LLMAJ evaluation

Owner	Ⓐ Andrii Cherniak
Tags	

💡 Computational budget: 600 ∗ 6 + 600 ∗ 2 = 4800 calls to Azure OpenAI.

The main purpose of this evaluation is to confirm using LLMAJ that DPO fine-tuning took place and the generated answers are aligned with the preferences.

I observe that base LLM does a good job, and the alignment effect is very small, if any. Two dpo-aligned LLMs have been evaluated: one with smaller and one is with larger rewards/margins obtained on the training data set. It is very difficult to confirm that larger reward/margin during training indeed results in better performing llm.

Without validation error plots it is impossible to diagnose if training was terminated early before proper convergence. At this moment I would suspect a potential error somewhere in the pipeline.

The most practical tool to assess answer quality for now is LLMAJ. We would like to confirm that LLMAJ can detect improvements between answers generated with tuned vs base llm. While we care about pregnancy-related domain, `Intel/orca_dpo_pairs` does not explicitly include any training pairs from that domain. Thus if we only judge answer generation on pregnancy- related QAs, `inability to detect any improvements` between tuned and base llm can be attributed to a variety of factors, and sadly is `not very helpful` with DPO evaluation. At the same time, if we do `detect improvements` on one of LLMAJ dimensions on pregnancy-related domain, `it could be just by chance`, and we may be claiming undeserved victory.

However , we can select say a large `<question, chosen, rejected>` records from the `Intel/orca_dpo_pairs`, apply LLMAJ to score

1. `<question, chosen>` and `<question, rejected>`
2. `<question, generated answer(tuned llm)>`, `<question, generated answer(base llm)>`

then compute LLMAJ feature differences:

1. feature_i(chosen) - feature_i(rejected) == diff_cr
2. feature_i(answer, tuned llm) - feature_i(answer, base llm) == diff_tb

and show that `avg(diff_cr)` and `avg(diff_tb)` have the same sign.

In addition, I want to understand,

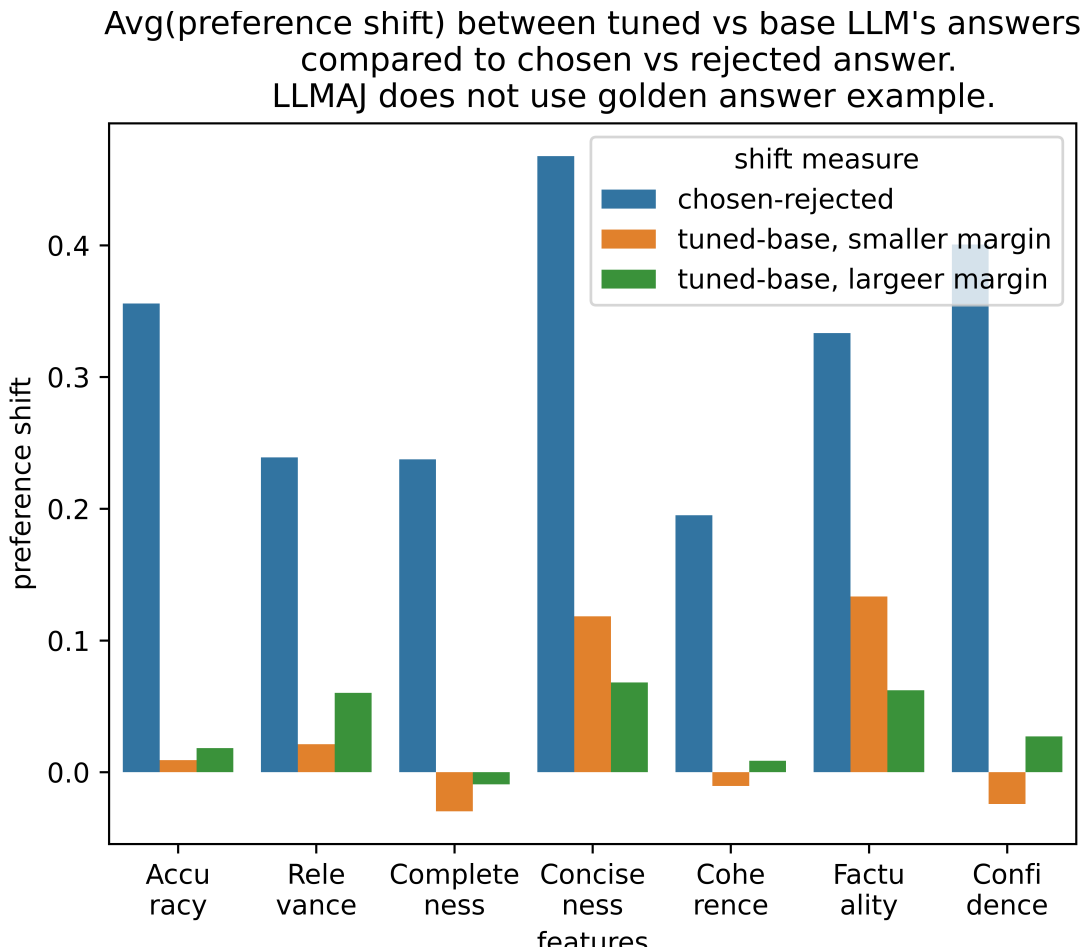
1. how much LLMAJ scores will shift if I use `chosen` answer as ground truth or reference.
2. does `rewards/margins` have any effect on the LLM's output
3. how does performance depend on the % of the training data DPO is being trained on

LLMAJ: raw scores conversion into preference

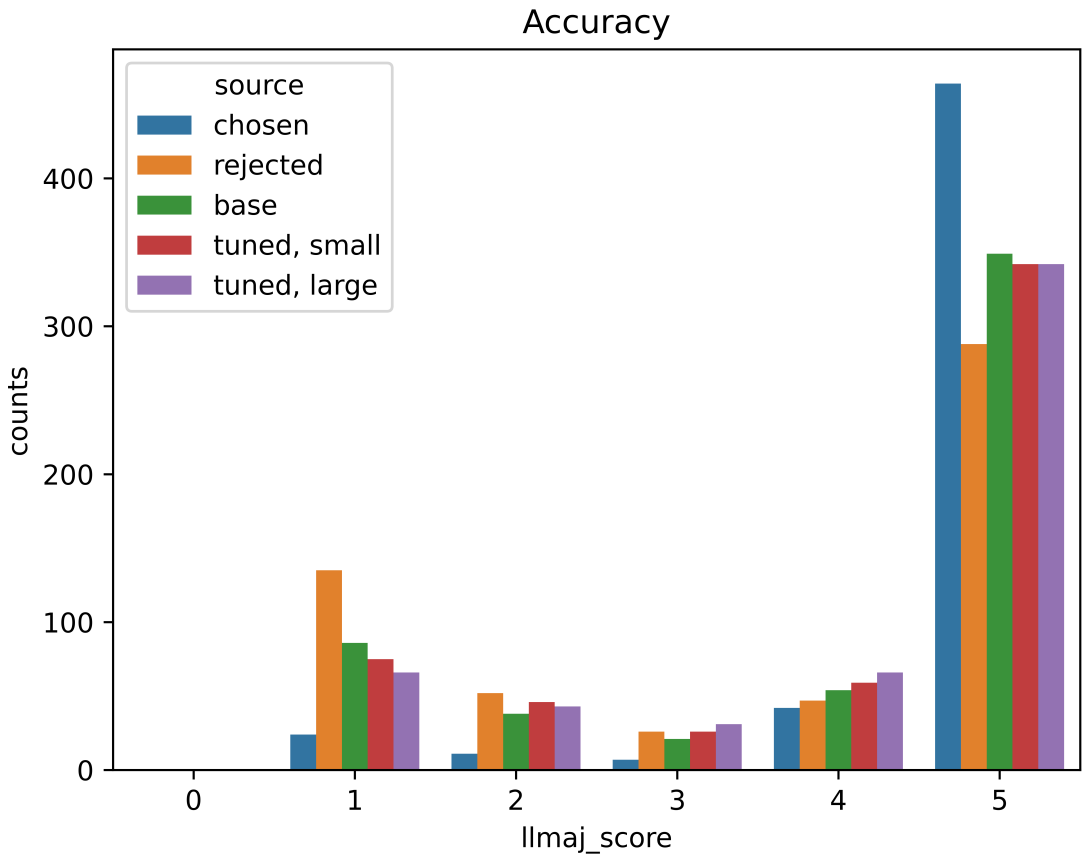
One important scoring detail that went unnoticed is that LLMAJ outputs scores [1..5] and 0 if the scoring is not applicable. `I filter out all results with score == 0`. While we care about raw LLMAJ scores, for DPO we care about comparison between raw scores produced by base and tuned LLM, aka preference between the answers. Preference takes values [-1, 0, 1], where

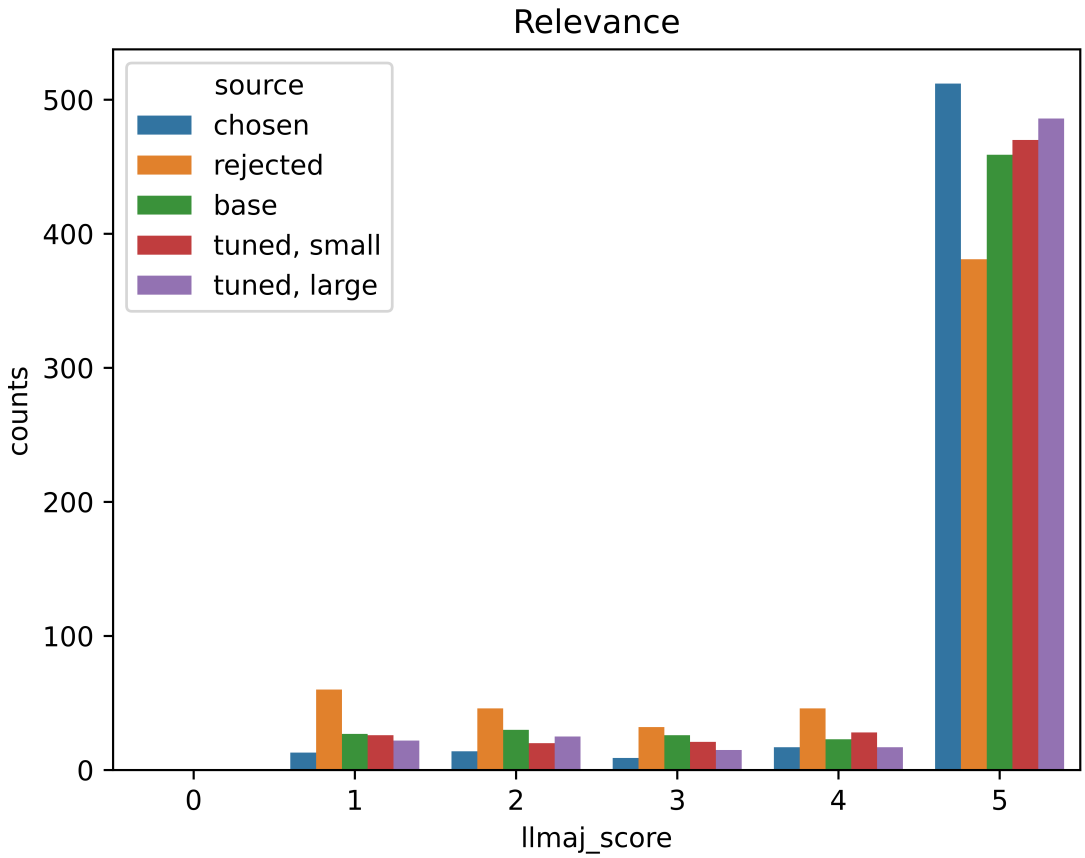
- preference == -1 when score 1 < score 2
- preference == 0 when score 1 == score 2
- preference == 1 when score 1 > score 2

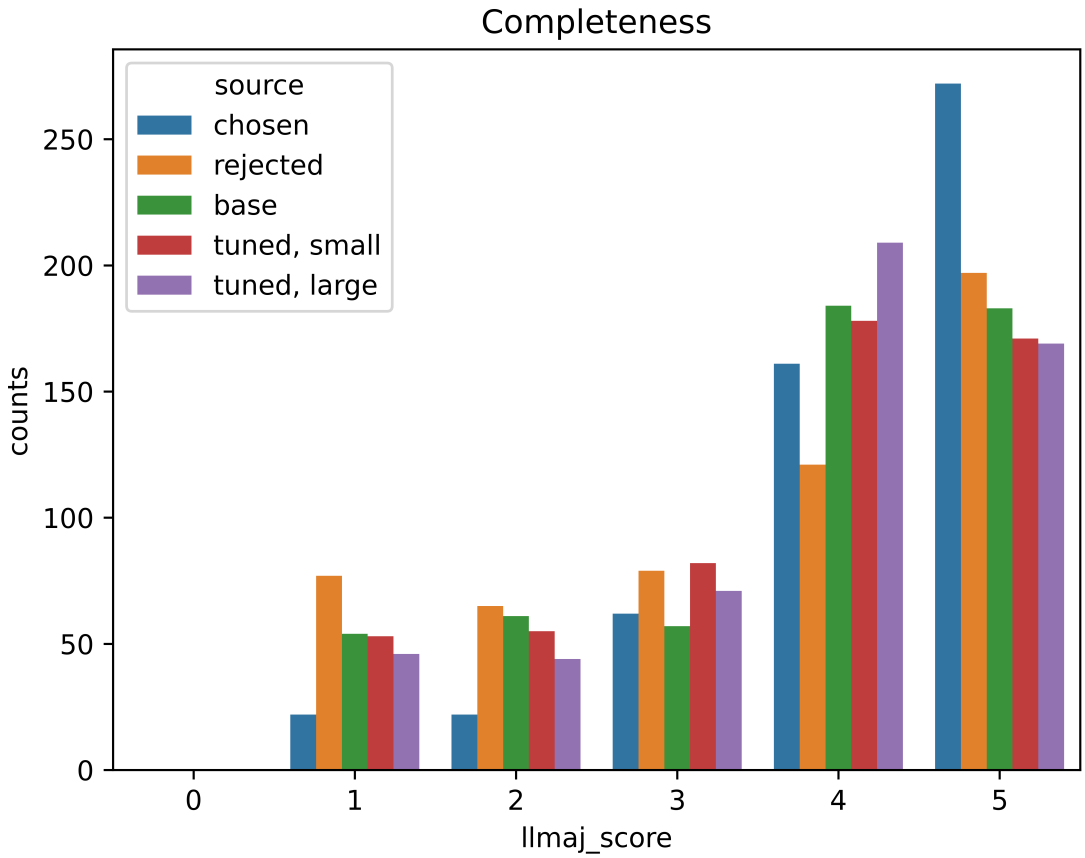
LLMAJ without using golden answer: 600 validation points

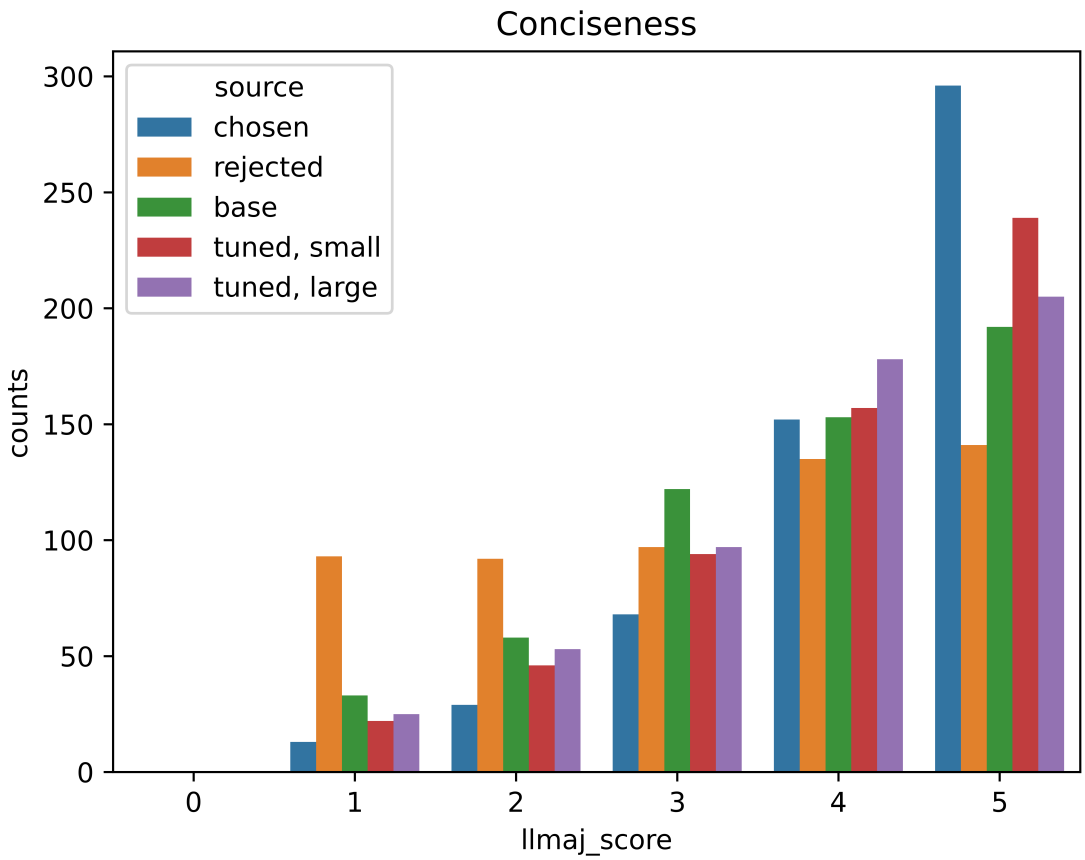


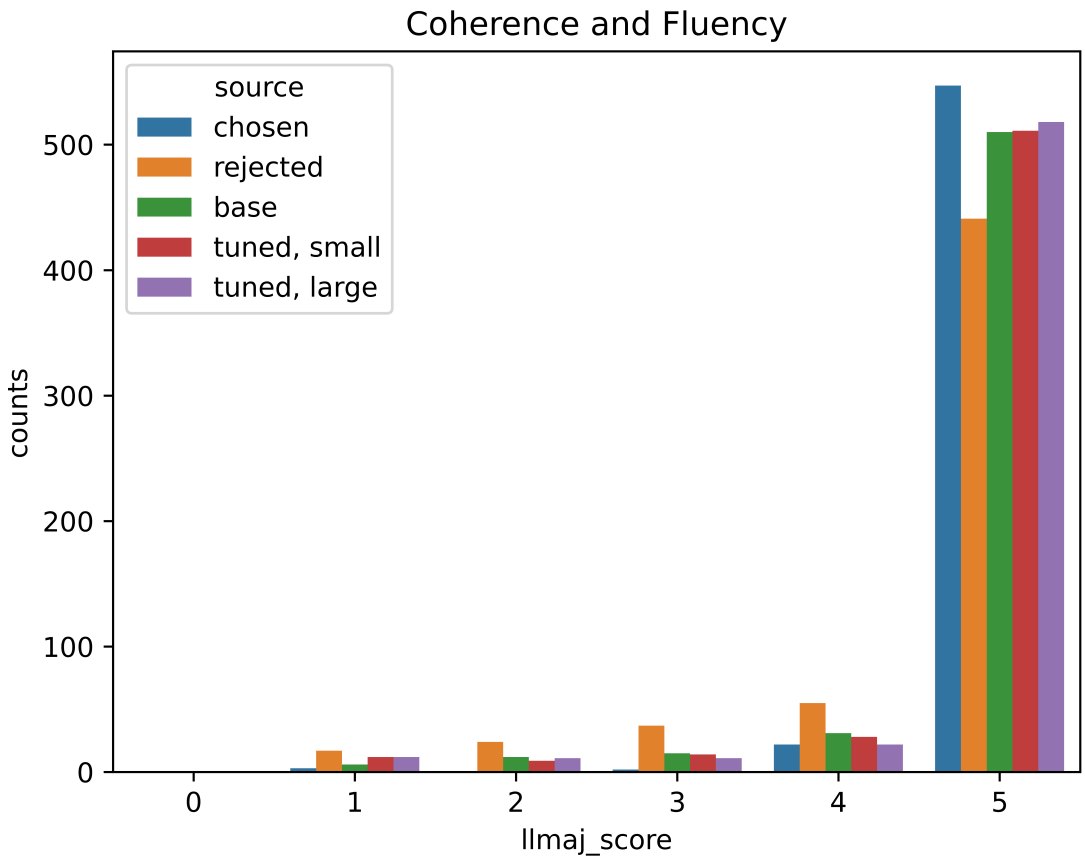
In order to understand why `we observe such a small shift between tuned and base LLM`, I am looking into raw score distribution . It appears that base LLM is doing a reasonable job to begin with, and that is why the expected lift is much smaller compared to `chosen vs rejected` .

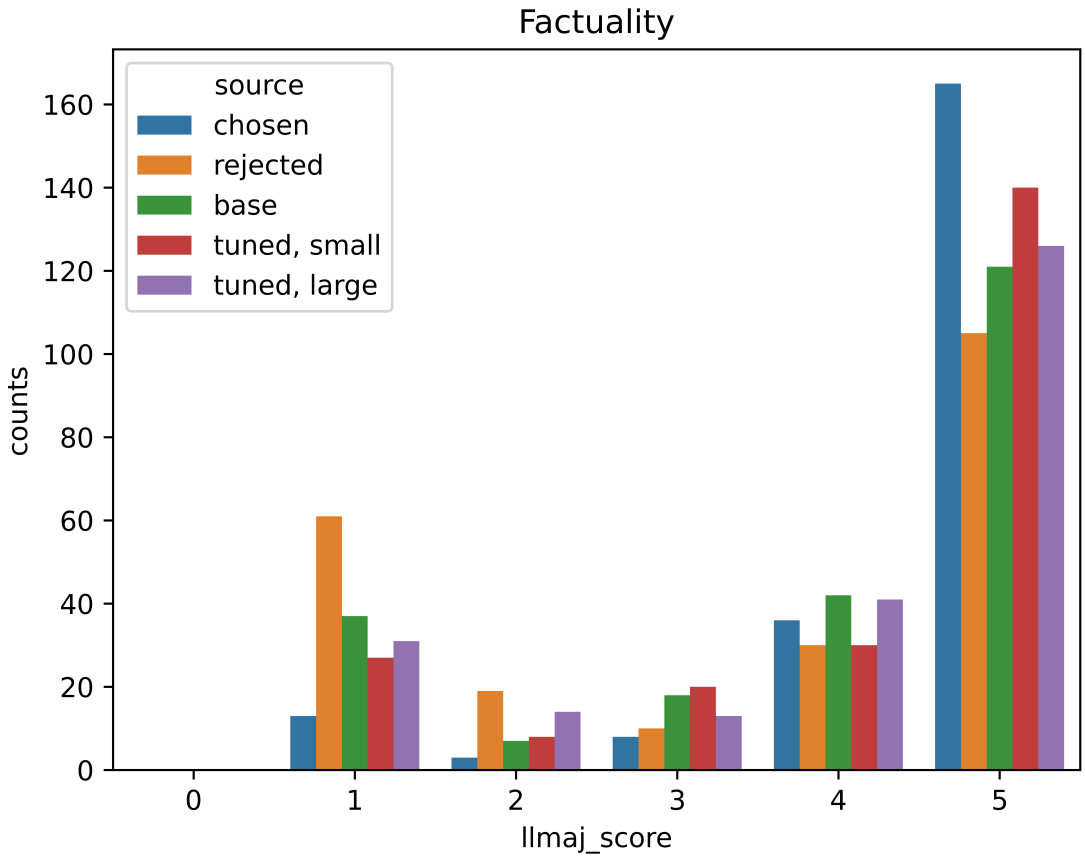

Accuracy
source
chosen
rejected
base
tuned, small
tuned, large
counts
0
100
200
300
400
0
1
2
3
4
5
llmaj_score

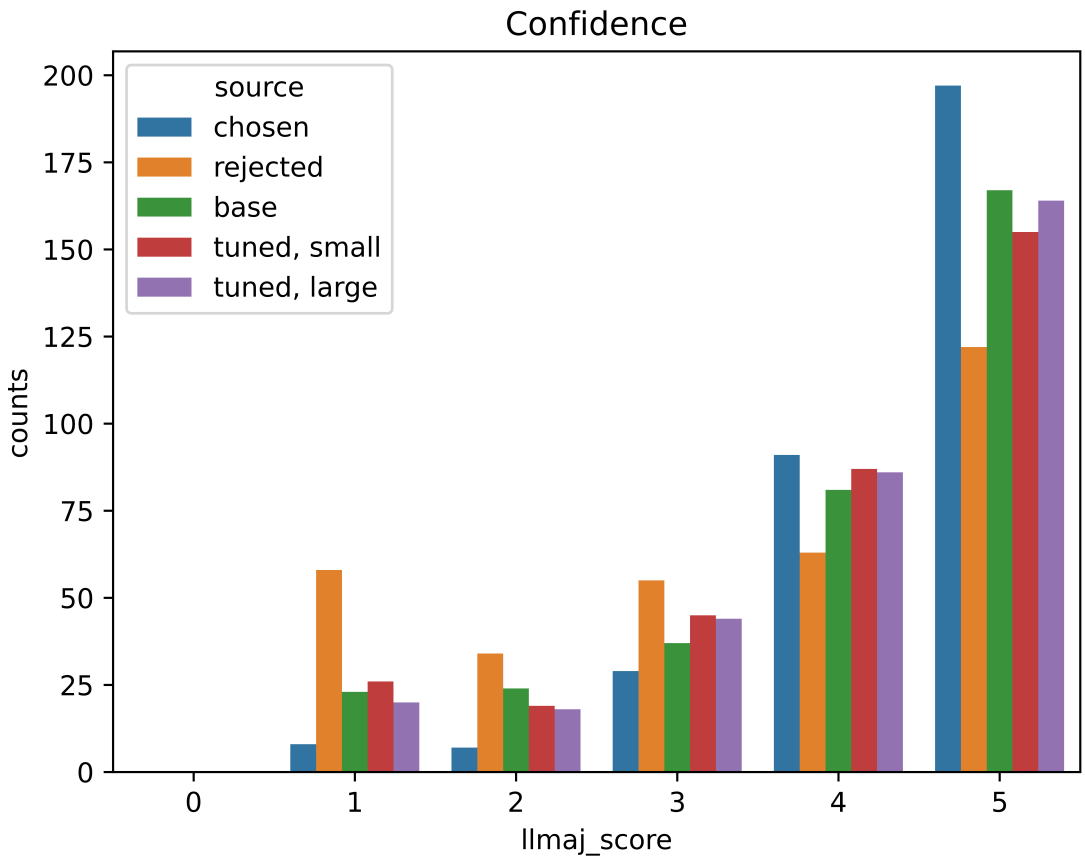

Relevance
source
chosen
rejected
base
tuned, small
tuned, large
counts
0
100
200
300
400
500
0
1
2
3
4
5
llmaj_score


Completeness
source
chosen
rejected
base
tuned, small
tuned, large
counts
0
50
100
150
200
250
0
1
2
3
4
5
llmaj_score

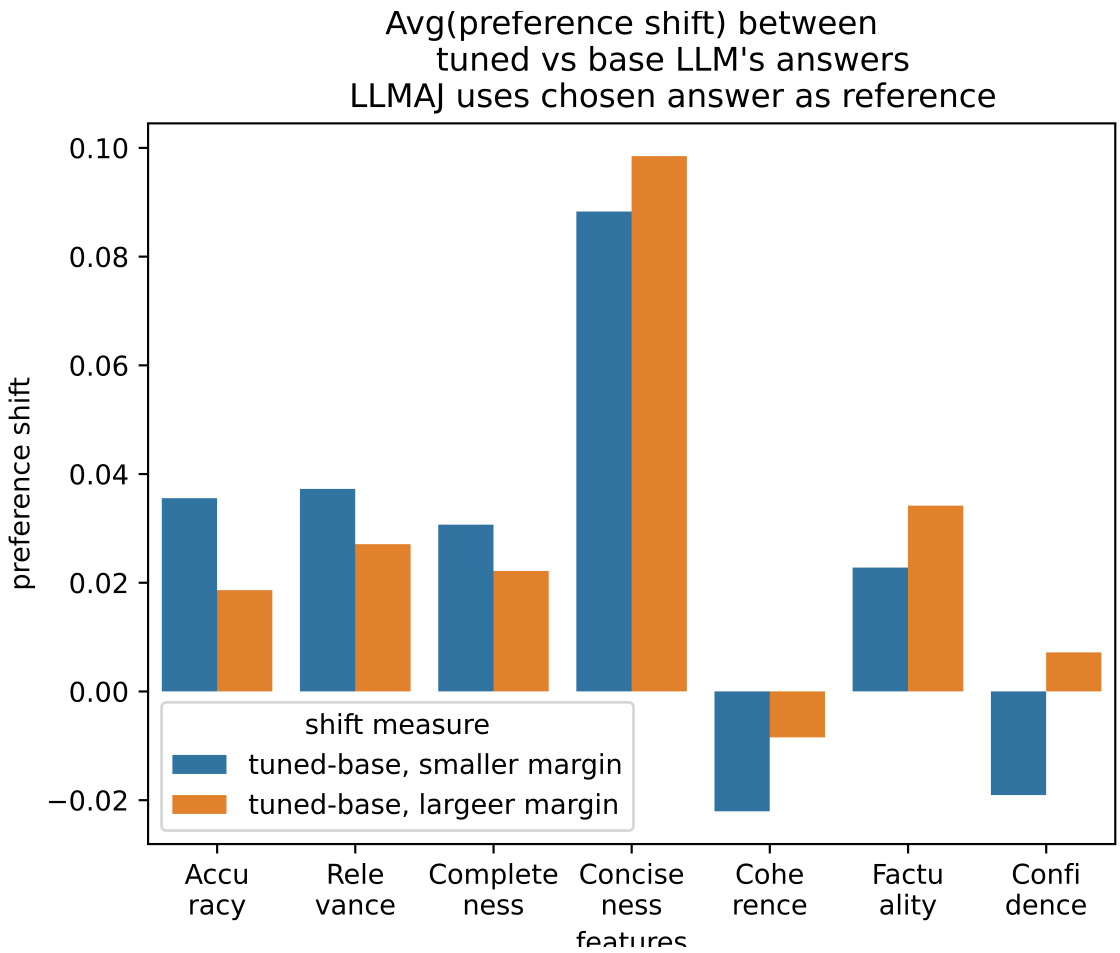

Conciseness
source
chosen
rejected
base
tuned, small
tuned, large
counts
300
250
200
150
100
50
0
0
1
2
3
4
5
llmaj_score

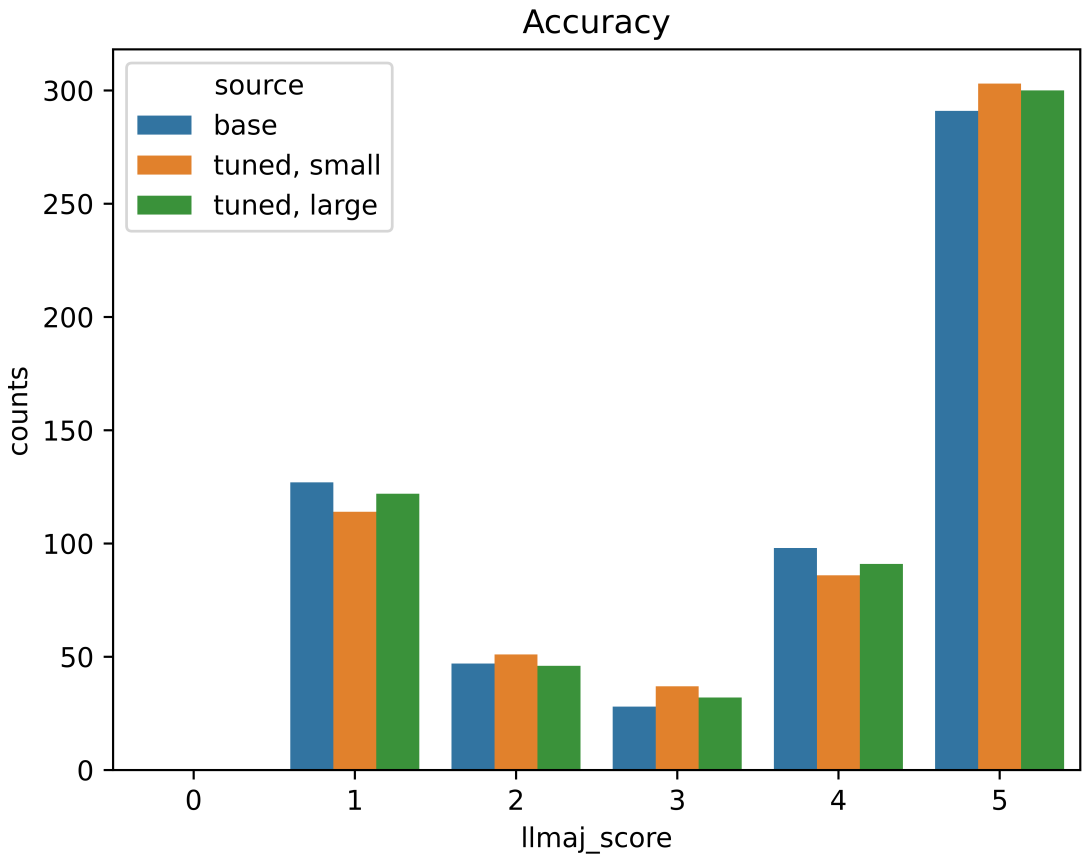

Coherence and Fluency
source
chosen
rejected
base
tuned, small
tuned, large
counts
0
100
200
300
400
500
0
1
2
3
4
5
llmaj_score

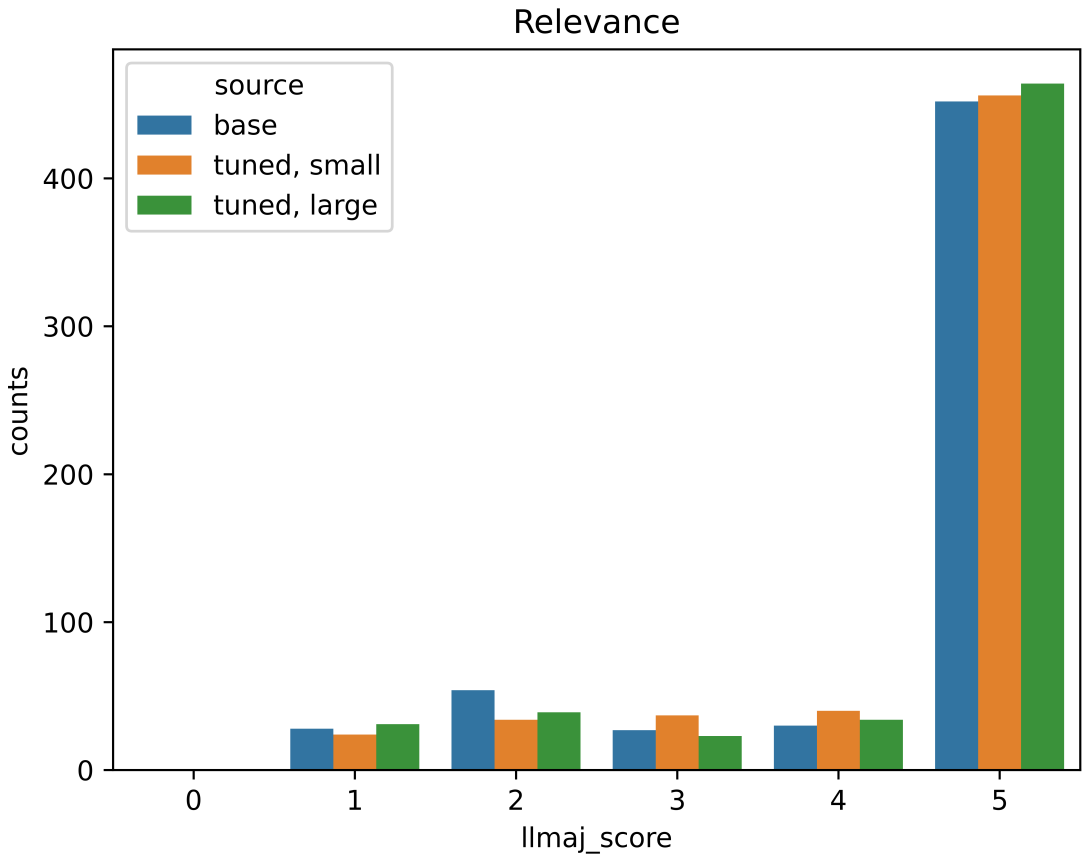

Factuality
source
chosen
rejected
base
tuned, small
tuned, large
counts
0
20
40
60
80
100
120
140
160
0
1
2
3
4
5
llmaj_score

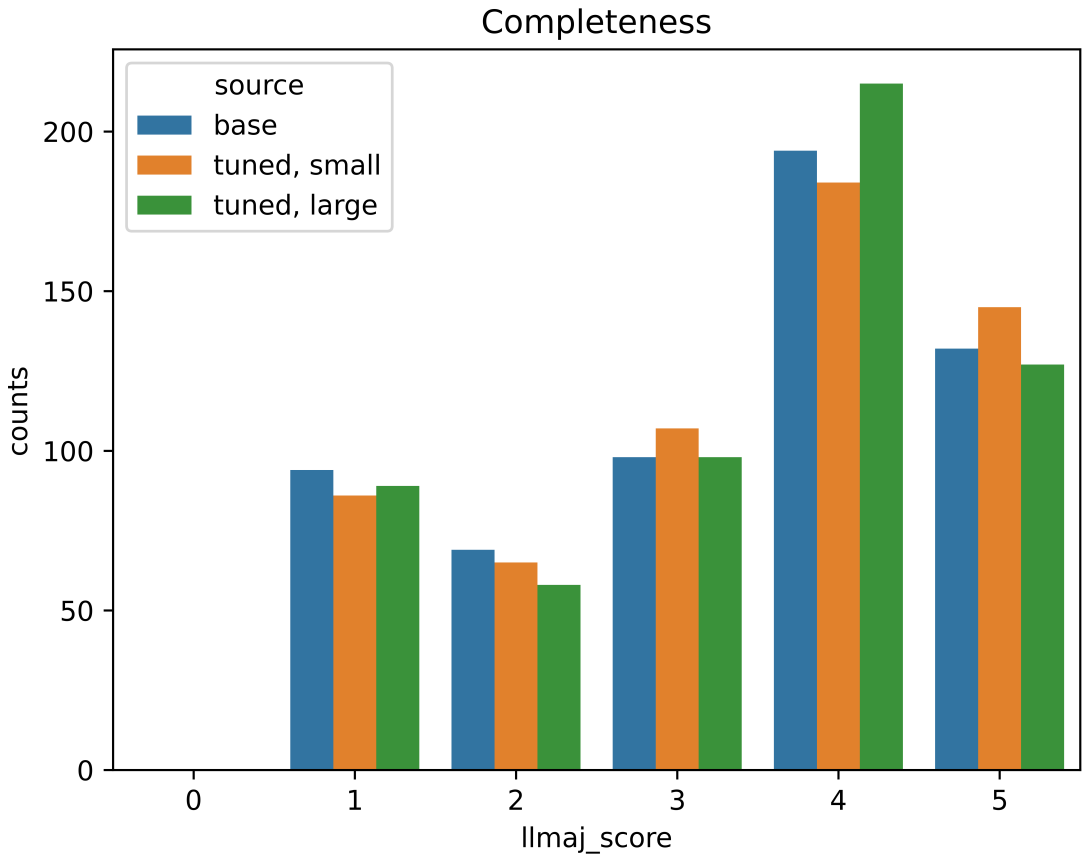


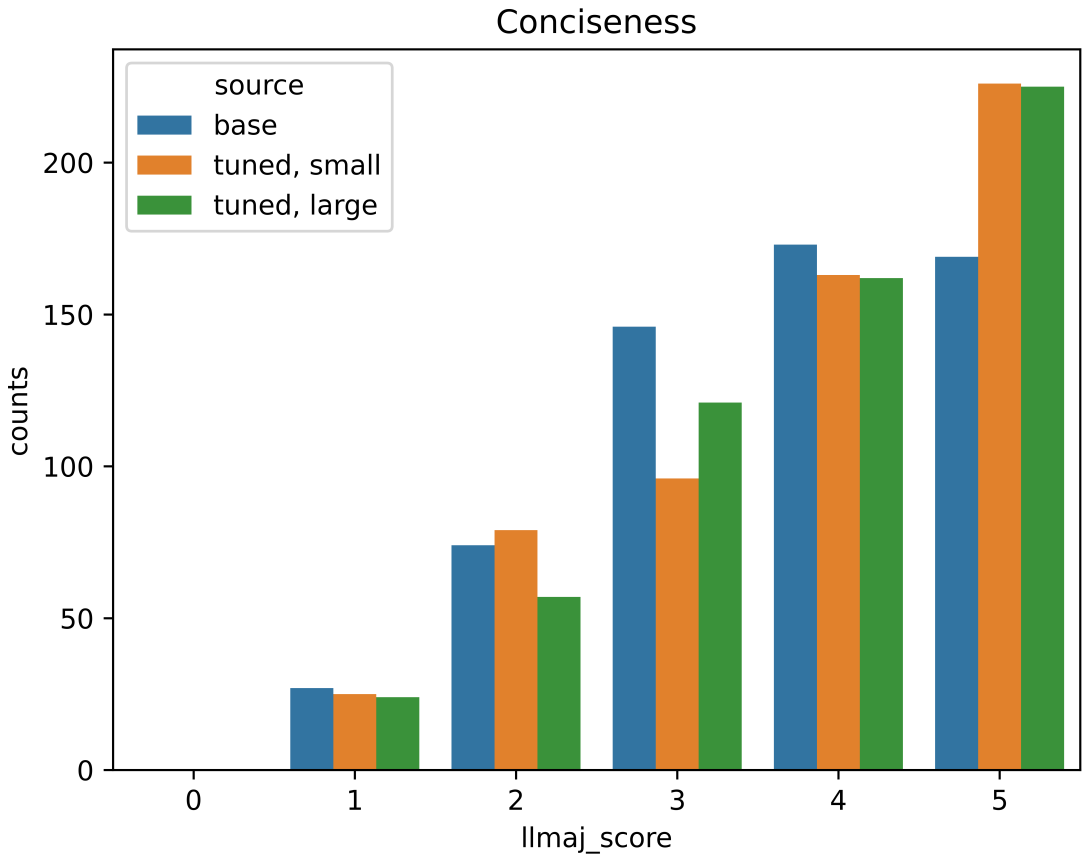
LLMAJ with golden answer: 600 validation points

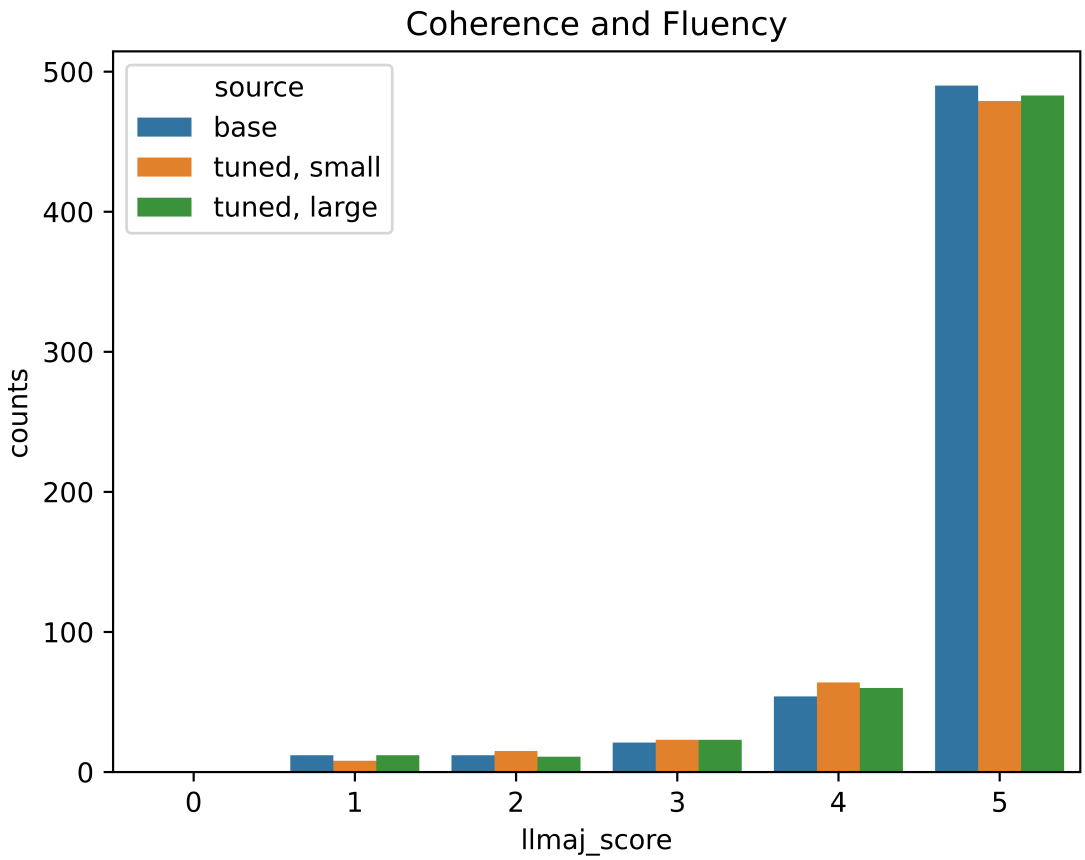

Avg(preference shift) between
tuned vs base LLM's answers
LLMAJ uses chosen answer as reference
preference shift
0.10
0.08
0.06
0.04
0.02
0.00
−0.02
shift measure
tuned-base, smaller margin
tuned-base, largeer margin
Accu
racy
Rele
vance
Complete
ness
Concise
ness
Cohe
rence
Factu
ality
Confi
dence
features

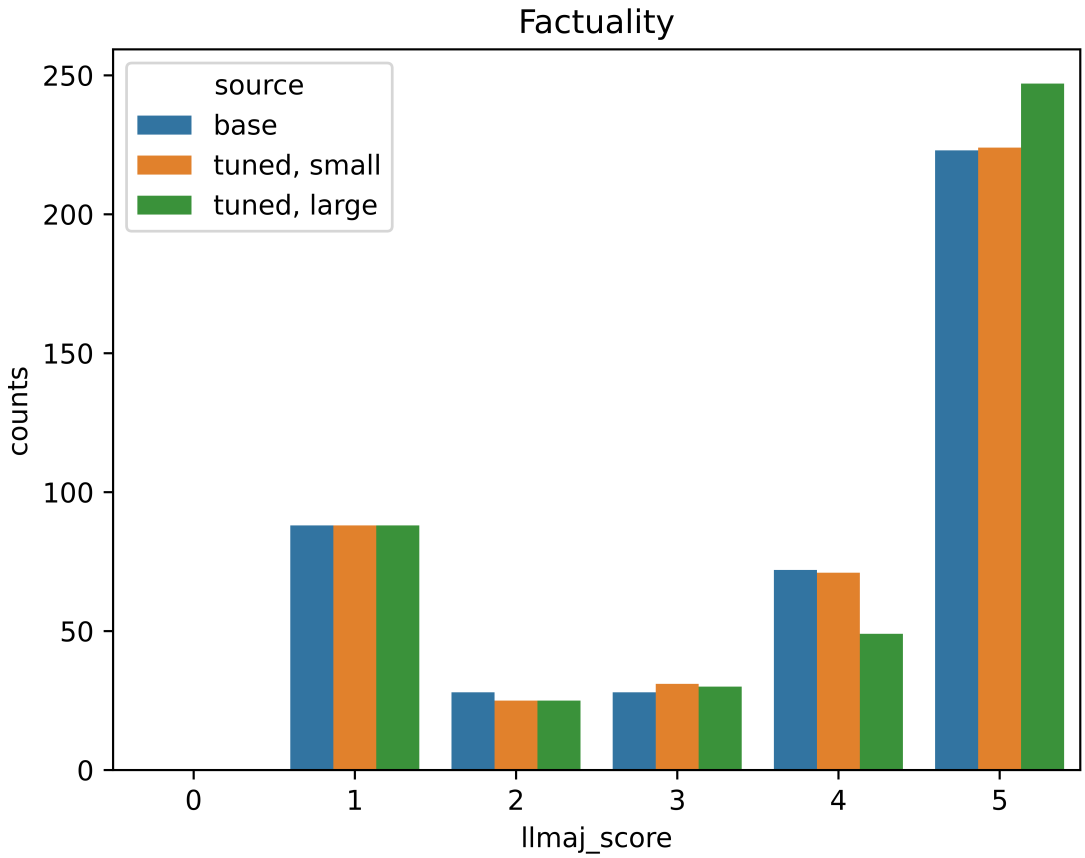

Accuracy
source
base
tuned, small
tuned, large
counts
300
250
200
150
100
50
0
0
1
2
3
4
5
llmaj_score

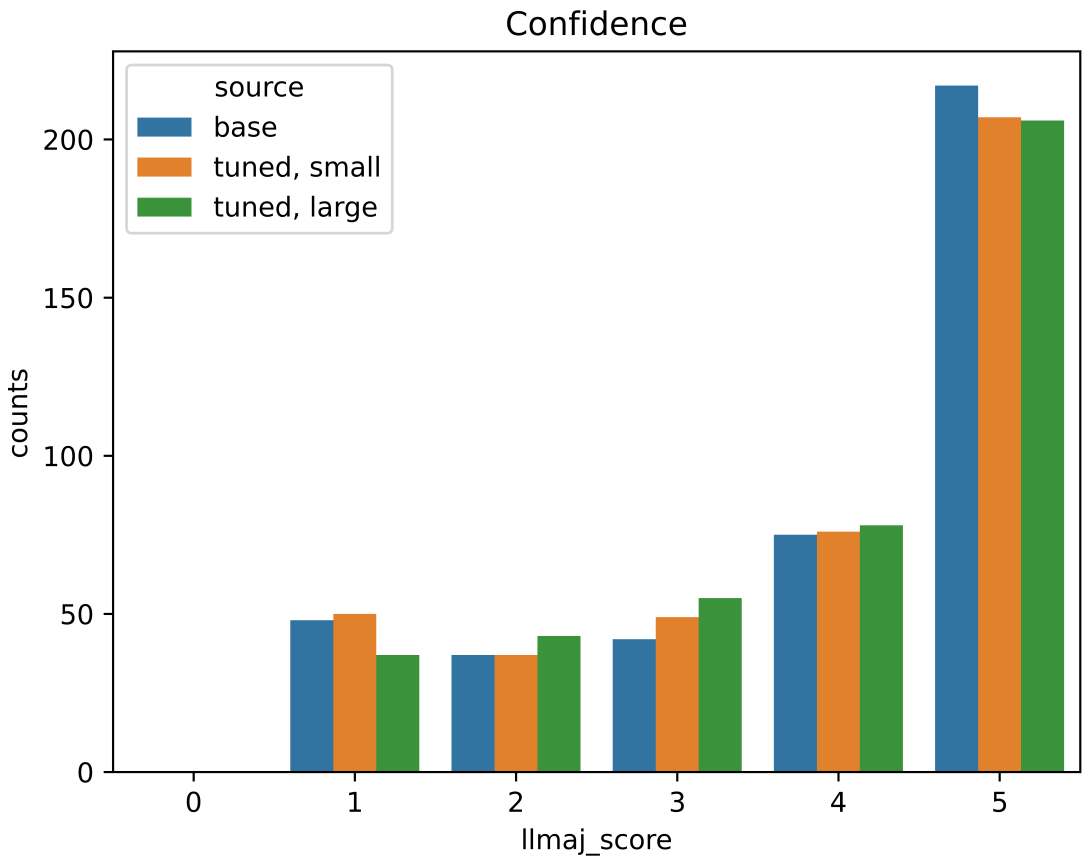

Relevance
source
base
tuned, small
tuned, large
counts
0
100
200
300
400
0
1
2
3
4
5
llmaj_score


Completeness
source
base
tuned, small
tuned, large
counts
0
50
100
150
200
0
1
2
3
4
5
llmaj_score


Conciseness
source
base
tuned, small
tuned, large
counts
0
50
100
150
200
0
1
2
3
4
5
llmaj_score


Coherence and Fluency
source
base
tuned, small
tuned, large
counts
500
400
300
200
100
0
0
1
2
3
4
5
llmaj_score


Factuality
source
base
tuned, small
tuned, large
counts
250
200
150
100
50
0
0
1
2
3
4
5
llmaj_score



Conclusions and next steps

It is impossible to say if the model had a chance to converge without validation error and reward plots that sadly I didn't generate during training. We may simply have a case of overfitting to the training data. We may potentially have error in loading and applying QLoRA adapter. The issue may be coming from the prompt format. Also the DPO data set may have been designed in such way that it primarily focuses on conciseness and we are chasing wild goose on other dimensions.

Ideally I would love to re-train the DPO using a simplified prompt , log evaluation metrics on every training step, and make sure there is no error in loading QLoRA adapter.

If we still do not see significant changes, maybe to just use https://huggingface.co/datasets/Anthropic/hh-rlhf for tuning and evaluate

improvements only on **helpfulness and harmlessness.**

After speaking with Alexei, a couple more thoughts emerged:

1. switch from QA evaluation prompt to instruction following prompt with LLMAJ as not every question in the DPO data set is about QA
2. compare with neural chat performance assuming that is our gold standard tuned LLM just to get a feel what is the performance we are targeting here
3. switch from [1..5] score scale to binary : good enough or not good enough