Linköping University | Department of Computer and Information Science

Master's thesis, 30 ECTS | Datateknik

2023 | LIU-IDA/LITH-EX-A--2023/077--SE

Modeling of human preferences without humans - a study on data augmentation for large language reward models

Modellering av mänskliga preferenser utan människor - en studie av dataförbättringar för stora språkbelöningsmodeller

Oskar Hallström

Supervisor : Marcel Bollmann
Examiner : Marco Kuhlmann

External supervisor : Julien Launay

LINKÖPING UNIVERSITY

Linköpings universitet
SE–581 83 Linköping
+46 13 28 10 00 , www.liu.se

Upphovsrätt

Copyright

Abstract

Recent large language models such as chatGPT, GPT-4, Claude, and Bard have gained enormous interest thanks to their impressive performance and helpfulness for humans. These large language models are pre-trained on huge amounts of text to predict the next word, and then further refined through reinforcement learning from human feedback. In this process, a reward model is first trained to capture human preferences and then used to provide a reward signal in the reinforcement learning loop. In this manner, the large language model is steered towards the behavior desired by humans. These reward models, not only used within reinforcement learning, are trained on ranked text data, which in its simplest form consists of samples with a prompt and a pair of two ranked continuations. As this data exists in limited amounts and is expensive to annotate for humans, automatic generation of ranked training data becomes an interesting area of research. By investigating how existing and new generation techniques for ranked data impact the accuracy of reward models with respect to a dataset based on human preferences, this thesis contributes with new insight on the impact of these generation techniques as well as highlighting their potential. In particular, this thesis introduces AI distractors and AI continuation pairs. When combined with previous techniques, AI distractors give an improvement over these techniques, resulting in that replacing a partition of human-ranked data with generated data can give slightly better performance than with the human-ranked data as is. With AI continuation pairs, this thesis takes a first rudimentary step into observing the capabilities of reward models trained on no data ranked by humans to model human preferences.

Acknowledgments

First and foremost, I want to thank my colleagues at LightOn. Special thanks to Axel Marmet, with whom I have worked closely during my time at LightOn and learned a lot from. Big thanks to Iacopo Poli for feedback on ideas I have had during this thesis, as well as interesting discussions and support. Thanks also to my supervisor Julien Launay, and to the other members of LightOn's R&D team, who have provided great support throughout my time at LightOn, especially during my development of the reward model training framework and ranked data pipeline. This includes Alessandro Capelli, Baptiste Pannier, Daniel Hesslow, and Guilherme Penedo. It has been very inspiring to work among such talented people. Lastly, a thank you to my academic supervisor Marcel Bollmann and examiner Marco Kuhlmann.

- Oskar Hallström, Paris 2023

Contents

List of Figures

List of Tables

Glossary

continuation The output of a large language model. 2

prompt The input provided to a large language model, for example a question or an instruction. 2

token A basic unit of text processable by large language models. 4

1 Introduction

In the following chapter, the reader is given an introduction to the thesis. In Section 1.1, the research ambitions of this thesis are motivated and justified. Subsequently, the aim of the thesis is stated in Section 1.2, and the research questions are described in Section 1.3. Lastly, the delimitations of the thesis are mentioned in Section 1.4.

1.1 Motivation

Recent large language models such as chatGPT, GPT-4, Claude, and Bard have gained an enormous interest thanks to their impressive performance for various natural language tasks such as question-answering and creative writing, as well as their perceived general helpfulness for humans. A significant part of the technical progress made since earlier and less mature large language models such as GPT-3 can be attributed to Reinforcement Learning from Human Feedback [27, 20, 2, 3, 13, 17], which enables optimization of large language models with the objective of maximizing perceived quality by humans.

In the center of the reinforcement learning loop is a reward model which assigns scores to the large language model's outputs, ideally giving high scores to outputs that are perceived as helpful to humans, and lower scores to outputs that are considered less helpful. With a reward model modeling human preferences, the reinforcement learning process consequently steers the large language model towards a behavior desired by humans. The usage of reward models is however not limited to reward assignment in reinforcement learning - another common usage is for selecting the best output out of several options at inference time, which is commonly referred to as reranking [17, 13]. Lastly, scores from reward models can serve as proxies for large language model performance according to human preferences [20], potentially a better proxy for perceived quality by humans than other common proxies such as validation loss and NLP benchmarks.

Reward models are trained on ranked datasets [41, 27, 20, 17, 13, 1, 2, 3], where each sample corresponds to two or more outputs ranked from best to worst for a given input. Such data is much more expensive to generate for humans than regular language model finetuning data, and thus also exists to a much lesser extent. Lately, large language models have shown the

potential to improve significantly by being finetuned on data generated from another larger language model [21], or even pruned data generated from the same model [37]. This process of leveraging competent large language models to generate training data has however not been as well explored for reward models. Only three previous works can be found on this subject in the literature [17, 3, 21], whereas none of them address and evaluate this type of data generation in the context of human preferences.

The current usefulness and future potential of reward models, together with the sparsity of ranked training data for language reward models, illustrate the need for more ranked data that can help improve reward models. This justifies this thesis's ambition to take a first step into filling the void of research on reward model training data generation in the context of human preferences, as well as to investigate other methods for generation of ranked data and training with such data.

1.2 Aim

The purpose of this thesis is to generate a better understanding of how existing and new strategies for generating ranked data impact the ability of a reward model to model human preferences, as well as to explore how the integration of such data into existing datasets can be optimized.

1.3 Research questions

To fulfill the aim of this thesis, three research questions will be investigated. These are centered around the three generation techniques listed below, further described in Subsection 4.1.2.

- **Random distractor:** Randomly sampled continuation added to an existing prompt and continuation in order to form a ranked data sample. Intends to make the reward model better at penalizing out of distribution continuations.
- **AI distractor:** AI-modified continuation added to an existing prompt and continuation in order to form a ranked data sample. The AI distractor is generated by slightly changing the existing continuation using a large language model. Ideally, this modification correspond to an altering of one or a few words; It could for instance be a change of a pronoun, swapping a word for an antonym, and other similar changes. This modification intends to make the reward model more attentive to details.
- **AI continuation pair:** AI-generated ranked continuation pair added to an existing prompt in order to form a ranked data sample. First, a ground-truth continuation is generated by prompting a large language model with the existing prompt. Then, the worse continuation is generated based on the ground-truth continuation, following the same method as for the generation of an AI distractor. Similarly to an AI distractor, an AI continuation pair intends to make the reward model more attentive to details, however without the need of a pre-existing continuation.

The first research question serves to give an initial understanding of how the already existing random distractor [13], as well as the AI distractor and AI continuation pairs which both are introduced in this thesis, affect accuracy according to human preferences. The second research question then investigates if training with distractors can be improved with curriculum learning, and the third research question intends to explore the potential of AI continuation pairs on their own. Below follows a list of these three research questions:

- **RQ1:** How does adding random distractors, AI distractors, and AI continuation pairs in training data impact accuracy on a dataset based on human preferences?
- **RQ2:** Can curriculum learning be applied to reward model training with distractors, where the difficulty of a sample is determined based on how it has been generated, in order to improve accuracy?
- **RQ3:** To what extent can reward models trained on only AI-generated continuation pairs model human preferences, in terms of accuracy on a dataset based on human preferences?

Accuracy refers to the percentage of samples in the test set from the ranked summarization dataset by Stiennon et al.[27] for which the reward model prefers the same continuation as the annotators.

1.4 Delimitations

Firstly, the study on reward model performance is limited to training and evaluation on one single task and dataset. The dataset used throughout this thesis is a summarization dataset of ranked summaries introduced and used by Stiennon et al. [27]. Even though the reward model is tested in a reinforcement learning setting as part of verifying that the reward model functions properly during the implementation stage, the performance consequences of different data choices are never evaluated through integrating the reward model in any process to augment large language model performance.

2 Background

This chapter intends to provide the reader with the necessary background to understand the work conducted in this thesis. First, large language models are briefly introduced in Section 2.1, then reward models and important aspects of those are described in Section 2.2.

2.1 Large language models

In the standard case, large language models are simply put large neural networks based on the transformer architecture [34] pre-trained on enormous amounts of text data to predict the next token at a given time step [31, 33, 19, 6, 14]. In practice, this prediction is given by the final layer of a large language model, which outputs a probability for each token in the vocabulary. Later in this thesis, the final layer is referred to as the projection layer.

Consider V as the the vocabulary size and $\mathbf{w}$ as an one-hot coded vector such that $\mathbf{w_n} \in \{0,1\}^V$ and $\|\mathbf{w_i}\| = 1$ for all $i \in \mathbb{Z}^+$. $\mathbf{w_n}$ then represents the token at a given time step n, where the index of the one in $\mathbf{w}$ signifies the given token's position in the vocabulary. The output of a large language model's projection layer can then be argued as the following probability at time step n:

$$P(\mathbf{w_{n+1}}|\mathbf{w_n}, ..., \mathbf{w_1})$$

According to the same premises, a reward language model's projection layer can then be said to give the following output (S_n) at time step n:

$$S_n = f(\mathbf{w_n}, \mathbf{w_{n-1}}, ..., \mathbf{w_1}), S_n \in \mathbb{R}$$

After pre-training, the models can be adapted in several ways to perform better on downstream tasks. Examples of such techniques prevalent in previous research are instruction tuning [38, 7, 25, 15] , objective adaptation [30, 35], and reinforcement learning from human or AI feedback [27, 20, 2, 3, 17, 13]. As this thesis focuses on reward modeling, which is central to the reinforcement learning process, reinforcement learning for large language models is further described in Subsection 2.3.

GPT-3 [5] is a large language model released in 2020 that is frequently cited in subsequent research on large language models. Despite being based on the transformer architecture [34], it is slightly different from the original transformer as the encoder part of the transformer has been omitted [5]. Originally, the transformer was created with an encoder-decoder structure: The encoder transforms an input sequence of tokens into a sequence of vector representations, which in turn is input into the decoder that finally outputs a sequence of tokens [34]. Wang et al. [35] studies the performance differences between the encoder-decoder architecture and the casual decoder-only architecture used by GPT-3. Contrary to encoder-decoder models, decoder-only models can only condition their outputs on past tokens. Nevertheless, the decoder-only architecture is used by most recent large language models, and has shown to give better performance on unseen tasks (*zero-shot generalization*) directly after pre-training than encoder-decoder architecture [35].

2.2 Reward models for natural language

The following subsection gives an explanation of the architecture, training, and applications of reward models based on large language models.

2.2.1 Architecture

Most commonly, reward models for natural language are built using the same architecture as a large language model, with the exception being the final projection layer [12, 17, 41, 27, 20, 13]. In regular large language models, the projection layer outputs a scalar for all elements in the vocabulary, whereas a common approach for reward models is to replace it with a single scalar head that outputs a single score [27, 20].

2.2.2 Training

The shared architecture between regular large language models and reward models enables reward models to exploit the costly pre-training of large language models. With the exception of the final projection layer, the weights of a reward model can be initialized from the weights of an already trained large language model [1, 17, 41, 41, 27, 20, 13]. At its initialization, the reward model thus already possesses language modeling capabilities to a certain extent. The weights of the projection layer on the other hand, can for instance be randomly initialized [41, 27].

Rather than being trained on a dataset where a given text has a target value that the model is trained to predict, reward models are in the usual case trained on comparison data [13, 17, 41, 27, 20, 1, 2, 3]. In the domain of natural language, a sample of such comparison data consists of a single prompt and two or more continuations that have been ranked by quality. For each sample seen during training, the model is trained to output a higher score for the preferred continuation. A toy sample from a ranked dataset is displayed in Figure 2.1.

The most common loss function seen in literature [1, 2, 27, 20] for reward model training on ranked data is the following:

$$L_{REWARD} = \log\left(1 + e^{r(prompt, worse\ continuation) - r(prompt, preferred\ continuation)}\right)$$

where $r(x, y)$ is the score given by the reward model for prompt x and continuation y.

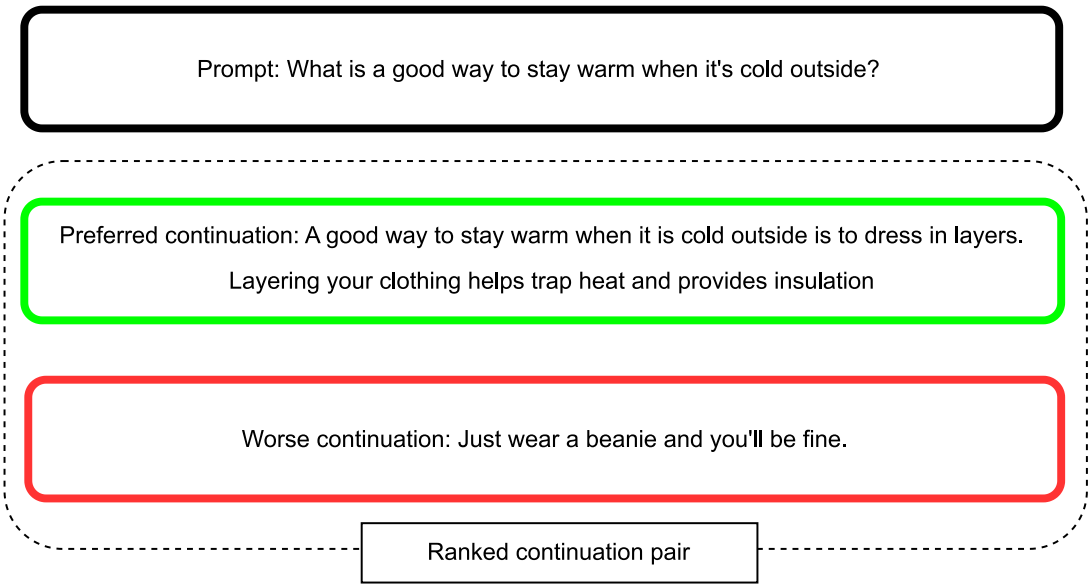


Figure 2.1: A toy sample from a ranked dataset that could be used to train a reward model.

Applications

In previous literature, reward models are found to be used for mainly two purposes in the domain of natural language. The first application is found during training time in reinforcement learning for language models [41, 27, 20, 17, 13, 3, 2], where the reward model is used to give rewards to the actor, in other words the large language model, as to guide the model towards the desired behavior. This is further described in Section 2.3.

The other common application for reward models for natural language is as a tool for ranking the quality of a large language model's outputs during inference time [17, 13]. By scoring a set of outputs for a given prompt, the reward model can give a ranking of the outputs which for instance could be used to decide what output to display to the user that made the prompt. With a reward model trained to model human preferences, the outputs can thus be ranked according to human preferences. The technique of ranking outputs with a reward model during inference time is sometimes referred to as reranking [17, 13].

Additionally, the scores from reward models can also be used for checkpoint selection of model checkpoints generated at different stages during finetuning [20].

2.3 Reinforcement learning from human feedback for large language models

Reinforcement learning is a method that aims to let an agent learn what action to take in a given situation in order to maximize its reward [28]. The mapping from a perceived state to an action is called a policy, which is learned by the actor through trying different actions and observing the subsequent reward [28]. A new family of reinforcement learning algorithms called proximal policy optimization was introduced in 2017 [26], which is used in several applications of reinforcement learning on language models [41, 27, 20, 2]. A recent extension to proximal policy optimization called natural language policy optimization [24] reduces the combinatorial action space, improving stability and performance. There are also examples of works [17, 13, 12] using other reinforcement learning algorithms such as synchronous advantage actor-critic [18].

Ziegler et al. [41] finetune a pre-trained GPT-2 model with reinforcement learning using a variant of proximal policy optimization based on human preferences. First, human labelers are asked to pick the best out of four outputs for a given input. A reward model is then trained based on given human preferences, which in turn is used to train the policy with proximal policy optimization [26]. Stiennon et al. [27] use a similar approach to the task of summarization, however with larger models and higher labeler-researcher agreement considering the rankings. First, they gather human feedback by letting annotators choose the best out of two summaries, this data is then used to train a reward model which in turn is used to train a policy with proximal policy optimization [26]. After having performed supervised finetuning on GPT-3, Ouyang et al. [20] follow a similar methodology to Ziegler et al. [41] and Stiennon et al. [27] to adapt the model further through reinforcement learning. They find that the resulting model outperforms GPT-3 despite having over 100 times fewer parameters.

Reinforcement learning from human feedback has been extended by other works to not only limit the feedback given during reinforcement learning to human preferences regarding helpfulness. Bai et al. [2] follow the reinforcement learning methodology of Stiennon et al. [27], however additionally including human preferences with respect to harmfulness. Harmfulness is also addressed by Perez et al. [12]. In contrast, they use a classifier trained on harmfulness data as reward signal during reinforcement learning to elicit harmful behavior for red teaming purposes. Glaese et al. [13], Menick et al. [17], and Bai et al. [3] all extend reinforcement learning from human feedback to be based on reward models not only trained on direct human rankings. This is made by introducing different methods to automatically generate ranked data. These approaches are further described in Section 3.1.

2.4 Curriculum learning

Formally introduced by Bengio et al. in 2009 [4], curriculum learning corresponds to the concept of training machine learning models on gradually more difficult data. The training starts on a small, easier, subset of the training data, and then tougher and tougher subsets are added until the whole training set is in use. This methodical ordering of examples can speed up convergence, as well as augment the final capacity of the trained network [36]. How the difficulty of a sample is decided is task-dependent, and thus varies case by case. As an example, Bengio et al. [4] uses word frequency to define difficulty in a language modeling experiment.

Bengio et al. [4] relate curriculum learning to human learning; The meaningful order of subjects taught in our educational system, where there is a gradual increase of complexity of the subjects presented, contributes to more efficient learning.

3 Related work

This chapter gives an overview of previous research related to the work conducted in this thesis. Section 3.1 describes related work with respect to generation of ranked data for reward model training, and Section 3.2 describes previous uses of curriculum learning in modern natural language processing.

3.1 Improvement of natural language reward models through data

In this section, previous efforts in the literature for the generation of ranked natural language data without explicit human involvement are described.

Askell et al. [1] introduces preference model pretraining, where "preference model" is simply the same as a reward model. This is performed by first gathering ranked preference data based on for instance votes on answers on internet forums such as Reddit, and then pre-train the reward model on this data. This pre-training is shown to increase sample efficiency initially during the subsequent training of the reward model, which is useful for cases where the data available for training of the reward model is limited. However, the utility of preference model pre-training decreases as the size of the regular reward model training data increases, causing the performance for reward models trained with and without preference model pre-training to converge to similar performance after around 10 000 continuation pairs [1].

Menick et al. [17] extend a question-answering dataset, where answers have been ranked by humans, by leveraging the fact-checking dataset FEVER [32] to generate additional ranked question-answering data. The FEVER dataset contains claims, which together with evidence have been classified either as *Supported*, *Refuted*, or *NotEnough*. These labels are used to generate ranked data in four different ways:

- Type A: First, claims are transformed into questions by inserting them into templates such as 'Is it correct to say that {claim}?'. Two options are then generated by pairing the evidence with an affirmative answer such as 'It is true', and a negative answer such as 'It

is false'. If the claim was labeled as supported, the affirmative option is marked as preferred, else the negative option is ranked higher. Additionally, they also construct option pairs where the preferred answer is the same as before, however, the lower ranked option has been swapped to the preferred answer but with a random quote instead of the actual evidence.

- Type B: Here, claims are transformed into questions through few-shot prompting their large language model Gopher. The answer options are simply the claim with evidence, or a negation of the claim such as 'It is not true that {claim}'. If the claim was labeled as supported, the answer with the claim is marked as preferred, else the negated claim is ranked higher. Just as for type A, they also construct option pairs where the preferred answer is the same as before, however with the lower-ranked option swapped to the preferred answer but with a random quote.

Menick et al. [17] verify the quality of the generated ranked pairs, and found them to correspond to their own rankings in 87 % of the cases through rating 50 comparisons. However, they never investigate the impact of adding this data to the reward model training. Furthermore, the method is limited to question answering and relies on the existence of datasets such as FEVER [32].

Glaese et al. [13] introduce distractors to generate ranked continuation pairs, which contrary to the extension technique used by Menick et al. [17] is task agnostic. To construct a ranked continuation pair using a distractor, the distractor is paired with a preferred continuation for a given prompt. In their work, the distractor is simply a randomly sampled continuation from the entirety of the preference training dataset, intending to make the reward model better penalize out of distribution data [13].

Even though Menick et al. [17] were the first to use a large language model in the process of generating ranked data, the large language model neither contributed to the construction nor the ranking of the continuations. Instead, Constitutional AI [3] is the first work to use a large language model in the ranking process when generating comparison data for training of reward models. Given a continuation pair for a prompt, they create a multi-question answering setting where the two continuations are the options. Then, a large language model is asked to select the best option according to a given principle for harmlessness. Though this data when used to train a reward model is shown to reduce harmfulness in the resulting policy after reinforcement learning with the given reward model as a signal, it deteriorates helpfulness according to human preferences [3]. They present validation accuracies for their reward model trained on both human-ranked data for helpfulness and AI-ranked data for harmfulness, but no results of a reward model trained solely on AI-ranked data are presented.

Peng et al. [21] generates comparison data by simply letting GPT-4 rate large language model outputs from 1-10, and then form ranked pairs based on these ratings. In their evaluation they find the resulting reward model to be consistent with direct evaluation with ChatGPT and GPT-4. However, they do not evaluate whether the generated ranked pairs and consequently the resulting reward model explicitly corresponds to human preferences.

3.2 Curriculum learning in modern natural language processing

In previous research in modern natural language processing where curriculum learning has been applied, there are mainly two decisions that differentiate different approaches from each other: how the difficulty of a training sample has been defined and how sampling of training data has been made in order to make the reward model see progressively more difficult

samples during training. To the best of the author's knowledge, no previous work has been made on curriculum learning for reward models. Consequently, this related work section is widened to describe curriculum learning in modern natural language processing.

Kocmi and Bojar [16] introduce a variation of curriculum learning for neural machine translation, where in difference to initial curriculum learning by Bengio et al [4], each sample is only seen once per epoch. The training samples are divided into bins based on how hard they are, for which the bin size decreases with increasing complexity. During training, data is sampled randomly without replacement from the bins that are active. Starting with only the easiest bin activated, the easiest unactivated bin is activated once all the active bins each have as many samples left as there are samples in the easiest inactive bin. They rank the difficulty of samples based on different criteria such as lexical features like sentence length. Some of their curriculum setups display a small improvement over the randomly ordered baseline.

Platanios et al. [23] extend the research on curriculum learning on neural machine translation. This is made by not only proposing a new method, but also by applying it to transformers, for which their curriculum setup gives both higher performance as well as reduced training time. They use similar difficulty measures as Kocmi and Bojar [16] for text, but with a different sampling strategy. Before sampling a new batch, they compute a model competence score based on the time for which the model currently has been trained. Then they sample data uniformly from all training samples with a difficulty inferior to the competence score to construct a batch.

Zhao et al. [40] use curriculum learning for knowledge-grounded dialogue generation, involving both BERT and GPT-2. They have two subsets of data for curriculum learning: the easier subset consisting of pseudo-generated ground truth, and a harder subset from the original training data. For each sample to be included in a mini batch during training, the subset to select the sample from is based on a sample from a Bernoulli distribution. If a zero is sampled, the sample is selected from the harder subset, and if a one is sampled, the sample is selected from the easier subset. As the Bernoulli distribution is parametrized by a parameter that shrinks over time, the model will gradually be exposed for harder samples more frequently as the training goes on.

Tay et al. [29] employ curriculum learning to improve the performance of a model for question answering. They have two notions of difficulty: answerability which depends on if the excerpts used to answer a question are retrieved based on the question (harder) or the answer (easier), and understandability which depends on the paragraph size, also referred to as chunk size. For each chunk size, they construct an easy set and a hard set in terms of answerability. They start with a random chunk size and a training set only consisting of easy samples. As long as the evaluation score improves, they continue to train on this dataset. When the score does not improve, a certain percentage of the dataset is replaced by harder samples. This loop continues until there are only hard samples in the training dataset. Once this happens, they repeat the process with a new chunk size until all chunk sizes have been handled.

Xu et al. [39] use a more agnostic method to evaluate the difficulty of samples, as no prior knowledge about the data is needed. However, this comes with the trade-off of being computationally expensive, making it harder to justify for compute-intense large language models. They partition the dataset into several subsets and then train a model on each subset. Then for each sample, all models except the one that has seen the sample during its training are used to calculate the sample's difficulty. This score could be calculated using a metric such as accuracy or MSE, and is calculated between the sample's ground truth and the inferred value for that sample, for all different models. Based on the score, the samples are put into

buckets based on their difficulty. The training is then carried out in as many stages as there are buckets. For each stage *i*, samples are sampled uniformly from the *i* easiest buckets. After the final stage, the model is trained on the entire dataset until convergence.

4 Method

The method chapter is divided into two parts: the implementation and its different stages are presented in Section 4.1, whereas the approach to training and evaluation is described in Section 4.2.

4.1 Implementation

4.1.1 Implementation and verification of training framework

A major part of the work conducted to obtain the final results of this thesis has been dedicated to the designing and engineering of a training framework for reward models. With LightOn's codebase for training of large language models as a starting point, the whole process from preparing and tokenizing ranked text data for the training, to the actual training and evaluation of the reward models, has been implemented.

Similarly to Stiennon et al. [27] and Ouyang et al. [20], the final projection layer has been replaced by a scalar head, thus giving one single scalar value. For the experiments in this thesis, the reward model is initialized from a 7B parameter pre-trained large language model, and the projection layer is initialized randomly from a normal distribution. More precisely, the model is open-sourced as Falcon-7B [11], which is a pre-trained large language model with an architecture based on GPT-3 [5]. As Stiennon et al. [27], the initial projection layer weights are sampled from the distribution $\mathcal{N}(0, \frac{1}{d+1})$, where d corresponds to the depth of the large language model. Also similar to most previous research [1, 2, 27, 20], the following loss function is used to train the reward model:

$$L_{REWARD} = \log\left(1 + e^{r(prompt, worse\ continuation) - r(prompt, preferred\ continuation)}\right)$$

where $r(x, y)$ is the score given by the reward model for prompt x and continuation y.

To initially ensure that the reward model worked properly, the training of reward models by Stiennon et al. [27] was reproduced with similar validation accuracies. In order to find the

best learning rate to get as close as possible to their results, a log-linear sweep was carried out for batch sizes 32, 64 and 128 over three different seeds. As Glaese et al. [13] and Menick et al. used linear warmup of the learning rate during reward model training, warmup was also part of the hyperparameter search space. To limit the search space, the two options evaluated were linear warmup during the first million tokens, or no warmup at all. Further details about the learning rate schedule are given in Subsection 4.2.3.

The optimal combination of batch size and learning rate (magnitude and with or without warmup) found then served as a starting point for the main experiments of this thesis, described further in Section 4.2. The reward model has also been validated by controlling that it functions properly with LightOn's reinforcement learning codebase, using Proximal Policy Optimization [26].

4.1.2 Data generation

In the following subsection the processes used to generate comparison data are described. As a basis for all these processes lays the distractor by Glaese et al. [13], in this thesis referred to as random distractor. This is motivated through its task-agnostic nature, easiness to implement as well as potential to refine. Out of the related work in Section 3.1, the generation technique of Menick et al. [17] was deemed inappropriate as it is limited to question answering. The method by Bai et al. [3] could also be argued as partially task-dependent as it only has been used to generate rankings for harmfulness data, and relies on specific principles to generate the rankings. With the right principles, their generation technique could take a step towards a more task-agnostic nature, however finding principles which not only are task-agnostic, but also possible for the model to make a good assessment of, was deemed out of scope for this thesis. Preference model pre-training by Askell et al. [1] constitutes another option. However, the utility of preference model pre-training has already been studied well with respect to performance on dataset based on human preferences. Additionally it is found that its utility decreases as the size of the regular reward model training data increases, causing the performance for reward models trained with and without preference model pre-training to converge to similar performance after around 10 000 continuation pairs [1]. Given the size of the base dataset used in this thesis, further combined with the generation techniques used to generate comparison data, the amount of data available makes preference model pre-training hard to justify. Peng et al. [21] takes a fully task-agnostic approach, enabling fully AI generated continuation pairs. However, the project had already started at the time of its release.

The generation of a distractor to form a continuation pair is based on a prompt and a corresponding continuation. The generated distractor then forms a ranked continuation pair together with the already existing continuation, where the existing continuation is ranked as better than the generated distractor with respect to the given prompt. In theory, this means that the generation of distractors could be used to transform a regular dataset for supervised fine-tuning of large language models into a ranked comparison dataset usable for training of reward models. The processes to generate the two different types of distractors handled in this thesis are further described in the remaining parts of this subsection. Thereafter, the generation of an AI continuation pair is explained. Random distractors are based entirely on research by Glaese et al. [13], whereas AI distractors and AI continuation pairs are ideas introduced through this thesis. In Figure 4.1, a toy sample with a prompt, a continuation, and the two distractor types is displayed. All AI-generated data used in this thesis have been generated using Falcon-40B-Instruct [10]. This is a 40B parameter large language model with an architecture broadly adapted from GPT-3 [5], which after regular pre-training has been adapted through instruction-tuning.

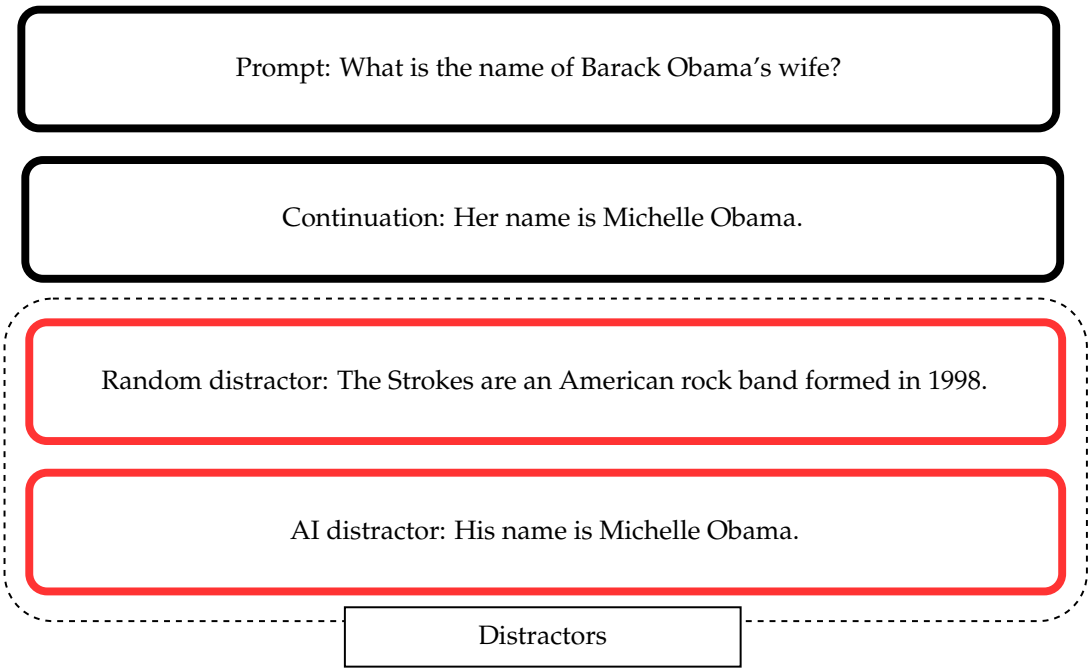


Figure 4.1: A toy sample showing what a random distractor and an AI distractor could look like.

Random distractors

In this thesis, a random distractor refers to the distractor introduced by Glaese et al. [13]. To add a worse continuation to a prompt and a continuation, the worse continuation is randomly sampled from a set of all continuations in the original dataset. The generation of a random distractor in this thesis follows their methodology, which is visualized in Figure 4.2.

AI Distractors

The generation of an AI distractor is, as for the random distractors, also based on a prompt and a continuation. However, this approach leverages a large language model to generate the worse continuation based on the ground truth continuation. In order to modify the ground truth continuation, an instruction together with some few-shot examples of slightly changing a continuation are prepended to the continuation to form a prompt. The prompt is then passed to the large language model, which returns a continuation. This generated continuation is compared in two ways with the original continuation that ought to be changed, in order to avoid two failure cases. If the generated continuation either has not changed at all from the original continuation, or changed to the extent that its total string length is more than 25 % longer or 20 % shorter than the original continuation, the new continuation is invalid. In that case the prompt is rerun through the large language model. This loop continues until all generations are valid.

The overarching process for generating an AI distractor is visualized in Figure 4.3, and the modification process that takes a continuation and turns it into an AI distractor is visualized in Figure 4.4. In comparison to the usage of random distractors that serves to make the reward model better at penalizing out of distribution answers [13], AI distractors are intended to make the reward model more attentive to details in the answer.

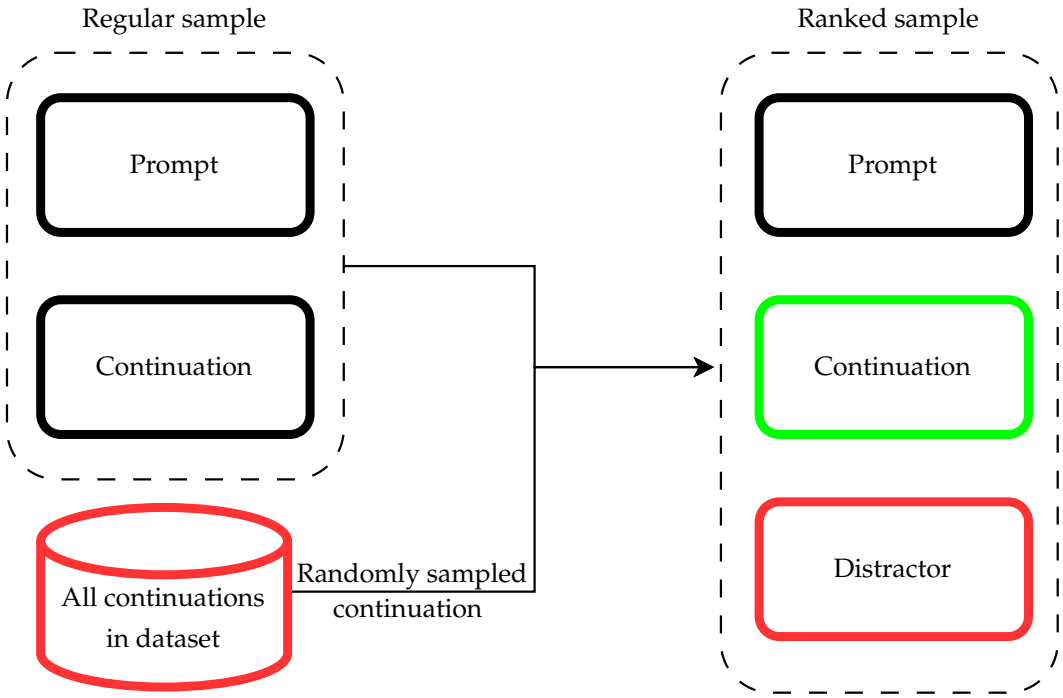


Figure 4.2: Visualisation of the generation of a ranked sample with a random distractor

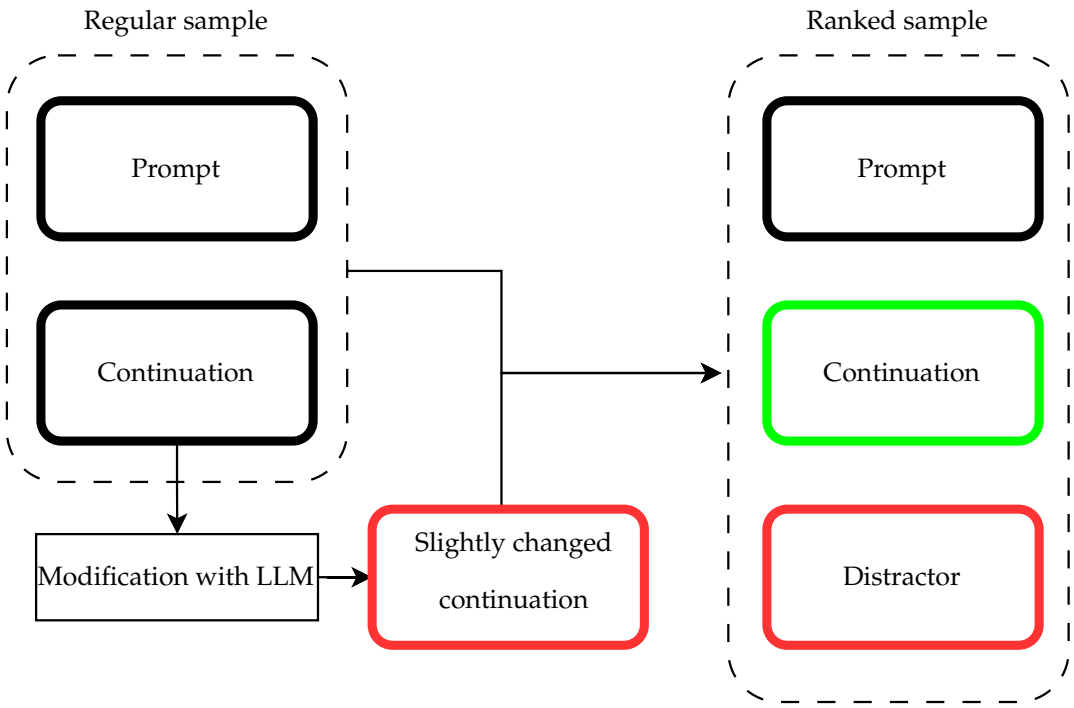


Figure 4.3: Visualisation of the generation of a ranked sample with an AI distractor

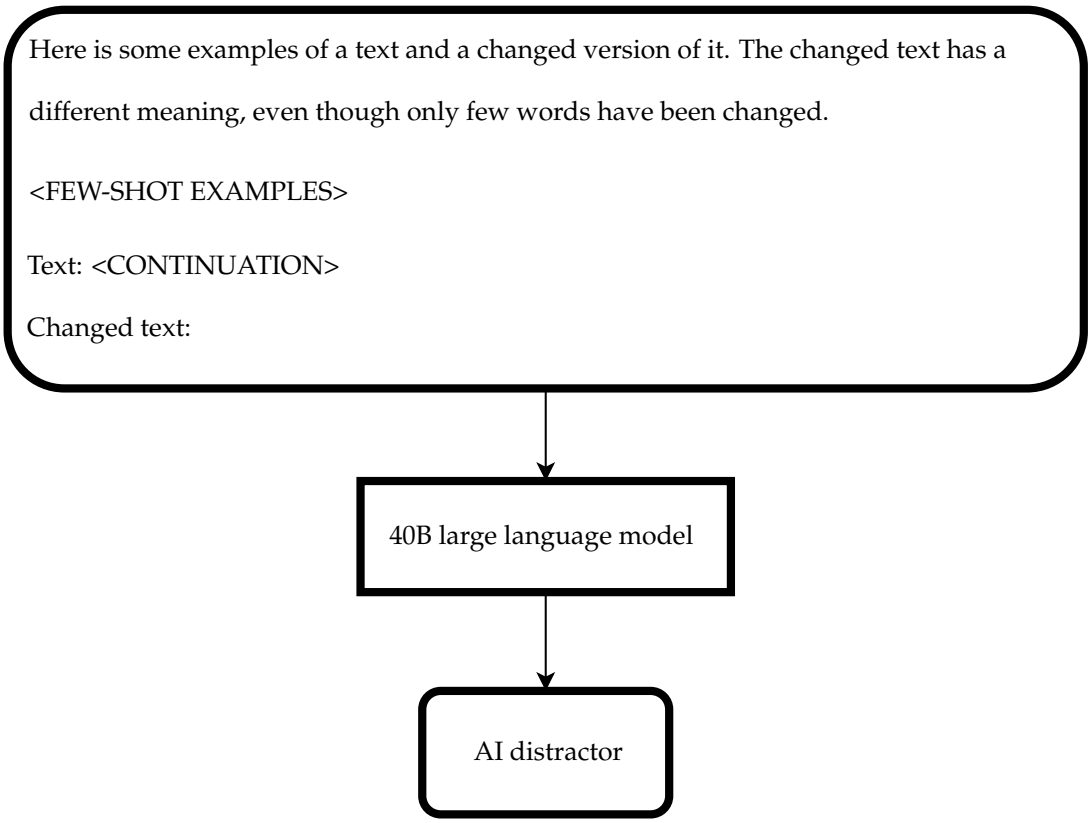


Figure 4.4: Visualisation of how a continuation is transformed into an AI distractor using a large language model. To construct the model input, the continuation is inserted into the given template. The few-shot examples used for all experiments are listed in Section A.1 in Appendix A.

AI Continuation pair

Contrary to the two previous methods for generating a ranked pair of continuations, where a distractor is added to an existing sample of a prompt and a continuation, the generation of an AI continuation pair only requires a prompt as a starting point. Based on the prompt, a large language model is few-shot prompted to generate the continuation that will be considered the preferred continuation in the final continuation pair. To form the worse continuation, an AI distractor is constructed following the same methodology as described above for AI distractors. The process of generating an AI continuation pair is visualized in Figure 4.5, whereas the inference process for transforming a prompt into a continuation is displayed in Figure 4.6.

4.2 Training and evaluation

4.2.1 Data details

As a basis for all experiments, the main summarization dataset used by Stiennon et al. [27] is filtered so that there are no duplicates of prompts in the dataset. A small amount of additional samples is then removed to get an appropriate factorable dataset size, enabling partitioning of the data into equally sized subgroups for the experiments. The resulting set then contains 14700 samples, which is significantly less than the unfiltered dataset. For the rankings in the unfiltered dataset, labelers agreed with each other 73% ± 2% of the time [27], which gives an indication of the accuracy that the reward models should be able to obtain. Similarly to Sti-

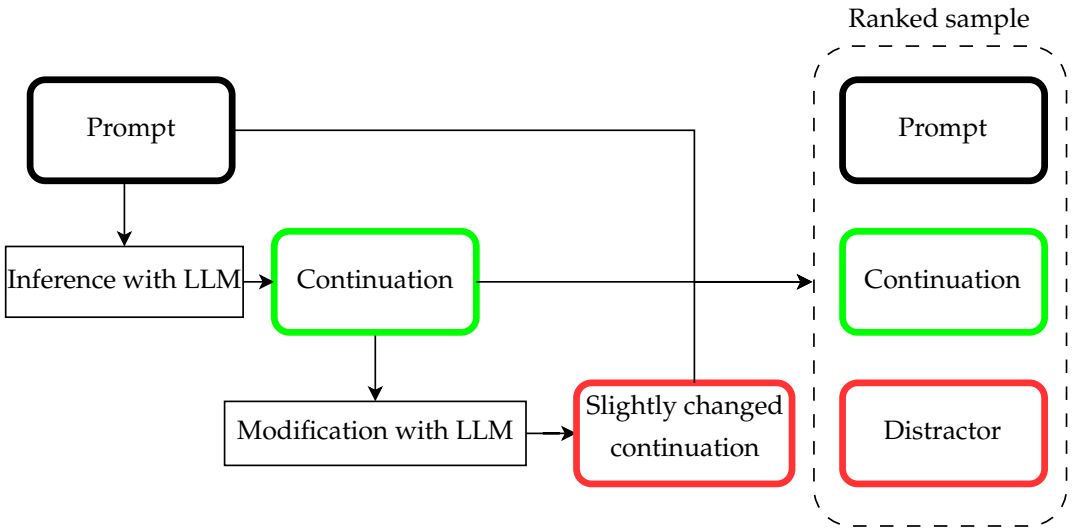


Figure 4.5: Visualisation of the generation of an AI Continuation pair.

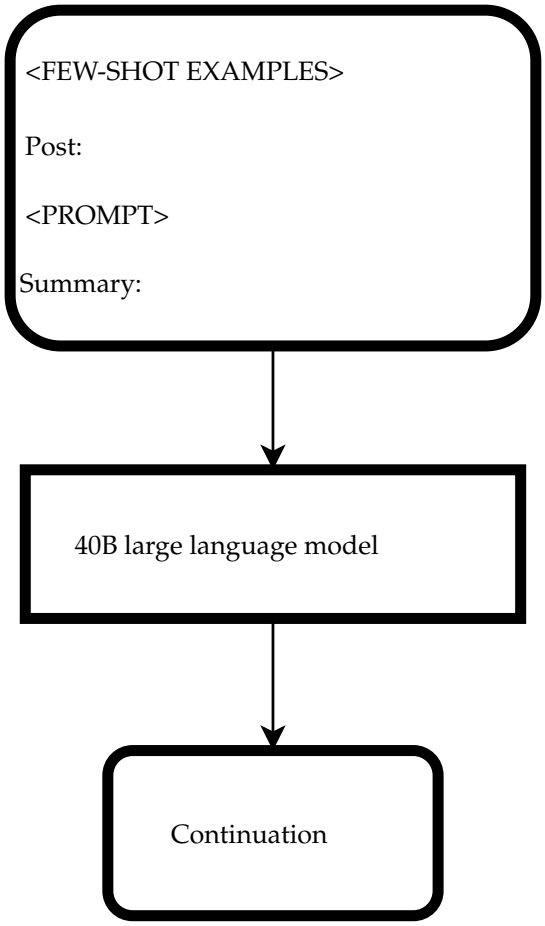


Figure 4.6: Visualisation of how a prompt is used to generate a continuation using a large language model. To construct the model input, the prompt is inserted into the given template. The few-shot examples used for all experiments are listed in Section A.2 in Appendix A.

ennon et al. [27], a validation set of 1911 samples was used for hyperparameter optimization, and then the final results were obtained on a test set of 50701 samples.

For the experiments, there are two approaches to integrating samples with generated continuation pairs into the base dataset: replacing samples or extending the dataset.

Replacing samples of the base dataset

In this setting, the number of samples is held constant, thus all data mixes with this setting contain 14700 samples. While the prompts remain the same as the base dataset, it is a partition of the continuation pairs that is replaced. In the case of a distractor, it is the worse continuation in the pair that is replaced by a distractor. The preferred continuation in the pair is kept, as well as used in the generation of the distractor. In the case of an AI continuation pair, the entire continuation pair is replaced.

In practice, it is unlikely that one would replace samples of a ranked dataset to improve performance, especially considering the shortage of ranked data and the expense of having humans ranking continuations. However, with a baseline with only human-ranked continuation pairs, including all data in the base dataset except the samples that are replaced in the experiments, the approach of replacement could be seen as simulating the following practical scenarios:

- There are additional prompts available for a dataset, with only one continuation per prompt. Distractors can then be used to generate continuation pairs for the additional prompts.
- There are additional prompts available for a dataset, however without any continuations. In this case, AI continuation pairs could be generated for the additional prompts.

As the results of replacing parts of the dataset with different types of generated continuation pairs can be compared with the results of training on the human base dataset as is, this experiment setting also allows insight into how automatic generation of continuation pairs for the new prompts compares to having annotators construct ranked continuation pairs for the new prompts.

For each experiment, a third (4.9k samples) of the base dataset is replaced by generated continuation pairs. To enable as fair comparisons as possible, it is always the same samples that are replaced. This partition of the data is substituted for four different types of generated data throughout the experiments: random distractors, AI distractors, mixed distractors (combination of random and AI distractors), and AI continuation pairs.

For the fully AI-generated continuation pairs, additional experiments where the continuations in the base dataset data are progressively replaced by AI continuation pairs are carried out. This progressive substitution corresponds to the seven datamixes listed in Table 4.1.

As to allow for insight into how adding AI continuations compares to not adding them at all, a human baseline is established through conducting training runs on only the human data from all these 7 datamixes, except for the datamix which only consists of AI continuation samples as it simply has no human data.

Table 4.1: Combinations of human samples and AI continuation pairs in the interpolation datamixes

Datamixes	
No. human samples	No. AI continuation pairs
14.70k	0.00k
12.25k	2.45k
9.80k	4.90k
7.35k	7.35k
4.90k	9.80k
2.45k	12.25k
0.00k	14.70k

Extending the base dataset

In this setting, no samples in the base dataset are replaced. Instead, the new samples are simply added, despite being based on existing samples in the dataset, resulting in that every prompt that has been used to construct a new ranked continuation pair will appear twice in the dataset. This setup intends to give insight into the scenario where there are no prompts without continuation pairs, and the extension of the dataset thus is based on existing prompts. For each type of new continuation pair, the base dataset is extended with generated continuation pairs corresponding to a third (4.9k samples), two thirds (9.8k samples),and all of the base dataset (14.7k samples). The base dataset is either extended with random distractors, AI distractors, or AI continuation pairs.

4.2.2 Curriculum learning

The data mix used for the curriculum learning corresponds to the base dataset with 14.7k samples, but for which 2450 human samples have been replaced by random distractor samples, and 2450 other human samples have been replaced with AI distractor samples. This combination of random and AI distractors is referred to as mixed distractors.

Despite various previous efforts in adapting curriculum learning to modern natural language processing discussed in Section 3.2, there are no successful previous work made that corresponds perfectly to the premises for this thesis.

Firstly, curriculum learning has not been used for large language reward model training in the literature previously. Secondly, as the reward model training is carried out over a single epoch as reward models are prone to overfitting if trained for multiple epochs [20], all methods that cannot be restricted to only exposing the model for each sample once during training are discarded. Thirdly, the purpose of the curriculum learning experiments in this thesis is to explore if the generated ranked data could be integrated more cleverly during training than simply distributing all sorts of generated data uniformly over the training stages. Thus these experiments rely on predefined levels of difficulty depending on the type of continuation pairs. Three different levels of difficulty have been identified in this thesis, where the simplest samples correspond to continuation pairs with random distractors, the medium-difficult samples correspond to continuation pairs with AI distractors, and the most difficult samples are continuation pairs ranked by humans. This further constrains the pool of previous work that could serve as a basis for this thesis' curriculum experiments.

Kocmi and Bojar [16] describe the most similar conditions as they train for one epoch, and that they have a sampling method compatible with more than two discrete levels of difficulty. However, their method requires the ranked subsets to decrease in size with increasing difficulty, which is not the case for the data used in this thesis.

Given the lack of previous work on curriculum learning for natural language with sampling algorithms applicable to the conditions in this thesis, the author has come up with three different simple sampling strategies, intending to give some initial insight in how sampling order of different types of generated data can impact reward model training. The sampling orders of data during training for these experiments, also referred to as curriculum setups, are visualized in Figure 4.7. Setup *C1* serves to model the simplest way possible to distribute the two types of generated data ordered by difficulty, whereas setup *C2* takes inspiration from how various previous work [16, 23, 40, 39, 29] progressively introduces harder samples during training. The proportion of human samples is kept constant for both *C1* and *C2* to enable a fair comparison between a direct and a progressive change of generated data type (or difficulty). Setup *C3* intends to give insight to the impact of progressively introducing the human (or hardest) data in the training.

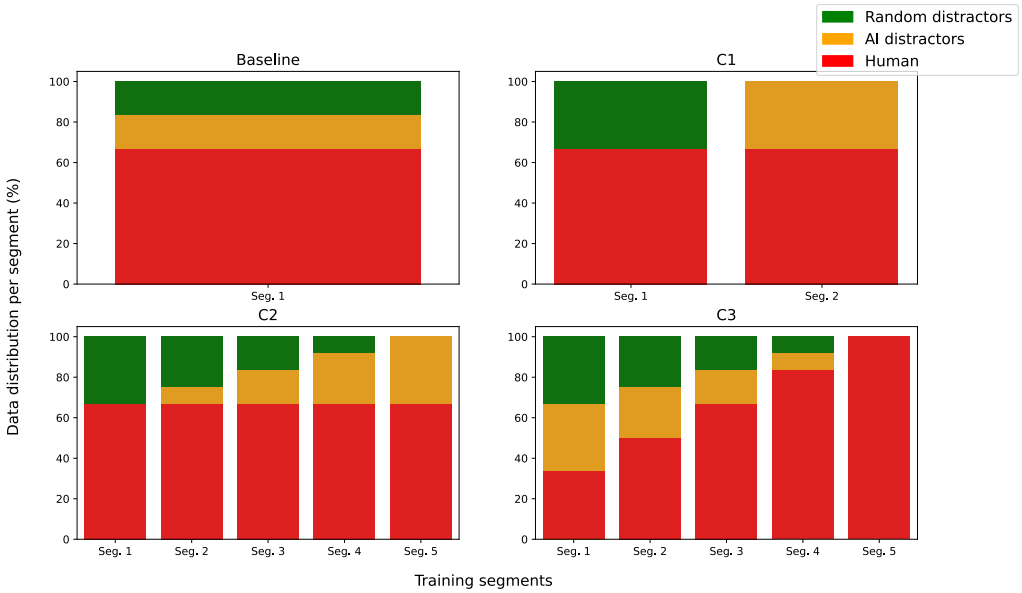


Figure 4.7: Visualisation of the curriculum setups, displaying the data distribution at different stages of training.

4.2.3 Training details

As the ordering of samples in a data set can have a significant impact on the results of fine-tuning large language models [8], all experiments are run over several seeds for data ordering. For hyperparameter optimization, three seeds are used, whereas the final runs on the test set are conducted over 5 seeds. Similar to previous research [27, 20, 2] , the reward model is only trained for one epoch as reward models are prone to overfitting [20]. As Stiennon et al. [27] and others [20, 13, 17], the learning rate is decayed with a cosine schedule for all experiments. Additionally, a constant learning rate is tried for the curriculum learning experiments, as results otherwise risk modeling the fact that a certain learning rate is better for a certain subtype of the data, rather than the consequences of data ordering by difficulty. Constant learning rates have also been used in previous literature describing reward model training [2, 3].

As mentioned previously in this chapter, the optimal batch size and learning rate found in the initial verification of the reward model through reproduction of the results of Stiennon et al. [27], are used as a starting point for the rest of the experiments. However, as the base dataset is significantly smaller than the dataset used for reproduction, a higher learning rate

might be better. In general, the following process is executed for each data mix, where the execution order of data mixes is ordered by decreasing order of the total number of samples in the datamix:

1. Run training with the optimal learning rate from previous experiments [1] over three seeds, both with and without warmup. For each of the two setups, calculate the mean accuracy over the different seeds. Note the highest validation accuracy and the hyperparameters used to obtain those results.
2. Increase the learning rate to the next higher value swept over in the initial log-linear sweep.
3. Run training with the new learning rate both with and without warmup similar to step 1. If a higher mean validation accuracy is given, note the hyperparameters used and start over from step 2. If the validation accuracy does not improve, the hyperparameter search is finished and the hyperparameter setting that has been noted as optimal is used to evaluate the data mix on the test set.

The exceptions, that do not necessarily follow the optimization methodology outlined above, are the runs of the data mixes that could be considered as interpolations of two other datamixes. These are 1) the data mix for replacement with mixed distractors (9.8k human + 2.45k AI distractors + 2.45k random distractors) which could be seen as an interpolation between the data mixes for replacement with AI distractors and random distractors, and 2) all data mixes for which a partition of the base dataset has been replaced with AI continuation pairs. Consider a data mix X, which can be seen as an interpolation of two data mixes Y and Z as follows:

$$X = \lambda Y + (1 - \lambda)Z,\ \lambda \in [0, 1]$$

If for an interpolated data mix with a given $\lambda = a$, the same hyperparameters are optimal both for a data mix with $\lambda > a$ as well as for a data mix with $\lambda < a$, it is assumed that the data mix with $\lambda = a$ also has the same optimal hyperparameters. To identify such cases, the execution of hyperoptimization experiments for data mixes that could be considered interpolations is ordered by decreasing order of $|\lambda - 0.5|$ for the data mixes. This adaptation to the hyperparameter optimization process is solely a measure to be more resourceful with computational resources.

The other exception is that for curriculum learning experiments, the optimal hyperparameters found for the baseline are also used for the curriculum setups.

[1] Optimal learning rate from previous experiments refers to the optimal learning rate for the smallest data mix containing at least all samples of the current data mix. If no such datamix exists, the optimal learning rate is considered to be the one used for the reproduction of the results of Stiennon et al. [27]

5 Results

The results chapter is divided into three different sections. In Section 5.1, the results for integrating AI distractors, random distractors, and AI continuation pairs into the base dataset through replacement are displayed. Then, the results when extending the base dataset with either AI distractors, random distractors, or AI continuation pairs are presented in Section 5.2. Lastly, Section 5.3 displays the results from the curriculum learning experiments. To enable better understanding of the results, the author makes the remark that the usage of bfloat16-precision[1] to calculate accuracies results in a discretization. Given the size of the test set being 50701 samples, two runs with the same accuracy with bfloat16-precision can differ by almost 200 samples in terms of number of correct predictions.

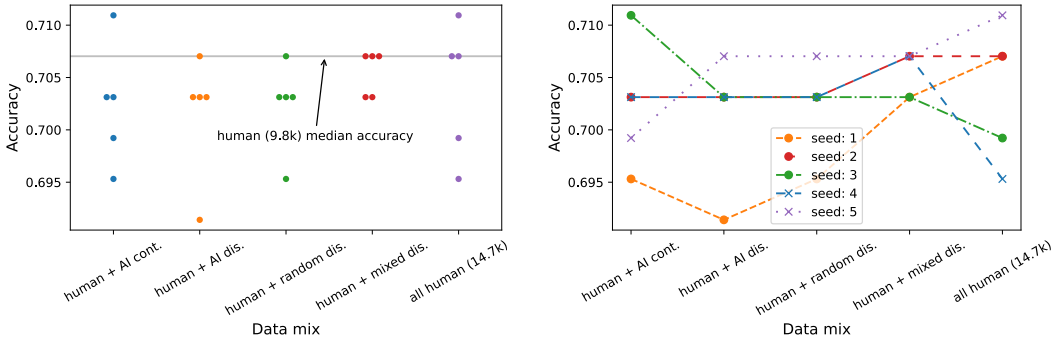
5.1 Replacing samples of the base dataset

The results for replacing samples of the base dataset with either AI distractors, random distractors, mixed distractors (50 % random distractors and 50 % AI distractors) or AI continuation pairs are displayed in Figure 5.1. While all replacement strategies give similar accuracies, replacement with mixed distractors exhibits the highest mean and median accuracy, in addition to achieving the highest accuracy across the majority of seeds. In Figure 5.2, the results for gradually replacing the base dataset with AI continuation pairs until there are only AI continuation pairs in the datamix are displayed. Compared to the baseline with samples only from the base dataset, adding AI continuation pairs never increases accuracy on the test set. However, training with only AI continuation pairs outperforms the random baseline with 50 % accuracy. The optimal hyperparameters found during hyperparameter optimization and thus the hyperparameters used to obtain these results are displayed in Table 5.1 and Table 5.2.

5.2 Extending the base dataset

The results for the extension experiments, where the base dataset is extended with either AI distractors, random distractors, or AI continuation pairs are displayed in Figure 5.3. Adding

[1] https://cloud.google.com/tpu/docs/bfloat16



(a) Performance

(b) Seed-wise performance

Figure 5.1: Performance over 5 seeds for different types of data mixes when integrating altered continuation pairs through replacing samples of the base dataset. For comparison purposes, both figures contains performances of the base dataset without replacement (*human 14.7k*). Subfigure 5.1a also contain the median accuracy on the base dataset without the samples that else are replaced (*human (9.8k)*). All data mixes except for the human baseline with 9.8k samples and the base dataset (*all human (14.7k)*) consist of 14.7k samples, where 4.9k samples have been replaced by samples with the given type of continuation pairs.

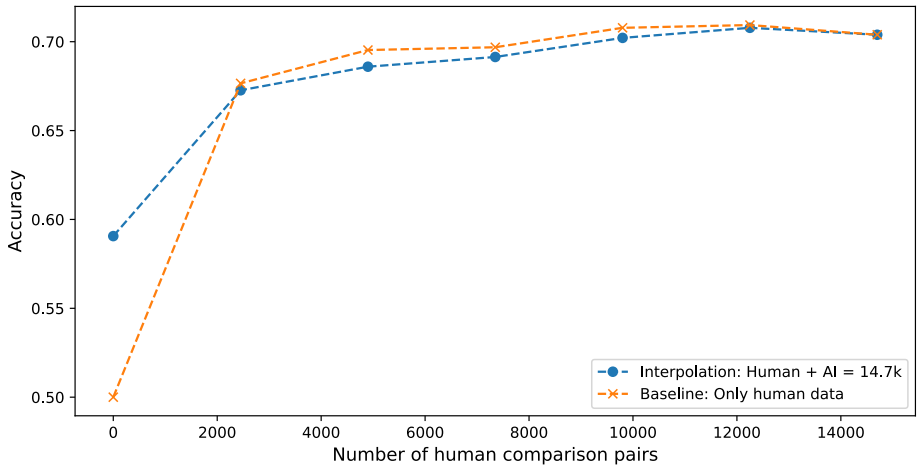


Figure 5.2: Mean accuracy over 5 seeds for gradually replacing the base dataset with AI continuation pairs until there are only AI continuation pairs in the data mix, which for all runs have the same total number of samples. For comparison purposes, a baseline with mean accuracies trained on only human data is plotted beside the interpolation between human and AI data. When there is no human data and thus no training data at all in the baseline dataset, a fully random performance corresponding to an accuracy of 50 % is assumed.

Table 5.1: Optimal hyperparameters found during optimization for human baselines

Hyperparameters		
Data mix	Learning rate	Warmup
Human 14.7k	1E-5	NO
Human 12.25k	1E-5	NO
Human 9.8k	1E-5	NO
Human 7.35k	2E-5	NO
Human 4.9k	2E-5	NO
Human 2.45k	4E-5	NO

Table 5.2: Optimal hyperparameters found during optimization for replacement experiments

Hyperparameters		
Data mix	Learning rate	Warmup
Human 9.8k + Random Distractors 4.9k	5E-6	YES
Human 9.8k + AI Distractors 4.9k	5E-6	YES
Human 9.8k + Mixed Distractors 4.9k	5E-6	YES
Human 12.25k + AI Continuation Pairs 2.45k	5E-6	NO
Human 9.8k + AI Continuation Pairs 4.9k	5E-6	NO
Human 7.35k + AI Continuation Pairs 7.35k	5E-6	NO
Human 4.9k + AI Continuation Pairs 9.8k	5E-6	NO
Human 2.45k + AI Continuation Pairs 12.25k	5E-6	NO
Human 0k + AI Continuation Pairs 14.7k	5E-6	NO

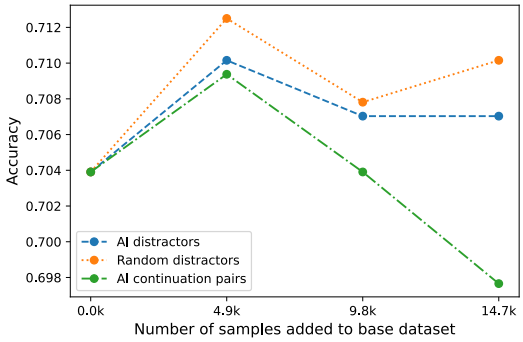
random distractors gives the highest mean accuracy as well as highest accuracy over the majority of seeds for all sizes of added data. The optimal hyperparameters found for the extension strategies are displayed in Table 5.3.

Table 5.3: Optimal hyperparameters found during optimization for extension experiments

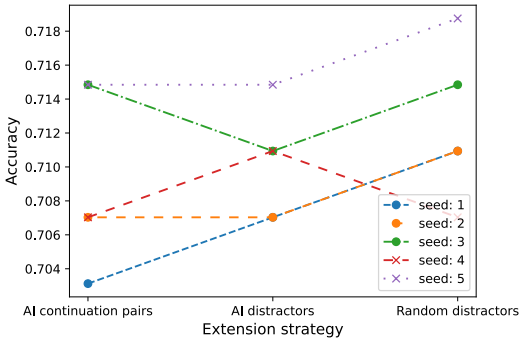
Hyperparameters		
Data mix	Learning rate	Warmup
Human 14.7k + Random Distractors 4.9k	5E-6	NO
Human 14.7k + AI Distractors 4.9k	5E-6	YES
Human 14.7k + AI Continuation Pairs 4.9k	5E-6	YES
Human 14.7k + Random Distractors 9.8k	2.5E-6	YES
Human 14.7k + AI Distractors 9.8k	2.5E-6	YES
Human 14.7k + AI Continuation Pairs 9.8k	2.5E-6	YES
Human 14.7k + Random Distractors 14.7k	2.5E-6	YES
Human 14.7k + AI Distractors 14.7k	2.5E-6	YES
Human 14.7k + AI Continuation Pairs 14.7k	2.5E-6	YES

5.3 Curriculum learning with distractors

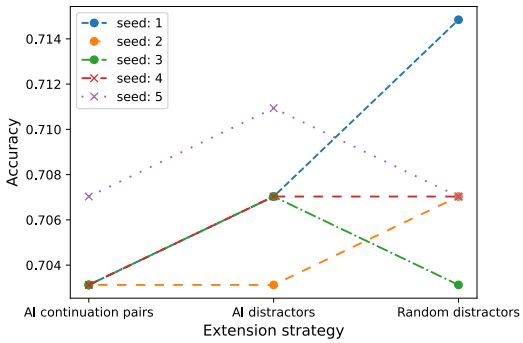
The results for the curriculum learning experiments are displayed in Figure 5.4c, and the optimal hyperparameters used to obtain these results are displayed in Table 5.4. For both learning rate schedules, setup *C1* is on par or better than the baseline, whereas setup *C3* gives the lowest accuracy. Setup *C2* is better than baseline with annealing cosine learning rate schedule, but worse than baseline with fixed learning rate.



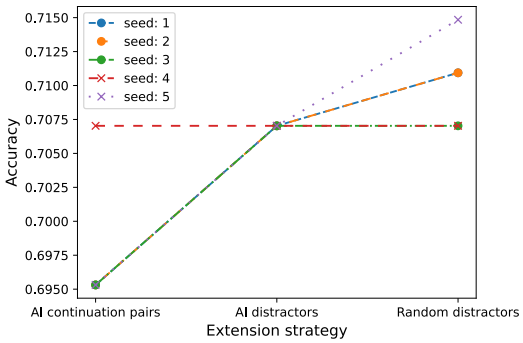
(a) Mean performance of all extension strategies



(b) Seed-wise performance - 4.9k added samples



(c) Seed-wise performance - 9.8k added samples

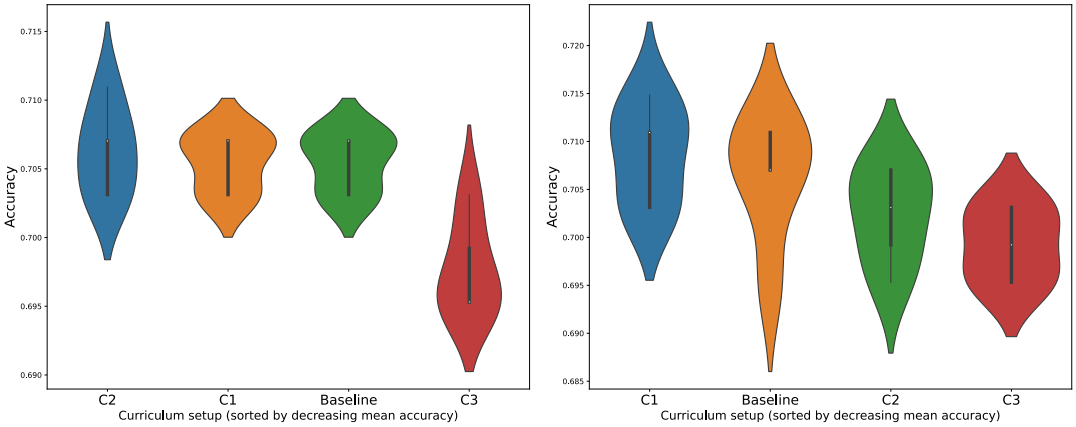


(d) Seed-wise performance - 14.7k added samples

Figure 5.3: Performance for all extension strategies over 5 different seeds. Subfigure 5.3a allows for comparison between the different strategies through displaying the mean accuracy for each strategy over all seeds. In Subfigures 5.3b, 5.3c and 5.3d, the performance is disaggregated over all seeds for each extension magnitude.

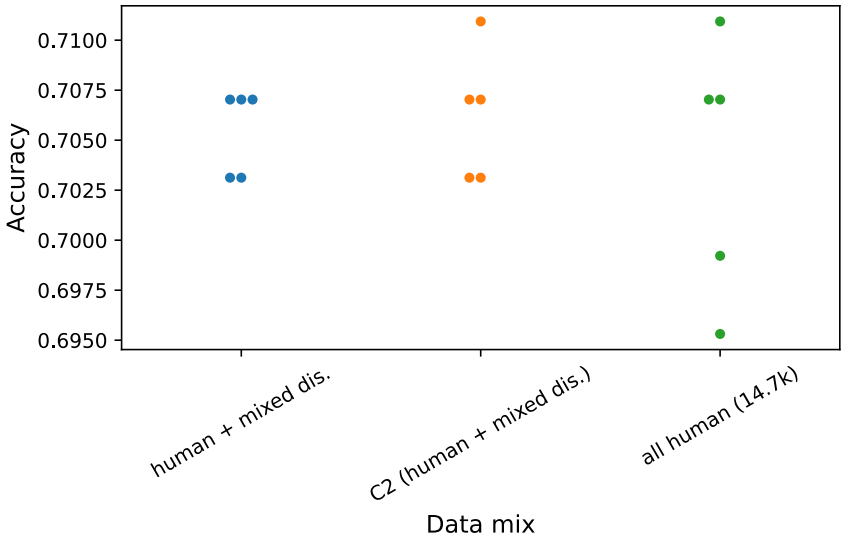
Table 5.4: Optimal hyperparameters found during optimization for curriculum experiments

Hyperparameters		
Data mix	Learning rate	Warmup
Human 9.8k + Random Distractors 2.45k + AI Distractors 2.45k	5E-6	NO
Human 9.8k + Random Distractors 2.45k + AI Distractors 2.45k (Fixed learning rate)	2.5E-6	-



(a) Performance distribution - cosine lr schedule

(b) Performance distribution - fixed lr



(c) Curriculum setups

Figure 5.4: Performance of different curriculum setups over 5 seeds. Subfigures 5.4a and 5.4b show the accuracy distributions for each setup with an annealing cosine schedule for learning rate (as for all other experiments) and a fixed learning rate respectively. Subfigure 5.4c displays how the curriculum setup with the highest mean accuracy with cosine decayed learning rate (*C2*) compare to the unordered baseline (*human + mixed dis.*) as well as the base dataset (*all human*). For reference, the distribution of the data types exposed to the reward model during training for the different curriculum setups are shown in 4.7.

6 Discussion

This chapter discusses the thesis from three different perspectives. Firstly, the results are discussed in Section 6.1, then the method is discussed in Section 6.2, and lastly societal and ethical aspects of this thesis are discussed in Section 6.3.

6.1 Results

In the following section, the results of this thesis are discussed. First, the results are discussed per section as they are presented in Chapter 5. Then, all results are discussed together in a concluding discussion in Subsection 6.1.4.

6.1.1 Replacing samples of the base dataset

In general, the performances of the different replacement strategies are all similar to the performance on the base dataset, as seen in Figure 5.1. In terms of mean accuracy, replacing 4.9k of the samples of the base dataset with mixed distractors gives a higher performance than the base dataset itself. This setting also produces the most consistent results over the different seeds, possibly indicating it being less dependent on data ordering than the other techniques. All the other replacement strategies have lower mean accuracy than training on the base dataset. However, the differences in mean accuracies are too small given the limited number of samples to draw any stronger conclusions. Only considering mean accuracies, the performance of the reward model does not change considerably depending on the replacement strategy used.

When disaggregating the results over the 5 different seeds there are some observations with respect to consistency over seeds that can be made. First of all, the data mix with AI distractors and the data mix with random distractors perform the same over all seeds except one, where the random distractor data mix performs slightly better. When these two distractors instead are mixed, the accuracy improves on the majority of seeds compared to both data mixes containing these distractors separately, whereas it stays the same on the remaining seeds. In addition to showcasing the potential of the AI distractors introduced in this thesis, this could indicate the usefulness of having diversity in the generated data that is added.

When observing other differences in seed-wise performance, there are no clear patterns that can be seen between the different replacement strategies.

From the interpolation results between the base dataset and AI continuation pairs, displayed in Figure 5.2, there are three interesting observations that can be made:

- **Potential of AI preferences:** Reward models trained on data for which the continuation pairs have been fully AI-generated is capable of modeling human preferences to a certain extent, as they achieve an accuracy of almost 60 %. Though significantly over the random performance of 50 %, this could still be considered a low accuracy. However, put in relation to the labeler agreement of 73% ± 2% of the summarization dataset by Stiennon et al. [27], which also corresponds to the accuracy a 7B reward model would get when trained on this dataset according to their experiments, the gap is not as big as it might seem. The method for generating AI continuation pairs is a rudimentary early step into AI-generated ranked data for reward model training intending to model human preferences, with many opportunities for refinement and improvement. Thus, this accuracy should be possible to improve further.

  Bai et al. [3] display the potential of AI-ranked data for harmfulness, as adding it to human preference data for helpfulness when training a reward model makes the resulting large language model after reinforcement learning using the reward model more harmless than it would have been without ranked harmfulness data. This thesis however provides results for AI-generated ranked data not restricted to harmfulness, and is further different and novel as it evaluates the performance of a reward model which only has been trained on AI-generated ranked data.

- **Unexpected deterioration of performance when increasing data mix size:** Test accuracy on the base dataset (14.7k human samples) is lower than the accuracy achieved when training on 12.25k and 9.8k human samples. There are several reasons that could explain this. In the method, hyperparameter optimization has been carried out separately for all data mixes with human data. It could be that the granularity of the hyperparameter optimization search space was not fine enough, potentially causing the hyperparameters to be much more optimal for some of the data mixes than others. This hypothesis is however made less likely by previous findings by Ouyang et al. [27], observing that reward model training is not very sensitive to learning rate changes.

  Another theory is that the partition of human data that is in the base dataset but not in the other sets is of lower quality, causing the performance to deteriorate. To test this theory, a simple ablation study is carried out where the human dataset of 9.8k samples is regenerated two times by removing other parts of the base dataset than the one removed for the main experiments in this thesis. For the two generations where the suspected part of the data is included and other parts are removed instead, the mean and the median of the accuracies for both regenerations are notably lower, to the extent that they are not higher than the mean and median of the accuracies of the base dataset anymore. There is thus reason to suspect the performance deterioration being attributable to data quality. The details of this extra study can be found in Appendix B.

- **Adding AI data does not improve accuracy:** Adding AI-generated ranked data to human data never improves accuracy on the test set in comparison to the baseline during the interpolation experiments. If the potential data quality issues described above would be due to bad prompts, this could also explain why AI-generated ranked pairs based on these prompts do not improve performance.

  Bai et al. [3] also find that adding AI-generated data decreases performance on the main task of the already existing data. However, the data added is for harmlessness whereas

the existing data is for helpfulness, between which they describe that a natural tension exists. This is not the case in this thesis, as both the existing data and generated AI continuations are for the task of summarization. Nevertheless, there is still a potential that the existing data and the introduced data of AI continuations cover different types of summaries as well as different aspects with respect to preferences. As discussed further in Subsection 6.2.3, the test set is limited to preferences with respect to the type of summarization found there, and it is possible that the AI-generated continuation pairs contribute to preferences with respect to other aspects. As Bai et al. [3] find that added AI-generated data improves performance for aspects covered by the generated data (harmfulness), it could be that the addition of AI continuation pairs improves the modeling of human preferences for other types of summaries and aspects of those that are not contained in the test set. This exposes the need for a more thorough validation of the preference modeling capabilities of the reward models.

6.1.2 Extending the base dataset

As seen in Figure 5.3, all data generation techniques manage to improve the performance in comparison to the base dataset when they are used to extend the dataset with 4.9k samples. Random distractors improve the performance the most considering mean accuracy over seeds, whereas AI distractors improve the performance the second most. For all seeds and the number of added samples, random distractors always have a performance on par with or better than AI continuation pairs. Furthermore, AI continuation pairs only outperforms AI distractors on one single run out of all runs for different seeds and numbers of added samples.

Similar to as discussed in the previous subsection, Subsection 6.1.1, the contribution of AI continuation pairs to the preference modeling capabilities of a reward model might not display the full contribution of AI continuation pairs as the continuations might be distributionally different from those in the base dataset. This could explain why the generated pairs with distractors perform better, as the preferred continuation always comes from the already existing dataset. That random distractors give better performance than AI distractors is less intuitive. The author hypothesizes that the random distractors possibly make the reward model better at distinguishing summaries that are simply well formulated from summaries that correspond well to the actual text to summarize. Reward model training with generated pairs including random distractors will expose the model for certain summaries twice or more, both as a preferred option for the prompt it was written for as well as the non-preferred option as a distractor for a prompt it has not been written for.

After adding 4.9k samples, adding more samples does not give better performance for any type of generated data pair. In fact, adding 14.7k samples of AI continuation pairs give even worse performance than without adding any data at all. As Ouyang et al. find reward models prone to overfitting if trained for more than one epoch [20], it could possibly be the repetition of prompts as well as continuations (depending on the method for generating the ranked pairs), that causes this degradation of performance when adding more generated data. The degradation seen could also be attributed to the fact the base dataset, most similar to the test set, becomes increasingly diluted when more data is added, and thus this data has less impact on the gradients used to update the reward model weights during training.

6.1.3 Curriculum learning with distractors

As seen in Figure 5.4, curriculum learning is able to give a slight improvement over the randomly ordered baseline for both learning rate schedules. As the baseline with mixed distractors performed the best out of all data mixes with 14.7k samples in the replacement exper-

iments, it consequently means that the curriculum setup *C2*, performing the best out of all setups with cosine learning rate schedule, performs the best out of all data mixes with 14.7k samples in this thesis. With a fixed learning rate, it is instead the *C1* setup that performs the best. However, the baseline and the setups *C1* and *C2* give such similar accuracies that the differences could be attributed to randomness, whereas it is more clear that setup *C3* lags behind the others.

As the baseline and the setups *C1* and *C2* all have the human data from the base dataset evenly distributed over all stages of the reward model training, the results could indicate that it is favorable to have a constant percentage of the main data when doing curriculum learning with distractors. The author hypothesizes that this could be due to the large difference in difficulty and other aspects of the different types of data in the dataset.

For a reward model to identify which of the continuations in a ranked continuation pair with a random distractor that is the distractor, advanced language modeling capabilities are likely not needed as the distractor generally will be notably distributionally shifted from the prompt. In the extreme case where a reward model would only be trained on random distractor data, the initial language modeling capabilities would probably vanish, making it harder for the model to afterward learn on the harder data. If it wasn't for the fact that reward models leverage language modeling capabilities from pre-trained large language models through their initialization [1, 41, 27, 20, 17, 13], initial learning on samples with random distractors could have been more useful. Instead, it is likely that the reward model will do best if there is always a significant part of the samples seen during training that needs such language capabilities, which in turn could explain the lower accuracy of setup *C3*. The importance of language modeling capabilities in reward model training is indicated by Askell at al. [1], showing significantly better results for the final accuracy if the reward model is trained to keep its language modeling capabilities during the preference model pre-training. In this case, through being trained on a combined loss for ranking and regular language modeling.

6.1.4 Overarching discussion of results

In the cases where different data mixes give test accuracies that are not notably different, it becomes more important to investigate how the data mixes impact the performance on data that is outside of the distribution of the test set. The concerns related to evaluating solely on one single dataset are further discussed in Subsection 6.2.3.

In both the extension experiments and curriculum experiments, there are indications that the accuracy on the test set will degrade if the data at any training stage is diluted with too much generated data. Additionally, the results from the replacement experiments show that instead of adding one sort of generated data, it can be better to add fewer samples of several types of data. This is indicated by the fact that the combination of AI distractors and random distractors (*mixed distractors*) perform better than all other replacement strategies.

Excluding mixed distractors from the generation strategies used in the replacement experiments, random distractors give the best performance for both replacement experiments and extension experiments

6.2 Method

In the following section, the method and its main limitations are discussed. Central to this discussion are the methodological concepts replicability discussed in Subsection 6.2.1, reliability discussed in Subsection 6.2.2, and validity discussed Subsection 6.2.3. Lastly, the sources are discussed in Subsection 6.2.4.

6.2.1 Replicability

Overall, the description of the method for carrying out the work performed in this thesis should be precise enough to enable reproduction of results. However, there are some weaknesses with respect to the initial implementation part concerning creating the framework for training reward models, as including all implementation details simply is out of scope for this thesis.

6.2.2 Reliability

Firstly, data order during training can have a substantial impact on training [8], which has negative consequences for reliability. Through running the experiments repeatedly over several seeds, the impact of data order can however be reduced. In the method, this issue is addressed as each hyperparameter optimization is run over three seeds, and the final results are obtained through running the experiments on 5 seeds. Ideally, the experiments would have been run over even more seeds, however there is a trade-off between cost and reliability as the training requires large computational resources and thus is expensive. In previous work, the number of seeds used is not always disclosed, but the examples that do exist make the choice of number of seeds in this thesis to be within reasonable scope. Stiennon et al. [27] use 3-10 seeds during hyperparameter optimization of their reward models, whereas Ouyang et al. [20] use 3 seeds for some experiments.

Secondly, the experiments rely on specific large language models, which both impact the data generation and the reward model's initialization. The quality of the data generated using the 40B parameter large language model naturally depend on the characteristics of that specific model, making it possible that another large language model gives different results. Similarly, the initialization of the reward model with weights from a 7B large language model makes the reward model dependent on the quality of the model used as starting point.

Thirdly, the results vary depending on what part of the base dataset that is excluded during the replacement experiments. As shown in Appendix B, the performance when excluding different parts of the base dataset can be notably different.

6.2.3 Validity

As this thesis has the ambition to generate a better understanding for how existing and new strategies for generating ranked data impact a reward model's ability to model human preferences, validity becomes a reasonable concern. First of all, human preferences are subjective by nature. There is thus no unequivocal ground truth to if something aligns with human preferences or not. The subjectivity of human preferences concerning natural language is further demonstrated by the labeler agreement for the ranked continuation dataset used in this thesis, which is 73% ± 2% [27].

In addition to the ambiguious underlying nature of preferences, the method of evaluating results on a single ranked dataset has other shortcomings. When using performance on a certain dataset with rankings based on human preferences as a proxy for human preference modeling, the notion of human preferences is constrained to the preferences of the labelers, as well as to preferences only considering the samples available in the dataset. Preferences with respect to aspects not covered in the dataset are consequently not taken into consideration in the performance metric. This issue is potentially even more apparent when generated data has been integrated in the training data. The reason for this is that generated preference data, AI continuation pairs in particular, are distributionally shifted from the original dataset and thus also the test set. Whereas this might make the reward model model human preferences better on a wider variety of data, the test performance might not be affected at all.

6.2.4 Source criticism

Research on large language models can be very costly, resulting in a limited number of institutions or organizations that have the financial means to contribute to the field. While the main sources used are up to date and reflect the latest advancements, there is a lack of diversity as previous research on reward models is dominated by a few big actors. Furthermore, experiments that are difficult and very expensive to reproduce may be less likely to be replicated, thus potentially endangering the ability to ensure that previous work is of high quality. Additionally, the papers by these dominating actors are not necessarily published in peer-reviewed journals, further threatening the academic credibility of these works. However, this is compensated for to some extent by the fact that the sources used are in general well-cited as a consequence of their impact and relevance to the field.

6.3 The work in a wider context

Better modeling of human preferences by reward models will enable better alignment of large language models to human preferences. In turn, this can make large language models more useful for humans and enable more automatization of tasks in society. Eloundou et al. [9] research how large language models can impact the industries and occupations in the United States, finding out that almost 20 % of the workforce can use large language models for the majority of their tasks.

However, as Ouyang et al. [20] point out, making large language models better according to human preferences, in their case better adapted to following instructions, can also make them easier to use for misuse. Brown et al. [5] exemplify potential areas of misuse, such as automatic generating of text for misinformation, spam, and phishing.

Allegedly, previous improvements of large language models, enabled by human annotation of text data, have involved outsourcing to workers operating under questionable working conditions [22]. With automatic generation of data on the other hand, such as generation with the techniques explored in this thesis, the dependency on humans and thus potentially labor under poor working conditions could decrease.

As previously discussed in Subsection 6.2.3, the ambiguous nature of human preferences makes it important to consider which human preferences the reward model actually models. If reward models fail to model different groups of people to an equal extent, large language models then risk being optimized for certain groups, displaying the risk that automatization might happen at the expense of increased discrimination.

7 Conclusion

The initial aim of this thesis was to generate a better understanding of how existing and new strategies for generating ranked data impact the ability of a reward model to model human preferences, as well as to explore how the integration of such data into existing dataset can be optimized. In order to achieve this ambition, the following research questions have been investigated:

- **RQ1:** How does adding random distractors, AI distractors, and AI continuation pairs in training data impact accuracy on a dataset based on human preferences?
- **RQ2:** Can curriculum learning be applied to reward model training with distractors, where the difficulty of a sample is determined based on how it has been generated, in order to improve accuracy?
- **RQ3:** To what extent can reward models trained on only AI-generated continuation pairs model human preferences, in terms of accuracy on a dataset based on human preferences?

Through experiments with replacing samples of a base dataset of human preferences as well as adding samples to this base dataset, the impact of different generation strategies have been investigated which ultimately has given a response to **RQ1**. When added in the best way, amount, and proportion to existing training data, all generation strategies manage to improve the test accuracy for the resulting reward model. For the replacement strategies, all generation techniques have very similar accuracies, all close to the base dataset where no sample has been replaced. Through mixing AI distractors introduced in this thesis with the already existing random distractors for replacing samples, the replacement strategy with both the best and most consistent results is found, outperforming the base dataset with human samples instead of the mixed distractors. For the extension experiments, the smallest extension for all generation strategies gives the greatest improvement in test accuracy. Random distractors improve the accuracy the most for all magnitudes of extensions, whereas AI continuation pairs perform the worst with respect to test accuracy for extension experiments.

However, the experiments also display cases where the addition of generated ranked data worsen validation accuracy.

Through testing different curriculum learning setups, **RQ2** is possible to answer, though with limited certainty. It appears that curriculum learning indeed has the potential to improve the integration of distractors in training as the unordered baseline is outperformed by different curriculum setups for both learning rate schedules. However, this is with a minimal marginal, indicating the need for further investigations to find out if the successful curriculum learning setups tested in this thesis are beneficial in general or if these slight improvements only were due to randomness.

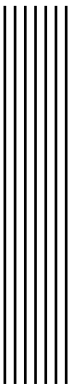
Lastly, **RQ3** is answerable as the experiments conducted in this thesis has shown that reward models trained on data for which the continuation pairs have been fully AI-generated achieve an accuracy of almost 60 % on the test dataset with human preferences, indicating a capability to model human preferences to a certain extent. These results mark a first step into evaluating accuracy with respect to human preferences for reward models trained on only AI-generated rankings, shining light on the potential for the in this thesis presented AI continuation pairs if they would successfully be developed further.

As all generation techniques studied in this thesis are task agnostic, the results could be generalizable to other tasks than summarization. However, certain tasks such as creative writing, where changing details using AI distractors do not necessarily lead to worse continuations, could require some adaptation of how a continuation is modified with a large language model into a distractor.

7.1 Future work

Future work should address the delimitations of this thesis. First of all, future research should address the research questions of this thesis for other datasets and tasks, in order to see if the insights of this thesis generalize over these axes. Secondly, future research require a more thorough evaluation of reward models to improve the validity of the results. As the accuracy on a single test potentially only displays the tip of the iceberg of the reward model performance, a more thorough evaluation could help in making better data decisions, especially in the cases where there is currently little difference in accuracy for different setups. This involves comparing the performance of large language models that have been aligned with different reward models during reinforcement learning, as well as finding more diverse preference datasets to base accuracy calculations on. Related to this, future research should address what human preferences the model actually models, as a step towards ensuring that the resulting reward model does not favor preferences of certain people more than others. Thirdly, future research should address the reliability issues faced in this thesis, where potentially varying data quality decreases the reliability of results. More robustness with respect to how the data is partitioned would allow higher reliability, and thus higher quality of the research overall.

Furthermore, this thesis presents promising results for both techniques of combining generated data as well as integrating these combinations cleverly into training through curriculum learning. As these outperform other setups, further research should be made on different ways of combining generated data and how such techniques can be improved. Additionally, the generation of AI continuation pairs is at its first iteration, and it already has shown abilities to model human preferences to a certain extent. Thus, further research should capitalize on this potential by investigating how the usage of large language models to generate ranked data can be improved in order to train reward models giving better modeling of human preferences.



Bibliography

[1] Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, et al. "A general language assistant as a laboratory for alignment". In: *arXiv preprint arXiv:2112.00861* (2021).

[2] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. "Training a helpful and harmless assistant with reinforcement learning from human feedback". In: *arXiv preprint arXiv:2204.05862* (2022).

[3] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. "Constitutional AI: Harmlessness from AI Feedback". In: *arXiv preprint arXiv:2212.08073* (2022).

[4] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. "Curriculum Learning". In: *Proceedings of the 26th Annual International Conference on Machine Learning*. ICML '09. Montreal, Quebec, Canada: Association for Computing Machinery, 2009, pp. 41–48. ISBN: 9781605585161. DOI: 10.1145/1553374.1553380. URL: https://doi.org/10.1145/1553374.1553380.

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

[6] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. "Palm: Scaling language modeling with pathways". In: *arXiv preprint arXiv:2204.02311* (2022).

[7] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. "Scaling instruction-finetuned language models". In: *arXiv preprint arXiv:2210.11416* (2022).

[8] Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. "Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping". In: *arXiv preprint arXiv:2002.06305* (2020).

[9] Tyna Eloundou, Sam Manning, Pamela Mishkin, and Daniel Rock. "Gpts are gpts: An early look at the labor market impact potential of large language models". In: *arXiv preprint arXiv:2303.10130* (2023).

[10] *Falcon-40B-Instruct*. https://huggingface.co/tiiuae/falcon-40b-instruct. Accessed: 2023-05-30.

[11] *Falcon-7B*. https://huggingface.co/tiiuae/falcon-7b. Accessed: 2023-05-30.

[12] Deep Ganguli, Liane Lovitt, Jackson Kernion, Amanda Askell, Yuntao Bai, Saurav Kadavath, Ben Mann, Ethan Perez, Nicholas Schiefer, Kamal Ndousse, et al. "Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned". In: *arXiv preprint arXiv:2209.07858* (2022).

[13] Amelia Glaese, Nat McAleese, Maja Trębacz, John Aslanides, Vlad Firoiu, Timo Ewalds, Maribeth Rauh, Laura Weidinger, Martin Chadwick, Phoebe Thacker, et al. "Improving alignment of dialogue agents via targeted human judgements". In: *arXiv preprint arXiv:2209.14375* (2022).

[14] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. "An empirical analysis of compute-optimal large language model training". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 30016–30030.

[15] Srinivasan Iyer, Xi Victoria Lin, Ramakanth Pasunuru, Todor Mihaylov, Dániel Simig, Ping Yu, Kurt Shuster, Tianlu Wang, Qing Liu, Punit Singh Koura, et al. "OPT-IML: Scaling Language Model Instruction Meta Learning through the Lens of Generalization". In: *arXiv preprint arXiv:2212.12017* (2022).

[16] Tom Kocmi and Ondřej Bojar. "Curriculum Learning and Minibatch Bucketing in Neural Machine Translation". In: *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*. Varna, Bulgaria: INCOMA Ltd., Sept. 2017, pp. 379–386. DOI: 10.26615/978-954-452-049-6_050. URL: https://doi.org/10.26615/978-954-452-049-6_050.

[17] Jacob Menick, Maja Trebacz, Vladimir Mikulik, John Aslanides, Francis Song, Martin Chadwick, Mia Glaese, Susannah Young, Lucy Campbell-Gillingham, Geoffrey Irving, et al. "Teaching language models to support answers with verified quotes". In: *arXiv preprint arXiv:2203.11147* (2022).

[18] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. "Asynchronous methods for deep reinforcement learning". In: *International conference on machine learning*. PMLR. 2016, pp. 1928–1937.

[19] OpenAI. "GPT-4 technical report". In: *arXiv* (2023).

[20] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll 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 F Christiano, Jan Leike, and Ryan Lowe. "Training language models to follow instructions with human feedback". In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh. Vol. 35. Curran Associates, Inc., 2022, pp. 27730–27744. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf.

[21] Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. "Instruction tuning with gpt-4". In: *arXiv preprint arXiv:2304.03277* (2023).

[22] Billy Perrigo. "Exclusive: OpenAI Used Kenyan Workers on Less Than $2 Per Hour to Make ChatGPT Less Toxic". In: *TIME* (Jan. 18, 2023). URL: https://time.com/6247678/openai-chatgpt-kenya-workers/ (visited on 06/17/2023).

[23] Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom M Mitchell. "Competence-based Curriculum Learning for Neural Machine Translation". In: *Proceedings of NAACL-HLT*. 2019, pp. 1162–1172.

[24] Rajkumar Ramamurthy, Prithviraj Ammanabrolu, Kianté Brantley, Jack Hessel, Rafet Sifa, Christian Bauckhage, Hannaneh Hajishirzi, and Yejin Choi. "Is Reinforcement Learning (Not) for Natural Language Processing?: Benchmarks, Baselines, and Building Blocks for Natural Language Policy Optimization". In: *arXiv preprint arXiv:2210.01241* (2022).

[25] Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, et al. "Multitask Prompted Training Enables Zero-Shot Task Generalization". In: *International Conference on Learning Representations*. 2022.

[26] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. "Proximal policy optimization algorithms". In: *arXiv preprint arXiv:1707.06347* (2017).

[27] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. "Learning to summarize with human feedback". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin. Vol. 33. 2020, pp. 3008–3021. URL: https://proceedings.neurips.cc/paper/2020/file/1f89885d556929e98d3ef9b86448f951-Paper.pdf.

[28] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[29] Yi Tay, Shuohang Wang, Anh Tuan Luu, Jie Fu, Minh C Phan, Xingdi Yuan, Jinfeng Rao, Siu Cheung Hui, and Aston Zhang. "Simple and Effective Curriculum Pointer-Generator Networks for Reading Comprehension over Long Narratives". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2019, pp. 4922–4931.

[30] Yi Tay, Jason Wei, Hyung Won Chung, Vinh Q Tran, David R So, Siamak Shakeri, Xavier Garcia, Huaixiu Steven Zheng, Jinfeng Rao, Aakanksha Chowdhery, et al. "Transcending scaling laws with 0.1% extra compute". In: *arXiv preprint arXiv:2210.11399* (2022).

[31] Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. "Lamda: Language models for dialog applications". In: *arXiv preprint arXiv:2201.08239* (2022).

[32] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. "FEVER: a Large-scale Dataset for Fact Extraction and VERification". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. 2018, pp. 809–819.

[33] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. "Llama: Open and efficient foundation language models". In: *arXiv preprint arXiv:2302.13971* (2023).

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

[35] Thomas Wang, Adam Roberts, Daniel Hesslow, Teven Le Scao, Hyung Won Chung, Iz Beltagy, Julien Launay, and Colin Raffel. "What Language Model Architecture and Pretraining Objective Works Best for Zero-Shot Generalization?" In: *International Conference on Machine Learning*. PMLR. 2022, pp. 22964–22984.

[36] Xin Wang, Yudong Chen, and Wenwu Zhu. "A Survey on Curriculum Learning". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.9 (2022), pp. 4555–4576. DOI: 10.1109/TPAMI.2021.3069908.

[37] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. "Self-Instruct: Aligning Language Model with Self Generated Instructions". In: *arXiv preprint arXiv:2212.10560* (2022).

[38] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. "Finetuned Language Models are Zero-Shot Learners". In: *International Conference on Learning Representations*. 2022.

[39] Benfeng Xu, Licheng Zhang, Zhendong Mao, Quan Wang, Hongtao Xie, and Yongdong Zhang. "Curriculum learning for natural language understanding". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 2020, pp. 6095–6104.

[40] Xueliang Zhao, Wei Wu, Can Xu, Chongyang Tao, Dongyan Zhao, and Rui Yan. "Knowledge-Grounded Dialogue Generation with Pre-trained Language Models". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2020, pp. 3377–3390.

[41] Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. "Fine-tuning language models from human preferences". In: *arXiv preprint arXiv:1909.08593* (2019).

A Few-shot prompting of 40B large language model

In this appendix, the few-shot examples used to prompt the large language model during generation of AI distractors and AI continuation pairs are shown. In Section A.1, all few-shot examples inserted into the template for modification of a continuation into an AI distractor, visualized in Figure 4.4, are written down. Similarly, all few-shot examples inserted into the template for generating a continuation from a prompt, visualized in Figure 4.6, are given in Section A.2.

Similarly to how the few-shot examples are separated by enter + '===' + enter in the following two sections, they are also separated that way when inserted into the respective template.

The few-shot examples in Section A.1 are written by the author of this thesis, whereas the few-shot examples in Section A.2 are simply samples taken from a part of the summarization dataset by Stiennon et al. [27] that was not included either in training, validation or test data in this thesis. The preferred continuation of these samples is what is used as the example summary.

A.1 Modification with large language model

Text: The machine learning intern was skillful, and had several offers from different companies. He himself attributes his success to his great upbringing in Switzerland.
Changed text: The machine learning scientist was very skillful, and had several offers from different universities and research institutions. She herself does not attribute her success to his great upbringing in Sweden.
===
Text: Joe, Peter and I sailed thousands and thousands of miles in our new boat on the nile.

Who knew sailing could be this fun?
Changed text: Joe and I drove thousands of miles in his new truck on highway one.

Who knew driving a car could be this fun?
===

Text: It was a long day in Stockholm.
Changed text: It was a short day in Paris.
===
Text: 14 days without water. Will she survive?
Changed text: 14 minutes without any water at all. Will I survive?
===
Text: Spain is so warm this time of the year.

It is going to be great with a long visit to the sun.
Changed text: Spain is so cold this time of the winter.

It would have been great with a short visit to the sun.
===
Text: I wanna break free from my everyday life.
Changed text: I want to continue my everyday life as usual.
===
Text: He said he was crazy, whatever they said he would agree. I was happy to hear that.
Changed text: He said I was easy-going, whatever he said I would agree. I was extremely happy to hear that.
===

A.2 Inference with large language model

Post:
My girlfriend and I have been dating for 2 years. I have researched borderline personality disorder and I believe she exhibits the majority of the symptoms. I am nervous about bringing this up to her for a few reasons.

I'm not a doctor and I am far from certain about this. Also, she already has a negative self image and I don't want to make that even worse by telling her the person closest to her thinks she has a problem. She tends to over think things and I know if I told her this, it would constantly bother her for a long period of time.

I am considering not mentioning the borderline personality disorder part and just recommending she begin attending therapy again. She went to a few sessions a year ago but then stopped going.

Summary:
I'm worried my girlfriend has borderline personality disorder and want to know if I should bring it up to her and/or if I should just advise her to start therapy again.
===
Post:
About two years ago, I split with an ex who owed me money. The money is a separate matter, but her response to me starting legal proceedings in regard to it was to file police reports against me for harassment among other things.

I was directly contacted by the police department in regard to one report. In the context of the conversation with the officer, it was apparent that my ex had fabricated at least some things. I asked the officer for a copy of the report and I was told I couldn't have one.

It may not have been a good decision, but in light of the reports and bad experiences

with my ex in general, I decided to just drop everything and move on with my life.

My concern now, however, is that her reports may show on background checks, as I'm currently looking for a new job. Is there a way I can view these reports?

Summary:
Ex filed multiple police reports against me. Might show/be discovered during background checks, looking for advice on how to view these reports/possibly gain a new job.
===
Post:
So.. TIFU about fifteen minutes ago, when I was doing my 2000 word essay for school. I'm just sitting there, procrastinating away, about 400 words in. I have a glass of water in my hand, and I just think to myself, 'If I was to just.. give this glass of water.. the slightest tip.. I would have the perfect excuse for not doing this essay' (Keep in mind that this is very late at night and I'm incredibly sleep deprived). I have one hand supporting my face, and in one hand the glass of water. I start playing with the glass of water, tilting it so that it's almost tipping all the water out, then straightening it, then tipping it again (I'm so freakin bored). Guess what happens next? I accidentally tip half the fucking glass all over my keyboard. I PANIC SO FUCKING MUCH. I stand up frantically and hit my knee under the table (pretty hard), and I fall back onto my chair. I'm having a panic attack now. I grab the laptop, tip all the water out of it, shake it and wipe the keyboard on my bed. It's fucked. It's DEFINTELY fucked. I'm DEFINTELY FUCKED. But didn't I want this? I still have to do the damn assignment, but now I'll have an excuse for an extension. I don't think that a week's extension was worth a thousand dollars and all my work that I've done all semester. Then I realise.. whew.. that's all right.. my semester's work is fine.. it's all on Dropbox. It's all on Dropbox. It's all on Dropbox? IT'S ALL ON DROPBOX! ALL MY FUCKING WORK IS ON DROPBOX!

kill me right now

Summary:
tried to get out of an assignment, ended up completely flooding my laptop with water, got fucked really bad. But it's on Dropbox so I can still do it
===
Post:
Hey, this is just a very minor thing. I'm a 16 year old male, 6 ft, 155 lbs, white. I swim competitively, so I am in a lot of contact with water. I had some pain about a week ago, bought some swimmer's ear drops, and it cleared up. Pain came back in both ears yesterday, and I was taking ibuprofen so I could go through finals without distractions. I went to see the doc today and she seemed pretty calm about the whole thing. She said the ears weren't infected, and that the Eustachian tubes were probably blocked. I've taken 2 tablets of pseudoephedrine HCl as a decongestant, and the pain in one ear has lessened a good deal. The remaining pain still is apparent when I swallow and it does feel like there is pressure on the ear. After all of this backstory, I was just wondering is there are any remedies to help lessen this pain or address the problem. For example, would any from this list
work well or are there any to avoid for my state?

Summary:
Ear pain, on decongestant, looking for something to help lessen the pain besides that. Any help appreciated.
===

B Data quality ablation studies

B.1 Purpose

In this appendix a study on the performance of a reward model trained on 9.8k human samples, depending on what part of the base dataset of 14.7k human samples that has been removed, is carried out. This study is motivated by the unexpected results in Section 5.1, where a reward model trained on 9.8k human samples outperforms a reward model trained on the base dataset consisting of 14.7k human samples. This study serves to give indications to whether this could be due to the data excluded from the base dataset to form the 9.8k human samples data mix, in this appendix referred to as the *possibly distracting partition*.

B.2 Setup

For this experiment, the base dataset is split into three parts, where one of the parts corresponds to the *possibly distracting partition*. The *possibly distracting partition* is then combined with the other parts, one at a time, to form the two data mixes that will be used to run the two additional experiments in this appendix. These two data mixes, referred to as *shifted_1* and *shifted_2*, are used to train reward models with the same setup (including hyperparameters, number of seeds and so on) as the original data mix with 9.8k samples.

B.3 Results

In Table B.1, the results of the two additional experiments are put in relation to the performance on the base dataset and the original 9.8k samples.

Table B.1: Mean performance for reward models over 5 seeds

Data mix	Mean accuracy	Median accuracy
Human 9.8k (*original*)	0.7078	0.7070
Human 9.8k (*shifted_1*)	0.6992	0.6953
Human 9.8k (*shifted_2*)	0.7039	0.7031
Human 14.7k (*base dataset*)	0.7039	0.7070

B.4 Insight

As both data mixes with 9.8k samples including the *possibly distracting partition*, thus *shifted_1* and *shifted_2*, perform worse than the original data mix with 9.8k samples both with respect to mean and median accuracy, the ablation study could indicate a possibility that the *possibly distracting partition* is of worse quality than the rest of the data in the base dataset. Furthermore, the base dataset outperforms both 9.8k human data mixes with the *possibly distracting partition* included, which in difference with the unexpected results in Section 5.1 is in line with what one would expect.