RL FOR LLMS: FINAL PROJECT PROPOSALS

WS2023-24

Polina Tsvilodub
Department of Linguistics
University of Tübingen
polina.tsvilodub@uni-tuebingen.de

January 10, 2024

This document contains final group project formalia, structure guidelines for your own project ideas and specific project proposals which can be completed for 6 and 9 ECTS. You are more than welcome to propose your own ideas, or to sign up for one of the proposed projects. Please find formal project requirements below.

The outlined projects are roughly grouped into various topics based on contents covered in the class. You are welcome to add your own thoughts and think critically about these ideas, too. Following the deadlines, formal guidelines, especially the meeting and submission requirements are a strict requirement for passing the class (unless other arrangements were met with me).

The final project can take on two formats:

1. practical experimental work wherein you would test LLMs on various things, or analyse model representation or maybe even train a model (such projects will require at least some coding)
2. a very detailed and well-motivated report outlining a plan for a study. This format is well-suited for groups which don't want to code, if you want to take on a new idea (e.g., this can be the result of your own project idea) or if the ideas is technically infeasible to implement. The more open-ended projects from this document can be submitted in this format, too (see Section 4.4). This report should be in the style of a study preregistration (research plan specification); more information can be found here: `https://www.cos.io/initiatives/prereg`

1 Project formalia

- sign up for groups of 3-5 members and fill out the project sign up form by January 24th 11:59am here: `https://forms.gle/2hNFG758oK1wj9AE7`
- if you would like to pursue your own project, you will be required to submit your proposal (see below for guidelines) in the form. **One** form completion per group is sufficient.
- if you are having trouble forming a group, please contact me.
- please sign up for **one obligatory meeting / group with me** (sign up on Moodle available soon). More meetings can be scheduled as needed, of course.
- you are welcome to take on technically advanced own project ideas; if these require high-performance compute, please see Section 3.1 below. I will provide technical support as needed.
- project submission deadline: **March 31st 23:59**
- materials to submit: well-documented GitHub repo (if applicable) and an ACL paper style write up of your project (short for 6 ECTS, long for 9 ECTS). Please use the following paper template: `https://www.overleaf.com/latex/templates/acl-2023-proceedings-template/qjdgcrdwcnwp`
  - Please follow best practices for structuring and documenting your GitHub repository as suggested, e.g., here (only sensibly applicable aspects for your project should be taken into account): `https://djnavarro.net/slides-project-structure/#1`, `https://www.freecodecamp.org/news/how-to-write-a-good-readme-file/`, `https://vimeo.com/412835411`

- for brevity of expression in the following proposals, base pretrained LLMs will be called **base LMs**, RL fine-tuned LLMs will be called **RL-LMs**.

2 Structure of proposals for own projects

The following describes requirements for specification of your own project ideas. Specifically, if you have your own idea, please submit **1 document** when you sign up for your group and final project which follows the following structure and contains the following information. The submission deadline is January **24th 11:59am** via the Google form referenced above.

Project proposal checklist:

- short task definition
- example of task, if applicable
- motivation (e.g., against current literature; no need for long literature review, max 0.5 page)
- short definition of concrete hypotheses you plan to investigate, if applicable
- data description (data source, size, collection procedure if applicable); input and desired output specification
- evaluation: score and procedure specification
- rough step-by-step project timeline
- indication whether the project will be a report (see above) or an implementation
- desired number of ECTS (please consider the amount of work for 6 or 9 ECTS in the following proposals for a rough benchmark)

For inspiration on task proposals, see e.g., this or other tasks of the SemEval workshop: `https://sites.google.com/view/legaleval/home`, as well as the proposals below.

3 Technical resources

For your hands-on projects, you will need to use or even train language models. The default model of choice for the projects should be Llama-2 which is provided by Meta [Touvron et al., 2023] and shipped both as a base LLM and as an RLHF-tuned -chat version in three sizes (7B, 13B and 70B parameters). The following describes how Llama-2 can be loaded on Colab (or locally) **for inference**. In particular, 7B and 13B models (both base and chat) can be run on the Colab GPU with *quantized* weights. This means that the model weights are converted to a lower numerical precision format for inference which allows to significantly reduce the memory required for running the models without significant performance deterioration. The HuggingFace library provides utilities for loading their models in such a format. An example for loading Llama-2-7b is provided below. For loading and running the other models, only model name needs to be adjusted respectively.

Before accessing Llama-2 via HuggingFace, registration is required. Please complete the following steps also provided here: `https://huggingface.co/blog/llama2#using-transformers`

1. please register for Llama-2 use at Meta as described here: `https://huggingface.co/meta-llama`
2. create a HuggingFace account, if you don't have one yet. Please note that it needs to be run under the same email as the email with which you requested access to Llama-2. You can sign up here: `https://huggingface.co/welcome`
3. navigate to your user profile acnd find your HuggingFace access token under settings/tokens. You will need to create a token and copy it for the next step.
4. in the environment in which you will run Llama, execute the following code:

   ```
   pip install transformers # if you don't have the library yet
   huggingface-cli login
   ```

   This will prompt you to enter the token from the previous step. Please paste it and hit enter. (Colab might ask you whether you want to save the token as a git credential; just enter n and hit enter).
5. once your registration at Meta is approved, you are all set to access Llama. The following code provides a starting point for loading the model and an example for generation. Once the model is loaded, you can use it in regular fashion with the HuggingFace utils as is needed for your project.

```
#pip install transformers, accelerate, bitsandbytes, sentencepiece

from transformers import LlamaForCausalLM, LlamaTokenizer, AutoTokenizer
import torch

tokenizer = AutoTokenizer.from_pretrained("meta-llama/Llama-2-7b-hf")
model = LlamaForCausalLM.from_pretrained(
    "meta-llama/Llama-2-7b-hf",
    load_in_4bit=True,
    device_map="auto",
    bnb_4bit_use_double_quant=True,
    bnb_4bit_quant_type="nf4",
    bnb_4bit_compute_dtype=torch.float16,
)
```

However, other well-motivated model options can be used, as well.

3.1 HPC cluster access

In case your project requires more compute resources than is provided on Colab or that you have access to privately within the group, there is high-performance compute cluster BwUniCluster 2.0 provided by the state of Baden-Württemberg which can be used by members of the University of Tübingen. Information on signing up, available hardware and usage can be found here: `https://wiki.bwhpc.de/e/BwUniCluster2.0`. Please contact me if you need such compute resources. I will also provide technical assistance with the set-up as much as I can, if needed.

If your project involves training a model, we might consider using a more efficient fine-tuning method QLora: `https://huggingface.co/blog/4bit-transformers-bitsandbytes`. More technical details can be discussed with me by individual groups.

4 Project proposals

The projects indicate the number of credits that can be gained upon completing different numbers of subtasks in a given project, but, in principle, the exact task composition can be further discussed. 6 ECTS subtasks are listed under **work steps** in green; for 9 ECTS, the 6 ECTS tasks and, additionally, tasks listed in orange are required. Where no color is used, the steps hold for all types of projects.

The main goal of the class and the projects as the practical part is to address the broad question "How does RL fine-tuning affect LMs, and how can RL be applied to study language?". Taking a multi-faceted approach to this question, the available projects are grouped into the following domains:

- a 'behavioral perspective' (focusing on I/O; includes calibration analyses and elicitation methods based on log probabilities of answer options)
- a more 'mechanistic' perspective (including mechanisms of training, analysis of embeddings)
- additionally, some projects are designed to facilitate such research in the first place (e.g., dataset collection).

This grouping also attempts to help guide your own ideas.

4.1 Lingustic benchmark performance: Testing effects of RL [Behavioral evals]

- **Idea**: Intuitively, RL might affect the performance of resulting LMs on standard linguistic benchmarks because it shifts the model's performance towards more conversational, helpful/harmless/honest language. That is, linguistic knowledge which arises from standard pretraining might deteriorate upon RL fine-tuning (as measured on standard linguistics benchmarks) compared to base LMs.

  These inuitions are not contradicted by various exploratory work. In particular, my own exploratory work on simple reference games as well as linguistic tasks tasks suggests that log probabilities of different options on benchmarks or simple linguistic tasks decrease with RL fine-tuning, compared to base versions of the models, and 'unexpected' tokens can accumulate more probability mass. Exploratory calibration analyses of RL-tuned models have shown worse calibration [Kadavath et al., 2022, OpenAI, 2023], suggesting potential performance deterioration. Note that major papers on fine-tuned models usually test them on BIGBench or MMLU (e.g.,

OpenAI [2023], Bai et al. [2022a]), not necessarily on standard linguistic benchmarks as was common for base LMs. Therefore, analysing performance on standard linguistics benchmarks might also fill this gap. Next to probability based analyses, "behavioral" results in the sense of analysing free generation data of different models migth reveal intriguing results about performance effects of RL. For instance, they might reveal where the probability mass is shifted to upon RL (since the same answer options might receive lower probability in absolute terms compared to the base models; cf. `http://tinyurl.com/4kzek8s3`).

- **Hypotheses**: The following testable research questions are derived from the intuitions above and could be tested (and possibly augmented by own ideas) in this project.
  - syntactic performance: RL-LMs might perform worse on syntactic tasks like predicting grammaticality of sentences, choosing grammatical continuations of sentences (e.g., subject-verb agreement, filler-gap dependencies etc).
    - Beyond standard benchmarks: it could be that performance changes especially on constructions which are ungrammatical yet conventionalized or (more) acceptable and, therefore, endorsed by annotators (cf. Leivada and Westergaard [2020]). This is just a suggestion, not necessarily required for the project completion.
  - semantic performance (very coarse-grained notion of semantics, partially operationalized as NLU): RL-LMs might perform worse on semantic tasks like NLI, semantic similarity identification, coreference resolution, QA, ... However for more "conversational" benchmarks, RL-LMs might beat base LMs.
  - pragmatic performance: under the assumption that 'natural' conversational language that would be endorsed by annotators is rich in pragmatic phenomena like presuppositions and conversational implicatures, RL-LMs might perform better on benchmarks testing such phenomena than base LMs.
- **Work steps**:
  - use Llama 7b vs Llama 7b chat and Llama 13b vs Llama 13b chat
  - generate zero-shot results for the following datasets:
    - SyntaxGym [syntax]
    - NLI, QA, coreference, COLA: SuperGLUE [syntax/semantics]
    - conversational QA: CoQA [semantics]
    - semantic similarity: STS dataset [semantics]
    - presuppositions and implicatures: ImpPres benchmark [pragmatics]
    - implicatures and conversational reasoning: Grice dataset [pragmatics]

    You will easily find them and the respective papers on the internet, they are all freely accessible. Some datasets need specification of prompting / coming up with how exactly to elicit results; this is intended as part of the project.
  - compare against different prompting (few-shot, possibly varying the composition of few-shot prompt (e.g., number of examples from different benchmark categories))
  - methods:
    - log probability based based computation of task accuracy for both models, statistical comparison of results (see Homework 2, Exercise 3 for the method)
    - let the models freely generate responses and inspect them manually, and, ideally try to map them onto benchmark responses. Analyse interesting patterns and observations, if there are any.
  - write up results in a short/long ACL paper and submit together with well-documented repository.

4.2 RM component annotation [Dataset collection]

- **Idea**: Current approaches to fine-tuning with RL heavily rely on a reward model which captures human preferences for generically helpful and honest responses (harmlessness is not the focus of this project). That is, the notion of helpfulness is treated in a blackbox manner (although a few models' fine-tuning included usage of rules in various ways; cf. Bai et al. [2022b], Glaese et al. [2022]). However, what constitutes helpfulness and honesty might look very different depending on the task and the context (for example, a helpful answer to a question like "how do I start running?" might be much longer and more subjective than the answer to a question like "how do I turn on an iPhone?"). In order to train and test models which could take into account these fine-grained differences, a break-down of aspects which might constitute helpfulness (and possibly honesty) as well as datasets where single examples are annotated with respect to relevance of these aspects are required.

  The goal of this project is to annotate a multi-task dataset with respect to whether each of the following components is relevant for the input task at hand. Additionally, a base or supervised fine-tuned LM can be used

to sample candidate responses for tasks in the dataset, which would be ranked and annotated with whether the preferred answer indeed is better than the rejected option according to each helpfulness / honesty aspect.The following aspects into which helpfulness and honesty can be broken down are proposed:

  - length: Is the length of the answer appropriate? [help.]
  - appropriateness of the structure of the answer: Is the answer well structured? [help.]
  - informativity: Is the amount of information in the answer appropriate for the task at hand (not too much, not too little)? [help.]
  - manner: Is the answer respectful and friendly? [help. / harm.]
  - coherence: Is the answer coherent (on-topic, avoiding irrelevant off-topic information)? [help.]
  - support by evidence: Is important information in the answer backed up by evidence or arguments? [hon.]
  - correctness: is the answer factually correct? [hon.]
  - (un)certainty expression: is the answer stated as a verified fact which can be relied upon? [hon.]

  Additional aspects and ideas are welcome.

- **Work steps**:
  - Annotate **0.25** of the `databricks-dolly-15k` dataset [Conover et al., 2023] as described above. The dataset provides human responses for samples on the following tasks: brainstorming, classification, closed QA, generation, information extraction, open QA, and summarization. It consists of 15k inputs (task instruction, sometimes a context for the task) and human-written outputs.
    Example datapoint and annotation to be performed in this project:
    * *Input (General QA):* How do I start running?
    * *Output:* Make sure you get comfortable running shoes and attire. Start with achievable goal in mind like a 5K race. If you never ran before, start gradually from a walk, to brisk walk, light jog aiming for 15-30mins initially. Slowly increase your running time and distance as your fitness level improves. One of the most important things is cool down and gentle stretching. Always listen to your body, and take rest days when needed to prevent injury.
    * *Annotation:* relevant components: informativity, manner, coherence, support by evidence, uncertainty expression
    * *Annotation:* irrelevant components: length, appropriateness of structure, correctness
  - create the 0.25 split of the dataset evenly by task.
  - during annotation, if possible, formulate regularities about your annotation strategy (e.g., when did you consider an answer better according to aspect X?)
  - ideally, have at least two group members annotate a few datapoints and discuss your agreement.
  - based on your annotations, run some descriptive statistics to determine whether there are difference by task type with regard to aspects you deem important.
  - take, e.g., a Llama-2, Mistral or a Pythia model, sample a few outputs for the 0.25 subset you generated, and rank them according to each helpfulnes/honesty components you deem relevant. Have someone else in your group do a generic helpfulnes/honesty preference ranking of the sample answers.
  - do some descriptive statistics on the annotations to find out whether the generic preferences coincide with component-wise preferences. This might elucidate whether our intuitive generic preferences which are commonly gathered in RLHF datasets coincide with what we would consider as helpful and honest according to theoretical considerations.
  - write up results in a short/long ACL paper and submit together with well-documented repository and the dataset.

4.3 Reward collapse: Empirical investigation [Behavioral/mechanistic evals]

- **Idea**: In class, we discussed the paper by Song et al. [2023] which argues that currently standard techniques of training reward models (RM) based on response ranking data generate a continuous (unimodal) distribution of rewards for outputs of different task types. However, intuitively, we would expect different distributions of rewards. Specifically, Song et al. [2023] argue that for closed-ended tasks like factual QA, we would expect a bimodal reward distribution assigning low rewards to incorrect answers and high rewards to correct answers. For open-ended tasks, the reward distribution would be rather continuous. This phenomenon is described as reward collapse and attributed to the currently standard utility function applied during RM training. However, during the lecture some aspects of these paper were criticized, and it was mentioned that it is unclear to which extent this is an issue in real-world tasks, as the paper proposed solutions to the reward collapse problem based on a synthetic task.

The idea of this project would be to verify how the rewards are distributed empirically on a closed- vs. open-ended task (under trained reward models).

- **Hypotheses**: if the predictions of the paper are borne out empirically, we will observe a unimodal / normal distribution of rewards assigned by RMs trained with a standard utility function for both types of tasks.
- **Work steps**:
  - use the RM `OpenAssistant/reward-model-deberta-v3-large-v2`: `https://huggingface.co/OpenAssistant/reward-model-deberta-v3-large-v2`. The model was trained on the following datasets: webgpt_comparisons, summarization from human feedback, synthetic instructu-GPT-J pairwise, Anthropic hh RLHF.
  - gather reward model predictions on a closed ended task dataset—BoolQ (SuperGLUE)—from which both correct and incorrect responses can be derived. Specifically, look at rewards for correct and incorrect labels, given the question and the context, while varying how the response is represented (e.g., 0 / 1, Yes / No, True / False).
  - gather reward model predictions on an open-ended task dataset which already contains preferred and rejected answers: since the Anthropic hh dataset was used for training, use e.g. this dataset (but **try to check source again, so that it is not in the RM training data**): `AlekseyKorshuk/hh-rlhf-pairwise`
  - run descriptive stats on the reward and check if the reward distributions are polarized or more continuous. Check if the rewards match "ground truth" results (i.e., are higher on true / preferred answers than on incorrect/rejected ones.)
  - in order to estimate the reward distribution on which base models are fine-tuned with RL, use Llama-2 7B base to generate sample answers on BoolQ and the hh-rlhf-pairwise datasets, use the RM to generate rewards for these samples and also do some manual annotation to see if the RM predictions align with your inuitions.
  - run descriptive stats on the reward and check if the reward distributions are polarized or more continuous. Check if the distribution aligns with your annotations.
  - possible extensions (not needed for passing project): test more reward models; actually train a reward model with a proposed utility function on Anthropic hh dataset and test with the procedure described above
  - write up results in a short/long ACL paper and submit together with well-documented repository (and the generated dataset).

4.4 Calibration & expression of uncertainty [Behavioral evals]

- **Idea:** Calibration of LMs and whether they can (correctly) indicate uncertainty about their generations is an important question touching upon the reliability and correctness of LM generations. In class, work on calibration by Kadavath et al. [2022] was discussed at length which showed that big base LMs (by Anthropic) are well-calibrated in the sense that probabilities assigned to answer options on BIGBench and MMLU (and other benchmark datasets) correlate well with the correctness probability of these options (across trials). For RL-LMs, several observations have been made suggesting that calibration deteriorates with RL finetuning [OpenAI, 2023, Kadavath et al., 2022], but little consistent comparisons have been performed. Another line of work has focused on actually including indications of uncertainty in LM generations [Mielke et al., 2022, Lin et al., 2022], but all of this work focuses on factual correctness and calibration on abstractive tasks, not on context-dependent expressions, and little comparison to human behavior regarding knowledge and certainty expression has been done. Again, RL-LMs have not been under close inspection in this regard. Finally, an issue related to the correctness and confidence of statements generated by LLMs is so-called hallucination. One intuitive solution to avoid hallucinations is to fine-tune LMs with RL where responses like "i don't know" also receive high rewards, but it has been discussed that it would be rather difficult to not make the models evasive with this procedure. However, there have been few datasets or approaches how to include "i don't know" in the first place.

  The ideas within this project are all calibration / uncertainty and constitute a bit of a brainstorm. Focus can be chosen depending on interest. The goals of the subprojects are to:

  1. provide more comprehensive understanding of calibration of RL-LMs by replicating the first part of the analysis by Kadavath et al. [2022] on a few of the same datasets (e.g., MMLU, TriviaQA) on Llama-2 base vs. chat.
  2. investigate LMs' zero- or few-shot performance on naturalistic uncertainty expressions in-context. Specifically, in this project the model will be tested on contexts discribing situations where the exact state of the

world is uncertain. Given these contexts, we will test whether expressions "certainly", "possibly" and "probably" (not) are used in a human-like way depending on the context (e.g., Llama 7b base or chat can be tested). The project will replicate the experiment by Herbstritt and Franke [2016]. This will extend beyond previous work (e.g., Mielke et al. [2022]) by focusing on contextually induced uncertainty (rather than factual QA) and by comparing the results to human data.

3. try to construct a dataset of inputs and outputs which both contain informative responses and expressions of "i don't know". In particular, an extant dataset could be combined with manually constructed examples, e.g., with questions about things after the training data cutoff for a particular model, or where an evasive response would be more polite. This project idea is completely open-ended to your ideas and creativity.

- **Hypotheses:** the testable questions for two of the subpprojects would be:
  1. If previous observations are further borne out, we will observe worse calibration on RL-LMs on benchmark datasets compared to their base LM counterparts. However, strategies like increased sampling temperature could improve calibration (cf . Kadavath et al. [2022], but note that this was done for evaluating the correctness of particular answer option, given a question).
  2. if the tested model is well-aligned with human behavior, the probabilities assigned to the different uncertainty expressions in the different contexts will match human data from the reported experiment.
- **Work steps:**
  Calibration:
  - replicate the calibration experiment by Kadavath et al. [2022] on Llama-2 7b base and chat. Select a few of the same benchmarks, get the probabilities of the different response options, bin them following the reported procedure, and compute the frequencies of correct responses in each bin.
  - compare the calibration plots of the base and the chat model
  - run the same analyses for 13b and 70b models. Instead of the 70b model (which will require more intricate compute set up), one or two other analyses from the paper can be replicated

  Uncertainty expressions (6 ECTS project only):
  - read the paper by Herbstritt and Franke [2016], construct the reported contexts in a text-based manner, get the probabilities of the different expressions in each context under chosen Llama model. Compare results by context type between a base and an RL model, compare to reported human results (ask me for human data).

  Dataset construction:
  - this project is open to your ideas. One way to start could be to define criteria when it might be appropriate to say "i don't know".

  For all projects, write up results in a short/long ACL paper and submit together with well-documented repository (and the generated datasets, if applicable).

4.5 Investigating collapse onto RL fine-tuning data distribution [More 'mechanistic' evals]

- **Idea:** When LMs are fine-tuned, the fine-tuning datasets are usually constructed so as to reflect the target task on which the LM will be deployed, and, ideally, so as to reflect the distribution of the anticipated test data. However, when it comes to RL fine-tuning data, little attention has been paid to the question of how exactly the composition of the fine-tuning data influences the generalization / behavior at inference time. There have been a few observations of the model's tendencies to collapse onto certain output patterns across different inputs (cf. `http://tinyurl.com/4kzek8s3`), but due to the closed-source nature of many fine-tuning datasets, the relation to the training data is underexplored.

  The idea of this project is to investigate how statistics of the fine-tuning dataset (i.e., the dataset on which RL fine-tuning is run) affect the performance of the fine-tuned model. A model trained for summarization with RLHF (via PPO) will be used to this end. The project will focus on the RL fine-tuning data and does not take into account the stats / effects of the reward modeling dataset. The investigated statistics are approached intuitively and are intended to cover syntactic, lexical, formal and content related aspects of the dataset. Specifically, the fine-tuning dataset should be analysed with respect to:
  - sentence length distribution
  - syntactic structure (approximated, e.g., by common sequences of POS tags)
  - frequencies of different POS
  - frequencies of the lemmata

  - sentiment distribution of the texts
  - distribution of topics in the texts
- **Hypotheses**: We might expect that the model overfits to properties that were dominant in the fine-tuning dataset. This means, test performance will be better when the inputs exhibit similar properties as the training dataset, and when tested out-of-distribution, performance might be worse. It would be interesting to test if the intuitive quality of responses matches the quality predicted by the reward model (i.e., check for reward hacking). If analysis of the predicted outputs on the training dataset will be conduceted, we could expect that the same properties may appear on the test set outputs as for the training set outputs.
- **Work steps**:
  - use the RL-LM summarization model `CarperAI/openai_summarize_tldr_ppo`, was trained following Stiennon et al. [2022] by Castricato et al. [2023].
  - use the fine-tuning train and test splits of the dataset `CarperAI/openai_summarize_tldr` which contains Reddit posts which have to be summarized.
  - get information on the datasets with respect to the points above:
    * sentence length, syntactic structure (POS can be retrieved with e.g. spacy), lemma distribution (lemmatization can also be done with spacy), sentiment analysis (some pretrained model from HuggingFace could be used for that)
    * topic analysis (can be done e.g., with the package BERTtopic)
  - run descriptive stats on the information retrieved above (e.g., plot distributions, check if they match between train and test datasets).
  - generate predictions for the portion of the test split where the main statistics match the training statistics from the model (50 examples suffiecient). Annotate your intuitive quality judgement of the outputs. Get reward model predictions for the outputs.
  - generate predictions for the portion of the test split where the main statistics MISmatch the training statistics from the model (50 examples suffiecient). Annotate your intuitive quality judgement of the outputs. Get reward model predictions for the outputs.
  - compare the rewards and your own judgements. Does the model perform worse on the portion of the test split which is OOD relative to the train split?
  - generate model outputs on the train dataset split and do the same analyses as on the inputs. Conduct the same analyses with the test split outputs and compare whether the statistics (by-aspect) match. Does the trained model learn to generate outputs with certain properties (on the test set) as it did during finetuning on the training set?
  - take the model predictions for the portion of the test split where the main statistics MISmatch the training statistics (see above). Analyse the properties of the predictions. Check if these differ from the in-distribution test output properties. Does the model predict outputs with the same "property footprint" irrespectively of input properties? Such an out-of-distribution test item could comprise, for instance, a text to be summarized which is on a topic not (commonly) to be found in the train split.
  - write up results in a short/long ACL paper and submit together with well-documented repository (and the generated dataset).

4.6 Mechanistic interpretability [Mechanistic evals]

- **Idea**: Mechanistic interpretability is a developing area in NLP which strives to understand functional roles of different mechanisms neural LMs makes use of for solving different tasks. However, most of the research in this domain so far considered only base LMs like BERT and GPT-2 [e.g., Merullo et al., 2023, McCoy et al., 2019, Yu et al., 2023, Meng et al., 2022]. It is unclear to whether these results hold if the model is further fine-tuned (e.g., with RL). As discussed in the lecture, certain components in the late layers of GPT-2 (the FFN) can be seen as representing abstract relations which e.g., help retrieving the capital of a city in the preceding context [Merullo et al., 2023]. RL-tuning is frequently applied specifically to the late layers of a base LM; intuitively, in order to adapt to the reward signal and improve performance, representations in the late layers in an RL-LM might change compared to the base LM (unless the model has other 'capacity' to adapt to the RL signal). Furthermore, RL might target the model's instruction-following and context sensitivity, thereby adding to its extractive performance.

  The goal of this project is to look at mechanistic aspects of an RL-LM compared to a base LM by replicating the work by Merullo et al. [2023]. Specifically, the task is to explore whether the FFN updates still carry relational information for abstractive tasks like the capital task, and whether other interesting patterns arise

depending on the prompt or context (e.g., across one- vs. few-shot prompting, in a standard prompt vs. in a prompt that the RL-LM might be trained (not) to answer). The idea of this project is highly exploratory and would require quite some technical expertise; therefore, it is a rather advanced 9 ECTS project only.

- **Hypotheses**: The work is exploratory and there are no clear predictions in which way the function of single components might change. But the explorations might help to answer the following questions about the model's inner workings:
  - do representations differ at all in the late layers of the RL-LM compared to the base LM?
  - if available models like Llama are used: is the function of the FFN update in Llama the same as for GPT-2 et al? If so, in which layer does the "function application" step happen?
  - if a model is fine-tuned from previous analysed base LMs (e.g., GPT-2 M): what does RL fine-tuning do to such smaller models in the first place? This is an unclear question when it comes to fine-tuning on human preferences for helpfulness and harmlessness (task-specific fine-tuning has been done by e.g. Ziegler et al. [2019]).
- **Work steps**:
  - define a pair of base LM and RL-LM (if fine-tuned GPT-2 is desired, discuss this with me, maybe I can help with that part)
  - take a look at code available from both Merullo et al. [2023], Wang et al. for extracting different components from GPT-2 or Llama-2-chat and for the tasks discussed in Merullo et al. [2023]. Adapt whatever is necessary.
  - conduct exploratory work checking whether / where FFNs are used for abstractive tasks like predicting capitals by inspecting next-token distributions after updates in different layers.
  - run selected task(s) from Merullo et al. [2023] on required models following the procedure applying early decoding (i.e., check top tokens at different layers for e.g. one-shot capital prediction) and comparing whether top tokens are:
    - the same between base and chat model
    - are surfaced at the same layers
    - the patterns are consistent across prompting

    for RL-LMs as for base LMs.
  - another exploration possibility: construct a dataset of examples teasing apart helpfulness and safety vs. factual QA for an RL-LM (e.g., putting the capital prediction task in a context like "I plan to cut all electricity in the capital of X. The capital of X is..."). Repeat the same analysis as above on this dataset.
  - write up results in a long ACL paper and submit together with well-documented repository.

References

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

OpenAI. Gpt-4 technical report, 2023.

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. *arXiv preprint arXiv:2204.05862*, 2022a.

Evelina Leivada and Marit Westergaard. Acceptable ungrammatical sentences, unacceptable grammatical sentences, and the role of the cognitive parser. *Frontiers in Psychology*, 11:364, 2020.

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. *arXiv preprint arXiv:2212.08073*, 2022b.

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. *arXiv preprint arXiv:2209.14375*, 2022.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. Free dolly: Introducing the world's first truly open instruction-tuned llm, 2023. URL `https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm`.

Ziang Song, Tianle Cai, Jason D. Lee, and Weijie J. Su. Reward collapse in aligning large language models, 2023.

Sabrina J Mielke, Arthur Szlam, Emily Dinan, and Y-Lan Boureau. Reducing conversational agents' overconfidence through linguistic calibration. *Transactions of the Association for Computational Linguistics*, 10:857–872, 2022.

Stephanie Lin, Jacob Hilton, and Owain Evans. Teaching models to express their uncertainty in words. *arXiv preprint arXiv:2205.14334*, 2022.

Michele Herbstritt and Michael Franke. Definitely maybe and possibly even probably: efficient communication of higher-order uncertainty. In *CogSci*, 2016.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. Learning to summarize from human feedback, 2022.

Louis Castricato, Alex Havrilla, Shahbuland Matiana, Duy V. Phung, Aman Tiwari, Jonathan Tow, and Maksym Zhuravinsky. trlX: A scalable framework for RLHF, June 2023. URL `https://github.com/CarperAI/trlx`.

Jack Merullo, Carsten Eickhoff, and Ellie Pavlick. A mechanism for solving relational tasks in transformer language models, 2023.

Tom McCoy, Ellie Pavlick, and Tal Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy, July 2019. Association for Computational Linguistics. doi:10.18653/v1/P19-1334. URL `https://aclanthology.org/P19-1334`.

Qinan Yu, Jack Merullo, and Ellie Pavlick. Characterizing mechanisms for factual recall in language models. *arXiv preprint arXiv:2310.15910*, 2023.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372, 2022.

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. *arXiv preprint arXiv:1909.08593*, 2019.

Tony T. Wang, Miles Wang, Kaivalya Hariharan, and Nir Shavit. Forbidden facts: An investigation of competing objectives in llama-2. In *ATTRIB and SoLaR Workshops*.