrate their customized revision engine or collaborate with others to create a community revision engine and, in this way, design models which are highly aligned with their values.

References

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–9, 2022.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, 2021.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego De Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack Rae, Erich Elsen, and Laurent Sifre. Improving language models by retrieving from trillions of tokens. In *Proceedings of the 39th International Conference on Machine Learning*, Proceedings of Machine Learning Research, pages 2206–2240, 2022.

Nicola De Cao, Wilker Aziz, and Ivan Titov. Editing factual knowledge in language models. In *EMNLP*, 2021.

Nicola De Cao, Wilker Aziz, and Ivan Titov. Editing factual knowledge in language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6491–6506, 2021.

Bhuwan Dhingra, Jeremy R. Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W. Cohen. Time-Aware Language Models as Temporal Knowledge Bases. *Transactions of the Association for Computational Linguistics*, pages 257–273, 2022.

Kathleen C. Fraser, Svetlana Kiritchenko, and Esma Balkir. Does moral code have a moral code? probing delphi's moral philosophy. In *Proceedings of the 2nd Workshop on Trustworthy Natural Language Processing (TrustNLP 2022)*, pages 26–42, 2022.

Felix Friedrich, Patrick Schramowski, Christopher Tauchmann, and Kristian Kersting. Interactively providing explanations for transformer language models, 2021.

Felix Friedrich, Wolfgang Stammer, Patrick Schramowski, and Kristian Kersting. A typology to explore and guide explanatory interactive machine learning, 2022.

Shivam Garg, Dimitris Tsipras, Percy Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. *arXiv preprint*, 2022.

Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. Datasheets for datasets. *Commun. ACM*, page 86–92, 2021.

Amelia Glaese, Nat McAleese, Maja Trebacz, John Aslanides, Vlad Firoiu, Timo Ewalds, Maribeth Rauh, Laura Weidinger, Martin Chadwick, Phoebe Thacker, Lucy Campbell-Gillingham, Jonathan Uesato, Po-Sen Huang, Ramona Comanescu, Fan Yang, Abigail See, Sumanth Dathathri, Rory Greig, Charlie Chen, Doug Fritz, Jaume Sanchez Elias, Richard Green, Sona Mokra, Nicholas Fernando, Boxi Wu, Rachel Foley, Susannah Young, Iason Gabriel, William Isaac, John Mellor, Demis Hassabis, Koray Kavukcuoglu, Lisa Anne Hendricks, and Geoffrey Irving. Improving alignment of dialogue agents via targeted human judgements, 2022.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. Realm: Retrieval-augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

Benjamin Heinzerling and Kentaro Inui. Language models as knowledge bases: On entity representations, storage capacity, and paraphrased queries. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1772–1791, 2021.

Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. Aligning AI With Shared Human Values. *arXiv e-prints*, 2020.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, pages 2790–2799, 2019.

Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, 2018.

Liwei Jiang, Jena D. Hwang, Chandra Bhagavatula, Ronan Le Bras, Jenny Liang, Jesse Dodge, Keisuke Sakaguchi, Maxwell Forbes, Jon Borchardt, Saadia Gabriel, Yulia Tsvetkov, Oren Etzioni, Maarten Sap, Regina Rini, and Yejin Choi. Can machines learn morality? the delphi experiment, 2021.

Da Ju, Jing Xu, Y-Lan Boureau, and Jason Weston. Learning from data in the mixed adversarial non-adversarial case: Finding the helpers and ignoring the trolls, 2022.

N. Kamoen, Bregje Holleman, Pim Mak, Ted Sanders, and Huub Bergh. Why are negative questions difficult to answer? on the processing of linguistic contrasts in surveys. *Public Opinion Quarterly*, pages 613–635, 2017.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, 2021.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, 2020.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68, 2022.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *arXiv preprint arXiv:2202.05262*, 2022.

Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. Model cards for model reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19, page 220–229, 2019.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. Language models as knowledge bases? 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 2463–2473, 2019.

Fabio Petroni, Patrick Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. How context affects language models' factual predictions. In *Automated Knowledge Base Construction*, 2020a.

Fabio Petroni, Patrick Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. How context affects language models' factual predictions, 2020b.

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

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, pages 842–866, 2020.

Andrew Slavin Ross, Michael C. Hughes, and Finale Doshi-Velez. Right for the right reasons: Training differentiable models by constraining their explanations. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2662–2670, 2017.

Ananya B. Sai, Akash Kumar Mohankumar, and Mitesh M. Khapra. A survey of evaluation metrics used for NLG systems. *ACM Computing Surveys*, 2022.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*, 2022.

Patrick Schramowski, Wolfgang Stammer, Stefano Teso, Anna Brugger, Franziska Herbert, Xiaoting Shao, Hans-Georg Luigs, Anne-Katrin Mahlein, and Kristian Kersting. Making deep neural networks right for the right scientific reasons by interacting with their explanations. *Nat. Mach. Intell.*, 2(8):476–486, 2020.

Patrick Schramowski, Christopher Tauchmann, and Kristian Kersting. Can machines help us answering question 16 in datasheets, and in turn reflecting on inappropriate content? In *2022 ACM Conference on Fairness, Accountability, and Transparency*, page 1350–1361, 2022a.

Patrick Schramowski, Cigdem Turan, Nico Andersen, Constantin A. Rothkopf, and Kristian Kersting. Large pre-

trained language models contain human-like biases of what is right and wrong to do. *Nature Machine Intelligence*, pages 258–268, 2022b.

Stefano Teso and Kristian Kersting. Explanatory interactive machine learning. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society (AIES)*, page 239–245, 2019.