HoneyBee: Progressive Instruction Finetuning of Large Language Models for Materials Science

Yeajin Lee

May 8, 2024

AIML@K
KOREA UNIVERSITY

Contents

Background

- **Problem 1**

  : Data scarcity
  in materials science domain

- **Problem 2**

  : Limited presence of specialized
  language models

➡️ **MatSci-Instruct**

: A Two-Step Framework for Trustworthy

Instruction-Data Generation

➡️ **HoneyBee**

: A High-Performance LLaMa-Based Model

Progressively Trained via MatSci-Instruct

Method

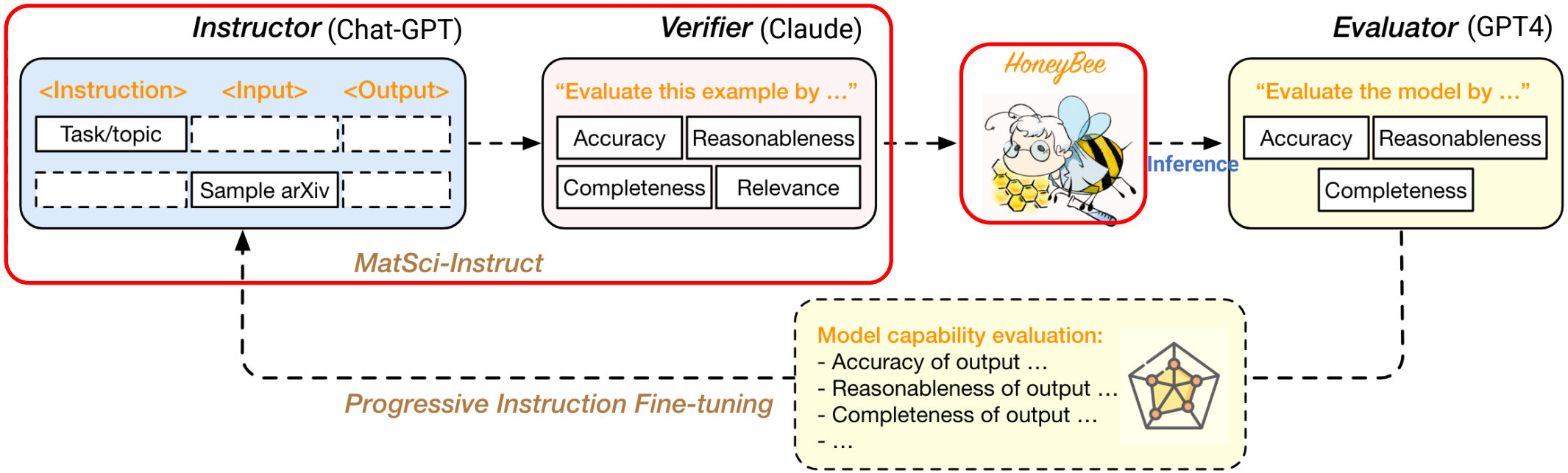
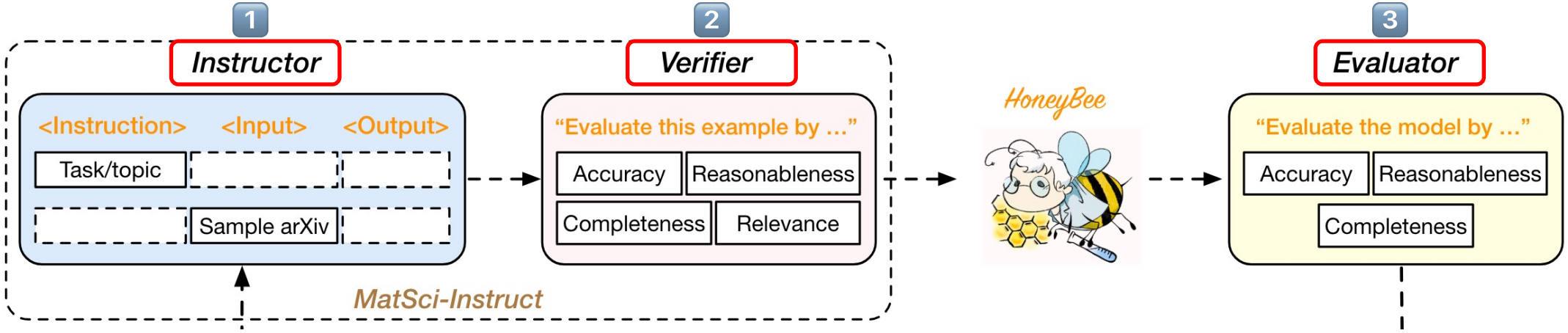


Figure 2: MatSci-Instruct and HoneyBee training workflow.

- Connect **HoneyBee** to **MatSci-Instruct** with a refinement-feedback loop to progressively generate new data and finetune HoneyBee based on its training status

MatSci-Instruct

: An innovative, domain-agnostic methodology that leverages the power of large language models (LLMs) to generate specialized instruction sets for subsequent model finetuning



- The number of instructions gets reduced in later stages of the progressive-refinement-feedback loop mainly due to greater emphasis on quality

Instructor Module

MatSci-Instruct Example

Instruction: How was the ME coupling down to the 2D limit verified in the study?

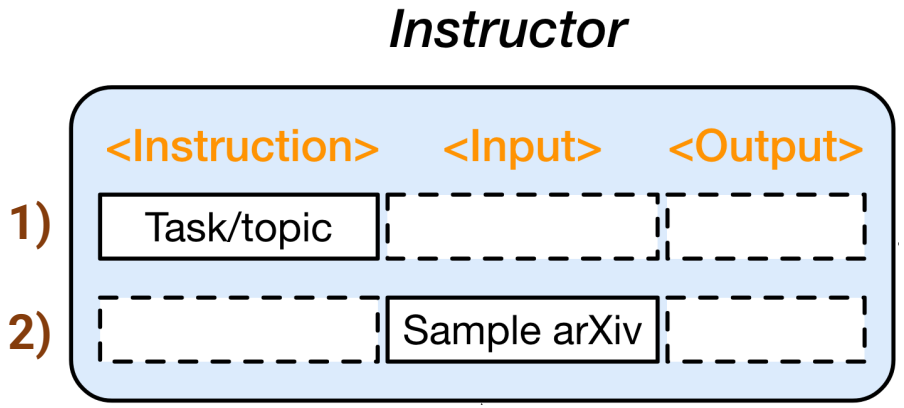
Input: [...] In particular, stripy antiferromagnets often exhibit broken crystal and magnetic symmetries, which may bring about the magnetoelectric (ME) effect and enable the manipulation of intriguing properties and functionalities by electrical means. [...] This work reports the ME effect in the 2D stripy antiferromagnetic insulator CrOCl down to a single layer. By measuring the tunneling resistance of CrOCl on the parameter space of temperature, magnetic field, and applied voltage, we verified the ME coupling down to the 2D limit and unraveled its mechanism. [...]

Output: The ME coupling down to the 2D limit was verified by measuring the tunneling resistance of CrOCl on the parameter space of temperature, magnetic field, and applied voltage in the study.

Figure 1: Example instruction generated by the MatSci-Instruct process

- Embodied by ChatGPT
- Performs the generation of material science instruction-data
- This module employs a concise instruction prompt schema composed of **three elements**
  - **<Instruction>** : the task using a standardized NLP task set
  - **<Input>** : contains the relevant data
  - **<Output>** : generates a pertinent response to the task
- These are manually verified before they are utilized in structured prompts

Instructor Module



1) Open-ended task :

<Input> field remains intentionally blank

➡️ Test generative abilities of the model under uncertainty and promotes more varied outcomes

2) Content-based instruction-data generation :

Select a random open-access paper from the materials science category on arXiv and extract a specific fragment to fill the <Input> field.

➡️ Aligns the generated instruction-data more closely with practical, domain-specific contexts

Verifier Module

- Use **Claude**
- Our evaluation is based on four dimensions:
  - **Accuracy** : evaluated by comparing it with known facts or credible sources
  - **Relevance** : assessed by determining how directly it relates to materials science
  - **Completeness** : to ensure the instruction-data comprehensively address the given task,
    inclusive of all sub-questions
  - **Reasonableness** : about logical consistency
- Identifies any low-quality data that falls below a predetermined threshold.

➡️ **Ensures the use of high-quality data in model fine-tuning**

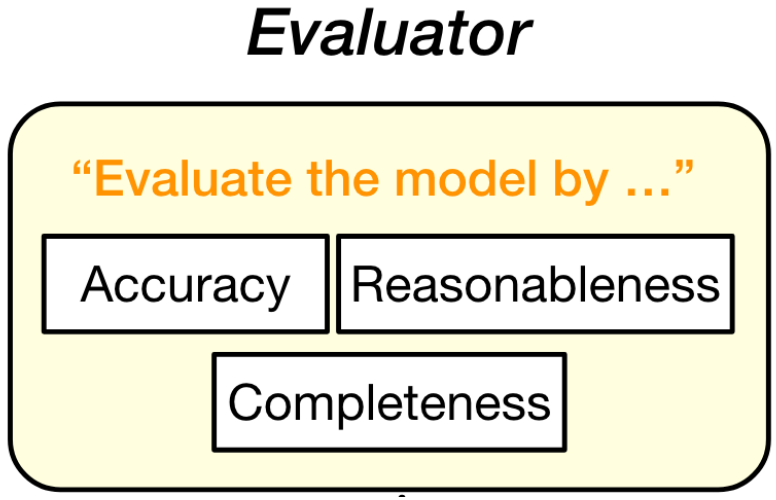
- Example a set of prompts -

"Evaluate accuracy of the given text by comparing with known facts or credible sources. This involves checking the accuracy of any claims or statements made in the text, and verifying that they are supported by evidence. The next line directly provide the text. {output_text} Please return a score ranging from 0 to 100, with 0 being the worst and 100 being the best. Please use the strictest grading standard. The score should be in JSON format with a field name of 'score'. You should not output any other information or text."

Appendix E : LLM Prompts

Evaluator Module

- Assesses the output of the HoneyBee language model
  - **Accuracy**, **Completeness**, and **Reasonableness**
- No longer consider relevance at this stage since the verification step filtered out all instructions with little relevance to materials science
- Use **GPT-4**
- Identification of poorly formulated instructions according to the performance of the HoneyBee model
- These instructions are then passed back to the Instructor for additional iterative refinement.



HoneyBee

: A model for materials science using a progressive finetuning technique
to convert a standard LLaMa model to a specialized model in material science

- Progressive finetuning process for the language model is based on **LoRA**

- **Instructor + Verifier models** act as **teacher model** and **HoneyBee model** acts as **student model**
  - **The student model** will continually learn from the instruction-data and undergo testing during the learning process, allowing us to monitor its performance in real-time
  - Finetuning process continues for a set number of epochs with early stopping if the student model converges to a given loss value

Experiments1

- MatSci-Instruct Evaluation

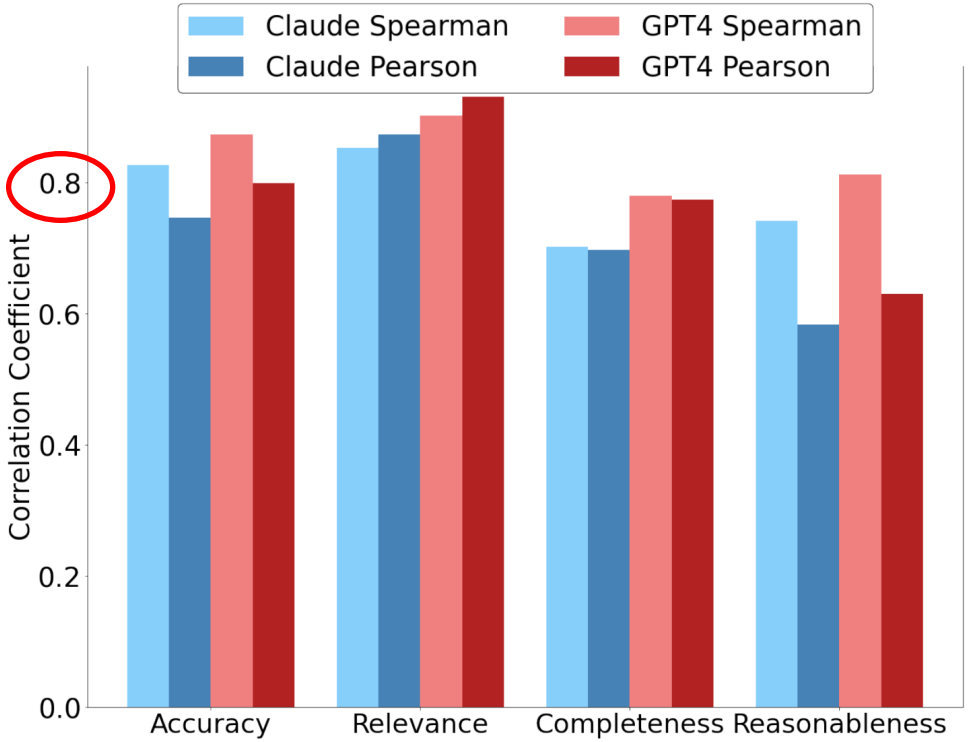


Figure 4: Correlation between human evaluation and LLM evaluation

- Evaluation with human experts on the trustworthiness of the instructions generated by MatSci-Instruct
- Measure agreement between the human experts and the LLMs(Claude, GPT4) by calculating Spearman and Pearson correlation coefficients between the scores along each of the dimensions.

➡️ Overall coefficient as high as 0.8 when compared to manual evaluation.

➡️ The ability of MatSci-Instruct to generate trustworthy

Experiments2

- HoneyBee Task Evaluation

Model	Accuracy	Completeness	Reasonableness
Zero-Shot LLMs			
Chat-GPT	92.55	98.74	99.84
Llama-7b	78.81	90.36	97.64
Llama-13b	84.22	91.22	98.33
Alpaca-7b	81.35	92.01	98.49
Alpaca-13b	86.24	92.17	98.80
HoneyBee without Verfication			
HoneyBee-7b	85.42	93.24	98.49
HoneyBee-13b	88.76	93.99	98.93
HoneyBee with MatSci-Instruct			
HB-7b-Stage1	88.81	93.42	99.07
HB-7b-Stage2	89.99	94.84	99.64
HB-7b-Stage3	91.95	95.78	99.90
HB-13b-Stage1	94.17	94.42	99.40
HB-13b-Stage2	96.42	95.42	99.78
HB-13b-Stage3	98.11	97.00	99.89

Table 2: Evaluation results for various LLMs based on performance on MatSci-Instruct data

- **HoneyBee** gets progressively better with each iteration of MatSci-Instruct for both HoneyBee-7b and HoneyBee-13b
- **HoneyBee without verification** also outperforms LLaMA and Alpaca LLMs of equal size on specialized materials science instruction-data
- **HoneyBee-13b** closely matches, and in some exceeds, the evaluation performance of Chat-GPT
- **HoneyBee-13b** is more parameter efficient than GPT-3

Experiments3

- HoneyBee Performance on MatSci-NLP

- MatSci-NLP : A broad benchmark of materials science NLP tasks

Model	Named Entity Recognition	Relation Extraction	Event Argument Extraction	Paragraph Classification	Synthesis Action Retrieval	Sentence Classification	Slot Filling	Overall (All Tasks)
Low-Resource Finetuning on MatSci-NLP								
MatSciBERT (Gupta et al., 2022)	0.707 0.470	0.791 0.507	0.436 0.251	0.719 0.623	0.692 0.484	0.914 0.660	0.436 0.194	0.671 0.456
MatBERT (Walker et al., 2021)	0.875 0.630	0.804 0.513	0.451 0.288	0.756 0.691	0.717 0.594	0.909 0.614	0.548 0.273	0.722 0.517
HoneyBee-7b	0.787 0.644	0.852 0.518	0.551 0.389	0.741 0.641	0.792 0.617	0.991 0.711	0.529 0.391	0.749 0.559
HoneyBee-13b	0.860 0.748	0.921 0.578	0.653 0.486	0.761 0.658	0.853 0.662	0.998 0.743	0.554 0.401	0.80 0.611

➡️ Both HoneyBee-7b and HoneyBee-13b perform best overall, outperforming MatBERT and MatSciBERT

➡️ **HoneyBee** shows better performance without requiring pretraining on materials science textual data.

Zero-Shot LLM Performance								
LLaMA-7b (Touvron et al., 2023)	0.042 0.064	0.094 0.013	0.160 0.042	0.279 0.218	0.052 0.013	0.096 0.087	0.142 0.010	0.208 0.064
LLaMA-13b (Touvron et al., 2023)	0.057 0.066	0.109 0.016	0.042 0.054	0.233 0.189	0.039 0.009	0.079 0.074	0.138 0.008	0.1 0.059
Alpaca-7b (Taori et al., 2023)	0.031 0.018	0.053 0.037	0.029 0.009	0.375 0.294	0.179 0.129	0.180 0.180	0.139 0.039	0.141 0.101
Alpaca-13b (Taori et al., 2023)	0.053 0.046	0.016 0.035	0.111 0.072	0.310 0.237	0.442 0.278	0.375 0.334	0.110 0.015	0.202 0.145
Chat-GPT (OpenAI, 2022)	0.063 0.052	0.232 0.145	0.204 0.203	0.433 0.450	0.300 0.183	0.320 0.318	0.368 0.280	0.274 0.233
Claude (Bai et al., 2022)	0.063 0.048	0.232 0.143	0.195 0.169	0.442 0.467	0.280 0.177	0.329 0.326	0.393 0.305	0.276 0.234
GPT-4 (OpenAI, 2023)	0.189 0.121	0.445 0.432	0.453 0.353	0.679 0.522	0.743 0.677	0.788 0.689	0.502 0.483	0.543 0.468
Zero-Shot HoneyBee with MatSci-Instruct								
HoneyBee-7b-Stage1	0.173 0.148	0.138 0.120	0.196 0.096	0.380 0.207	0.592 0.208	0.416 0.334	0.292 0.105	0.301 0.174
HoneyBee-7b-Stage2	0.243 0.166	0.199 0.145	0.237 0.123	0.440 0.301	0.612 0.289	0.467 0.345	0.344 0.176	0.363 0.221
HoneyBee-7b-Stage3	0.267 0.190	0.245 0.178	0.290 0.189	0.490 0.343	0.688 0.342	0.490 0.365	0.393 0.289	0.409 0.271
HoneyBee-13b-Stage1	0.369 0.256	0.301 0.224	0.389 0.265	0.500 0.379	0.701 0.378	0.512 0.402	0.467 0.334	0.463 0.320
HoneyBee-13b-Stage2	0.391 0.299	0.367 0.290	0.437 0.303	0.576 0.411	0.765 0.401	0.557 0.461	0.508 0.379	0.514 0.363
HoneyBee-13b-Stage3	0.429 0.372	0.412 0.346	0.481 0.378	0.611 0.467	0.801 0.429	0.589 0.503	0.578 0.423	0.557 0.417

Table 3: Low-resource finetuning and zero-shot evaluation results for various HoneyBee on MatSci-NLP tasks.

Experiments4

- HoneyBee Case Study

- We perform a case study to further understand the capabilities and limitations of the various LLMs we studied, including HoneyBee, Alpaca, and Chat-GPT.
  - Open-ended Instruction - “Analyze the properties and applications of electronic materials”
  - Content-based Instruction

➡️ **HoneyBee-13b** generally produces outputs of the same quality as Chat-GPT while other models generally produce lower quality outputs.

Input (Open-ended query) : **“Analyze the properties and applications of electronic materials”**

- **Chat-GPT:** Electronic materials are a class of materials that possess unique properties that make them suitable for use in electronic devices and technologies. These materials are characterized by their electrical conductivity, optical properties, magnetic properties, and thermal properties. The electrical conductivity of electronic materials is a critical property for materials used in electronic devices such as transistors, diodes, and integrated circuits. Semiconductors such as silicon and germanium are widely used in microprocessors and solar cells because of their unique electrical properties. Optical properties are also important for electronic materials used in displays, touch screens, and flat panel displays. Optoelectronic materials like indium tin oxide (ITO) and gallium nitride (GaN) possess unique optical properties that make them ideal for use in these devices. Magnetic properties are important for electronic materials used in magnetic sensors, hard drives, and magnetic memory. Magnetic materials like iron, cobalt, and nickel possess unique magnetic properties that make them ideal for use in these devices. Lastly, thermal properties are important for electronic materials used in high-temperature electronic devices such as microprocessors and power electronics. Materials like diamond and silicon carbide possess unique thermal properties that make them ideal for use in these devices.

- **HoneyBee-13b:** Electronic materials have unique properties such as electrical conductivity, optical properties, magnetic properties, and thermal properties, making them suitable for use in electronic devices. Semiconductors such as silicon and germanium are used in microprocessors and solar cells due to their unique electrical properties. Optoelectronic materials like indium tin oxide (ITO) and gallium nitride (GaN) possess unique optical properties, ideal for use in displays, touch screens, and flat panel displays. Magnetic materials like iron, cobalt, and nickel are used in magnetic sensors, hard drives, and magnetic memory due to their unique magnetic properties. Materials like diamond and silicon carbide possess unique thermal properties ideal for use in high-temperature electronic devices. The properties of electronic materials are critical for modern electronic devices, and research is ongoing to discover new materials with even more unique properties.

Conclusion

- **MatSci-Instruct** : **Iterative instruction generation method for materials science**
  - Provides a valuable framework for generating instruction-data to progressively finetune LLMs
  - Poor instruction-data is refined based on feedback from an Evaluator leading to higher quality instruction-data

- **HoneyBee** : **A state-of-the-art large language model for materials science**
  - The first billion- parameter scale language model that is specialized in materials science.
  - Outperforms current state-of-the-art general language models (LLaMa, Alpaca) and materials science BERT- based language models (MatBERT, MatSciBERT) in various materials science NLP tasks
  - Performance improvement with each successive MatSci-Instruct stage.

That’s It, Folks!