

# ARES

An Automated Evaluation Framework for Retrieval-Augmented Generation Systems

**Presentation by:**

Maja Andrzejczuk, Piotr Bielecki, Jakub Kasprzak, Maciej Orłowski

# About the paper

Link: <https://aclanthology.org/2024.naacl-long.20/>

Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers).

Authors:

- Jon Saad-Falcon (Stanford University),
- Christopher Potts (Stanford University),
- Omar Khattab (Stanford University),
- Matei Zaharia (Databricks & UC Berkeley)




# Problems with RAG evaluation

- The process of RAG evaluation often relies on manual human work of domain experts;
- This is time-consuming and expensive;
- There are also less expensive model-based methods, but they badly adapt to different evaluation contexts;
- Hence the need for an automated, resource-efficient method.



# Related works

The authors list several other frameworks used to evaluate RAGs:

- **AutoCalibrate** – uses an LLM judge to align to human preferences, but does not provide statistical guarantees for the accuracy of evaluations;
  - **EXAM** – evaluates using exam questions. This requires substantial human input data;
  - **RAGAS** – the most competitive and does not require as much human input data, but struggles with cross-domain applications.
- 

# ARES – bird's-eye view

- ARES stands for an **A**utomated **RAG** **E**valuation **S**ystem;
- It generates LLM judges which help evaluate different RAG components;
- Ares evaluates the following:
  - context relevance
  - answer faithfulness
  - answer relevance
- Compared to existing methods, not only is it more efficient, but it also provides a more accurate evaluation;
- The efficiency gains are especially significant when evaluating multiple RAG systems with ARES.



# ARES – bird's-eye view

For an input, ARES requires the following:

- In-domain passage set;
- Human preference validated set of ~150 data points;
- Few-shot (five or more) examples of in-domain queries and answers.



# ARES – bird's-eye view



**Step #1: LLM Generation of Synthetic Dataset:** Generate synthetic queries and answers from in-domain passages



**Step #2: Preparing LLM Judges:**  
Train LLM judges with synthetic data



**Step #3: Ranking RAG Systems with Confidence Intervals:** Use LLM judges to evaluate RAG systems with PPI + human labels

Source: the paper



# Stage 1: Synthetic dataset

- LLM generates queries and answers, based on passages;
- It utilises the few-shot examples provided by the user;
- This results in query-passage-answer triples;
- Low-quality generated samples are filtered out;
- Both positive and negative examples are generated, in the same amount.





# Negative samples generation

The authors use two strategies for generating negative samples:

1. Weak negative generation
2. Strong negative generation

Half of all negative samples are generated with the first method, and half with the other.



# Weak negative generation

For context relevance negatives:

- A synthetic query is generated based on some passage, and then unrelated passages are randomly sampled.

For answer relevance/faithfulness negatives:

- A synthetic query is generated based on some passage, and then answers from other queries are randomly sampled.



# Strong negative generation

For context relevance negatives:

- A synthetic query is generated based on some passage, and then other in-domain passages **from the same document** are randomly sampled.

For answer relevance/faithfulness negatives:

- A synthetic query is generated based on some passage, and then the model purposely generates an answer that contradicts the correct one.



## Stage 2: LLM judges

- Synthetic data generated in stage 1 is used to fine-tune LLM judges;
- There are three separate judges to evaluate the three metrics:
  - context relevance,
  - answer faithfulness,
  - answer relevance;
- Each judge acts as a binary classifier for its metric;
- Human preference validation set is also used to fine-tune them.



## Stage 3: Evaluation

- Query-document-answer triples are sampled from RAG systems we wish to evaluate
- They are then evaluated by the three LLM judges trained in stage 2;
- However, the final score is not just the accuracy in terms of the metrics described above, as it would be based on unlabelled data with predictions made by LMM judges trained on synthetic data;
- Hence the need for an alternative approach.



## Stage 3: Evaluation – PPI Scores

- PPI stands for prediction-powered inference;
- It produces confidence intervals for the accuracy metrics;
- Labelled data (human preference) is used to construct confidence intervals;
- Unlabelled data is used to tighten the confidence intervals;
- The authors use  $\alpha = 0.05$  for confidence intervals.



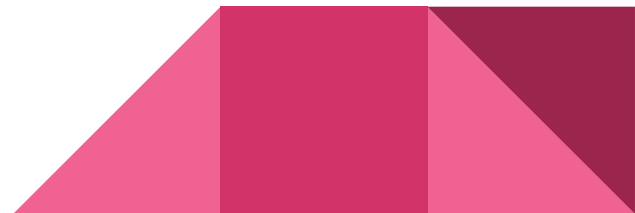
<https://www.youtube.com/watch?v=9P6JRvA5B9o>

# Experiments

# Experiment 1: Setup


Datasets from KILT and SuperGLUE benchmarks are used:

- Natural Questions (KILT),
- HotpotQA (KILT),
- FEVER (KILT),
- Wizards of Wikipedia (KILT),
- MultiRC (SuperGLUE),
- ReCoRD (SuperGLUE).





# Experiment 1: Mocked RAG systems

- RAG systems to evaluate are mocked using the datasets;
  - Negative samples are generated using, among others, unrelated Wikipedia articles;
  - Nine mocked RAG systems were created like this, with accuracies ranging from 70% to 90%, with a step of 2.5%;
  - To compare the real accuracy with ARES's estimation, Kendall's tau correlation is used;
  - Results are also compared with other state-of-the-art evaluation systems, such as RAGAS.
- 

# Experiment 1: Mocked RAG systems

	ARES Ranking of Pseudo RAG Systems											
	NQ		HotpotQA		WoW		FEVER		MultiRC		ReCoRD	
	C.R.	A.R.	C.R.	A.R.	C.R.	A.R.	C.R.	A.R.	C.R.	A.R.	C.R.	A.R.
Kendall's Tau for Sampled Annotations	0.83	0.89	0.78	0.78	0.78	0.83	<b>0.89</b>	<b>0.89</b>	0.83	0.83	0.72	0.94
Kendall's Tau for RAGAS	0.89	0.89	<b>0.94</b>	0.89	0.94	0.94	0.72	0.61	0.83	<b>0.94</b>	<b>0.89</b>	0.44
Kendall's Tau for GPT-3.5 Judge	0.89	0.94	0.67	<b>0.94</b>	0.94	0.89	0.78	0.78	0.83	0.89	0.83	<b>0.94</b>
Kendall's Tau for ARES LLM Judge	0.89	<b>1.0</b>	0.89	<b>0.94</b>	0.94	<b>1.0</b>	0.83	0.72	<b>0.94</b>	0.83	0.78	0.83
Kendall's Tau for ARES	<b>0.94</b>	<b>1.0</b>	<b>0.94</b>	<b>0.94</b>	<b>1.0</b>	<b>1.0</b>	<b>0.89</b>	0.78	<b>0.94</b>	0.89	0.83	0.89
RAGAS Accuracy	31.4%	71.2%	17.2%	76.0%	36.4%	77.8%	23.7%	69.2%	16.1%	75.0%	15.0%	72.8%
GPT-3.5 Judge Accuracy	73.8%	95.5%	75.3%	71.6%	84.3%	85.2%	60.4%	59.6%	72.4%	60.3%	81.0%	65.8%
ARES Accuracy	79.3%	97.2%	92.3%	81.3%	85.7%	96.1%	88.4%	78.5%	85.8%	82.7%	67.8%	92.3%

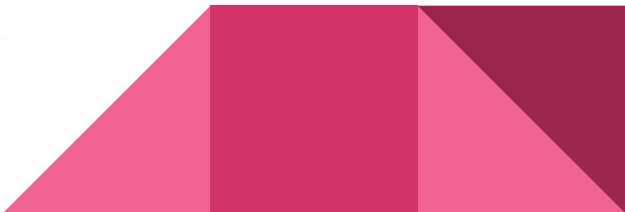
Source: the paper

## Experiment 2: Answer Faithfulness

This was a separate experiment conducted on AIS benchmark on two datasets.

	WoW	CNN / DM
ARES Split Prediction	0.478	0.835
Correct Positive/Negative Split	0.458	0.859
ARES Judge Accuracy	62.5%	84.0%
Evaluation Set Size	707	510
Human Preference Data Size	200	200

Source: the paper



## Experiment 3: Real-life RAG systems

- The authors also wanted to evaluate ARES on real-life systems;
- This experiment was performed on ten different RAG systems;



## Experiment 3: Real-life RAG systems

	ARES Ranking of Real RAG Systems					
	NQ		WoW		FEVER	
	C.R.	A.R.	C.R.	A.R.	C.R.	A.R.
Kendall's Tau for Sampled Annotations	0.73	0.78	0.73	0.73	0.73	0.82
Kendall's Tau for RAGAS	0.82	0.82	0.73	0.82	0.73	0.87
Kendall's Tau for GPT-3.5 Judge	0.82	0.87	0.82	0.82	0.64	0.87
Kendall's Tau for ARES LLM Judge	0.91	<b>0.96</b>	<b>0.91</b>	<b>1.0</b>	0.73	0.87
Kendall's Tau for ARES	<b>1.0</b>	<b>0.96</b>	<b>0.91</b>	<b>1.0</b>	<b>0.82</b>	<b>1.0</b>
RAGAS Accuracy	35.9%	68.2%	44.4%	80.1%	21.4%	75.9%
GPT-3.5 Accuracy	80.5%	91.2%	81.2%	83.5%	61.3%	54.5%
ARES Accuracy	85.6%	93.3%	84.5%	88.2%	70.4%	84.0%

Source: the paper

# Strengths and limitations

Pros	Cons
Works successfully in cross-domain applications	Judges struggle when switching languages
Requires very little human annotations	The annotations it requires need to be performed by domain experts for specialised use cases
Overall better than other SOTA frameworks	Requires substantial computational power to use





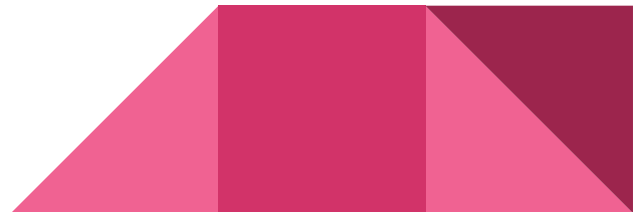
# Extras

# Quiz





# Notebook





Thank you for your attention