Small Language Models Improve Giants by Rewriting Their Outputs

Giorgos Vernikos, Arthur Brazinskas, Jakub Adamek, Jonathan Mallinson, Aliaksei Severyn, Eric Malmi

EACL 2024

Introduction: In-context Learning

Large Language Models have demonstrated impressive capabilities!

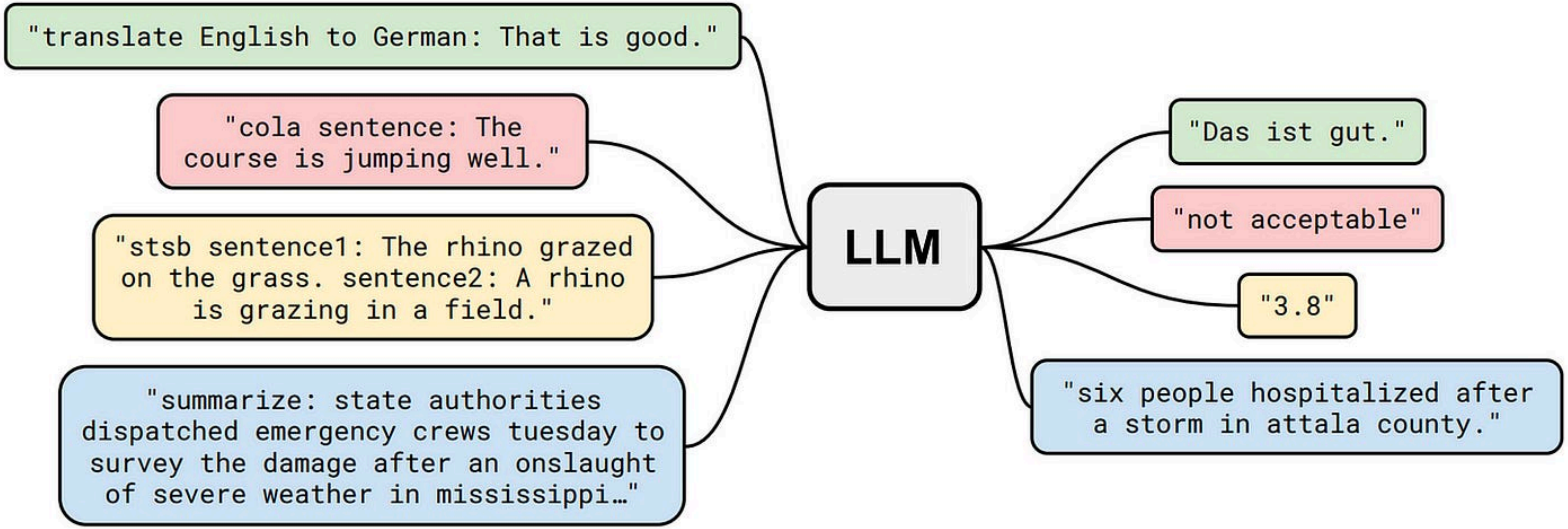


Figure from [Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer]

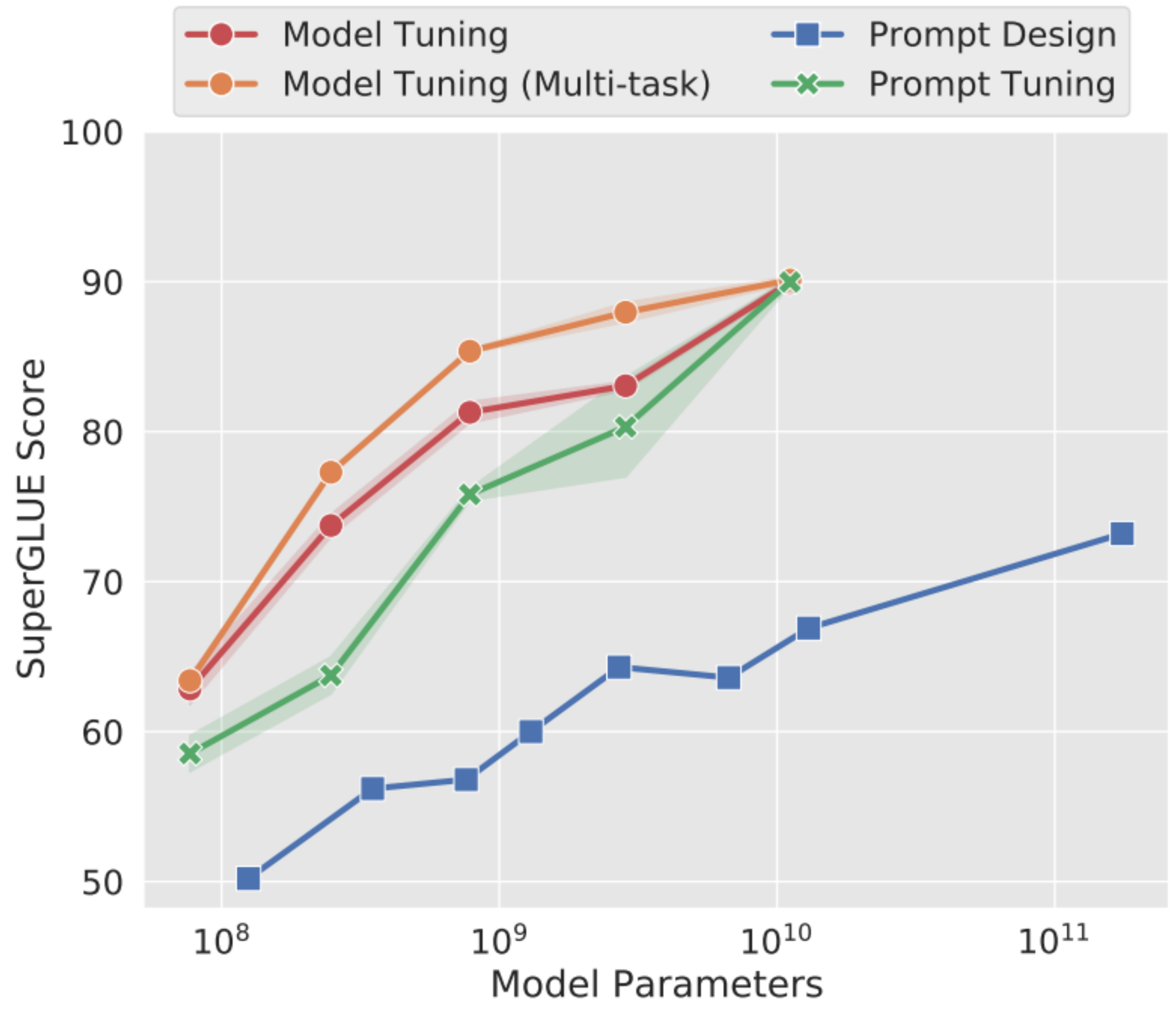
Downsides of in-context learning

1. **Sensitivity** to the description [Webson & Pavlick, 2022], selection [Liu et al., 2022] and ordering [Lu et al., 2022] of in-context examples

2. **Poor performance** compared to fine-tuned models [Lester et al., 2021; Xu et al., 2023]

Methods	MNLI-m	MNLI-mm	SST-2	QNLI	MRPC	QQP	CoLA	RTE	Avg.
GPT-3.5 ICL	80.80	82.39	91.39	80.52	60.05	81.64	60.51	86.28	81.32
RoBERTa-Large	88.68	89.47	96.44	94.07	83.09	92.11	64.55	87.00	88.68

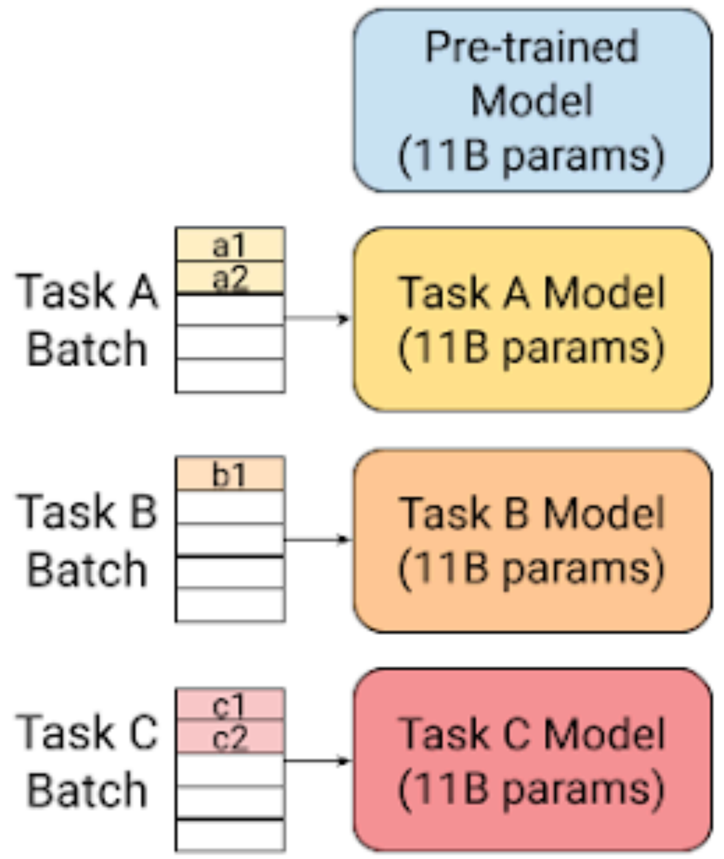
Table 2: Experimental results on GLUE (Wang et al., 2019) development set. The metric for CoLA is Matthews Correlation and all other tasks use accuracy.



Introduction: Parameter-Efficient Fine-tuning

How can we fine-tune LLMs?

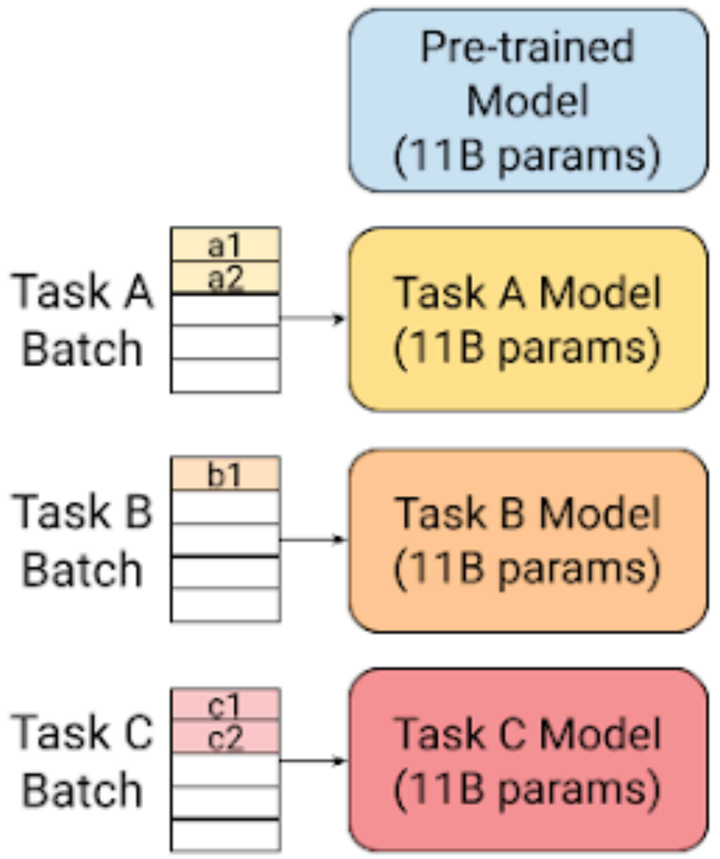
Full fine-tuning



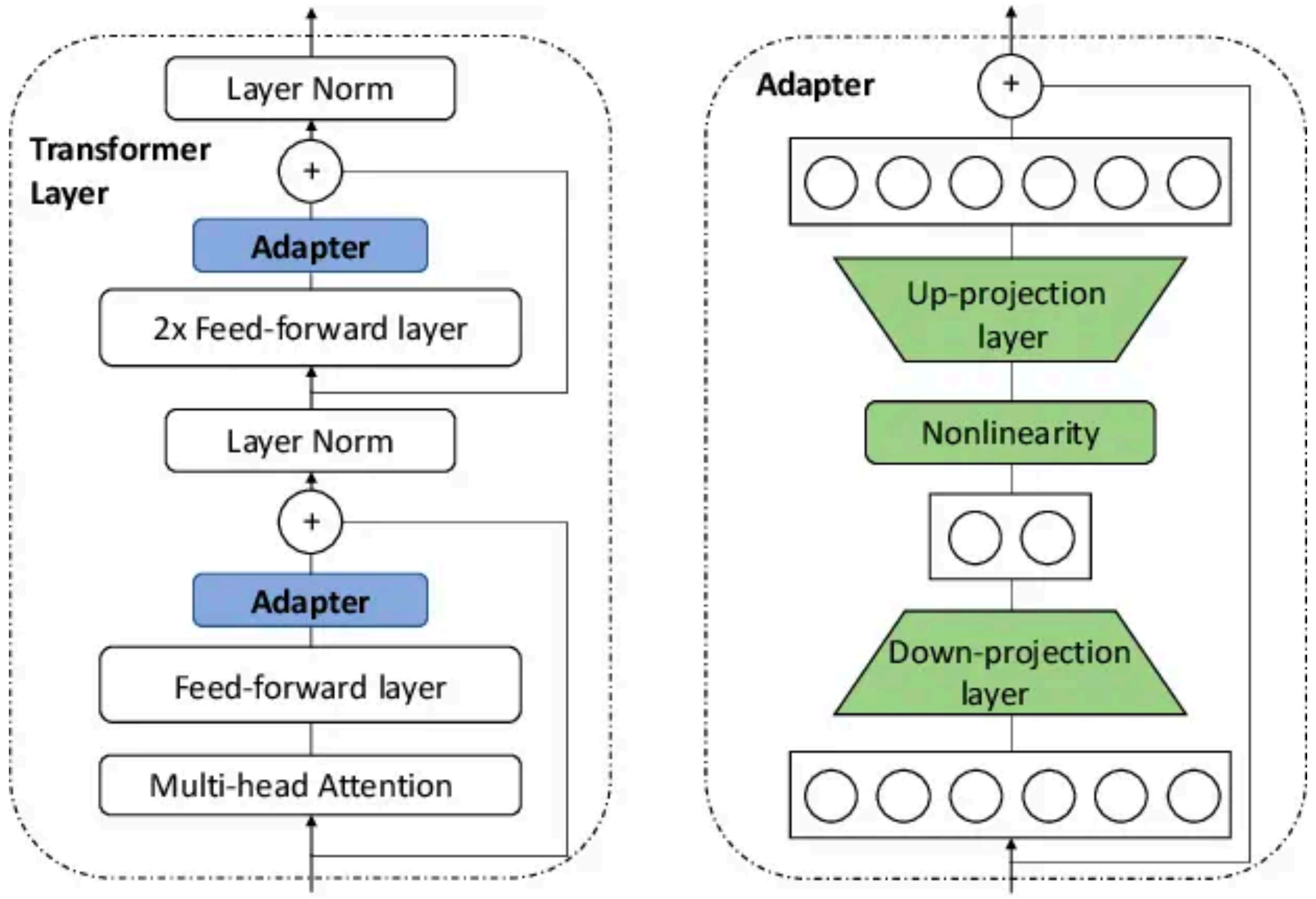
Introduction: Parameter-Efficient Fine-tuning

How can we fine-tune LLMs?

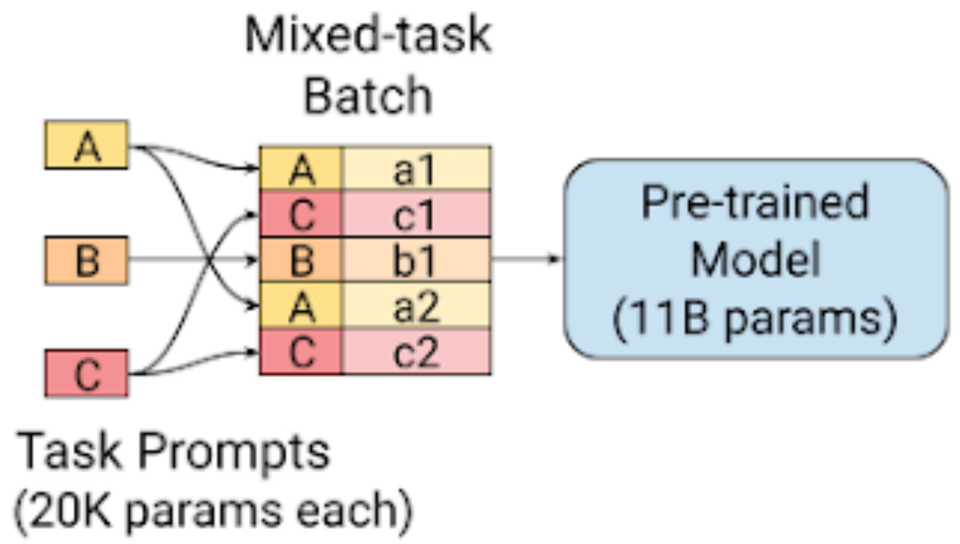
Full fine-tuning



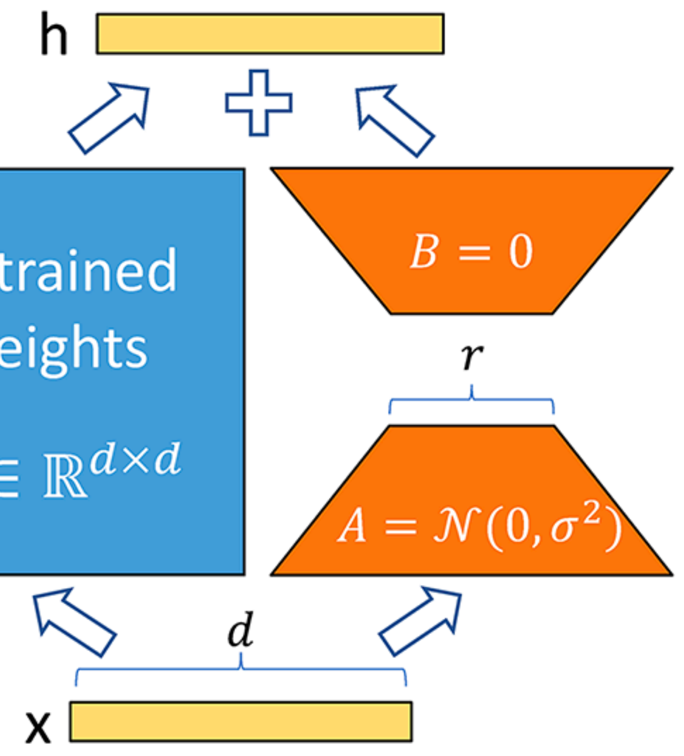
Adapters [Houlsby et al., 2019]



Prompt tuning [Lester et al., 2021]



LoRa [Hu et al., 2022]



Introduction: Parameter-Efficient Fine-tuning

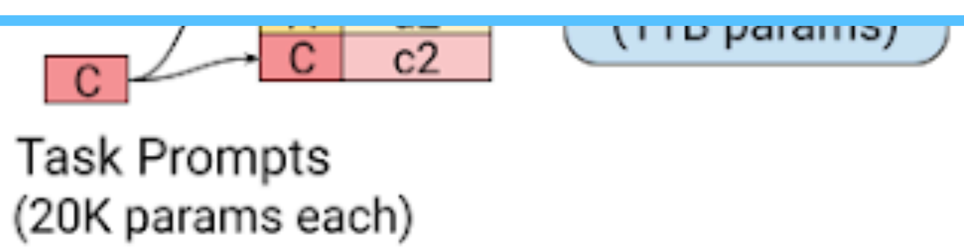
How can we fine-tune LLMs?

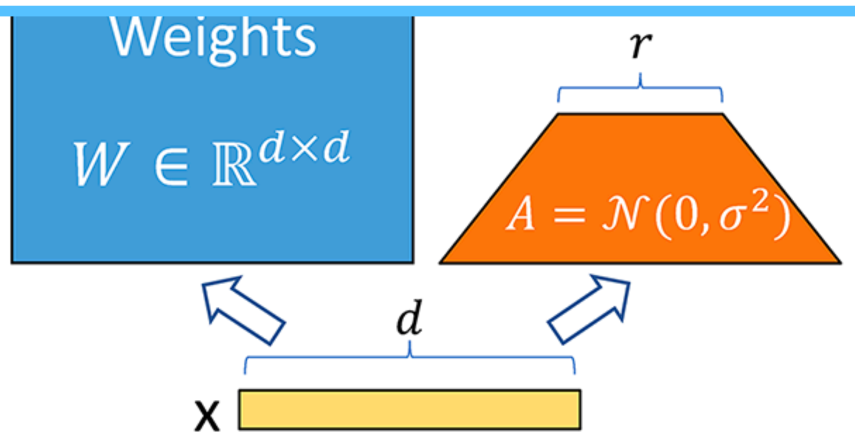
Full fine-tuning

Adapters [Houlsby et al., 2019]

However these methods still require:

- **computational resources** to load and update the model
- **access to the model's weights**





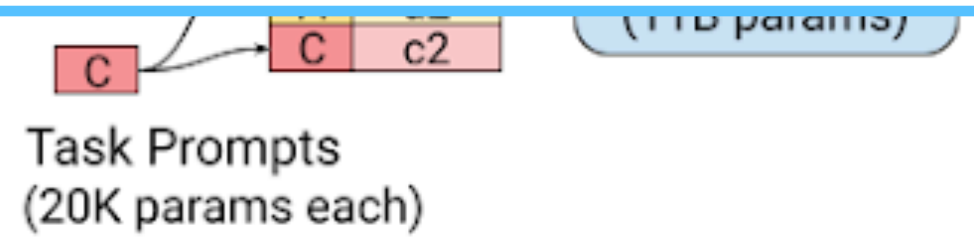
Introduction: Parameter-Efficient Fine-tuning

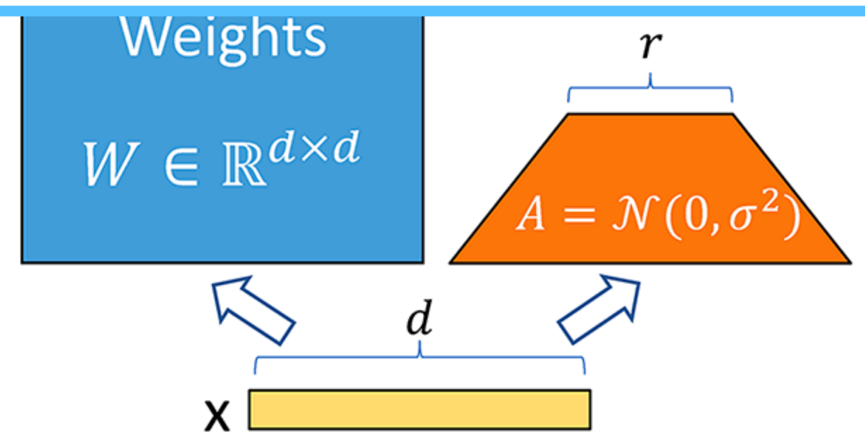
How can we fine-tune LLMs?

Full fine-tuning **Adapters** [Houlsby et al., 2019]

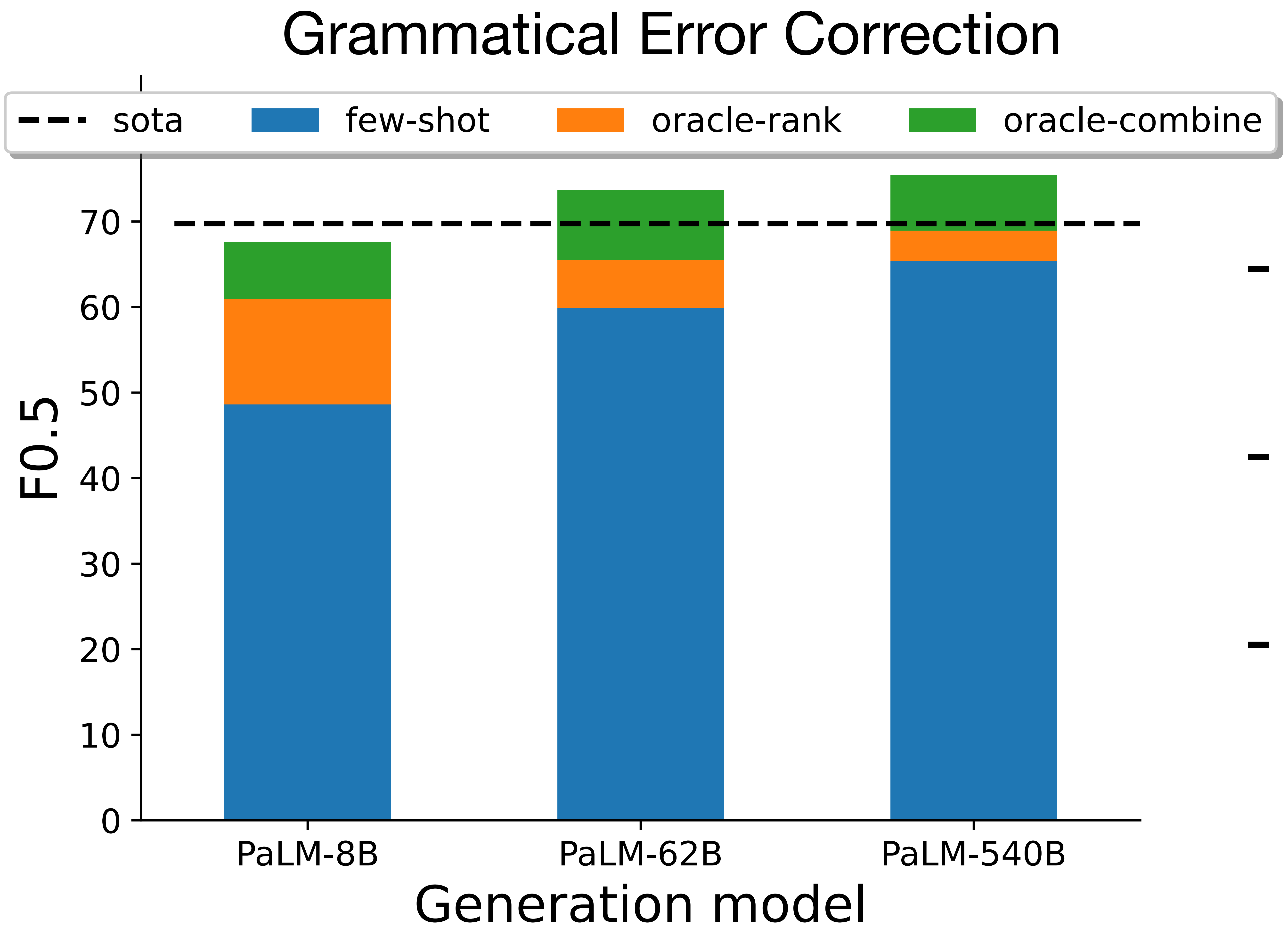
We propose **LMCor**:

- **compact model** that corrects the predictions of **LLMs**
- leverages **only the outputs of the LLMs**



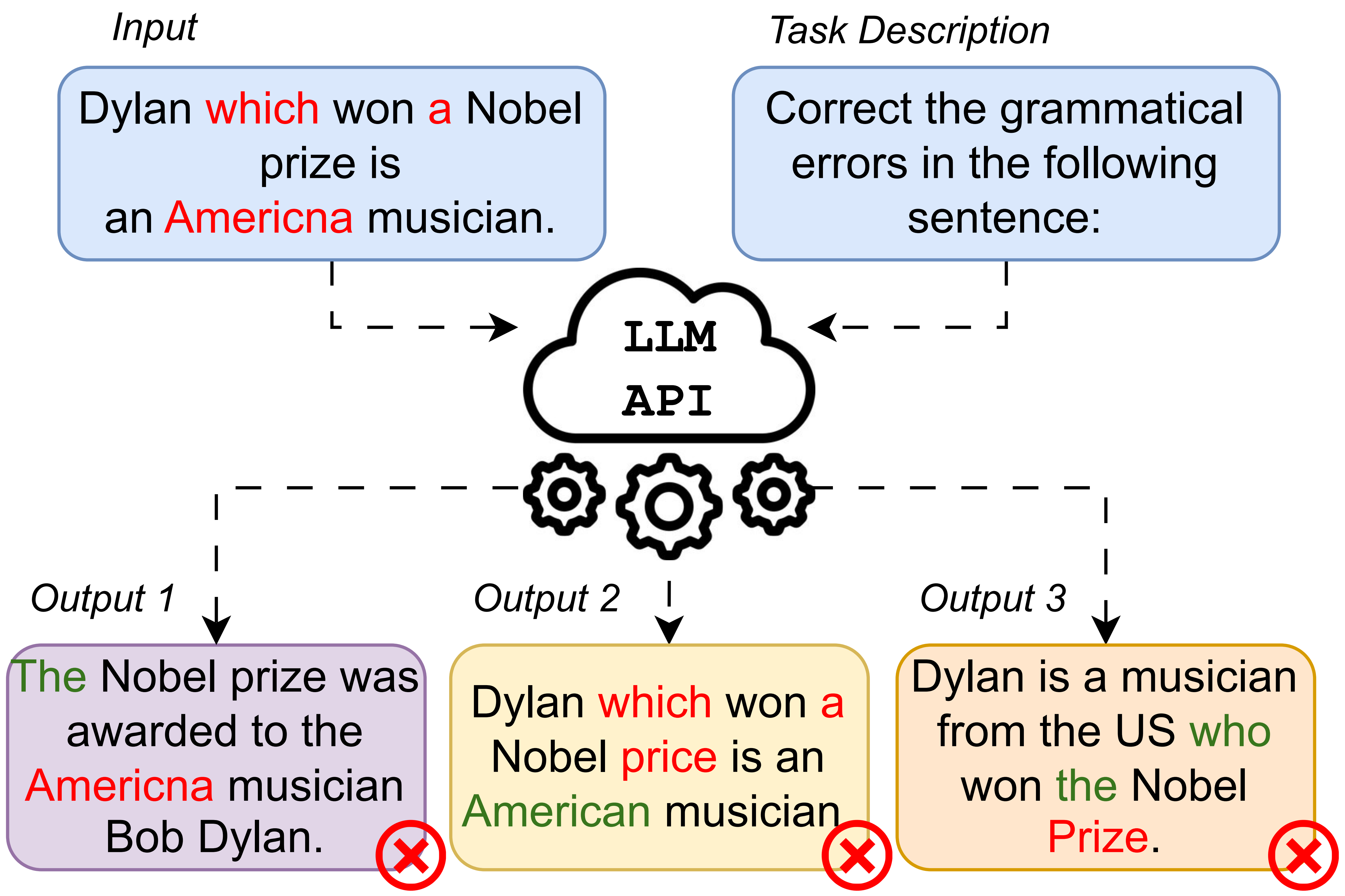


Approach: Motivation



- **Few-shot** prompting is competitive but underperforms state of the art (sota)
- **Sampling** and **ranking** multiple outputs shows moderate improvements
- **Combining** sampled outputs leads to significant performance gains

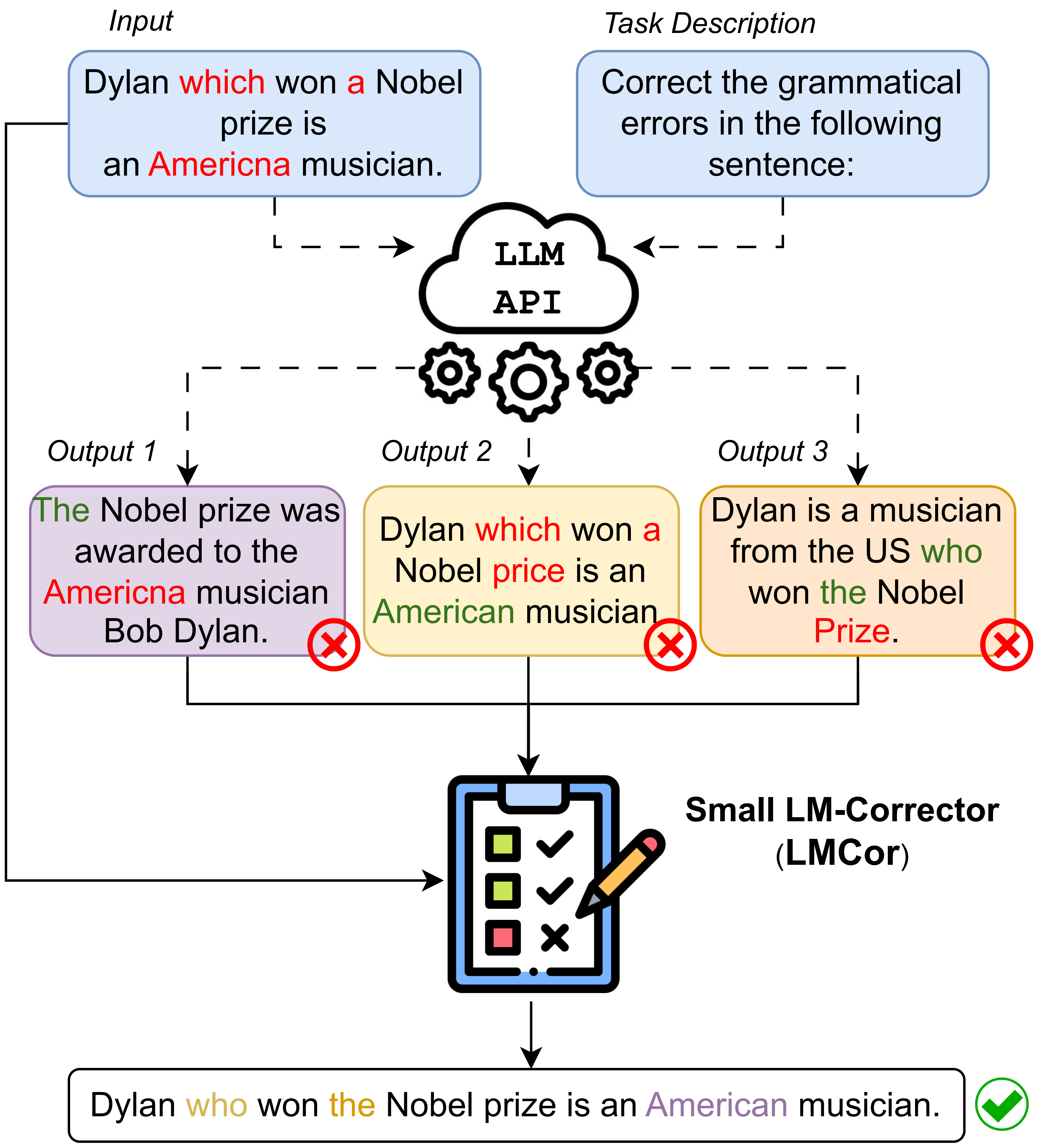
Approach: LM-Corrector



1. We generate multiple outputs from the LLM (API) through few-shot prompting

💡 *Generated outputs have complementary strengths and weaknesses* 💡

Approach: LM-Corrector

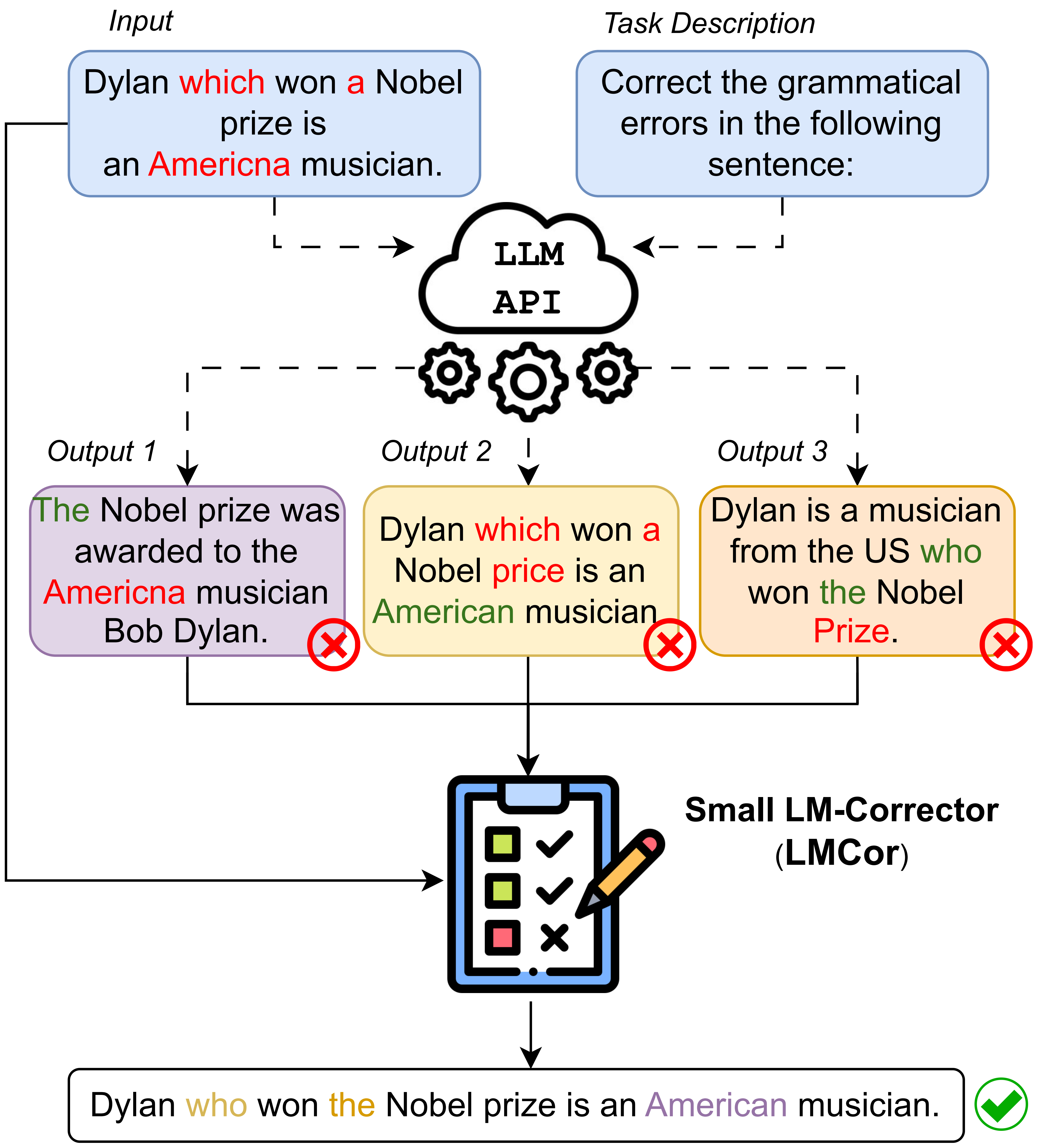


1. We generate multiple outputs from the LLM (API) through few-shot prompting

💡 *Generated outputs have complementary strengths and weaknesses* 💡

2. We feed the input & candidates to a smaller model, the **LM-Corrector** (**LMCor**) to synthesize a refined output.

Approach: LM-Corrector



- LMCor is trained on the <u>task-specific dataset</u> *augmented* with candidates generated by the LLM
- LMCor learns to *rank*, *edit* and ***combine*** the LLM-generated candidates
- LMCor can be **much smaller** than the LLM
- Our approach **does not require access to the weights** of the LLM

Experiments & Results: Datasets and Models

- 4 natural language generation tasks:
  (i) Grammatical error correction: CoNLL-14 (60k examples)
  (ii) Data-to-text generation: E2E NLG (35k examples)
  (iii) Summarization: XSum (204k examples)
  (iv) Machine translation: En->De WMT22 (200k examples)

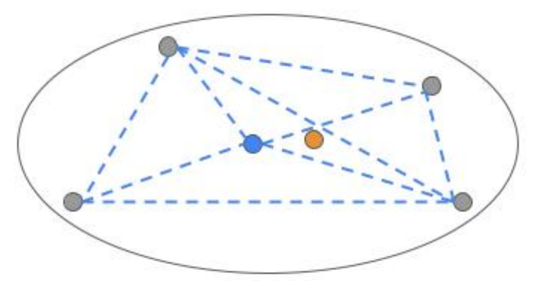
- LLMs: PaLM-62B for (i)-(iii) and XGLM-2.9B for (iv)
- Candidates: Greedy decoded + 4 sampled outputs
- Models: T5-base (250M)

Experiments & Results: Methods

T5-base (FT) — Standard fine-tuning of T5-base on the task-specific dataset

LLM (FS) — Prompting the LLM with few (5) shots

Sampling & Reranking



MBRD-Sim-LCS — Reranking with minimum Bayes risk decoding (MBRD) using longest common subsequence (LCS) as the utility function

Oracle Reranker — Reranking using an oracle that selects the best candidate

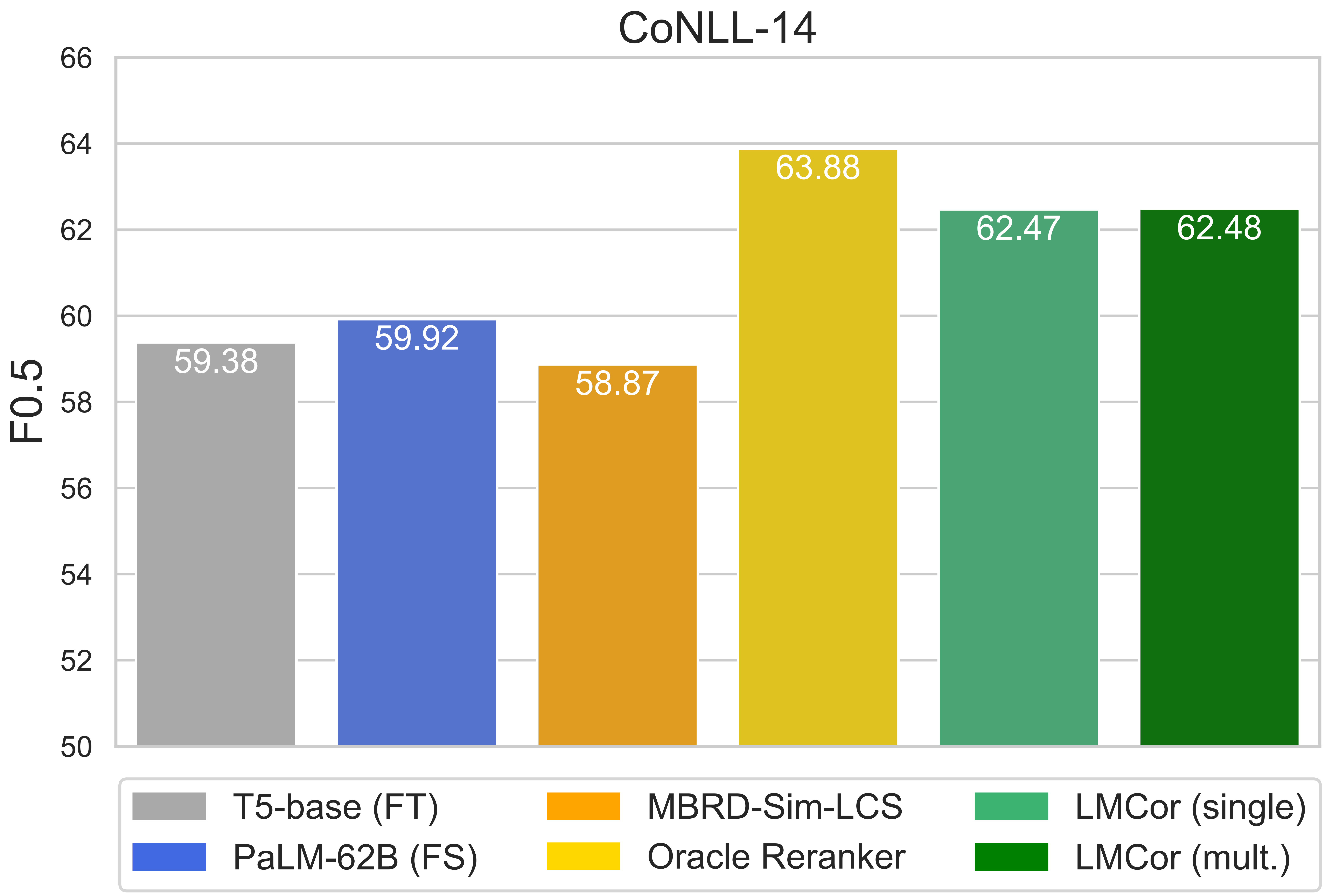
LMCor (ours)

LMCor (single)

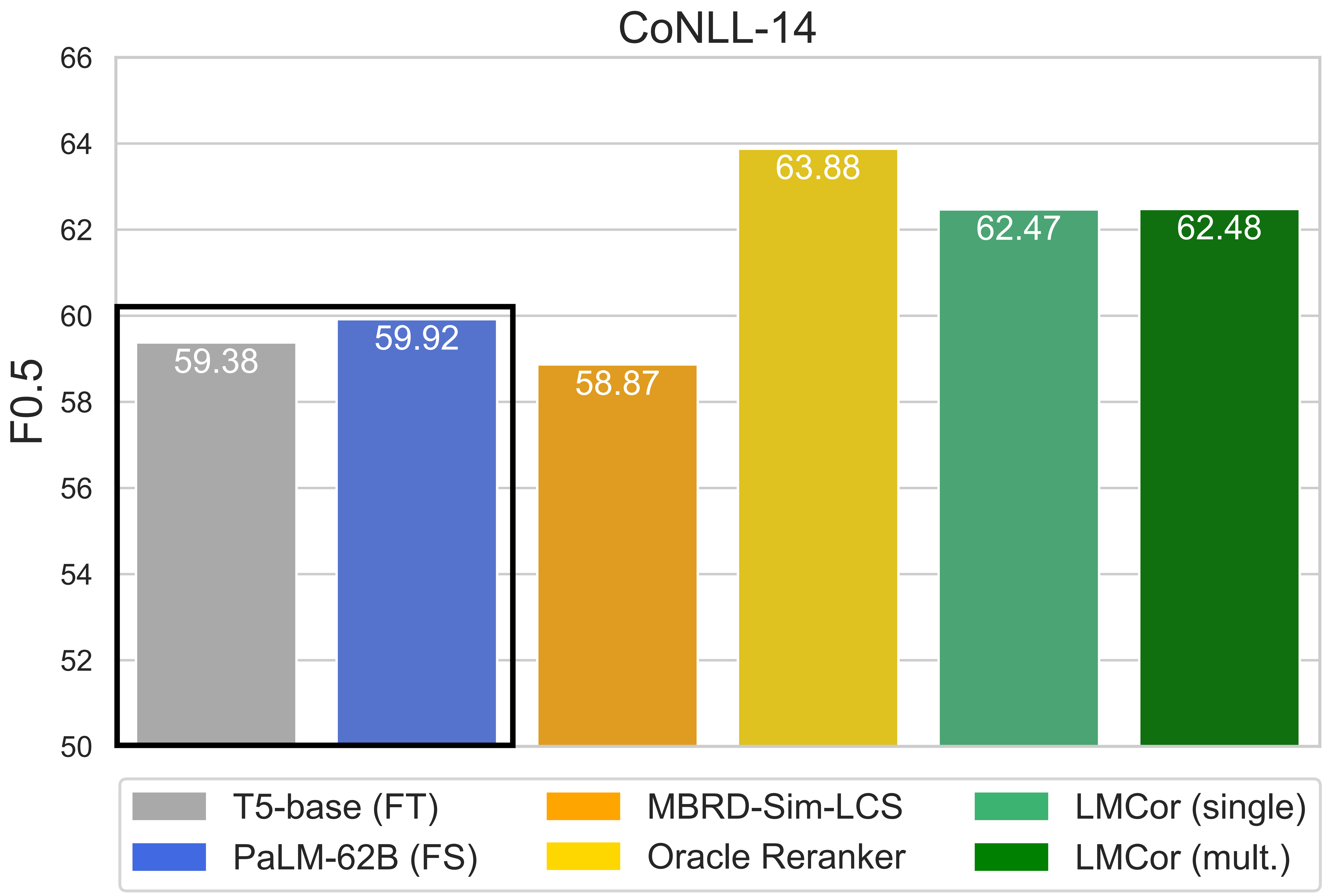
LMCor (mult.)

Fine-tuning a T5-base on the task-specific dataset *augmented* with a single or multiple candidates (mult.) sampled from the LLM

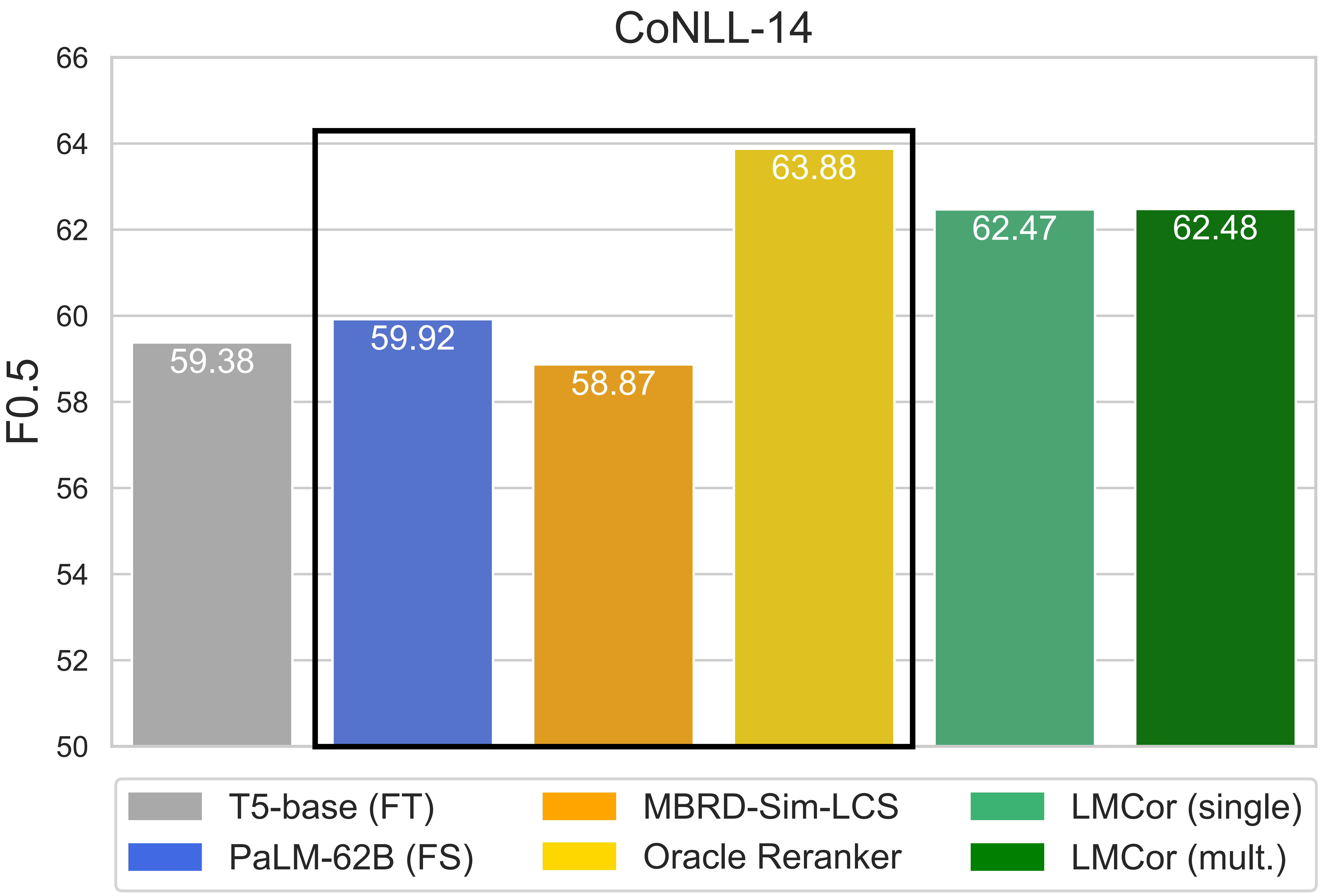
Experiments & Results: Grammatical Error Correction



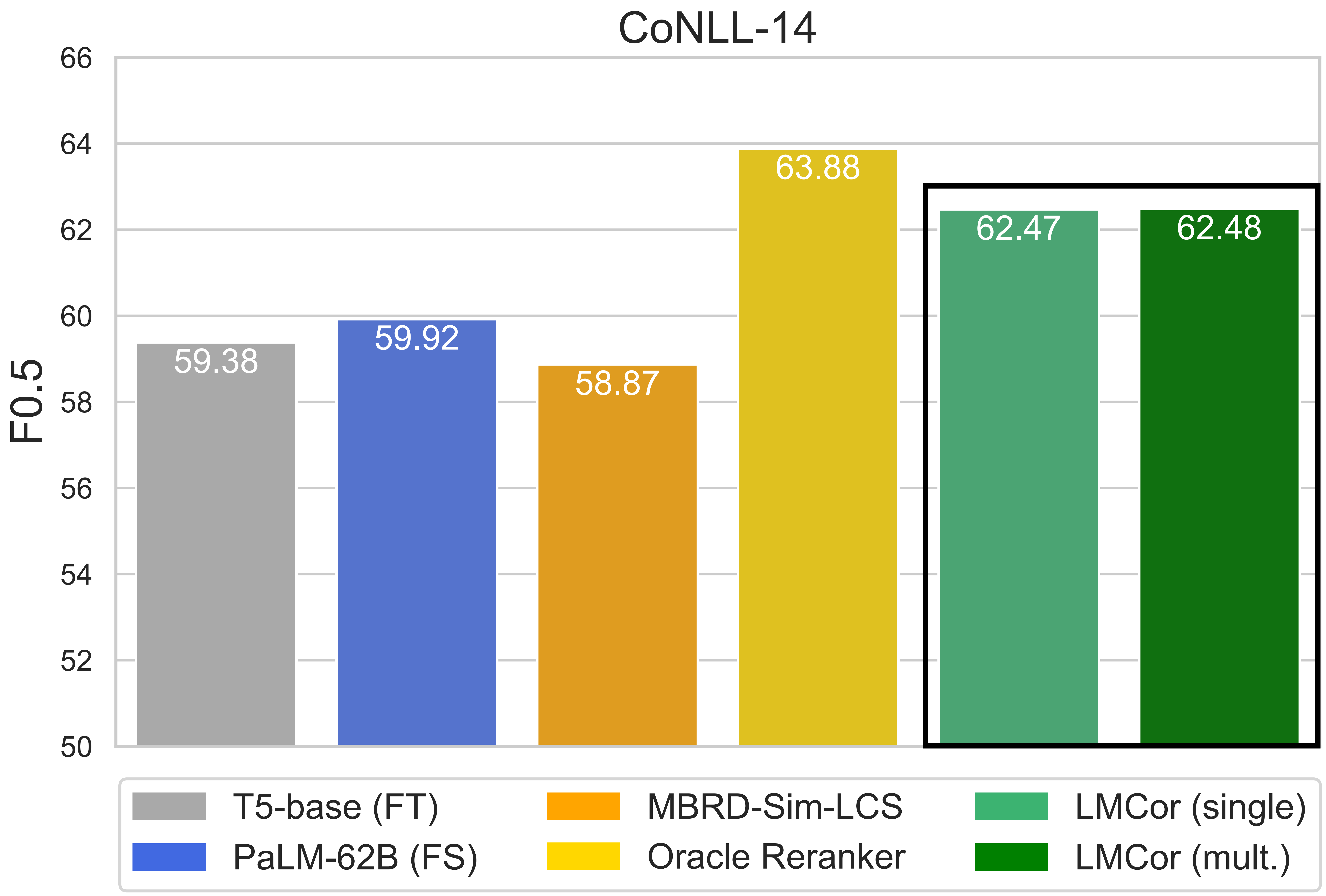
Experiments & Results: Grammatical Error Correction



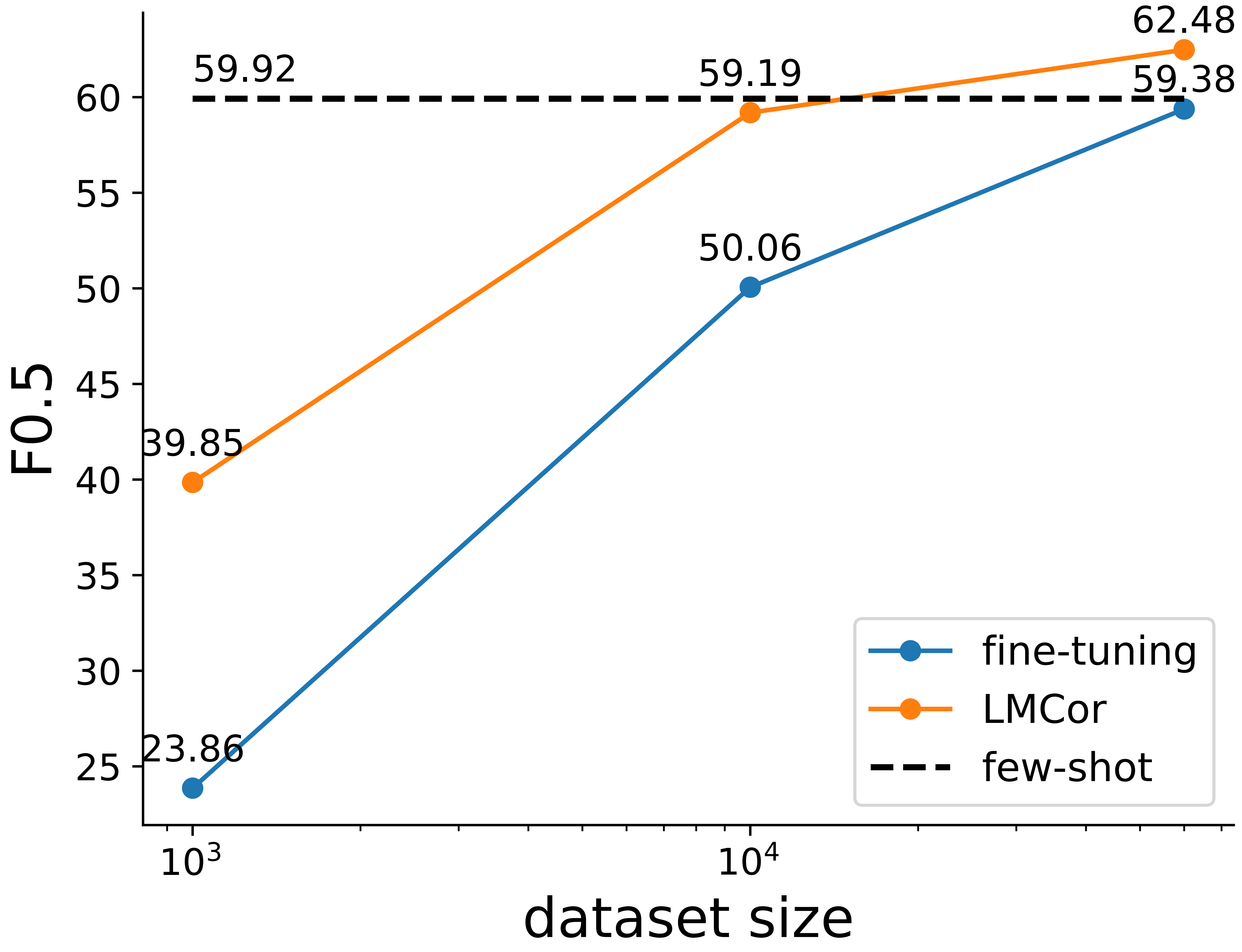
Experiments & Results: Grammatical Error Correction



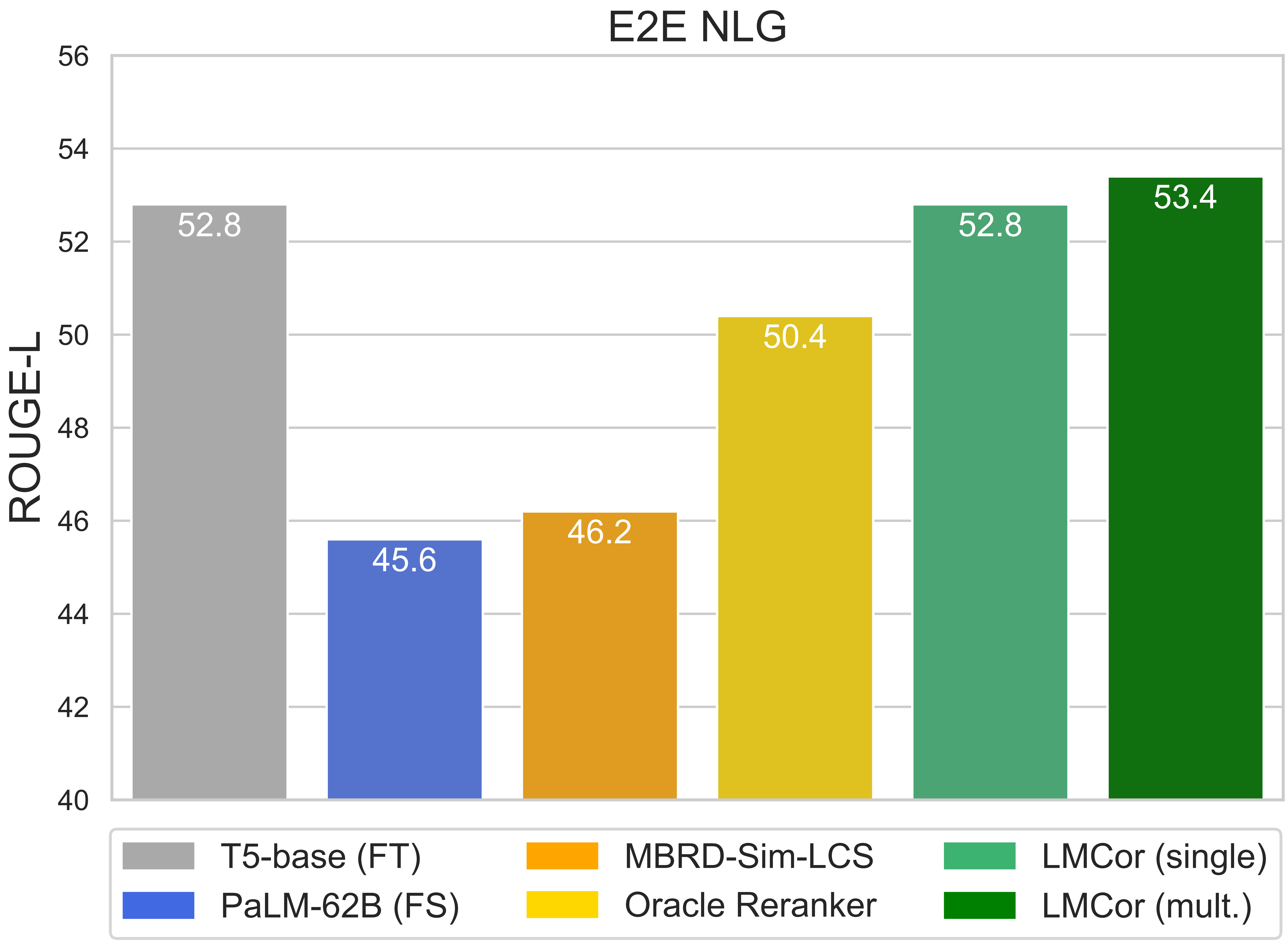
Experiments & Results: Grammatical Error Correction



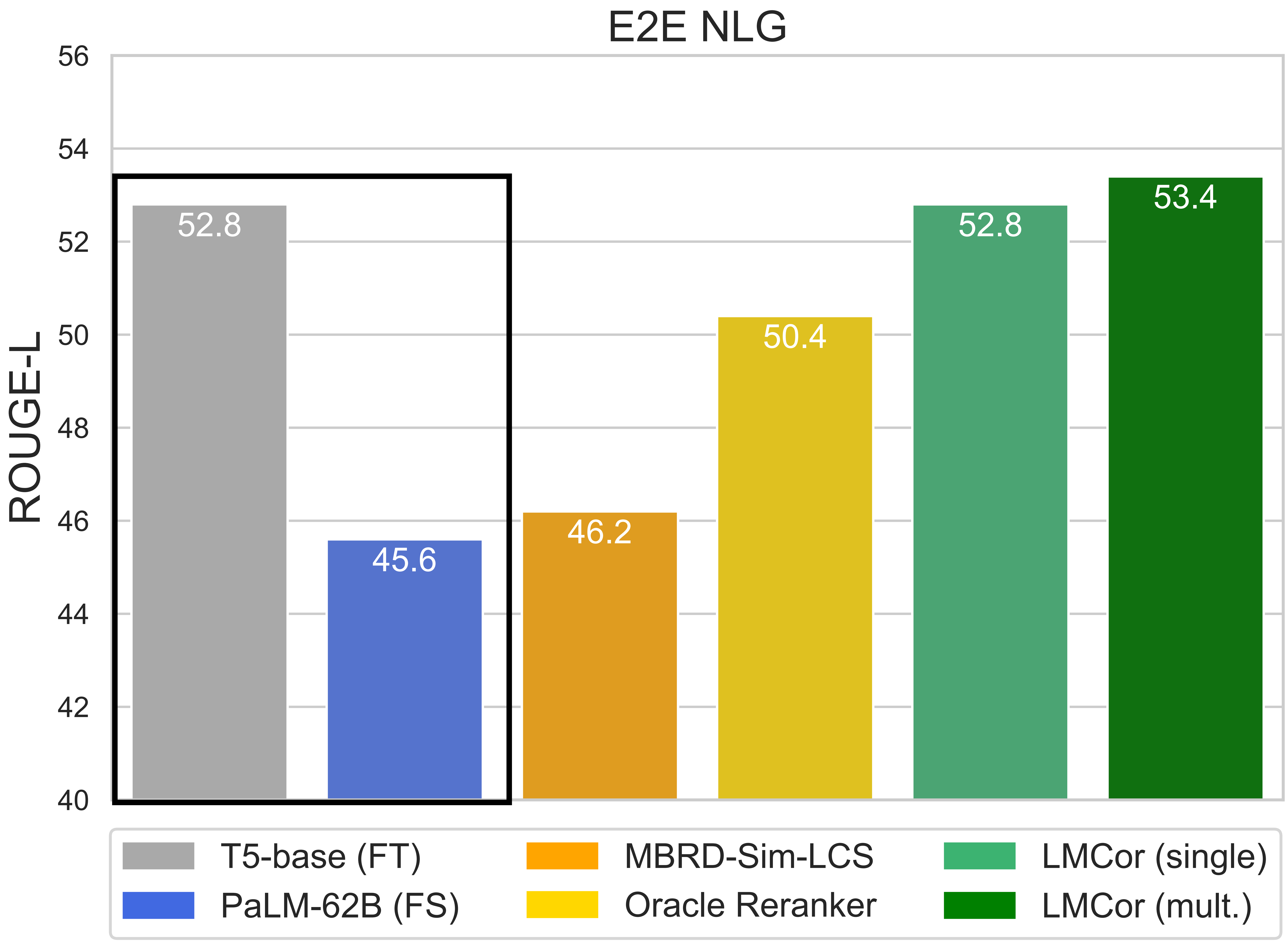
Experiments & Results: Grammatical Error Correction



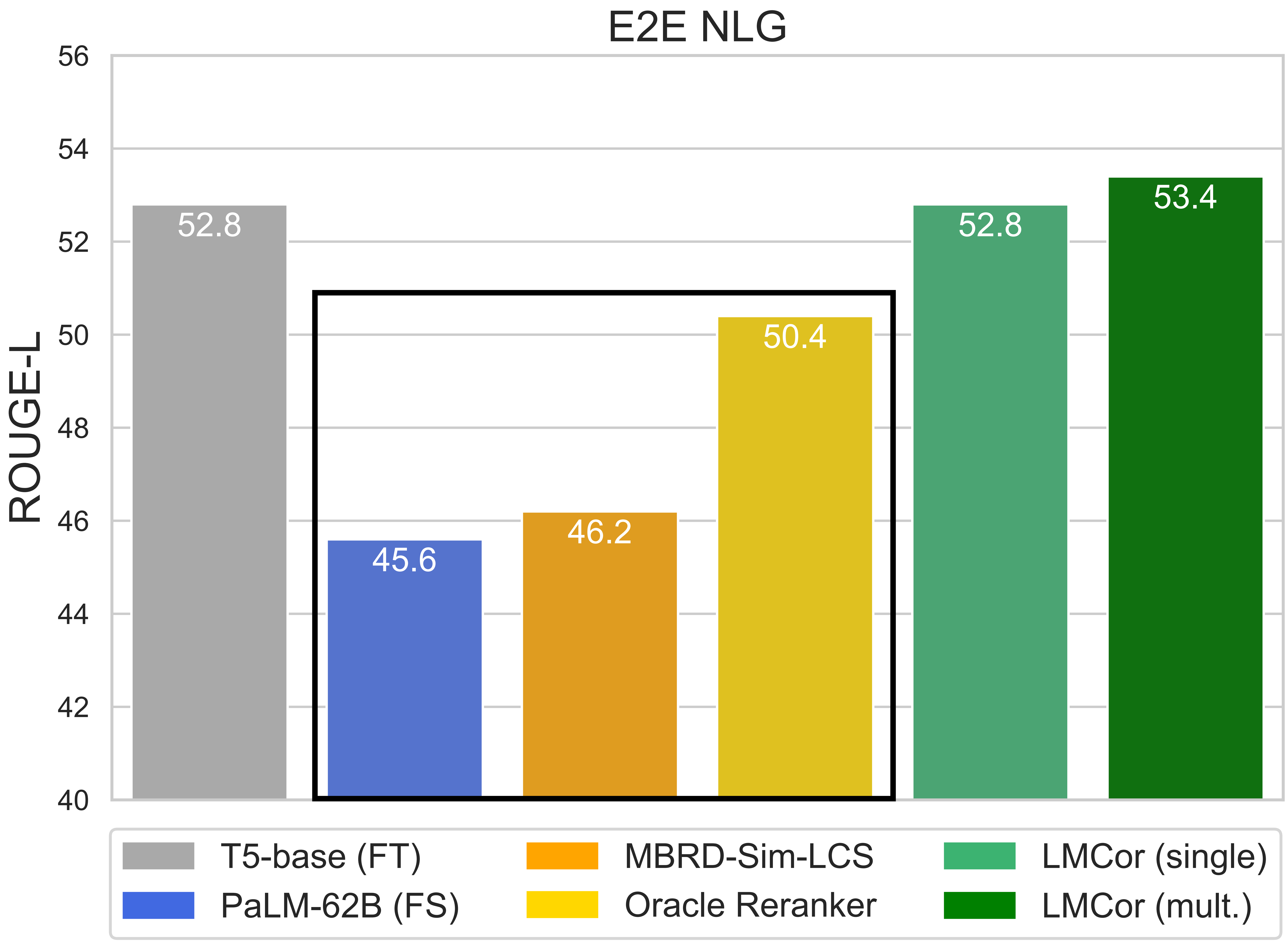
Experiments & Results: Data-to-text Generation



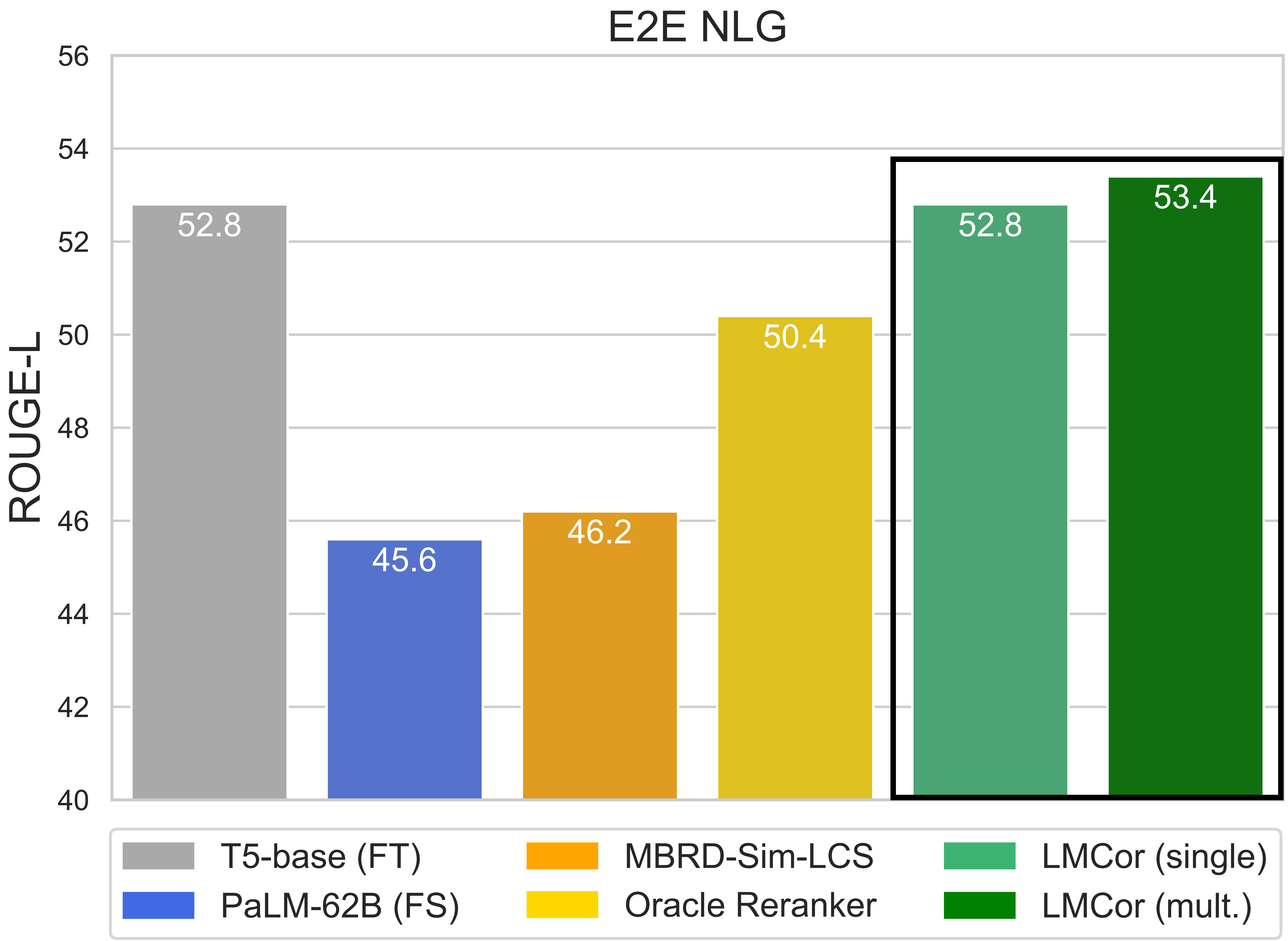
Experiments & Results: Data-to-text Generation



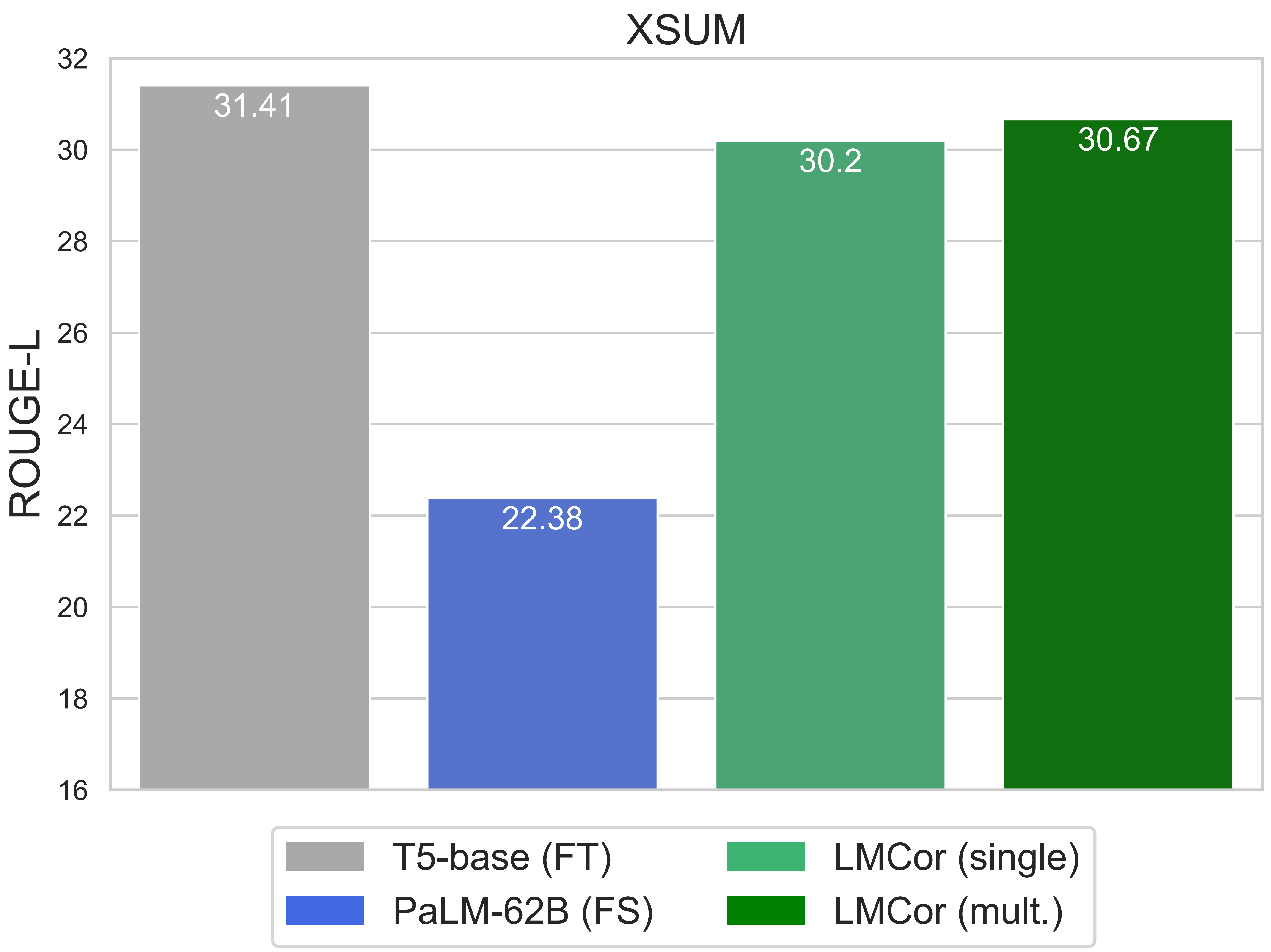
Experiments & Results: Data-to-text Generation



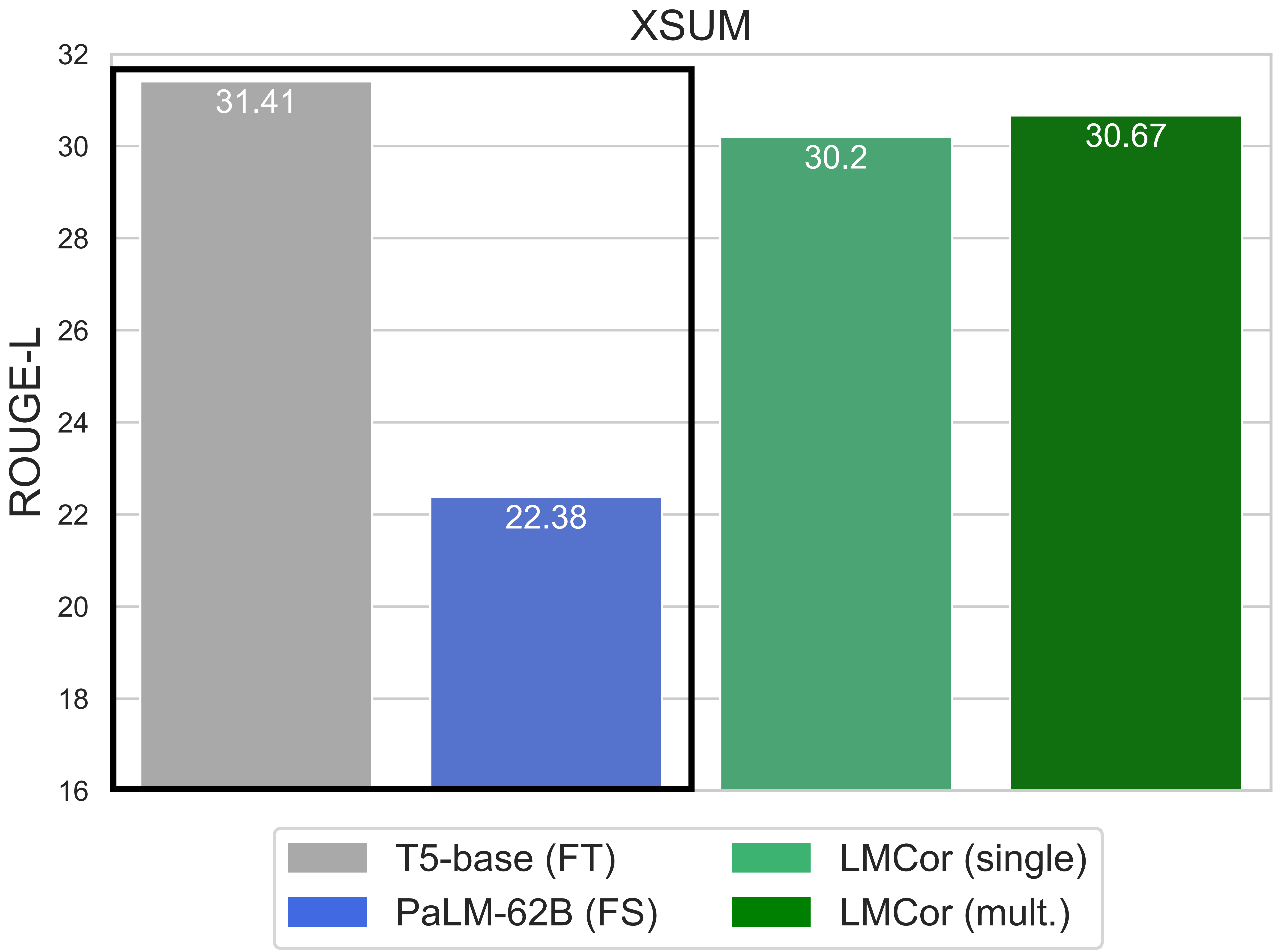
Experiments & Results: Data-to-text Generation



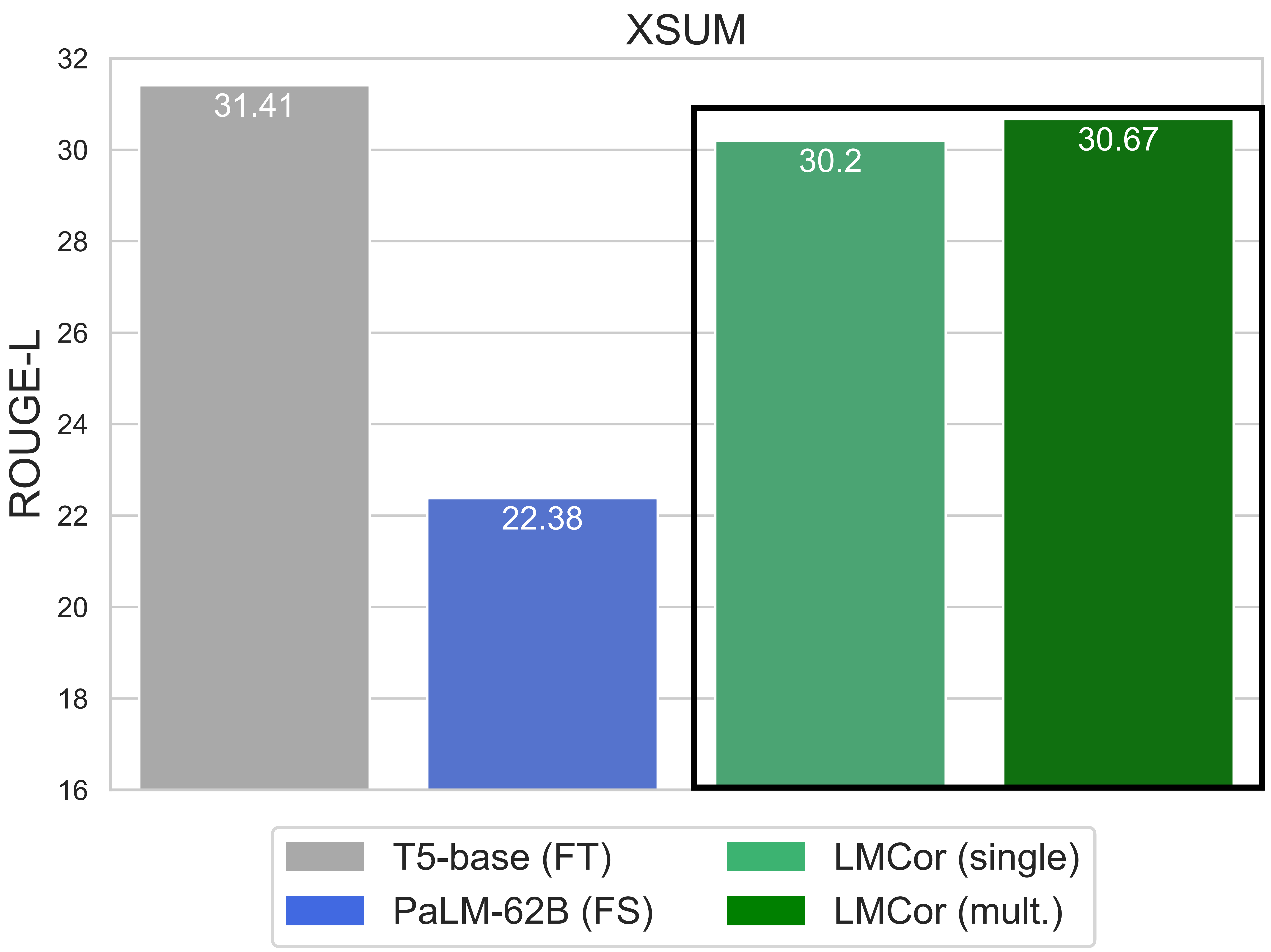
Experiments & Results: Summarization



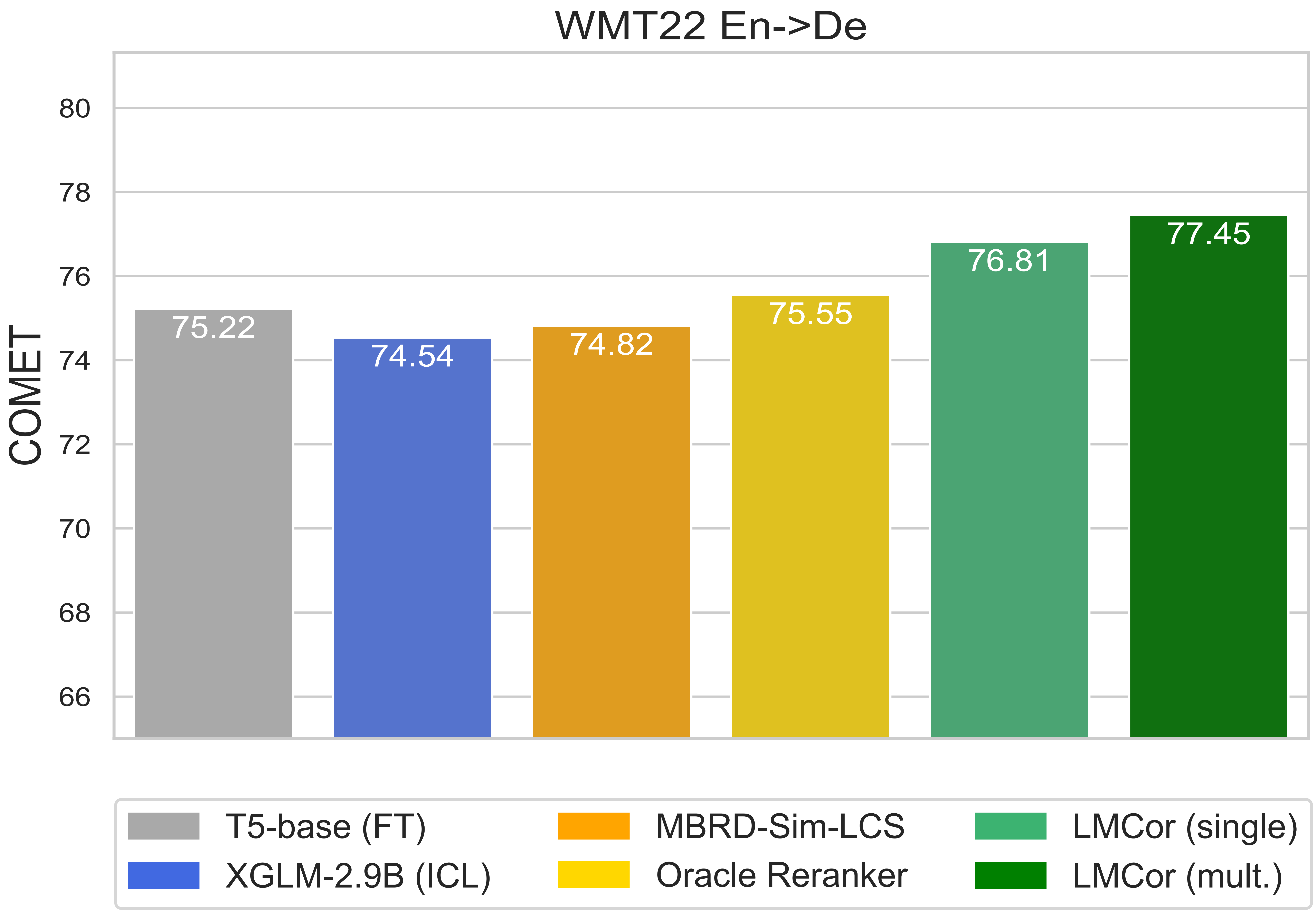
Experiments & Results: Summarization



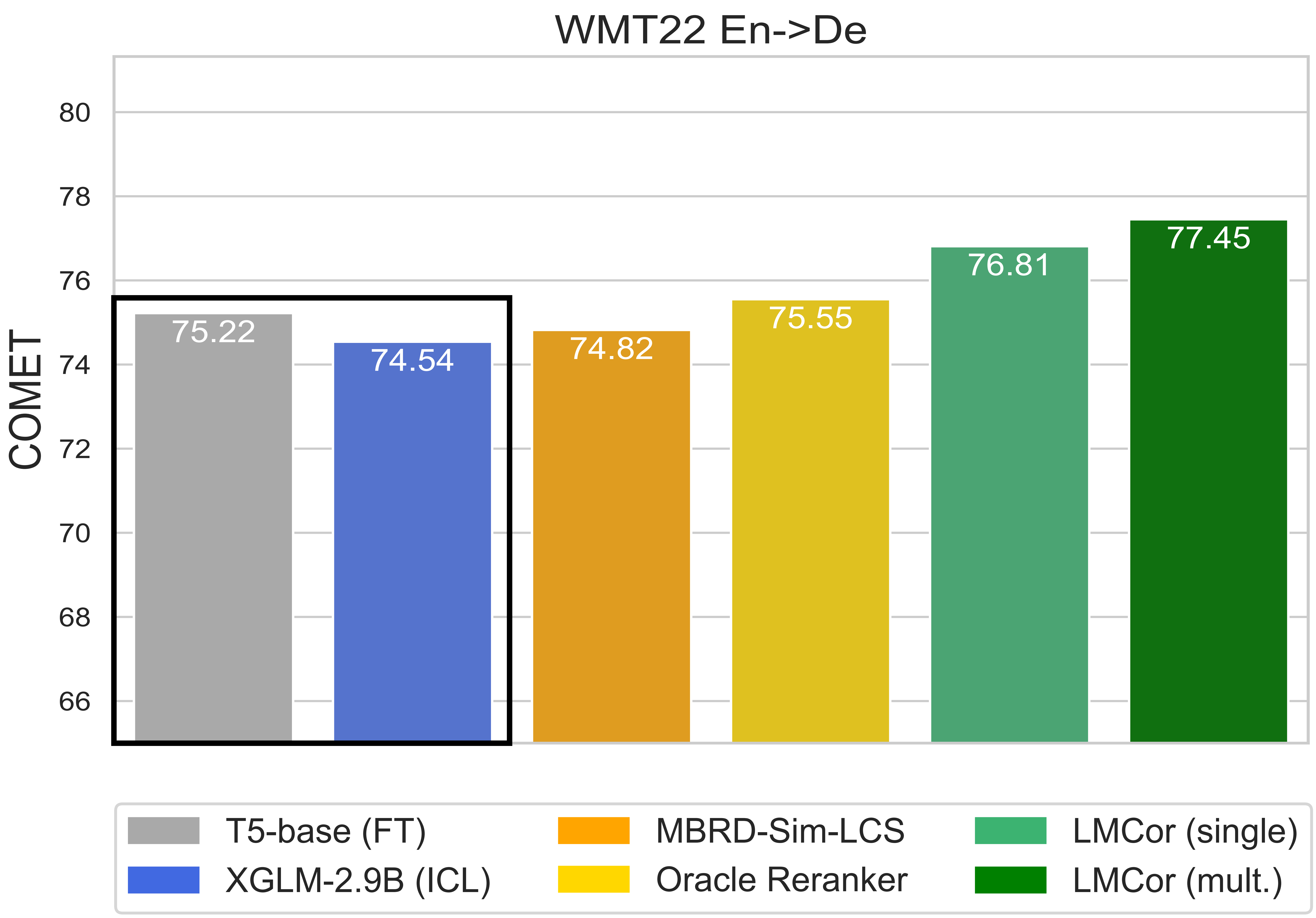
Experiments & Results: Summarization



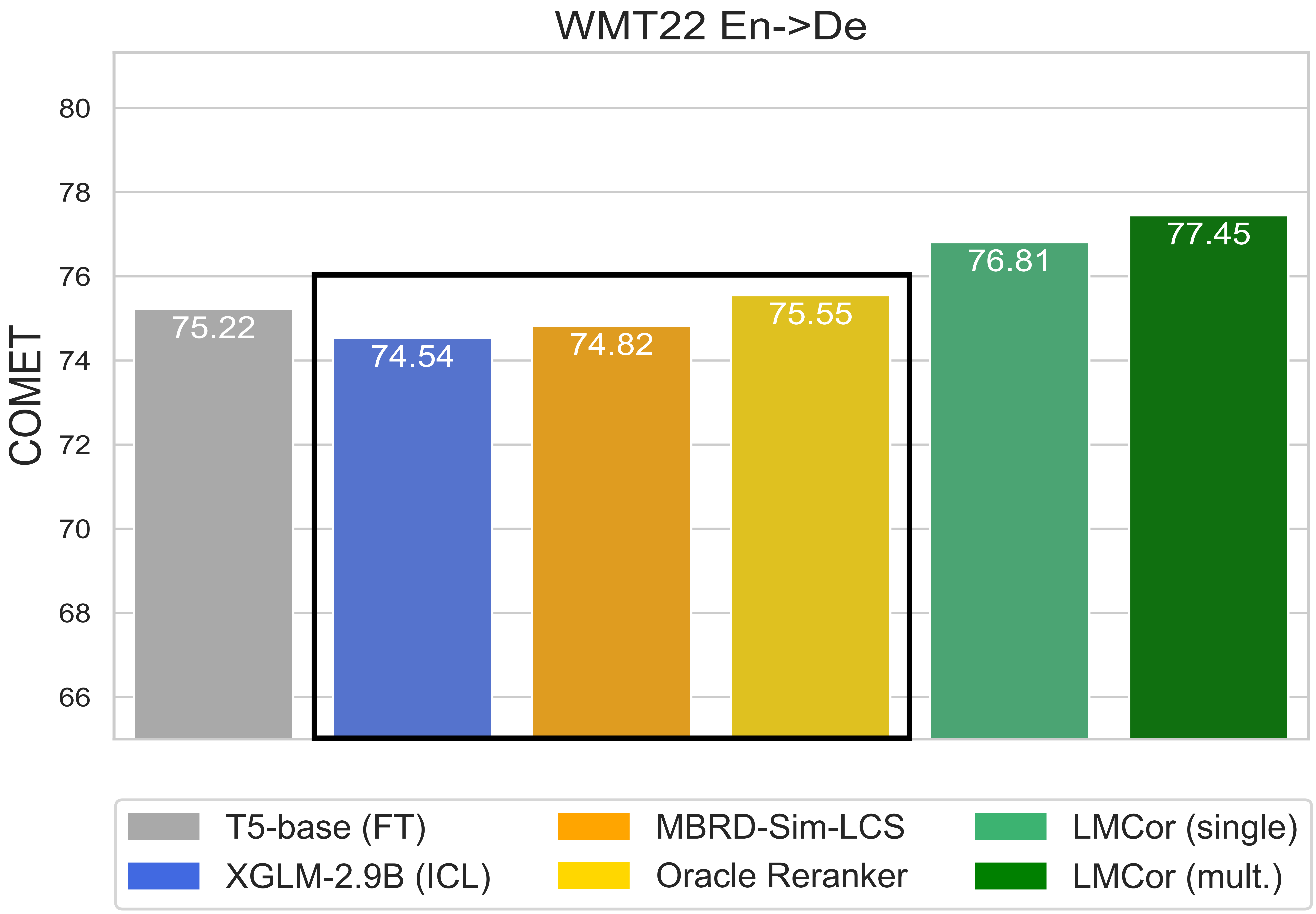
Experiments & Results: Machine Translation



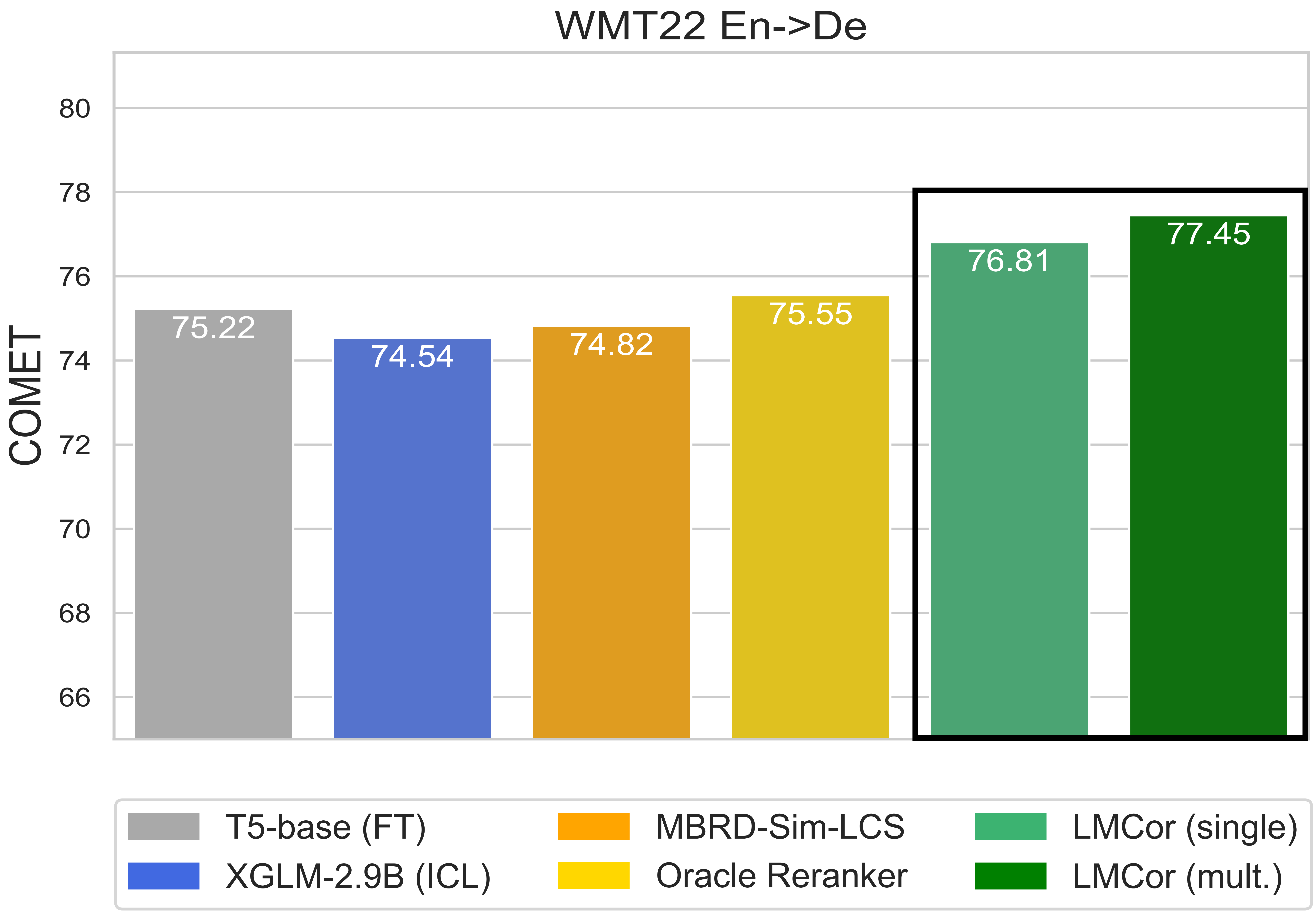
Experiments & Results: Machine Translation



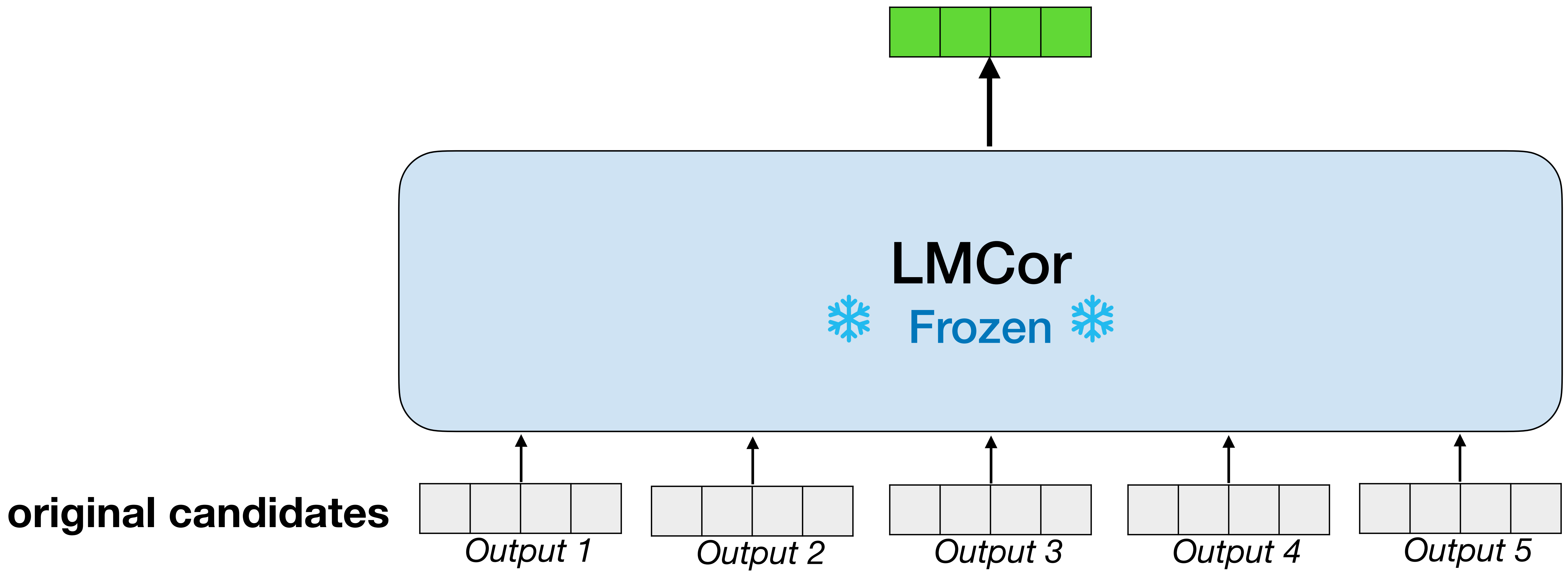
Experiments & Results: Machine Translation



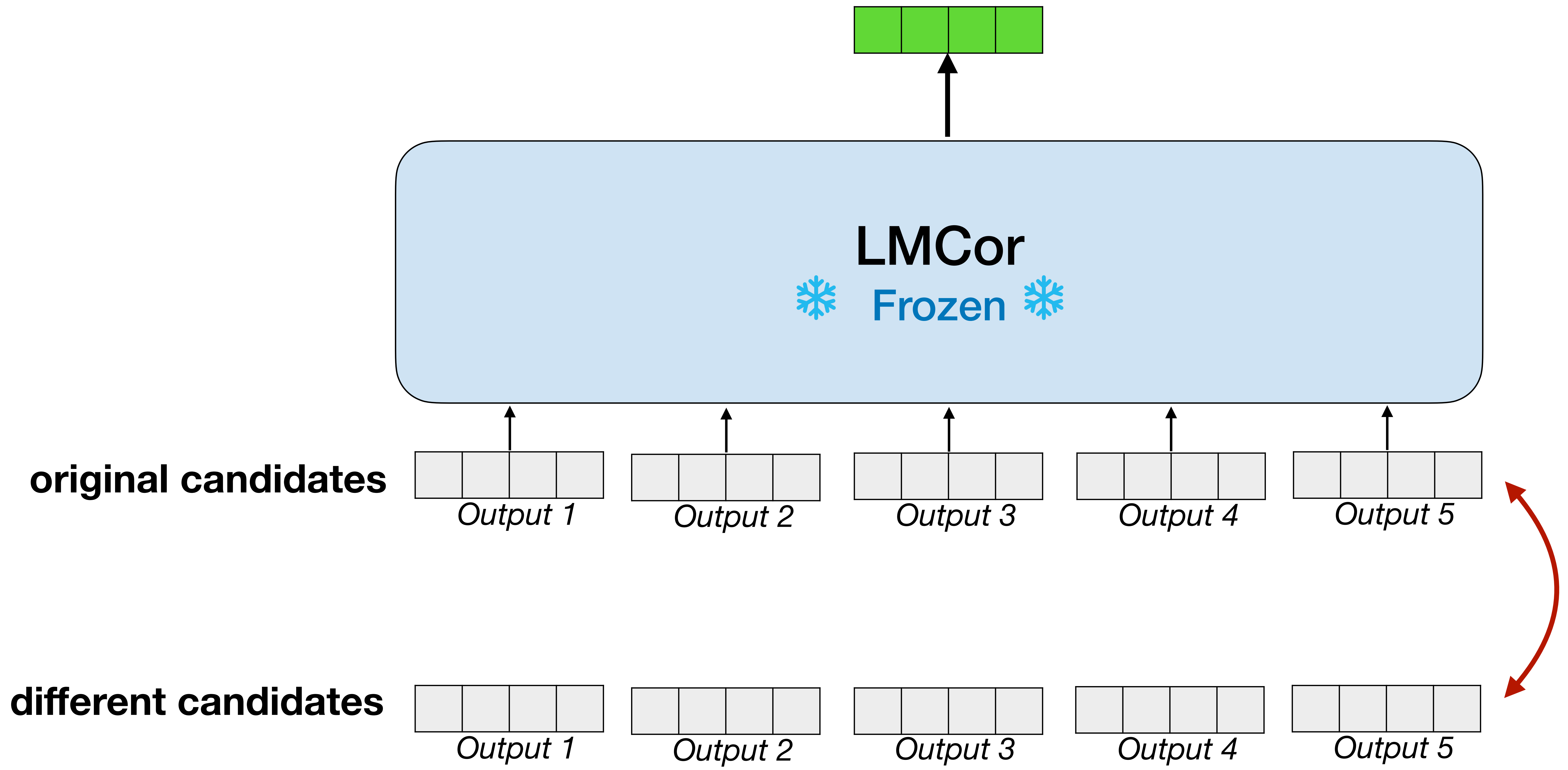
Experiments & Results: Machine Translation



Robustness: Pipeline

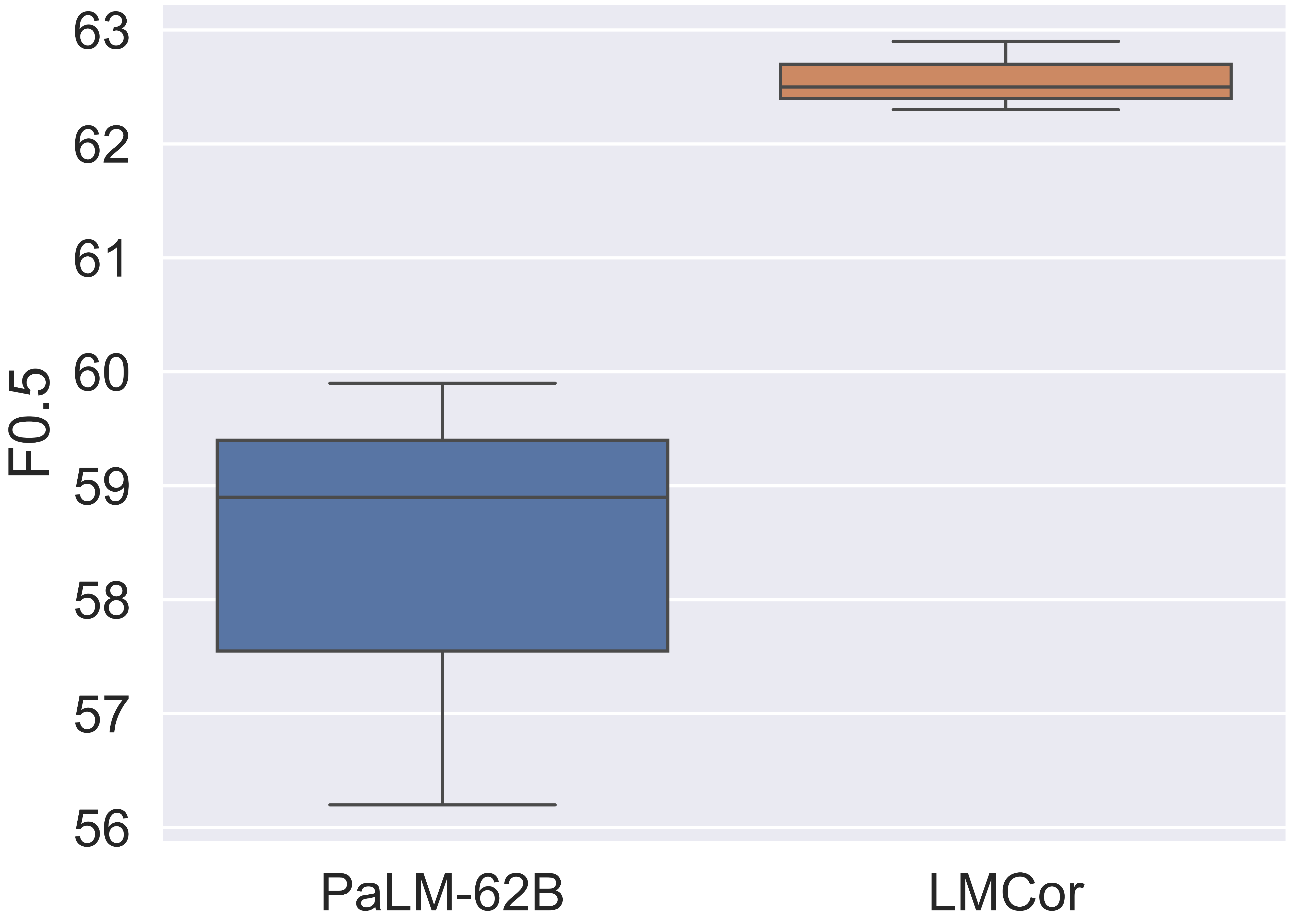


Robustness: Pipeline



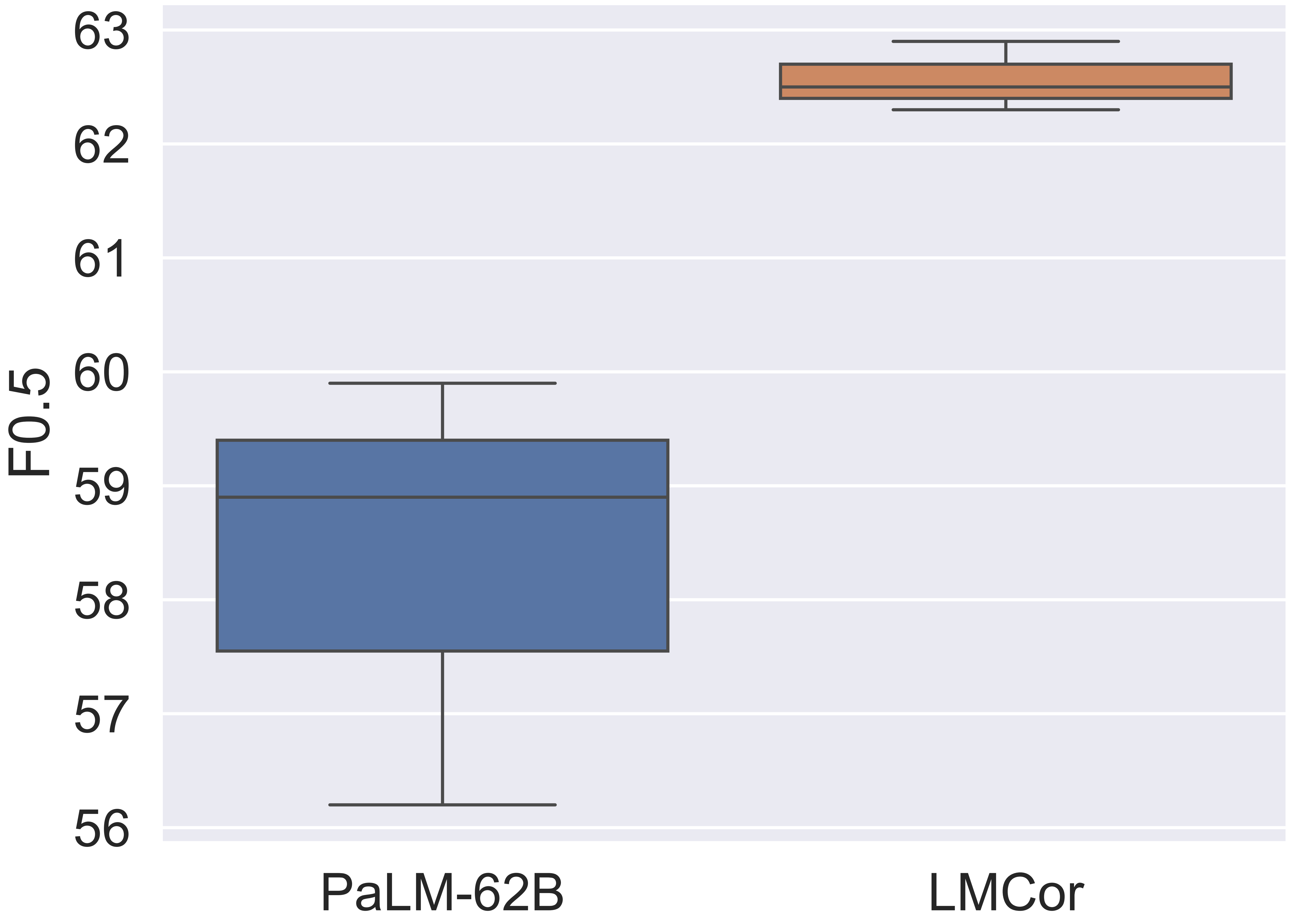
Robustness: Different prompts

Robustness to prompt variability using *three sets of demonstrations* for GEC



Robustness: Different prompts

Robustness to prompt variability using *three sets of demonstrations* for GEC



LMCor mitigates the need for extensive prompt engineering!

Robustness: Different LLMs

Applying the LMCor to different *LLMs* without retraining

same family, different scale

T5-base		59.38	
PaLM (ICL)	*8B*	*62B*	*540B*
	48.62	59.92	**65.37**
+ LMCor (single)	61.40	**62.48**	63.55
+ LMCor (mult.)	**61.89**	**62.47**	65.16

CoNLL-14

Robustness: Different LLMs

Applying the LMCor to different *LLMs* without retraining

same family, different scale

T5-base	59.38		
PaLM (ICL)	*8B*	*62B*	*540B*
	48.62	59.92	**65.37**
+ LMCor (single)	61.40	**62.48**	63.55
+ LMCor (mult.)	**61.89**	**62.47**	65.16

CoNLL-14

different family, different scale

Model	**R-2**	**R-L**
GPT3-Codex (ICL)*	34.2	44.4
+ MBRD-BLEURT*	36.4	46.5
+ LMCor (mult.)	**44.8**	**53.0**

E2E NLG

Robustness: Different LLMs

Applying the LMCor to different *LLMs* without retraining

same family, different scale

T5-base	59.38		
PaLM (ICL)	*8B*	*62B*	*540B*
	48.62	59.92	**65.37**
+ LMCor (single)	61.40	**62.48**	63.55
+ LMCor (mult.)	**61.89**	**62.47**	65.16

CoNLL-14

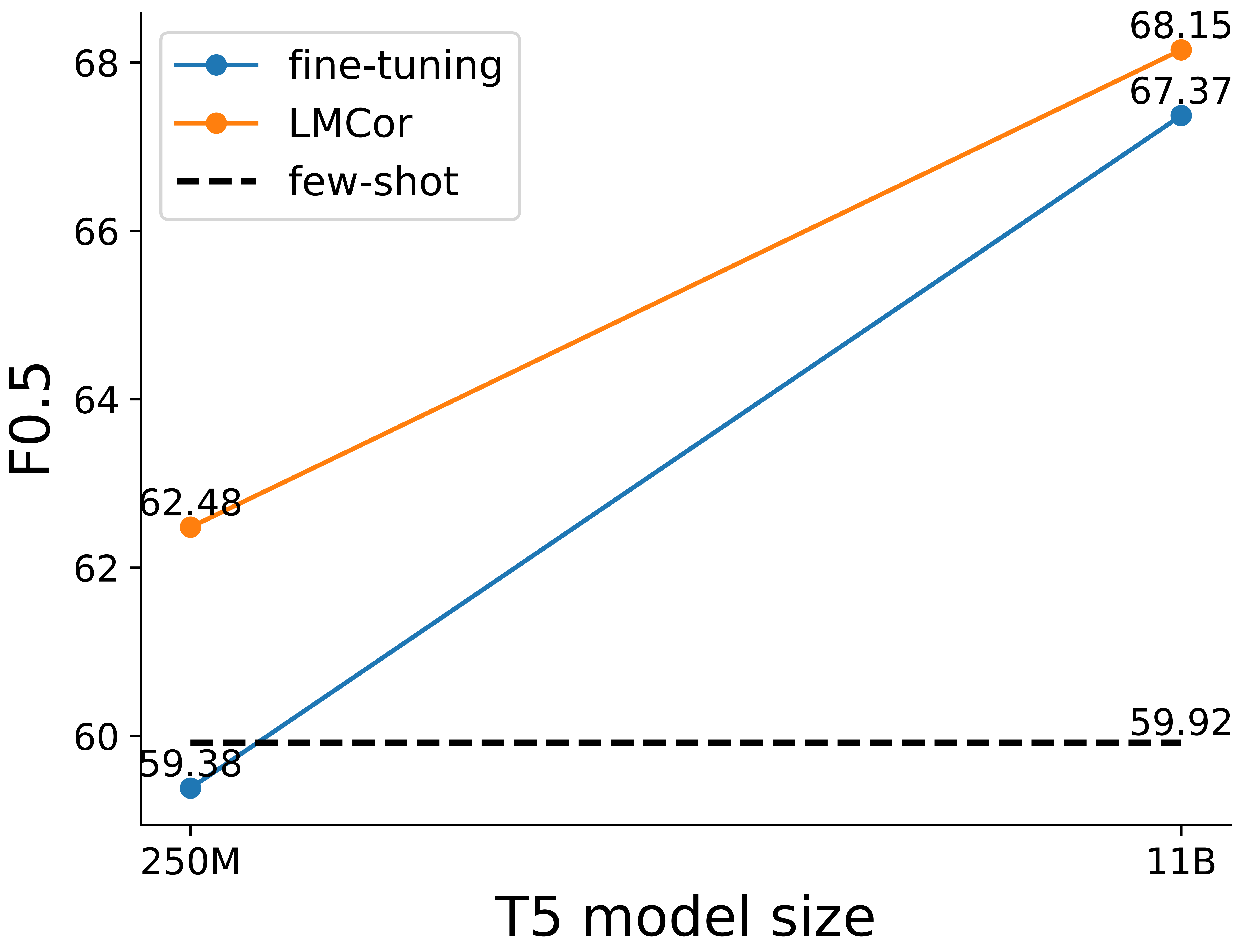
different family, different scale

Model	**R-2**	**R-L**
GPT3-Codex (ICL)*	34.2	44.4
+ MBRD-BLEURT*	36.4	46.5
+ LMCor (mult.)	**44.8**	**53.0**

E2E NLG

LMCor seamlessly integrates with various LLMs!

Analysis: Scaling the corrector



Conclusion

LMCor:

- a <u>compact</u> model that <u>improves</u> the performance of LLMs on specific tasks by correcting their outputs, <u>without access to their weights</u>
- <u>multiple candidates</u> improve task performance and robustness
- a small LMCor can improve the outputs of an LLM <u>x250</u> its size
- can be used as a plug-and-play module for <u>different LLMs</u>

Paper: https://arxiv.org/abs/2305.13514

Code: https://github.com/GeorgeVern/lmcor

Thank you!

@gvernikos

https://georgevern.github.io/

Additional Results: Data-to-text generation

E2E NLG

Model	**R-2**	**R-L**
T5-base	45.3	52.8
PaLM-62B* (FT)	45.2	–
PaLM-540B* (FT)	<u>45.3</u>	52.3
PaLM-62B (ICL)	35.1	45.6
+ MBRD-Sim-LCS	35.7	46.2
+ Oracle Reranker	37.1	50.4
+ LMCOR (single)	44.8	<u>52.8</u>
+ LMCOR (mult.)	**45.6**	**53.4**

Additional Results: Summarisation

XSum

Model	**R-1**	**R-2**	**R-L**
T5-base	**38.64**	16.98	31.41
PaLM-62B* (FT)	–	18.5	–
PaLM-540B* (FT)	–	**21.2**	**36.5**
PaLM-62B (ICL)	28.18	10.50	22.38
PaLM-540B (ICL)	29.88	11.75	23.83
+ LMCOR (single)	36.98	16.41	30.20
+ LMCOR (mult.)	<u>37.62</u>	<u>16.50</u>	<u>30.67</u>

Additional Results: Machine Translation

WMT22 En->De

Model	BLEU	COMET	BLEURT
T5-base	23.32	75.22	64.57
XGLM-2.9B (ICL)	17.32	74.54	66.47
+ MBRD-Sim-CLS	18.01	74.82	66.73
+ Oracle Reranker	21.21	75.55	66.90
+ LMCOR (single)	<u>24.51</u>	<u>76.81</u>	<u>67.23</u>
+ LMCOR (mult.)	**25.15**	**77.45**	**68.41**

Analysis: Correcting task-specific models

XSum

Model	**R-1**	**R-2**	**R-L**	**BLEU**
Pegasus (FT)	45.48	**23.88**	38.18	16.72
+ LMCOR	**45.76**	23.78	**38.28**	**17.00**

Analysis: Importance of the source

E2E NLG

Model	**R-2**	**R-L**
PaLM-62B (ICL)	35.1	45.6
+ LMCor	**45.6**	**53.4**
- source sentence	44.5	53.1