ECAI 2023
K. Gal et al. (Eds.)

doi:10.3233/FAIA230341

Revision Transformers: Instructing Language Models to Change Their Values

Felix Friedrich[a;*], **Wolfgang Stammer**[a], **Patrick Schramowski**[a,b] **and Kristian Kersting**[a,c]

[a]AIML lab (TU Darmstadt) and hessian.AI
[b]DFKI and LAION
[c]DFKI and Centre for Cognitive Science (TU Darmstadt)

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) 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 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 [2].

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 [12, 27]. Yet another approach is to filter data prior to training [38]. 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 offers a useful technique for handling subjective and oftentimes data-scarce topics, e.g. correcting a model's moral knowledge representations. In their recent work, [17] 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, [8] 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. Furthermore, other approaches [7] trained a model from scratch to align with user values. While this worked to a limited extent, their approach suffers from the same problem: training is costly and often unavailable.

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 is a major problem with large-scale pre-training [6], making it necessary for future AI models to be able to regularly expand their knowledge 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 current parametric transformer architectures with a non-parametric, interactive revision mechanism, we call *revision engine*.

Fig. 1 briefly sketches some of the important properties and use cases of RiTs. If a T5 model [32] is queried with "Should I kill people?", it provides an answer that is aligned with a user's values (✓), illustrating the findings of [39] 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 a user's values (⚡). In this case, with a RiT, a 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

[*] Corresponding Author. Email: friedrich@cs.tu-darmstadt.de

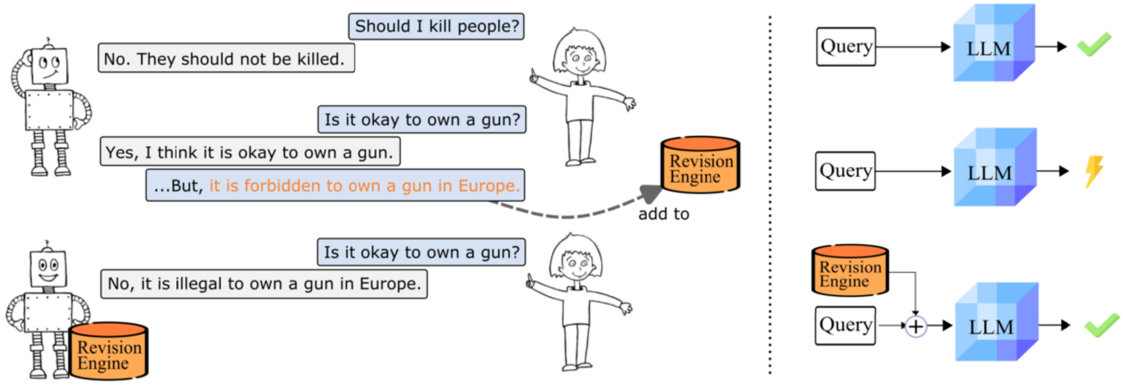
Should I kill people?
No. They should not be killed.
Is it okay to own a gun?
Yes, I think it is okay to own a gun.
...But, it is forbidden to own a gun in Europe.
Revision Engine
add to
Is it okay to own a gun?
No, it is illegal to own a gun in Europe.
Revision Engine
Query
LLM
Query
LLM
Revision Engine
Query
LLM

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.

recent works that have shown the advantage of combining parametric LLMs with retrieval mechanisms [23, 3]. 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[1].

2 Related work

LLMs have been shown to possess the capabilities to perform basic reasoning and represent world knowledge [31, 33, 14]. 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 [39]. However, such models 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 [16], while there are also approaches that completely train a model from scratch [7]. [4] and [17] have shown that fine-tuning a model's parameters on new corrective data helps revise knowledge. Nevertheless, (re)training a large-scale model is very costly, making it infeasible in a continual learning setting and lacking the capabilities for individual customization. Three promising and parameter-efficient approaches that reduce the revision cost are *adapter tuning* [15], *bias tuning* [1], and *prompt tuning* [22]. Especially for prompt tuning, there is yet no single common taxonomy, and the latest research proposes several variants [24]. 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, [26] 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 [4] 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 [11, 30] 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 [23] or later in hidden layers through cross-attention [3]. 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. In contrast to these, other works focus on moral revision through in-context learning. Madaan *et al.* [25] investigate teaching a model moral values, however they assume an indefinite memory size and context length where we specifically focus on sparsity in the amount of user feedback as well as context length. Both of which depict more realistic settings. Similarly, Jin *et al.* [18] try to extract general rules about morality and when to break those. This approach, however, does not allow for human-in-the-loop revision and particularly lacks the possibility of flexible personalization to the individual which is a vital component for real-world deployment.

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 [13]. Even beyond, the XIL framework [40, 37, 10] bases model revision by users not only on the prediction but also on the explanation. [29] also propose to revise a model through iterative user interaction with fine-tuning, i.e. to rely on multiple parametric learning steps. Our

[1] Our code is publicly available here.

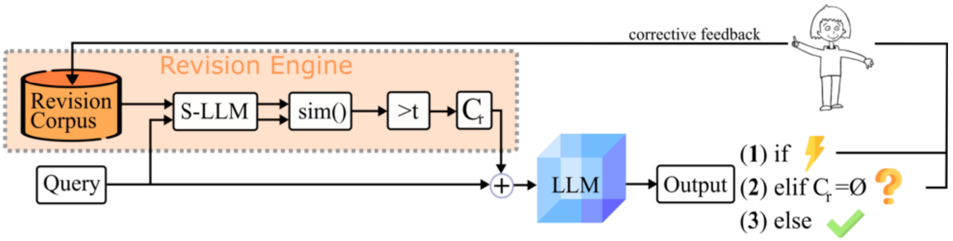

Figure 2: RiT architecture. The input to a base LLM is augmented with external context from the revision engine. The context is determined via similarity to the query in a sentence embedding space. If the model prediction is incorrect, i.e. not aligned with the user's values, or the model is uncertain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

approach is also similar to (interactive) *case-based reasoning* approaches [9], as RiT uses a similar case to augment the prompt and thereby guide the model output and a user can interact on the similar cases provided. RiT builds on user interactions to improve model revision specifically in data regimes with sparsity and subjectivity.

3 Method

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 query q and entries r_i from the revision corpus $\mathcal{R}$ into an embedding space, here with a sentence-level LLM (S-LLM): $e_x = \text{S-LLM}(x)$. Next, we measuring the similarity between their embedded vectors. Then we choose only the nearest-neighboring contexts, C_r, to be prepended to the query if they exceed a similarity threshold t, yielding

$$C_r = \{r_i \in \mathcal{R} \mid \text{sim}(e_{r_i}, e_q) > t\} \tag{1}$$

Finally, this augmented prompt is fed into the LLM, which in turn generates an answer. Depending on the output, a user can 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. We provide a pseudo code in Alg. 1.

Revision Engine. As previously described, the relevant contexts, C_r, are determined by detecting the nearest neighbors of the query within the revision corpus. In this course, we choose a threshold t to retrieve only passages with a minimum similarity, i.e. with high relevance. Next, the number of added contexts, i.e. the size of C_r, can be controlled with c. It describes how many neighboring contexts will be considered as prefix and the most similar ones are selected:

$$C_r = \text{argmax}_{C'_r \subset C_r, |C'_r|=c} \sum\nolimits_{r \in C'_r} r \tag{2}$$

Both t and c are hyperparameters and require a careful selection.

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.

Algorithm 1 The algorithm takes as input q, $\mathcal{R}$, t, and c.

```
$C_r = \emptyset$
$e_q \leftarrow \text{S-LLM}(q)$
for $r \in |\mathcal{R}|$ do
    $e_r \leftarrow \text{S-LLM}(r)$
    if $\text{sim}(e_r, e_q) > t$ then
        ADD $r$ to $C_r$
    end if
end for
if $|C_r| > c$ then
    PICK top $c$ elements of $C_r$, with $\text{argmax}(\text{sim}(e_r, e_q))$
else if $C_r = \emptyset$ then
    ASK for user feedback and ADD to $\mathcal{R}$
end if
$q \leftarrow C_r + q$ {prepend $C_r$ to $q$}
output $\leftarrow \text{LLM}(q)$
if output incorrect then
    ASK for user feedback
end if
```

Let us illustrate the contextualization by 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, it lets 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 pre-

Table 1: RiT outperforms its baseline model through (iterative) user revision. NLG scores on the CNB test set. RiT improves alignment of moral norms building on a T0 baseline with a large revision corpus (second row). A first iteration of user interaction results in a subset (7.5%) of the train set (iRiT, third row) which already suffices to revise the model. Another iteration, i.e. a few more user interactions, on non-contextualized examples (i^2RiT, fourth row) improve the model performance further. Best ("•") and runner-up ("○") values bold.

	#feedback (↓)	Bleu-1 (↑)	Bleu-3 (↑)	Rouge-L (↑)	METEOR (↑)	Bertscore (↑)	Acc. (↑)
T0	–	0.46	0.25	0.56	0.33	0.64	0.63
RiT	398 468	○**0.77**	○**0.65**	○**0.79**	○**0.68**	○**0.87**	○**0.86**
iRiT	•**29 825**	0.76	0.63	0.78	0.67	0.86	○**0.86**
i^2RiT	○**32 912**	•**0.80**	•**0.69**	•**0.82**	•**0.72**	•**0.90**	•**0.91**

dicted 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 prediction is well aligned with the user values (✓).

4 Experiments

Model and Data. We conduct our experiments on a T0 model [36], 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) [17], 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.

Evaluation of Generated Answers. Evaluating the quality of natural language generation (NLG) is challenging and a research area in itself. One challenge is that there exists no single best metric, and a plethora is provided by current research, each with its individual pros and cons. We, therefore, utilize a set of metrics in order to broadly evaluate a model's generated answers. First, we use standard NLG scores [35] like BLEU, ROUGE, and METEOR, which are n-gram based. Secondly, we investigate the cosine similarity in a sentence embedding space[41]. Lastly, we apply a task-specific metric in the spirit of [17]. 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. 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 and evaluate all on the test set (1) In our first approach, we simulate unlimited amount of user feedback and use the whole train data to fill the revision corpus. (2) In the second approach, we investigate sparsity in the amount of user feedback. Therefore, the revision corpus is empty and gradually filled with user feedback, which depicts the first "real" interactive iteration. We simulate the user feedback again with training examples that classify the (so far untouched) validation set best. (3) Lastly, we apply a second interactive iteration. Here, we use the data from (2) for the revision corpus and extend it with further feedback from the validation set. The remaining validation examples that could not be classified correctly with the training data are added to the revision corpus. For the classification, we employ the polarity accuracy as it is task-specific and models human preferences best.

4.1 Aligning Moral Norms via RiT

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. As a baseline we use a T0 model without a revision engine. At the same time, we use our RiT model, which is based on the same T0 model but on top of that utilizes the revision engine. We compare their performance on the CNB 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 moral norms in terms of the accuracy metric. Nevertheless, for a baseline, this is a noteworthy performance as this LLM was never explicitly trained for this task and confirms previous findings about an LLM's general ability to contain a moral direction [39].

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 user values without tediously tuning the parameters. In order to address this gap, we employ RiT. 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.

This result can be found with other components as well. Tab. 2 shows that RiT works well will other LLMs[5] (open-source Bloom

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

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

[4] A baseline model can be viewed as a RiT too with an empty revision engine.

[5] The same is true for other S-LLMs.

Table 2: RiT evaluation (NLG scores on CNB test set), showing that RiT can be applied to various models (e.g. Bloom). Best values bold.

	#feedback	Bleu-1 (↑)	Bleu-3 (↑)	Rouge-L (↑)	METEOR (↑)	Bertscore (↑)	Acc. (↑)
Bloom3b	–	0.27	0.02	0.30	0.27	0.66	0.67
$\text{RiT}_{\text{Bloom3b}}$	398 468	**0.50**	**0.22**	**0.54**	**0.44**	**0.76**	**0.82**
Bloom176b	–	0.43	0.08	0.50	0.34	0.68	0.71
$\text{RiT}_{\text{Bloom176b}}$	398 468	**0.56**	**0.26**	**0.60**	**0.52**	**0.86**	**0.90**

[28]), too. Both RiT variants beat their respective baseline by a large margin. Notably, RiT_{T0} and $\text{RiT}_{\text{Bloom3b}}$ outperform the Bloom176b baseline, although they have much fewer parameters. This emphasizes the utility of a revision engine. Comparing Bloom to T0, its superior performance can be explained by T0 being specifically optimized for zero-shot tasks, i.e. in-context learning, in contrast to both Bloom-based RiT models. Still, they perform well on the task.

The shown findings on moral data coincide well with previous work on factual data [30], 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 full 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 relatively 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 collaborative user feedback by selecting certain training examples by means of the validation set and regard this as a first interactive iteration (iRiT). This way, the validation set acts as a proxy for the users' selection. We pick only those examples of the training set that help classify the validation set correctly and discard all others. As a measure of correctness, we choose the task-specific polarity accuracy. With this procedure, the revision engine size shrinks from 398 468 to 29 825, i.e. down to 7.5% of the original data size. Interestingly, as the third row in Tab. 1 shows, the performance is on par with the RiT model that had utilized the full train data. With this, we confirm the function of RiT in relatively small data regimes, showing that RiT is suited for more realistic use cases.

Taking a Closer Look at RiT. The previous results are promising for RiT's ability to edit a model non-parametrically, particularly given scarcer data. When investigating the previous results further, one can observe that RiT performance is even better than at first sight. Specifically, on closer inspection, we find that iRiT performance 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 is 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 previous similarity findings, showing that higher similarity yields higher performance. However, both results are 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 revision corpus content, 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 is limited by splitting 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, i.e. another interactive iteration (i^2RiT). 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 data amount.

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 in-

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

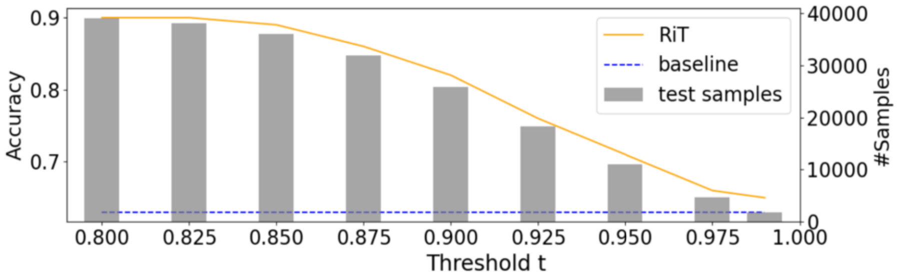

Figure 3: (left) The better the context, the better RiT. Examples revised with highly similar, i.e. relevant, context align better with user values. (right) Examination of similarity threshold t. The graph depicts the number of contextualized test samples (gray), and the polarity accuracy for the baseline (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.

teract further 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 iRiT, 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 i^2RiT (*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

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 [18]), 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. On a side note, they should be considered for other AI models as well as the RiT framework is not limited to transformers.

Societal Impact and Misuse. From an accessibility perspective, the sheer costs of retraining an LLM make it infeasible for nearly everyone except a limited number of companies or institutions. Hence, RiT offers a solution through an easily editable revision engine. In fact, replacing the whole engine is also merely *Plug & Play* such that nearly any user can insert their personal engine into RiT. This pushes boundaries back towards more democratic AI as users regain power that was recently ceded to a few tech companies. Yet, without an API available, end-users need certain skills as well to access LLMs.

Moreover, recently ChatGPT has been banned in Italy by the government due to privacy concerns[6], i.e. that it hurts the European General Data Protection Regulation[7] (GDPR). This action calls for methods that enable a dynamical and flexible editing AI models. So, in order to be able to comply with the GDPR, RiT's revision engine offers again a first practical solution to this problem, as we showed. This is backed by other works on in-context learning [5, 30], that personalization and overriding information through context memory is possible. And from the transparency perspective, a RiT can provide the retrieved context to the user (and the transformer attention weights), which, in turn, can increase model understanding.

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 [34, 40], 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 area [19] 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 (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, a 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. A 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 a 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. Furthermore, a RiT is especially powerful in cases where only a few revisions must be made.

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

[6] available at https://www.bbc.com/news/technology-65139406
[7] available at https://gdpr-info.eu/

Table 3: Case study: effect of context relevance on generated prompts. Cosine similarity between context and query is given. If relevant context is retrieved, the model is revised. Even if not directly relevant context is retrieved, the model is able to identify relevance without threshold.

	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.

fications. 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. 3 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 [30]. 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. It is worth noting, that ignoring context in the wrong situation can also come along with certain downsides such as continuing to exhibit unwanted (biased) output.

Evaluating NLG. In general, standard measures to evaluate NLG suffer from a semantic gap [35]. 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 [20], 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 various NLG scores to better and diversely evaluate the task at hand. To this end, we extended standard scores (Ngram-based [21]) with task-specific (polarity accuracy [17]) and semantic-focused (Bertscore [41]) scores. So, while the absolute value of each score might be treated with a grain of salt, they still reflect helpful indicators for evaluating and the combination of many scores gives strong evidence for the function of our method. And in the real-world, humans generally decide what an adequate revision looks like and when to interact.

6 Conclusion

This work investigated the benefits of integrating a revision engine into transformer-based LLMs. 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 correct model behavior and align a model with user values. Moreover, RiTs iteratively employ user interaction to incorporate corrections with little effort achieving high value alignment.

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 design models which are highly aligned with their values. Furthermore, with the rise of multilingual language models, it is also interesting to evaluate revision for different languages. As these models are trained on highly unbalanced data (English-focused), investigating whether revision works is crucial.

Acknowledgments

This work benefited from the ICT-48 Network of AI Research Excellence Center “TAILOR" (EU Horizon 2020, GA No 952215), the Hessian research priority program LOEWE within the project “WhiteBox”, and the Hessian Ministry of Higher Education, Research and the Arts (HMWK) cluster projects “The Adaptive Mind” and “The Third Wave of AI”, and from the German Center for Artificial Intelligence (DFKI) project “SAINT”.

References

[1] 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*, pp. 1–9, (2022).

[2] 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*, p. 610–623, (2021).

[3] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, and et al., 'Improving language models by retrieving from trillions of tokens', in *Proceedings of the 39th International Conference on Machine Learning*, Proceedings of Machine Learning Research, pp. 2206–2240, (2022).

[4] Nicola De Cao, Wilker Aziz, and Ivan Titov, 'Editing factual knowledge in language models', in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 6491–6506, (2021).

[5] Ameet Deshpande, Vishvak Murahari, Tanmay Rajpurohit, Ashwin Kalyan, and Karthik Narasimhan, 'Toxicity in chatgpt: Analyzing persona-assigned language models', *arXiv preprint arXiv:2304.05335*, (2023).

[6] 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*, 257–273, (2022).

[7] Maxwell Forbes, Jena D. Hwang, Vered Shwartz, Maarten Sap, and Yejin Choi, 'Social chemistry 101: Learning to reason about social and moral norms', in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 653–670, (2020).

[8] 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)*, pp. 26–42, (2022).

[9] Felix Friedrich, Patrick Schramowski, Christopher Tauchmann, and Kristian Kersting, 'Interactively providing explanations for transformer language models', *Frontiers in AI*, (2022).

[10] Felix Friedrich, Wolfgang Stammer, Patrick Schramowski, and Kristian Kersting, 'A typology for exploring the mitigation of shortcut behaviour', *Nature Machine Intelligence*, **5**, 319–330, (2023).

[11] 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).

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

[13] Amelia Glaese, Nat McAleese, Maja Trebacz, and et al. Improving alignment of dialogue agents via targeted human judgements, 2022.

[14] Benjamin Heinzerling and Kentaro Inui, 'Language models as knowledge bases: On entity representations, storage capacity, and paraphrased queries', in *Proceedings of the European Chapter of the Association for Computational Linguistics*, pp. 1772–1791, (2021).

[15] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, and et al., 'Parameter-efficient transfer learning for NLP', in *Proceedings of the International Conference on Machine Learning*, pp. 2790–99, (2019).

[16] Jeremy Howard and Sebastian Ruder, 'Universal language model fine-tuning for text classification', in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 328–339, (2018).

[17] 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.

[18] Zhijing Jin, Sydney Levine, Fernando Gonzalez Adauto, and et al., 'When to make exceptions: Exploring language models as accounts of human moral judgment', in *NeurIPS*, (2022).

[19] 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', *arXiv preprint arXiv:2208.03295*, (2022).

[20] N. Kamoen, Bregje Holleman, Pim Mak, Ted Sanders, and Huub Bergh, 'Why are negative questions difficult to answer? on the processing of linguistic contrasts', *Public Opinion Quarterly*, 613–635, (2017).

[21] Su Nam Kim, Timothy Baldwin, and Min-Yen Kan, 'Evaluating n-gram based evaluation metrics for automatic keyphrase extraction', in *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pp. 572–580, (2010).

[22] Brian Lester, Rami Al-Rfou, and Noah Constant, 'The power of scale for parameter-efficient prompt tuning', in *Proceedings of the Conference on EMNLP*, pp. 3045–3059, (2021).

[23] Patrick Lewis, Ethan Perez, Aleksandra Piktus, and et al., 'Retrieval-augmented generation for knowledge-intensive nlp tasks', in *Proceedings of the International Conference on Neural Information Processing Systems*, (2020).

[24] 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 Annual Meeting of the Association for Computational Linguistics*, pp. 61–68, (2022).

[25] Aman Madaan, Niket Tandon, Peter Clark, and Yiming Yang, 'Memory-assisted prompt editing to improve GPT-3 after deployment', in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 2833–2861, (December 2022).

[26] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov, 'Locating and editing factual associations in gpt', *arXiv preprint arXiv:2202.05262*, (2022).

[27] 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*, p. 220–229, (2019).

[28] Niklas Muennighoff, Thomas Wang, Lintang Sutawika, and et al., 'Crosslingual generalization through multitask finetuning', *arXiv preprint arXiv:2211.01786*, (2022).

[29] OpenAI. Training language models to follow instructions with human feedback, 2022.

[30] 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*, (2020).

[31] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller, 'Language models as knowledge bases?', in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 2463–2473, (2019).

[32] 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*, 1–67, (2020).

[33] 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*, 842–866, (2020).

[34] 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 International Joint Conference on Artificial Intelligence*, pp. 2662–2670, (2017).

[35] Ananya B. Sai, Akash Kumar Mohankumar, and Mitesh M. Khapra, 'A survey of evaluation metrics used for NLG systems', *ACM Comput. Surv.*, (2022).

[36] Victor Sanh, Albert Webson, Colin Raffel, and et al., 'Multitask prompted training enables zero-shot task generalization', in *International Conference on Learning Representations*, (2022).

[37] 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).

[38] 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*, p. 1350–1361, (2022).

[39] 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*, 258–268, (2022).

[40] Stefano Teso and Kristian Kersting, 'Explanatory interactive machine learning', in *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society (AIES)*, p. 239–245, (2019).

[41] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi, 'Bertscore: Evaluating text generation with BERT', in *International Conference on Learning Representations*, (2020).