LLM-BLENDER: Ensembling Large Language Models with Pairwise Ranking and Generative Fusion

ACL 2023

Dongfu Jiang, Xiang Ren, Bill Yuchen Lin

Allen Institute for Artificial Intelligence,

University of Southern California, Zhejiang University

Introduction

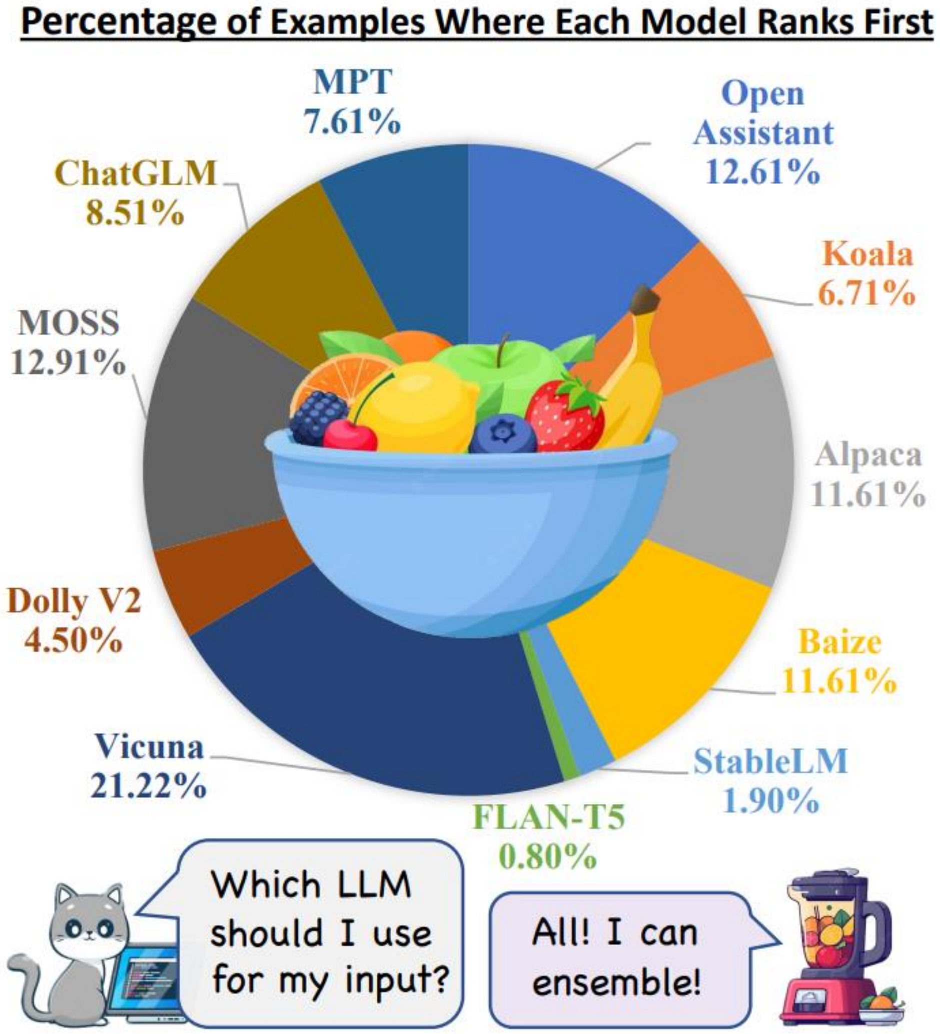
- **LLMs show a promising future for AGI**
  - Impressive performance in diverse tasks
  - Follow instructions, Access extensive, High-quality data

  - Closed-source LLMs
    - GPT-4, PaLM
    - Restricting insights into their architectures and training data

  - Open-source LLMs
    - Pythia, LLaMA, Flan-T5: Fine-tune models on custom instruction datasets
    - Alpaca, Vicuna, OpenAssistant, MPT: Smaller yet efficient LLMs

Introduction

• Motivation of ensembling LLMs



Open-source LLMs exhibit diverse strengths & weaknesses

- Optimal LLMs for different examples can significantly vary
- Variations in data, architectures, and hyperparameters

← **Pie Graph :** Distribution of best LLMs on 5,000 instructions that we collected

Combine unique contributions

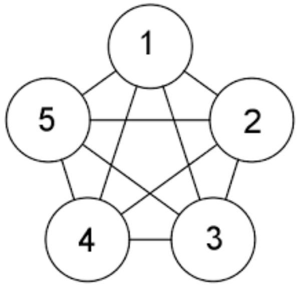
- Alleviate biases, errors, and uncertainties in individual LLMs
- Result in outputs better aligned with human preferences

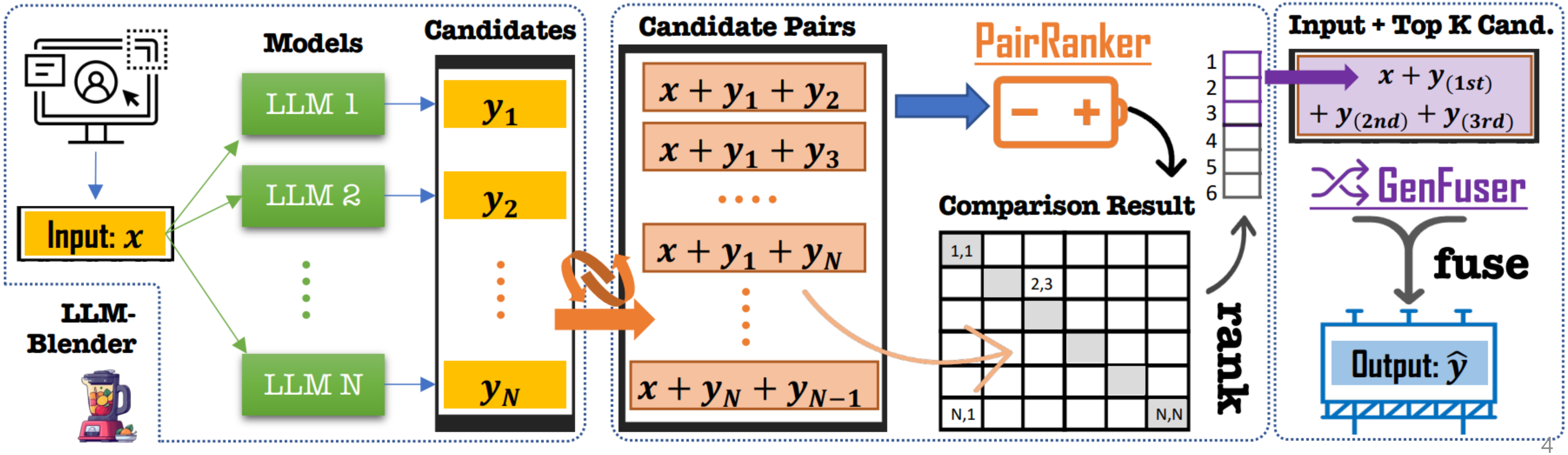
Introduction

- **LLM-BLENDER framework**
  - PairRanker
    - Create the $N(N-1)/2$ pairs of their outputs from N models
    - x and y as input for cross-attention encoder
  - GenFuser
    - Fuse the top K of the N ranked candidates and generate an improved output

e.g. Number of League Games

$N(N-1)/2$





Method

- **Problem Setup**
  - Model $\{\mathcal{M}_1, \ldots, \mathcal{M}_N\}$
  - Candidate output $\mathbb{Y} = \{y_1, \ldots, y_N\}$
  - Produces an output $\hat{y}$ for the input x, maximizing similarity $Q(\hat{y}, y; x)$
  - Maxmize similarity for test set $D_{\text{test}} = \{(x^{(i)}, y^{(i)})\}$

$$\sum_i Q(\hat{y}^{(i)}, y^{(i)}; x^{(i)})$$

  - Primary approaches for ensembling LLMs
    - Selection-based method
    - Generation-based method

Method

• Benchmark Dataset: MixInstruct


```
[{"id":"unified_chip2\/69962",
  "instruction":"",
  "input":"I've always wondered what the difference is between a skeptic and a denier.",
  "output":"A skeptic is someone who questions the validity of something, …",
  "candidates":[
              {"decoding_method":"top_p_sampling",
               "model":"oasst-sft-4-pythia-12b-epoch-3.5",
              "text":"A skeptic is someone who doubts or expresses doubt …",
        "scores":{
            "logprobs":-0.0240402222,
            "rougeL":0.2321428571,
            "rouge2":0.1272727273,
            "rougeLsum":0.2321428571,
            "rouge1":0.2857142857,
            "bleu":5.6561527509,
            "bertscore":0.7549101114,
            "bleurt":0.0506142341,
            "bartscore":-2.887932539
            }, … ],
   "cmp_results": "{\"alpaca-native,chatglm-6b\": \"A is better\",
                    \"alpaca-native,moss-moon-003-sft\": \"Same good\",
                    \"koala-7B-HF,dolly-v2-12b\": \"Same bad\""
```


LLMs

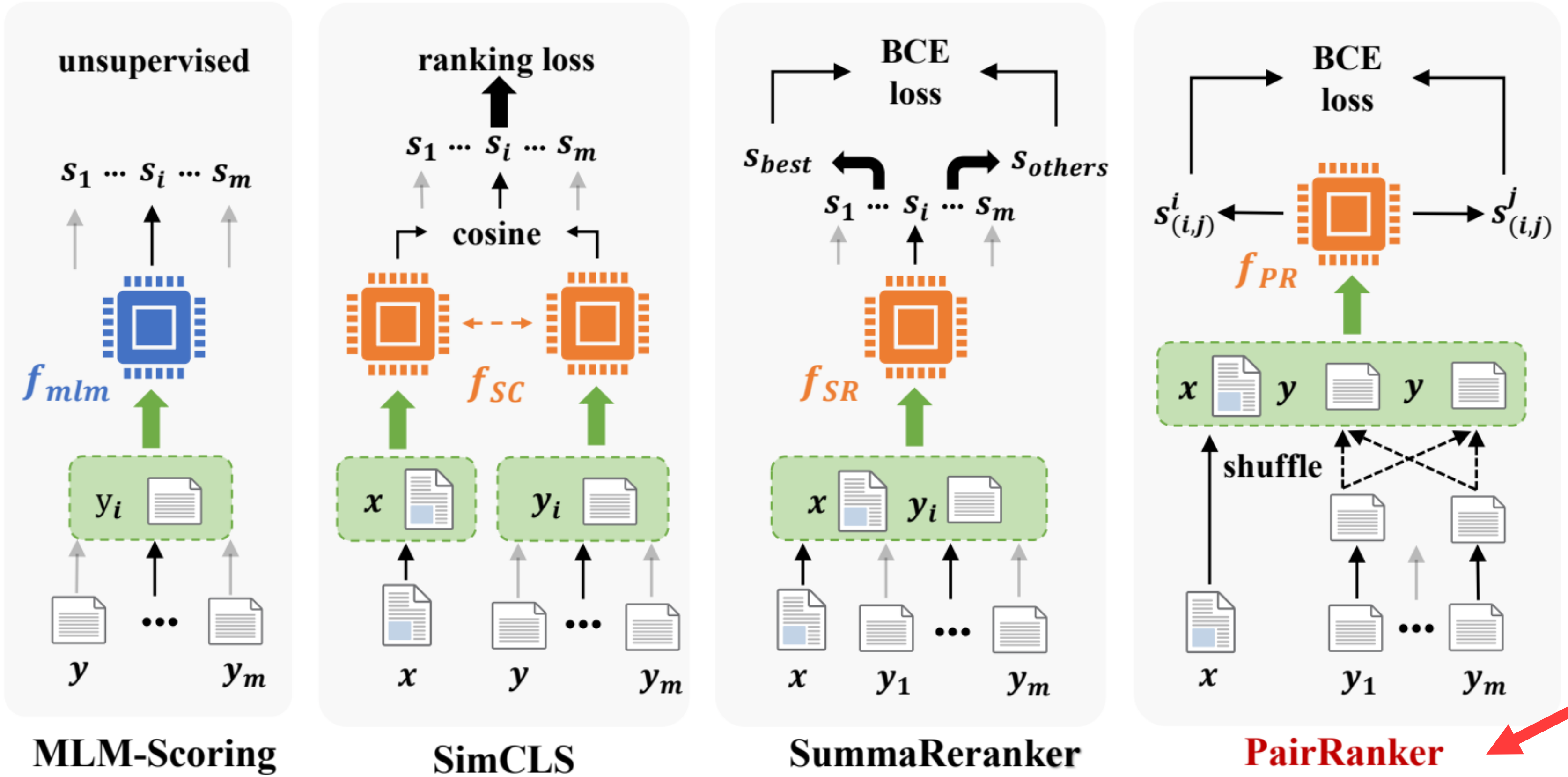
- Stanford Alpaca
- FastChat Vicuna
- Dolly V2, StableLM
- Open Assistant, Koala
- Baize, Flan-T5, ChatGLM
- MOSS, Moasic MPT

Data Source

Sources	#Examples	Source	I/O Tokens
Alpaca-GPT4	22,862	GPT-4	22 / 48
Dolly-15K	7,584	Human	24 / 53
GPT4All-LAION	76,552	ChatGPT	18 / 72
ShareGPT	3,002	ChatGPT	36 / 63
Total	110K	Mix	20 / 66

Method

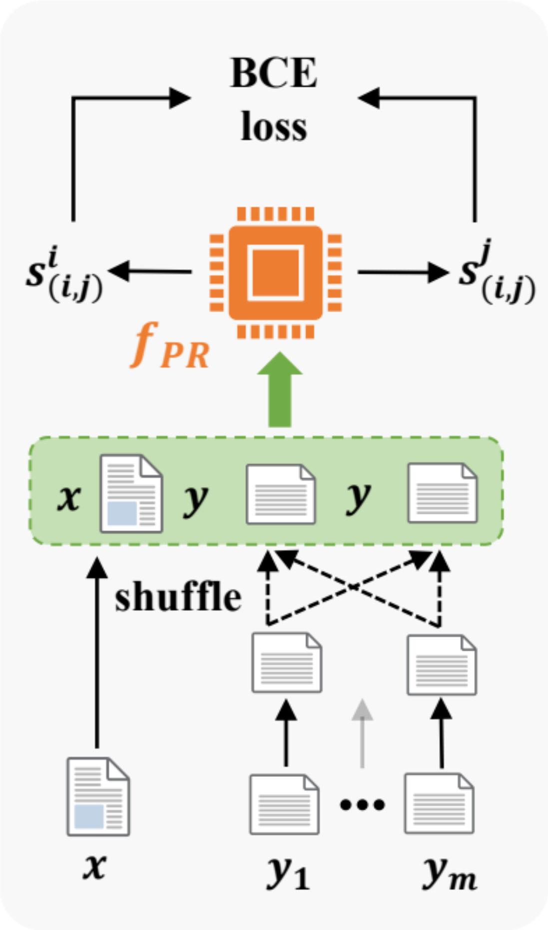
- **Individual Scoring (Pointwise Scoring)**
  - Insufficient for selection in the context of instruction-following tasks
  - Quality of outputs is generally high when LLMs are competitive
    (e.g, a few different words in shorter responses vary significantly in harmfulness)



Pairwise Scoring

Method

• Pairwise Scoring



PairRanker

PairRanker

- Create the $N(N-1)/2$ pairs of their outputs
- x and y as input for cross-attention encoder
- Model's confidence in thinking y_i is better than y_j $\quad s_{ij} = s^i_{(i,j)} - s^j_{(i,j)}$
- Sigmoid Function: σ $\quad \mathcal{L}_Q = -z_i \log \sigma(s^i_{(i,j)}) - (1 - z_j) \log \sigma(s^j_{(i,j)})$
- Multiple Q functions to optimize (e.g., BERTScore, BARTScore)

$$(z_i, z_j) = \begin{cases} (1,0), & Q(y_i, y) \geq Q(y_j, y) \\ (0,1), & Q(y_i, y) < Q(y_j, y) \end{cases}$$

- Take the average as the final multi-objective loss $\mathcal{L} = \sum \mathcal{L}_Q$

Method

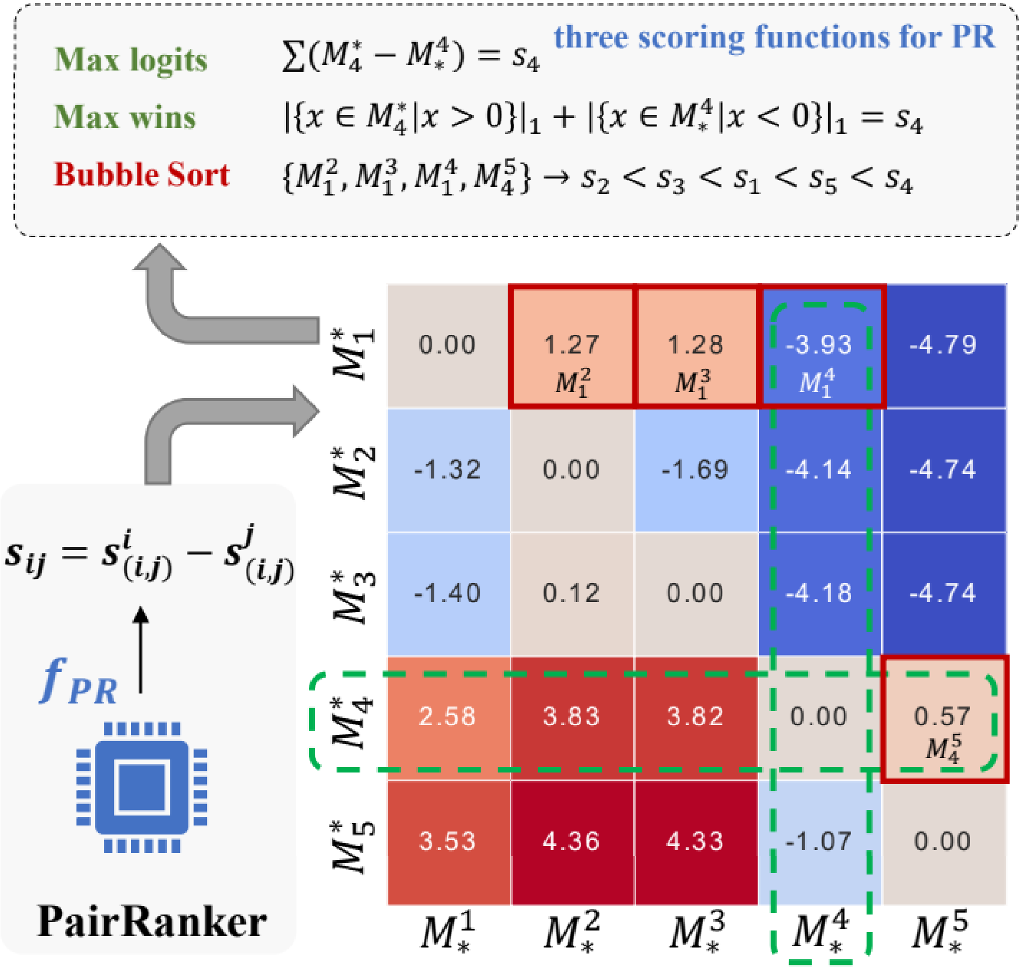
- **PairRanker Architecture: Embedding, Training**
  - Embedding
    - Concatenate segments sequentially with special tokens as separators

<s><source> x </s> <candidate1> yi </s> <candidate2> yj </s>

  - Training
    - Pass concatenated embeddings through a single-head layer
    - MLP with the final layer's each dimension represents a computed Q score
    - Instead of all the $N(N-1)/2$ pairs
      Randomly select some combinations from the candidate pool $\mathbb{Y}$
    - Shuffle the order of candidates within each training pair (x, y_i, y_j) and (x, y_j, y_i)

Method

• PairRanker Architecture: Inference



	M_*^1	M_*^2	M_*^3	M_*^4	M_*^5
M_1^*	0.00	1.27 M_1^2	1.28 M_1^3	-3.93 M_1^4	-4.79
M_2^*	-1.32	0.00	-1.69	-4.14	-4.74
M_3^*	-1.40	0.12	0.00	-4.18	-4.74
M_4^*	2.58	3.83	3.82	0.00	0.57 M_4^5
M_5^*	3.53	4.36	4.33	-1.07	0.00

Scoring Function

- MaxLogits:
  Sum(Horizontal rectangle) - Sum(Vertical rectangle)
- MaxWins:
  Count(Positive in horizontal rectangle) –
  Count(Negative in vertical rectangle)

- MaxLogits yields the best performance
- MaxLogits as the default aggregator for PairRanker

Bubble Sort

- $O(N^2)$ iterations for N candidates can be burdensome
- $N-1$ Comparisons
- Reduce the inference time complexity from $O(N^2)$ to $O(N)$

Method

- **GENFUSER**
  - Effectiveness of PairRanker is constrained from the candidate pool
    - Merging multiple top-ranked candidates
    - Generate a superior response by combining advantages while mitigating shortcomings

  - Overcome complementary strengths and weaknesses
    - Fuse the top K of the N ranked candidates and generate an improved output
    - Use separator tokens, such as <extra_id_i>
    - Fine-tune a Flan-T5-XL model to learn to generate

Experiment

- **Evaluation**
  - DeBERTa (400M) as backbone for PairRanker / GenFuser is based on Flan-T5-XL (3B)
  - e.g., Koala approximately 40% of examples' quality is as good as both OA and Vic

Category	Methods	BERTScore↑	BARTScore↑	BLEURT↑	GPT-Rank↓	≥ Vic(%)↑	≥ OA(%)↑	Top-3(%)↑
LLMs	Open Assistant (LAION-AI, 2023)	**74.68**	-3.45	**-0.39**	**3.90**	**62.78**	N/A	51.98
	Vicuna (Chiang et al., 2023)	69.60	**-3.44**	-0.61	4.13	N/A	**64.77**	**52.88**
	Alpaca (Taori et al., 2023)	71.46	-3.57	-0.53	4.62	56.70	61.35	44.46
	Baize (Xu et al., 2023)	65.57	-3.53	-0.66	4.86	52.76	56.40	38.80
	MOSS (Sun and Qiu, 2023)	64.85	-3.65	-0.73	5.09	51.62	51.79	38.27
	ChatGLM (Du et al., 2022)	70.38	-3.52	-0.62	5.63	44.04	45.67	28.78
	Koala (Geng et al., 2023)	63.96	-3.85	-0.84	6.76	39.93	39.01	22.55
	Dolly V2 (Conover et al., 2023)	62.26	-3.83	-0.87	6.90	33.33	31.44	16.45
	Mosaic MPT (MosaicML, 2023)	63.21	-3.72	-0.82	7.19	30.87	30.16	16.24
	StableLM (Stability-AI, 2023)	62.47	-4.12	-0.98	8.71	21.55	19.87	7.96
	Flan-T5 (Chung et al., 2022)	64.92	-4.57	-1.23	8.81	23.89	19.93	5.32
Analysis	Oracle (BERTScore)	**77.67**	-3.17	-0.27	3.88	54.41	38.84	53.49
	Oracle (BLEURT)	75.02	-3.15	**-0.15**	3.77	55.61	45.80	55.36
	Oracle (BARTScore)	73.23	**-2.87**	-0.38	3.69	50.32	57.01	57.33
	Oracle (GPT-Rank)	70.32	-3.33	-0.51	**1.00**	**100.00**	**100.00**	**100.00**
Rankers	Random	66.36	-3.76	-0.77	6.14	37.75	36.91	29.05
	MLM-Scoring	64.77	-4.03	-0.88	7.00	33.87	30.39	21.46
	SimCLS	**73.14**	-3.22	-0.38	3.50	52.11	49.93	60.72
	SummaReranker	71.60	-3.25	-0.41	3.66	**55.63**	48.46	57.54
	PairRanker	72.97	**-3.14**	**-0.37**	**3.20**	54.76	**57.79**	**65.12**
LLM-BLENDER	**PR (K = 3) + GF**	**79.09**	**-3.02**	**-0.17**	**3.01**	**70.73**	**77.72**	**68.59**

Experiment

• Ranking correlation with GPT-Rank

Ranking Methods	Pearson Correlation ↑	Spearman's Correlation ↑	Spearman's Footrule ↓
Random	0.00	0.00	48.27
BLEU	28.70	26.92	33.57
Rouge2	29.17	27.77	32.96
BERTScore	32.25	30.33	33.34
BLEURT	34.14	32.31	32.17
BARTScore	**38.49**	**36.76**	**30.93**
MLM-Scoring	-0.02	-0.01	47.16
SimCLS	39.89	38.13	29.32
SummaReranker	41.13	39.10	29.69
PairRanker	**46.98**	**44.98**	**27.52**

Evaluation Metric

- Bartscore gets the highest correlation with GPT-Rank

Pointwise Ranking & Pairwise Ranking

- MLM-Scoring still underperforms random permutations
- SummaReranker: Pearson Correlation (41.13), Spearman's Correlation (39.10)
- SimCLS: Spearman's Footrule distance (29.32)
- PairRanker achieves the highest correlation with GPT-Rank

Conclusion

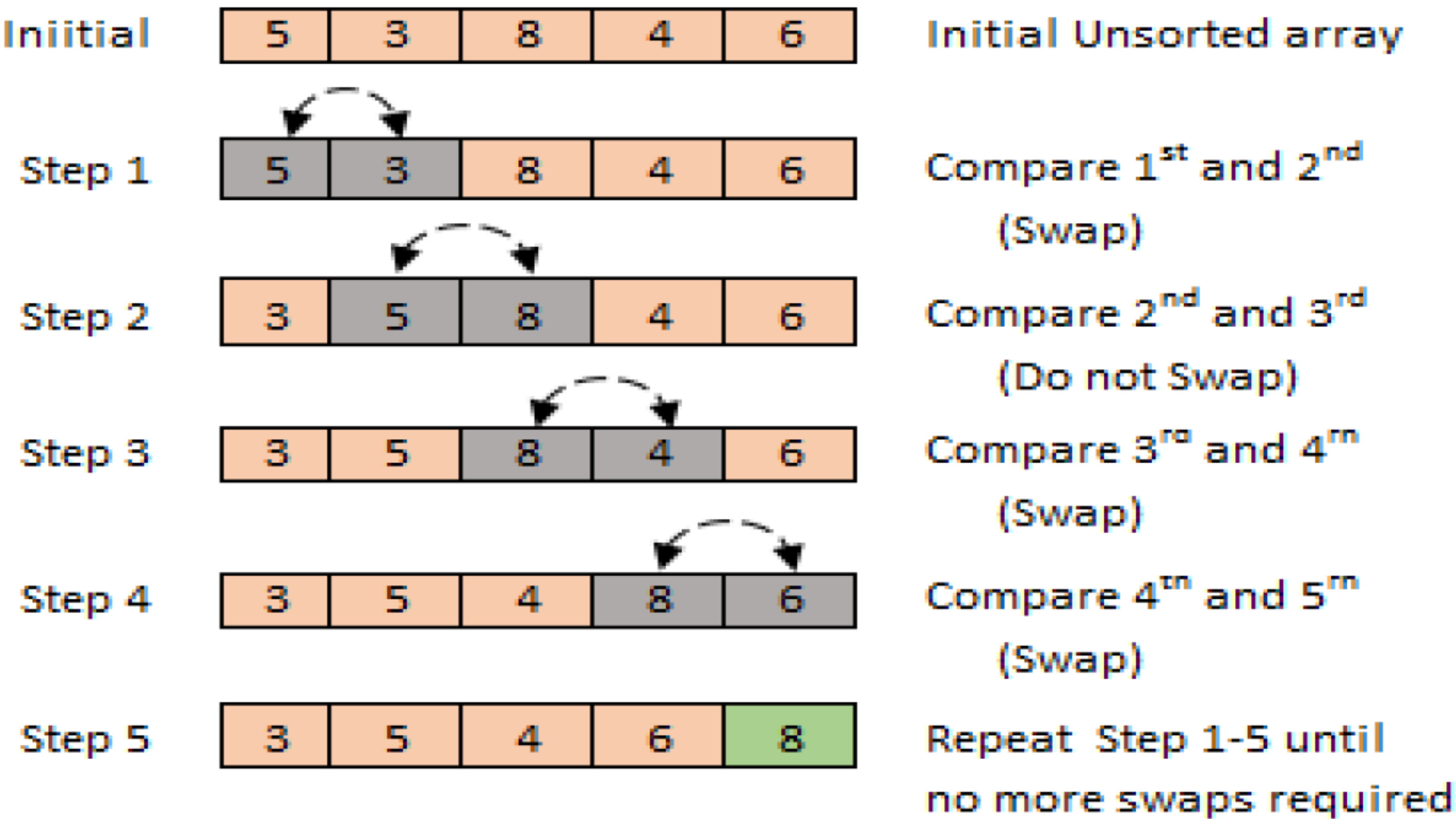
- **LLM-BLENDER**
  - Post-hoc LLM ensemble learning method
    - Ranking and fusing the outputs from multiple LLMs: PAIRRANKER & GENFUSER
    - Improve the overall results on various metrics
  - MixInstruct
    - Created benchmark dataset for evaluating ensembling methods
  - Toolkit
    - By open-sourcing our framework, aim to make it easier for others
    - Robustness, generalization, and enhanced accuracy in a wide variety of tasks
  - Future directions
    - For rapid adaptation to new specialized domains and data sources
    - Investigating the transferability of our ensembling approach to other domains and tasks

Conclusion

- **Limitations**
  - Less Efficiency
    - To get the optimal performance from PAIRRANKER
    - One may need to call the model $O(N^2)$ times for getting the full matrix
  - Human evaluation VS ChatGPT evaluation
    - Human evaluation is more reliable and comprehensive
    - We cannot afford large-scale human evaluation
    - Argue that our use of ChatGPT for evaluation is a good alternative

Appendix

- **Bubble Sort**



Dipak Kandel. Bubble Sort Algorithm | Algorithm | Flowchart | Data Structures. Researchgate. 2018