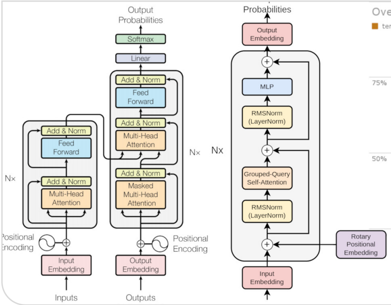
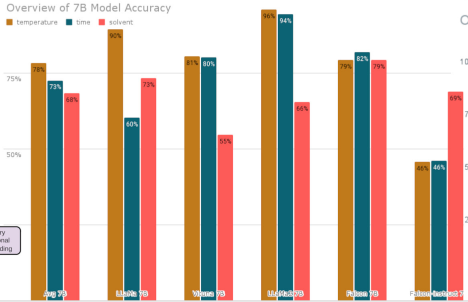
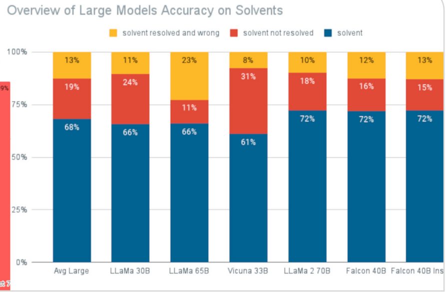
Benchmarking Large Language Models for Zero-Shot Automated Information Extraction from Scientific Literature

Felix Karg | 12. October 2023

Reviewer: T.T.-Prof. Dr. Pascal Friederich; Second Reviewer: Prof. Jan Niehues; Advisor: Tobias Schlöder







Motivation

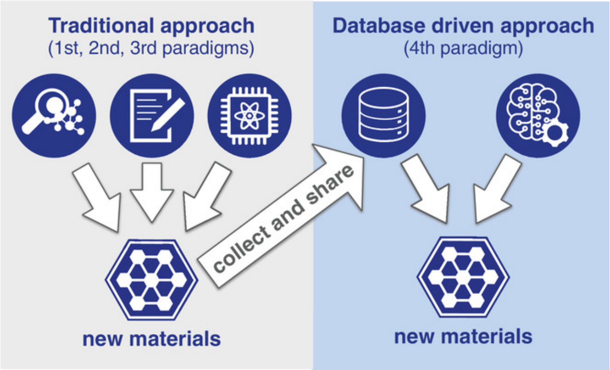


Image Source: [1]

Machine Learning (ML) models are increasingly used in screening steps for materials discovery and property prediction [2–4]. Yet, **most previous research is not available in a machine-readable format.**

Scientific Questions

There are three main questions this work aims to answer:

1. Can I demonstrate high accuracy in zero-shot automated information extraction from scientific literature using open-access Large Language Models (LLMs)?
2. How do currently available open-access LLMs compare for this task?
3. How easy is it to fine-tune open-access LLMs for this task? How much does the accuracy increase from fine-tuning?

While we're at it, create an automated pipeline for information extraction from unstructured text.

Scientific Questions

There are three main questions this work aims to answer:

1. Can I demonstrate high accuracy in zero-shot automated information extraction from scientific literature using open-access LLMs?
2. How do currently available open-access LLMs compare for this task?
3. How easy is it to fine-tune open-access LLMs for this task? How much does the accuracy increase from fine-tuning?

While we're at it, create an automated pipeline for information extraction from unstructured text.

Scientific Questions

There are three main questions this work aims to answer:

1. Can I demonstrate high accuracy in zero-shot automated information extraction from scientific literature using open-access LLMs?
2. How do currently available open-access LLMs compare for this task?
3. How easy is it to fine-tune open-access LLMs for this task? How much does the accuracy increase from fine-tuning?

While we're at it, create an automated pipeline for information extraction from unstructured text.

Scientific Questions

There are three main questions this work aims to answer:

1. Can I demonstrate high accuracy in zero-shot automated information extraction from scientific literature using open-access LLMs?
2. How do currently available open-access LLMs compare for this task?
3. How easy is it to fine-tune open-access LLMs for this task? How much does the accuracy increase from fine-tuning?

While we're at it, create an automated pipeline for information extraction from unstructured text.

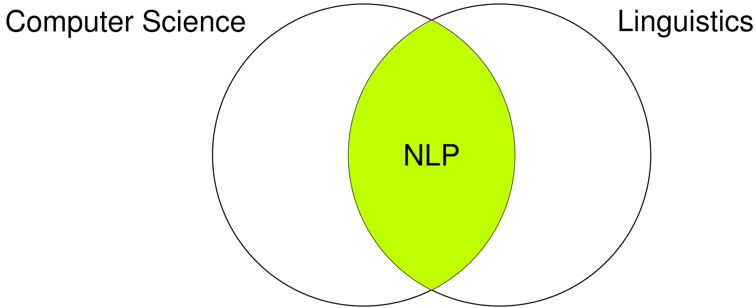
Scientific Questions

There are three main questions this work aims to answer:

1. Can I demonstrate high accuracy in zero-shot automated information extraction from scientific literature using open-access LLMs?
2. How do currently available open-access LLMs compare for this task?
3. How easy is it to fine-tune open-access LLMs for this task? How much does the accuracy increase from fine-tuning?

While we're at it, create an automated pipeline for information extraction from unstructured text.

Natural Language Processing



Goal: Make computers "understand" documents.

Information Extraction for Automated Experimentation

Information Extraction is the Natural Language Processing (NLP) task of extracting structured (machine-readable) information from unstructured text.

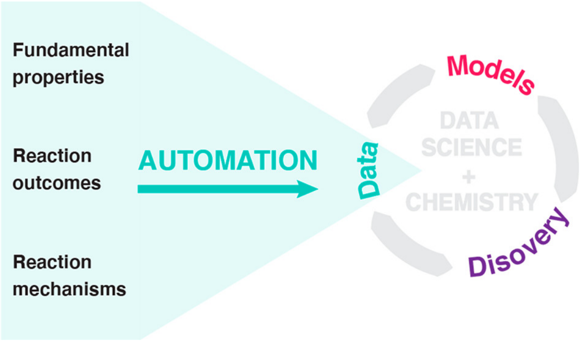


Image Source: [5]

Namend Entity Recognition

Named Entity Recognition (NER) is the NLP task of extracting structured (machine-readable) information from unstructured text.

Effects of the silica **MAT** content and temperature on the magnetic properties **PRO** of Fe4NiO8Zn **MAT** / O2Si **MAT** nanocomposites **DSC** have been studied by electron paramagnetic resonance **CMT** (EPR **CMT**) technique.

MAT stands for Materials, **PRO** stands for Material Property, **DSC** is Descriptor and **CMT** is Characterization method. The goal of NER is to automatically detect entities that fall into these pre-defined semantic types.
Example and partial description taken from [6] (supposedly taken from [7]), visualized using the `spaCy` python library [8].

Rule-Based Entity Recognition

Easy: Regular Expressions! ChemTagger [9], and others [10, 11] demonstrated that it works!
Except ...

- "The mixture was filtered and the filterate was kept at room temperature to obtained needle like colorless crystals of 1 after a month." [12]
- "... distilled water, and dried at ambient temperature to give 39 mg of ..." [13]
- "... was added into 1 mL *boiling methanol* solution of bipe ..." [14]
- ...

Rule-Based Entity Recognition

Easy: Regular Expressions! ChemTagger [9], and others [10, 11] demonstrated that it works!

Except ...

- "The mixture was filtered and the filterate was kept at *room temperature* to obtained needle like colorless crystals of 1 after a month." [12]

Rule-Based Entity Recognition

Easy: Regular Expressions! ChemTagger [9], and others [10, 11] demonstrated that it works!
Except ...

- “The mixture was filtered and the filterate was kept at *room temperature* to obtained needle like colorless crystals of 1 after a month.” [12]
- “... distilled water, and dried at *ambient temperature* to give 39 mg of ...” [13]
- “... was added into 1 mL *boiling methanol solution* of btpe ...” [14]
- ...

Rule-Based Entity Recognition

Easy: Regular Expressions! ChemTagger [9], and others [10, 11] demonstrated that it works! Except ...

- “The mixture was filtered and the filterate was kept at *room temperature* to obtained needle like colorless crystals of 1 after a month.” [12]
- “... distilled water, and dried at *ambient temperature* to give 39 mg of ...” [13]
- “... was added into 1 mL *boiling methanol solution* of btpe ...” [14]
- ...

Rule-Based Entity Recognition

Easy: Regular Expressions! ChemTagger [9], and others [10, 11] demonstrated that it works!
Except ...

- “The mixture was filtered and the filterate was kept at *room temperature* to obtained needle like colorless crystals of 1 after a month.” [12]
- “... distilled water, and dried at *ambient temperature* to give 39 mg of ...” [13]
- “... was added into 1 mL *boiling methanol solution* of btpe ...” [14]
- ...

Rule-Based Entity Recognition

Easy: Regular Expressions! ChemTagger [9], and others [10, 11] demonstrated that it works! Except ...

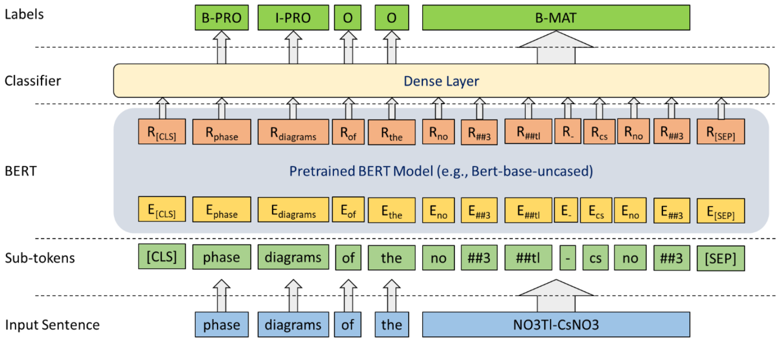
- "The mixture was filtered and the filterate was kept at *room temperature* to obtained needle like colorless crystals of 1 after a month." [12]
- "... distilled water, and dried at *ambient temperature* to give 39 mg of ..." [13]
- "... was added into 1 mL *boiling methanol solution* of btpe ..." [14]
- ...

Rule-Based Entity Recognition

Easy: Regular Expressions! ChemTagger [9], and others [10, 11] demonstrated that it works!
Except ...

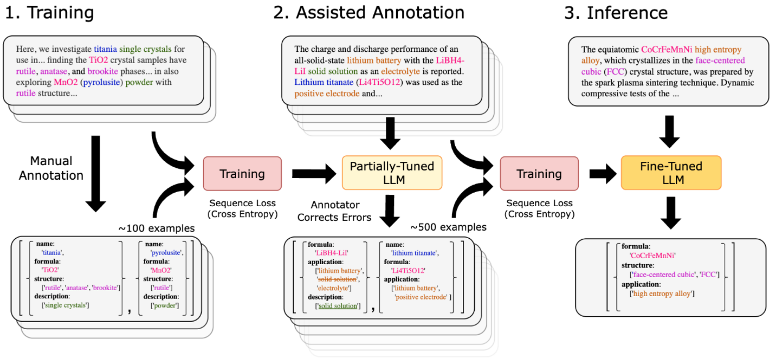
- “The mixture was filtered and the filterate was kept at *room temperature* to obtained needle like colorless crystals of 1 after a month.” [12]
- “... distilled water, and dried at *ambient temperature* to give 39 mg of ...” [13]
- “... was added into 1 mL *boiling methanol solution* of btpe ...” [14]
- ...

Language Models for Information Extraction



NER modeled as a sequence-to-sequence labeling problem can achieve high accuracy using Bidirectional Encoder Representation from Transformers (BERT)-based Language Models (LMs). Image Source: [6]

Large Language Models for Structured Information Extraction



Other work focused on Entity Relation extraction, with mixed results for NER.
Image Source: [15]

Basic Terminology

- **Token:** String of arbitrary length, usually 3-4 characters
  - Refer to my previous talk about the transformer architecture for more details on internals
- **Single-shot** / **Multi-shot:** Evaluation setting in which a LLM is being provided with one or multiple examples of the task to fulfill
- **Zero-shot:** Evaluation setting in which no task examples are provided, or the model has been fine-tuned for

Basic Terminology

- **Token:** String of arbitrary length, usually 3-4 characters
  - Refer to my previous talk about the transformer architecture for more details on internals
- **Single-shot** / **Multi-shot:** Evaluation setting in which a LLM is being provided with one or multiple examples of the task to fulfill
- **Zero-shot:** Evaluation setting in which no task examples are provided, or the model has been fine-tuned for

Basic Terminology

- **Token:** String of arbitrary length, usually 3-4 characters
  - Refer to my previous talk about the transformer architecture for more details on internals
- **Single-shot** / **Multi-shot:** Evaluation setting in which a LLM is being provided with one or multiple examples of the task to fulfill
- **Zero-shot:** Evaluation setting in which no task examples are provided, or the model has been fine-tuned for

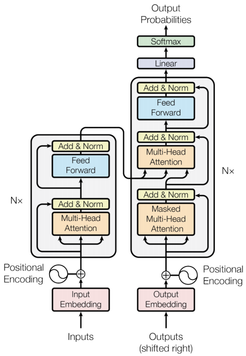
Basic Terminology

- **Token:** String of arbitrary length, usually 3-4 characters
  - Refer to my previous talk about the transformer architecture for more details on internals
- **Single-shot / Multi-shot:** Evaluation setting in which a LLM is being provided with one or multiple examples of the task to fulfill
- **Zero-shot:** Evaluation setting in which no task examples are provided, or the model has been fine-tuned for

Basic Terminology

- **Token:** String of arbitrary length, usually 3-4 characters
  - Refer to my previous talk about the transformer architecture for more details on internals
- **Single-shot** / **Multi-shot:** Evaluation setting in which a LLM is being provided with one or multiple examples of the task to fulfill
- **Zero-shot:** Evaluation setting in which no task examples are provided, or the model has been fine-tuned for

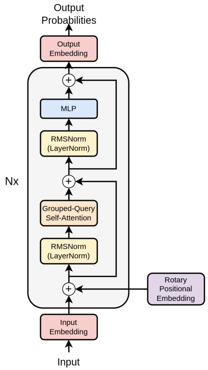
The Transformer Architecture



Original Transformer Architecture
Image Source: [16]

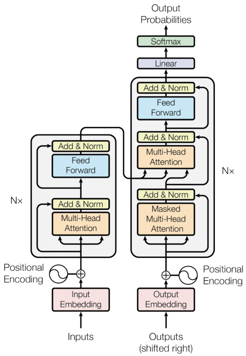
Most Prominent Changes:

- Activation Function: Swish Gated Linear Unit (SwiGLU) [17] instead of Rectified Linier Unit (ReLU)
- Positional Encoding: Rotary Positional Encoding (RoPE) [18], and *on each layer*
- Normalization with RMSNorm [19] *before* instead of after each layer
- Attention: Often a variant of Sparse Attention [20] or FlashAttention [21]
- Most Recently: The usage of Grouped Query Attention (GQA) [22]



Modern Transformer Architecture

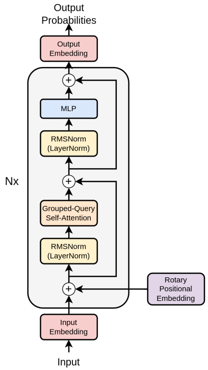
The Transformer Architecture



Original Transformer Architecture
Image Source: [16]

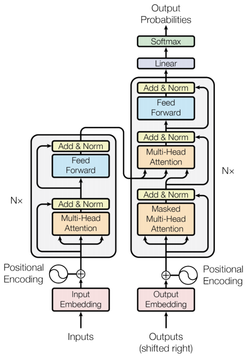
Most Prominent Changes:

- Activation Function: SwiGLU [17] instead of ReLU
- Positional Encoding: Rotary Positional Encoding (RoPE) [18], and *on each layer*
- Normalization with RMSNorm [19] *before* instead of after each layer
- Attention: Often a variant of Sparse Attention [20] or FlashAttention [21]
- Most Recently: The usage of Grouped Query Attention (GQA) [22]



Modern Transformer Architecture

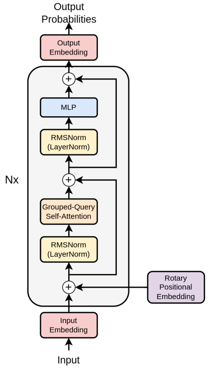
The Transformer Architecture



Original Transformer Architecture
Image Source: [16]

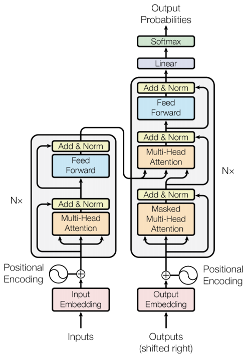
Most Prominent Changes:

- Activation Function: SwiGLU [17] instead of ReLU
- Positional Encoding: Rotary Positional Encoding (RoPE) [18], and *on each layer*
- Normalization with RMSNorm [19] *before* instead of after each layer
- Attention: Often a variant of Sparse Attention [20] or FlashAttention [21]
- Most Recently: The usage of Grouped Query Attention (GQA) [22]



Modern Transformer Architecture

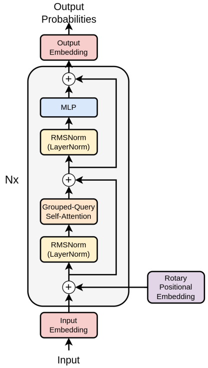
The Transformer Architecture



Original Transformer Architecture
Image Source: [16]

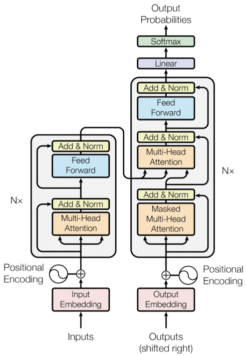
Most Prominent Changes:

- Activation Function: SwiGLU [17] instead of ReLU
- Positional Encoding: Rotary Positional Encoding (RoPE) [18], and *on each layer*
- Normalization with RMSNorm [19] *before* instead of after each layer
- Attention: Often a variant of Sparse Attention [20] or FlashAttention [21]
- Most Recently: The usage of Grouped Query Attention (GQA) [22]



Modern Transformer Architecture

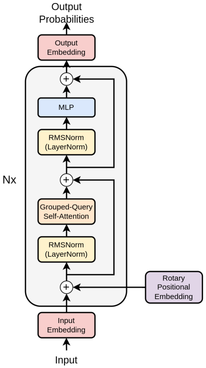
The Transformer Architecture



Original Transformer Architecture
Image Source: [16]

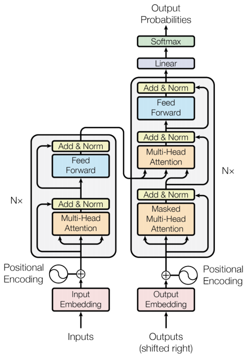
Most Prominent Changes:

- Activation Function: SwiGLU [17] instead of ReLU
- Positional Encoding: Rotary Positional Encoding (RoPE) [18], and *on each layer*
- Normalization with RMSNorm [19] *before* instead of after each layer
- Attention: Often a variant of Sparse Attention [20] or FlashAttention [21]
- Most Recently: The usage of Grouped Query Attention (GQA) [22]



Modern Transformer Architecture

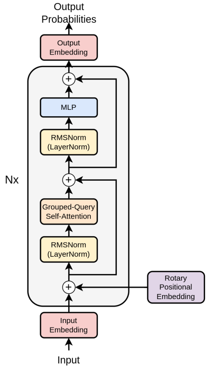
The Transformer Architecture



Original Transformer Architecture
Image Source: [16]

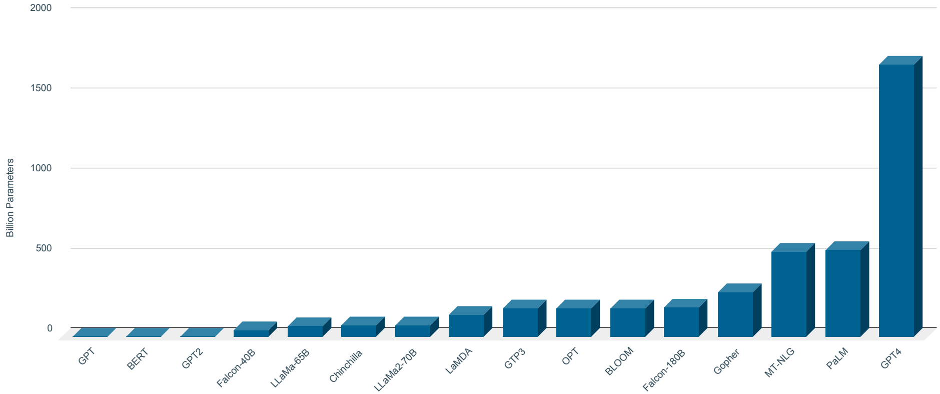
Most Prominent Changes:

- Activation Function: SwiGLU [17] instead of ReLU
- Positional Encoding: Rotary Positional Encoding (RoPE) [18], and *on each layer*
- Normalization with RMSNorm [19] *before* instead of after each layer
- Attention: Often a variant of Sparse Attention [20] or FlashAttention [21]
- Most Recently: The usage of Grouped Query Attention (GQA) [22]

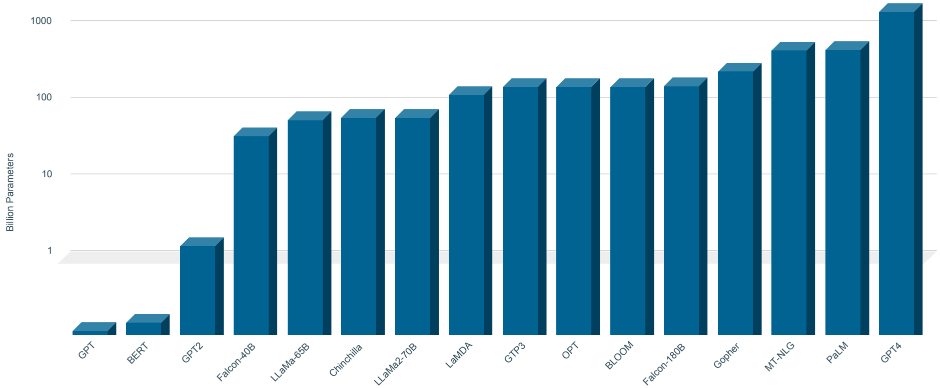


Modern Transformer
Architecture

Large Language Model Parameter Count



Large Language Model Parameter Count (logscale)



Training Large Language Models

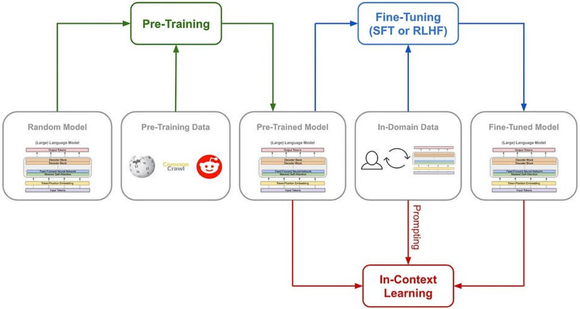


Image Source: [23]

Criteria for Models

1. It is possible to get the full model weights.
2. The selected models ought to be decently capable causal language models.
3. Ceteris paribus, a smaller model is better.

Criteria for Models

1. It is possible to get the full model weights.
2. The selected models ought to be decently capable causal language models.
3. Ceteris paribus, a smaller model is better.

Criteria for Models

1. It is possible to get the full model weights.
2. The selected models ought to be decently capable causal language models.
3. Ceteris paribus, a smaller model is better.

Criteria for Models

1. It is possible to get the full model weights.
2. The selected models ought to be decently capable causal language models.
3. Ceteris paribus, a smaller model is better.

Language Models

- LLaMa 7B, 13B, 30B, 65B
- Vicuna 7B, 13B, 33B
- LLaMa 2 7B, 13B, 70B
- Falcon 7B, 40B
- Falcon-instruct 7B, 40B

Language Models

- LLaMa 7B, 13B, 30B, 65B
- Vicuna 7B, 13B, 33B
- LLaMa 2 7B, 13B, 70B
- Falcon 7B, 40B
- Falcon-instruct 7B, 40B

Language Models

- LLaMa 7B, 13B, 30B, 65B
- Vicuna 7B, 13B, 33B
- LLaMa 2 7B, 13B, 70B
- Falcon 7B, 40B
- Falcon-instruct 7B, 40B

Language Models

- LLaMa 7B, 13B, 30B, 65B
- Vicuna 7B, 13B, 33B
- LLaMa 2 7B, 13B, 70B
- Falcon 7B, 40B
- Falcon-instruct 7B, 40B

Language Models

- LLaMa 7B, 13B, 30B, 65B
- Vicuna 7B, 13B, 33B
- LLaMa 2 7B, 13B, 70B
- Falcon 7B, 40B
- Falcon-instruct 7B, 40B

Language Models

- LLaMa 7B, 13B, 30B, 65B
- Vicuna 7B, 13B, 33B
- LLaMa 2 7B, 13B, 70B
- Falcon 7B, 40B
- Falcon-instruct 7B, 40B

Schema

The schema provided for the model to follow. Model output termination would happen after generation of a token for ‘`"`’ for strings or ‘`,`’ for numbers, or a number of other dedicated ‘end of generation’ tokens, e.g. `<EOS>`.

```
schema = {
    "type": "object",
    "properties": {
        "additive": {"type": "string"},
        "solvent": {"type": "string"},
        "temperature": {"type": "number"},
        "temperature_unit": {"type": "string"},
        "time": {"type": "number"},
        "time_unit": {"type": "string"},
    },
}
```

Prompt

Prompt used to generate output. `"{output}"` delineates where the model provides an answer.

```
prompt = "{paragraph}\nOutput result in the following JSON schema format:\n{schema}\nResult: {output}"
```

Output

Exemplary output based on the prompt and schema shown before.

```
output = {
    "additive": "acid",
    "solvent": "water",
    "temperature": 80,
    "temperature_unit": "C",
    "time": 24,
    "time_unit": "h",
}
```

Data Source

- SynMOF_M [3]
  - Publicly Accessible
  - Manually Extracted
  - 778 Labels
  - Temperature Information in °C
  - Timeframe (Durations) in h
  - Chemical Compounds via CID
- Corresponding Synthesis Paragraphs

Data Source

- SynMOF_M [3]
  - Publicly Accessible
  - Manually Extracted
  - 778 Labels
  - Temperature Information is in `°C`
  - Timeframe (Durations) in `h`.
  - Chemical Compounds via `cid`
- Corresponding Synthesis Paragraphs

Data Source

- SynMOF_M [3]
  - Publicly Accessible
  - Manually Extracted
  - 778 Labels
  - Temperature Information is in `°C`
  - Timeframe (Durations) in `h`.
  - Chemical Compounds via `cid`
- Corresponding Synthesis Paragraphs

Data Source

- SynMOF_M [3]
  - Publicly Accessible
  - Manually Extracted
  - 778 Labels
  - Temperature Information is in `°C`
  - Timeframe (Durations) in `h`.
  - Chemical Compounds via `cid`
- Corresponding Synthesis Paragraphs

Data Source

- SynMOF_M [3]
  - Publicly Accessible
  - Manually Extracted
  - 778 Labels
  - Temperature Information is in `°C`
  - Timeframe (Durations) in `h`.
  - Chemical Compounds via `cid`
- Corresponding Synthesis Paragraphs

Data Source

- SynMOF_M [3]
  - Publicly Accessible
  - Manually Extracted
  - 778 Labels
  - Temperature Information is in `°C`
  - Timeframe (Durations) in `h`.
  - Chemical Compounds via `cid`
- Corresponding Synthesis Paragraphs

Data Source

- SynMOF_M [3]
  - Publicly Accessible
  - Manually Extracted
  - 778 Labels
  - Temperature Information is in `°C`
  - Timeframe (Durations) in `h`.
  - Chemical Compounds via `cid`
- Corresponding Synthesis Paragraphs

Data Source

- SynMOF_M [3]
  - Publicly Accessible
  - Manually Extracted
  - 778 Labels
  - Temperature Information is in `°C`
  - Timeframe (Durations) in `h`.
  - Chemical Compounds via `cid`
- Corresponding Synthesis Paragraphs

Data Source

- SynMOF_M [3]
  - Publicly Accessible
  - Manually Extracted
  - 778 Labels
  - Temperature Information is in `°C`
  - Timeframe (Durations) in `h`.
  - Chemical Compounds via `cid`
- Corresponding Synthesis Paragraphs

Compound Equality: `cid`

Foreshadowing:

- ‘water’
- `cid` 962
- ‘Synonyms’: 319
- Includes ‘distilled water’ and ‘H2O’
- But not ‘distilled H2O’

Compound Equality: `cid`

Foreshadowing:

- ‘water’
- `cid` 962
- ‘Synonyms’: 319
- Includes ‘distilled water’ and ‘H2O’
- But not ‘distilled H2O’

Compound Equality: `cid`

Foreshadowing:

- ‘water’
- `cid` 962
- ‘Synonyms’: 319
- Includes ‘distilled water’ and ‘H2O’
- But not ‘distilled H2O’

Compound Equality: `cid`

Foreshadowing:

- 'water'
- `cid` 962
- 'Synonyms': 319
- Includes 'distilled water' and 'H2O'
- But not 'distilled H2O'

Compound Equality: `cid`

Foreshadowing:

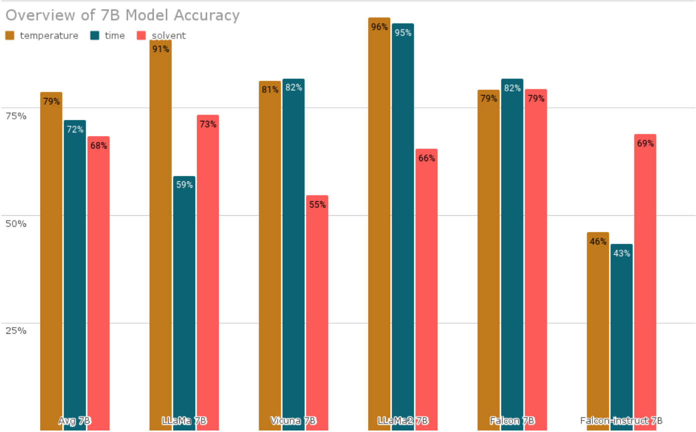
- ‘water’
- `cid` 962
- ‘Synonyms’: 319
- Includes ‘distilled water’ and ‘H2O’
- But not ‘distilled H2O’

Compound Equality: `cid`

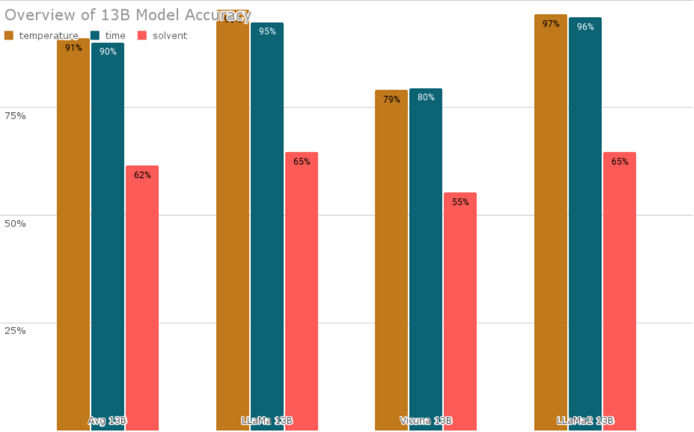
Foreshadowing:

- ‘water’
- `cid` 962
- ‘Synonyms’: 319
- Includes ‘distilled water’ and ‘H2O’
- But not ‘distilled H2O’

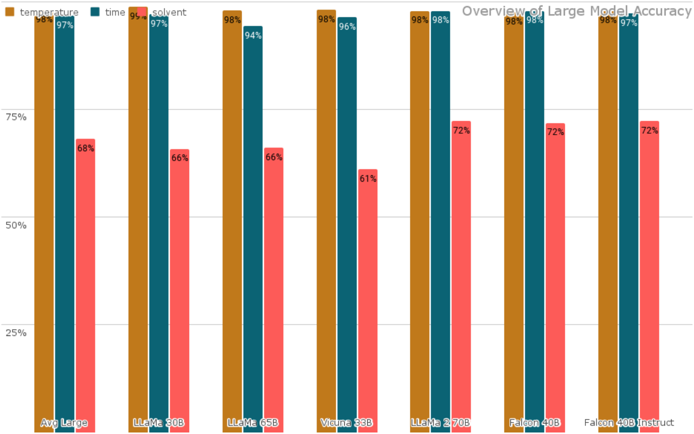
Accuracy Overview



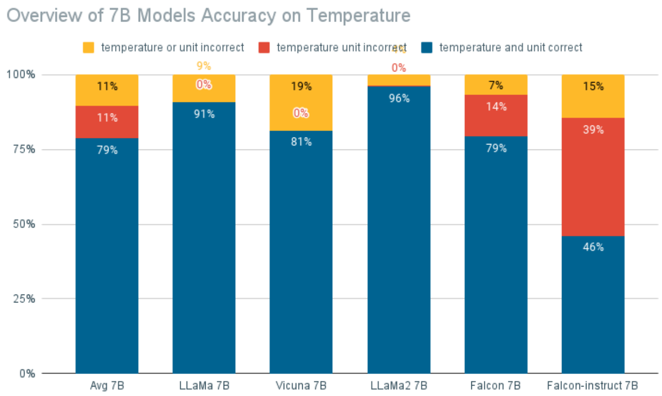
Accuracy Overview II



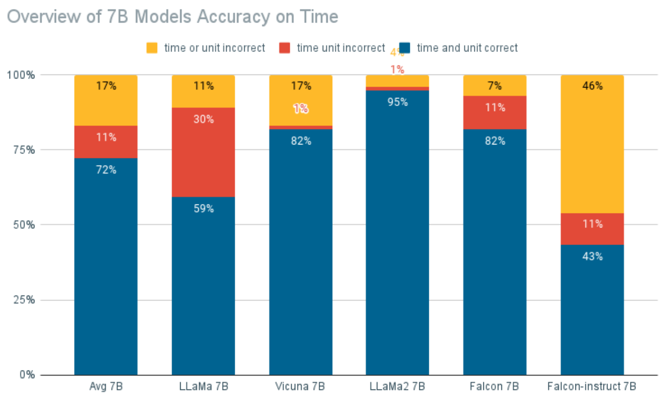
Accuracy Overview III



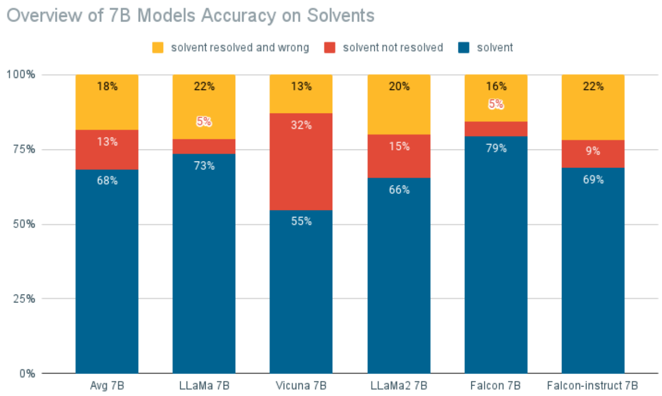
Unit Confusion



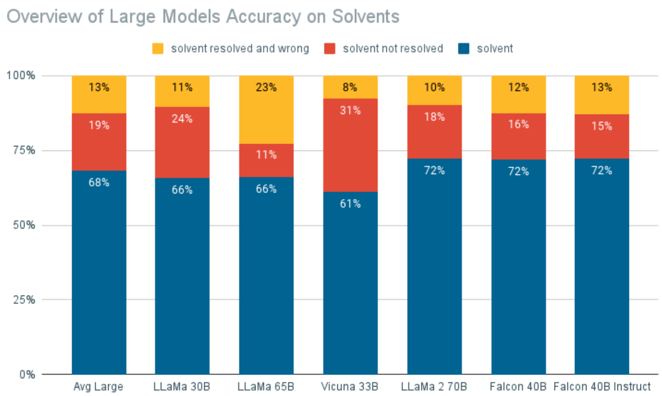
Unit Confusion II



Solvent Resolution I



Solvent Resolution II



Solvent Resolution?

One hypothesis: Models *are* getting more accurate, but there is a failure in resolving the compounds.

Remember ‘distilled H2O’?

This may be true in particular for the `solvent` `N,N-DIMETHYLACETAMIDE` (`cid` 31374), where the synthesis paragraphs contain none of its 125 synonyms in 34 cases (or about 4.37% of the dataset).

Karlsruhe Institute of Technology

Solvent Resolution?

One hypothesis: Models *are* getting more accurate, but there is a failure in resolving the compounds.

Remember ‘distilled H2O’?

This may be true in particular for the `solvent` `N,N-DIMETHYLACETAMIDE` (`cid` 31374), where the synthesis paragraphs contain none of its 125 synonyms in 34 cases (or about 4.37% of the dataset).

Solvent Resolution?

One hypothesis: Models *are* getting more accurate, but there is a failure in resolving the compounds.

Remember ‘distilled H2O’?

This may be true in particular for the `solvent` `N,N-DIMETHYLACETAMIDE` (`cid` 31374), where the synthesis paragraphs contain none of its 125 synonyms in 34 cases (or about 4.37% of the dataset).

Fine-Tuning: Excerpt 1

Excerpt of what could be found in a custom dataloader. `text` describes any string the model may be provided as input.
The tokenizer converts any string to a list of tokens and an attention mask, among other things.
Similar code can be found in tutorials and official sources, e.g. Microsoft [24]

```
text_encodings = tokenizer(text, ...)

return {
    "input_ids": text_encodings["input_ids"],
    "attention_mask": text_encodings["attention_mask"],
    "label": text_encodings["input_ids"],
}
```

Fine-Tuning: Failure 1

A model fine-tuned like this returns the following. The `"` where actually inserted during conversion to `json` from `jsonformer`.

```
output = {
    "additive": "",
    "solvent": "",
    "temperature": "",
    "temperature_unit": "",
    "time": "",
    "time_unit": "",
}
```

Fine-Tuning: Excerpt 2

- Using the HuggingFace `trl` (Transformer Reinforcement Learning) library
- `DataCollator` are used for batch-processing inputs
- `DataCollatorForLanguageModeling` abstracting away tokenization, uses `"text"`-key for training in other examples
- Specifically, the example uses `DataCollatorForCompletionOnlyLM`, deriving from it

Fine-Tuning: Excerpt 2

- Using the HuggingFace `trl` (Transformer Reinforcement Learning) library
- `DataCollator` are used for batch-processing inputs
- `DataCollatorForLanguageModeling` abstracting away tokenization, uses `"text"`-key for training in other examples
- Specifically, the example uses `DataCollatorForCompletionOnlyLM`, deriving from it

Fine-Tuning: Excerpt 2

- Using the HuggingFace `trl` (Transformer Reinforcement Learning) library
- `DataCollator` are used for batch-processing inputs
- `DataCollatorForLanguageModeling` abstracting away tokenization, uses `"text"`-key for training in other examples
- Specifically, the example uses `DataCollatorForCompletionOnlyLM`, deriving from it

Fine-Tuning: Excerpt 2

- Using the HuggingFace `trl` (Transformer Reinforcement Learning) library
- `DataCollator` are used for batch-processing inputs
- `DataCollatorForLanguageModeling` abstracting away tokenization, uses `"text"`-key for training in other examples
- Specifically, the example uses `DataCollatorForCompletionOnlyLM`, deriving from it

Fine-Tuning: Excerpt 2

- Using the HuggingFace `trl` (Transformer Reinforcement Learning) library
- `DataCollator` are used for batch-processing inputs
- `DataCollatorForLanguageModeling` abstracting away tokenization, uses `"text"`-key for training in other examples
- Specifically, the example uses `DataCollatorForCompletionOnlyLM`, deriving from it

Fine-Tuning: Failure 2

Error when providing `DataCollatorForCompletionOnlyLM` with a dataloader similar to those in examples. Counterintuitively, this is not a `KeyError`.

```
|------------- Traceback (most recent call last) ---------------|
|           ...                                                  |
|   372 |   model.train()  # put the model in training mode      |
| > 373 |   trainer.train()                                      |
...
ValueError: You should supply an encoding or a list of encodings
to this method that includes "input_ids", but you provided []
```

It also fails when manually tokenizing before the `DataCollator` (providing tokenized `"input_ids"` etc. as key, using this or a different `DataCollator`).

Conclusion

- Zero-shot automated information extraction from scientific literature was successfully demonstrated.
- Capabilities of different open-access LLMs where measured and compared. Furthermore, frequent mistakes where analyzed and provided insight in failure modes.
- Fine-Tuning was substantially harder than initially assumed, and eventually abandoned for this work.

Conclusion

- Zero-shot automated information extraction from scientific literature was successfully demonstrated.
- Capabilities of different open-access LLMs where measured and compared. Furthermore, frequent mistakes where analyzed and provided insight in failure modes.
- Fine-Tuning was substantially harder than initially assumed, and eventually abandoned for this work.

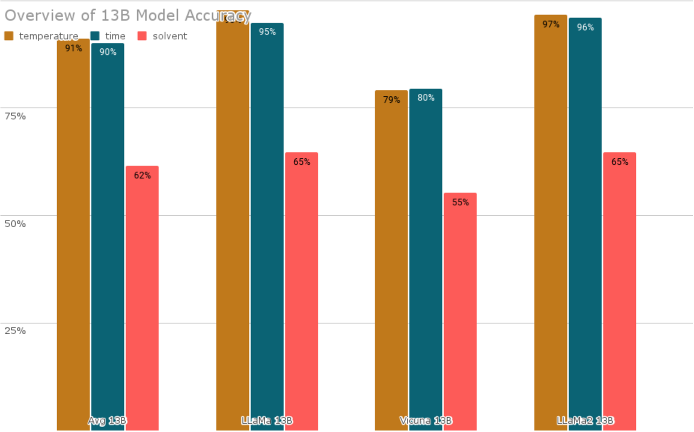
Conclusion

- Zero-shot automated information extraction from scientific literature was successfully demonstrated.
- Capabilities of different open-access LLMs where measured and compared. Furthermore, frequent mistakes where analyzed and provided insight in failure modes.
- Fine-Tuning was substantially harder than initially assumed, and eventually abandoned for this work.

Conclusion

- Zero-shot automated information extraction from scientific literature was successfully demonstrated.
- Capabilities of different open-access LLMs where measured and compared. Furthermore, frequent mistakes where analyzed and provided insight in failure modes.
- Fine-Tuning was substantially harder than initially assumed, and eventually abandoned for this work.

Surprises



Overview of 13B Model Accuracy
temperature
time
solvent
75%
50%
25%
91%
90%
62%
95%
65%
79%
80%
55%
97%
96%
65%
Avg 13B
LLaMa 13B
Vicuna 13B
LLaMa2 13B

Surprises: Implications

GPU requirements for 4-bit quantized LLaMA models

LLaMA Model	Minimum VRAM Requirement	Recommended GPU Examples
LLaMA-7B	6GB	RTX 3060, GTX 1660, 2060, AMD 5700 XT, RTX 3050
LLaMA-13B	10GB	AMD 6900 XT, RTX 2060 12GB, 3060 12GB, 3080, A2000

Image Source: [25]

Modern consumer hardware can achieve throughputs of 30 to 40 tokens per second, depending on the specific GPU used [25].

Outlook

A number of questions where answered, this newfound knowledge provides the opportunity to ask better questions.

- How many of the unresolved `solvent` cases where actually correct?
- What is the accuracy of a correctly modeled `additive`?
- How can the prompt be improved?
- How does zero-shot accuracy compare with fine-tuned models?
- How do these models compare with next-gen models such as GPT4 or Falcon-180B?
- How do LLMs compare to established masked language models for NER?

Outlook

A number of questions where answered, this newfound knowledge provides the opportunity to ask better questions.

- How many of the unresolved `solvent` cases where actually correct?
- What is the accuracy of a correctly modeled `additive`?
- How can the prompt be improved?
- How does zero-shot accuracy compare with fine-tuned models?
- How do these models compare with next-gen models such as GPT4 or Falcon-180B?
- How do LLMs compare to established masked language models for NER?

Outlook

A number of questions where answered, this newfound knowledge provides the opportunity to ask better questions.

- How many of the unresolved `solvent` cases where actually correct?
- What is the accuracy of a correctly modeled `additive`?
- How can the prompt be improved?
- How does zero-shot accuracy compare with fine-tuned models?
- How do these models compare with next-gen models such as GPT4 or Falcon-180B?
- How do LLMs compare to established masked language models for NER?

Outlook

A number of questions where answered, this newfound knowledge provides the opportunity to ask better questions.

- How many of the unresolved `solvent` cases where actually correct?
- What is the accuracy of a correctly modeled `additive`?
- How can the prompt be improved?
- How does zero-shot accuracy compare with fine-tuned models?
- How do these models compare with next-gen models such as GPT4 or Falcon-180B?
- How do LLMs compare to established masked language models for NER?

Outlook

A number of questions where answered, this newfound knowledge provides the opportunity to ask better questions.

- How many of the unresolved `solvent` cases where actually correct?
- What is the accuracy of a correctly modeled `additive`?
- How can the prompt be improved?
- How does zero-shot accuracy compare with fine-tuned models?
- How do these models compare with next-gen models such as GPT4 or Falcon-180B?
- How do LLMs compare to established masked language models for NER?

Outlook

A number of questions where answered, this newfound knowledge provides the opportunity to ask better questions.

- How many of the unresolved `solvent` cases where actually correct?
- What is the accuracy of a correctly modeled `additive`?
- How can the prompt be improved?
- How does zero-shot accuracy compare with fine-tuned models?
- How do these models compare with next-gen models such as GPT4 or Falcon-180B?
- How do LLMs compare to established masked language models for NER?

Outlook

A number of questions where answered, this newfound knowledge provides the opportunity to ask better questions.

- How many of the unresolved `solvent` cases where actually correct?
- What is the accuracy of a correctly modeled `additive`?
- How can the prompt be improved?
- How does zero-shot accuracy compare with fine-tuned models?
- How do these models compare with next-gen models such as GPT4 or Falcon-180B?
- How do LLMs compare to established masked language models for NER?

What are your Questions?

All code and artifacts can be found at
`https://github.com/fkarg/mthesis`.
A tagged commit marks the state of submission.

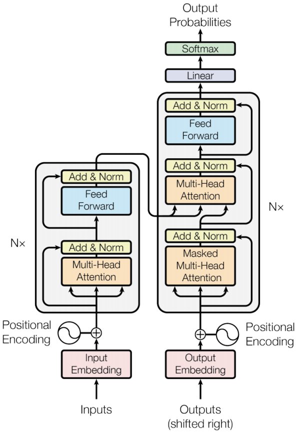


Image Source: [16]

Sources I

1. Himanen, L., Geurts, A., Foster, A. S. & Rinke, P. Data-Driven Materials Science: Status, Challenges, and Perspectives. en. *Advanced Science* **6,** 1900808. ISSN: 2198-3844. doi:10.1002/advs.201900808. https://onlinelibrary.wiley.com/doi/abs/10.1002/advs.201900808 (2023-10-10) (2019).
2. Saal, J. E., Oliynyk, A. O. & Meredig, B. Machine Learning in Materials Discovery: Confirmed Predictions and Their Underlying Approaches. *Annual Review of Materials Research* **50,** 49–69. doi:10.1146/annurev-matsci-090319-010954 (2020).
3. Luo, Y., Bag, S., Zaremba, O., Cierpka, A., Andreo, J., Wuttke, S., Friederich, P. & Tsotsalas, M. MOF Synthesis Prediction Enabled by Automatic Data Mining and Machine Learning**. en. *Angewandte Chemie International Edition* **61,** e202200242. ISSN: 1521-3773. doi:10.1002/anie.202200242. https://onlinelibrary.wiley.com/doi/abs/10.1002/anie.202200242 (2023-02-01) (2022).

Sources II

4. Choudhary, K., DeCost, B., Chen, C., Jain, A., Tavazza, F., Cohn, R., Park, C. W., Choudhary, A., Agrawal, A. & Billinge, S. J. Recent Advances and Applications of Deep Learning Methods in Materials Science. *npj Computational Materials* **8,** 59. doi:10.1038/s41524-022-00734-6 (2022).
5. Shi, Y., Prieto, P. L., Zepel, T., Grunert, S. & Hein, J. E. Automated Experimentation Powers Data Science in Chemistry. *Accounts of Chemical Research* **54,** 546–555. doi:10.1021/acs.accounts.0c00736 (2021).
6. Zhao, X., Greenberg, J., An, Y. & Hu, X. T. *Fine-Tuning BERT Model for Materials Named Entity Recognition.* in *2021 IEEE International Conference on Big Data (Big Data)* (2021-12), 3717–3720. doi:10.1109/BigData52589.2021.9671697.
7. Weston, L., Tshitoyan, V., Dagdelen, J., Kononova, O., Trewartha, A., Persson, K. A., Ceder, G. & Jain, A. Named Entity Recognition and Normalization Applied to Large-Scale Information Extraction from the Materials Science Literature. *Journal of chemical information and modeling* **59,** 3692–3702. doi:10.1021/acs.jcim.9b00470 (2019).

Sources III

8. Montani I spaCy, H. M. *Natural Language Understanding with Bloom Embeddings, Convolutional Neural Networks and Incremental Parsing. 2017.* 2017.
9. Hawizy, L., Jessop, D. M., Adams, N. & Murray-Rust, P. ChemicalTagger: A Tool for Semantic Text-Mining in Chemistry. *Journal of Cheminformatics* **3,** 17. ISSN: 1758-2946. doi:10.1186/1758-2946-3-17. https://doi.org/10.1186/1758-2946-3-17 (2023-02-01) (2011-05).
10. Beard, E. J., Sivaraman, G., Vazquez-Mayagoitia, A., Vishwanath, V. & Cole, J. M. Comparative Dataset of Experimental and Computational Attributes of UV/Vis Absorption Spectra. en. *Scientific Data* **6,** 307. ISSN: 2052-4463. doi:10.1038/s41597-019-0306-0. https://www.nature.com/articles/s41597-019-0306-0 (2023-02-20) (2019-12).
11. Huang, S. & Cole, J. M. A Database of Battery Materials Auto-Generated Using ChemDataExtractor. en. *Scientific Data* **7,** 260. ISSN: 2052-4463. doi:10.1038/s41597-020-00602-2. https://www.nature.com/articles/s41597-020-00602-2 (2023-02-20) (2020-08).

Sources IV

12. Vishnoi, P. & Murugavel, R. A Flexible Tri-carboxylic Acid Derived Zinc(II) 3D Helical Metal-Organic-Framework and a Cadmium(II) Interwoven 2D Layered Framework Solid. en. *Zeitschrift für anorganische und allgemeine Chemie* **640,** 1075–1080. ISSN: 1521-3749. doi:10.1002/zaac.201300677. https://onlinelibrary.wiley.com/doi/abs/10.1002/zaac.201300677 (2023-10-10) (2014).
13. Lin, Z., Jiang, F., Chen, L., Yuan, D. & Hong, M. New 3-D Chiral Framework of Indium with 1,3,5-Benzenetricarboxylate. *Inorganic Chemistry* **44,** 73–76. ISSN: 0020-1669. doi:10.1021/ic0494962. https://doi.org/10.1021/ic0494962 (2023-10-10) (2005-01).
14. Wang, N., Ma, J.-G., Shi, W. & Cheng, P. Two Novel Cd(II) Complexes with Unprecedented Four- and Six-Fold Interpenetration. en. *CrystEngComm* **14,** 5198–5202. ISSN: 1466-8033. doi:10.1039/C2CE25282A. https://pubs.rsc.org/en/content/articlelanding/2012/ce/c2ce25282a (2023-10-10) (2012-07).

Sources V

15. Dunn, A., Dagdelen, J., Walker, N., Lee, S., Rosen, A. S., Ceder, G., Persson, K. & Jain, A. Structured Information Extraction from Complex Scientific Text with Fine-Tuned Large Language Models. *arXiv:2212.05238.* doi:`10.48550/arXiv.2212.05238`. arXiv: `2212.05238 [cond-mat]`. `http://arxiv.org/abs/2212.05238` (2023-02-01) (2022-12).
16. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. & Polosukhin, I. Attention Is All You Need. *Advances in neural information processing systems* **30** (2017).
17. Shazeer, N. Glu Variants Improve Transformer. *arXiv preprint arXiv:2002.05202.* arXiv: `2002.05202` (2020).
18. Su, J., Lu, Y., Pan, S., Murtadha, A., Wen, B. & Liu, Y. RoFormer: Enhanced Transformer with Rotary Position Embedding. *arXiv:2104.09864.* arXiv: `2104.09864 [cs]`. `http://arxiv.org/abs/2104.09864` (2023-04-03) (2022-08).

Sources VI

19. Ba, J. L., Kiros, J. R. & Hinton, G. E. Layer Normalization. *arXiv:1607.06450.* doi:10.48550/arXiv.1607.06450. arXiv: 1607.06450 [cs, stat]. http://arxiv.org/abs/1607.06450 (2023-03-08) (2016-07).
20. Child, R., Gray, S., Radford, A. & Sutskever, I. Generating Long Sequences with Sparse Transformers. *arXiv:1904.10509.* doi:10.48550/arXiv.1904.10509. arXiv: 1904.10509 [cs, stat]. http://arxiv.org/abs/1904.10509 (2023-03-02) (2019-04).
21. Dao, T., Fu, D. Y., Ermon, S., Rudra, A. & Ré, C. Flashattention: Fast and Memory-Efficient Exact Attention with Io-Awareness. *arXiv preprint arXiv:2205.14135.* arXiv: 2205.14135 (2022).
22. Ainslie, J., Lee-Thorp, J., de Jong, M., Zemlyanskiy, Y., Lebrón, F. & Sanghai, S. GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints. *arXiv preprint arXiv:2305.13245.* arXiv: 2305.13245 (2023).

Sources VII

23. Ghosh, B. *Empowering Language Models: Pre-training, Fine-Tuning, and In-Context Learning.* en. 2023-06. https://medium.com/@bijit211987/the-evolution-of-language-models-pre-training-fine-tuning-and-in-context-learning-b63d4c161e49 (2023-10-10).
24. *DeepSpeedExamples/Applications/DeepSpeed-Chat/Training/Utils/Data/Data_utils.Py at Bae2afb8417697407ffe7cf6a21388a840679059 · Microsoft/DeepSpeedExamples.* en. 2023. https://github.com/microsoft/DeepSpeedExamples/blob/bae2afb8417697407ffe7cf6a21388a840679059/applications/DeepSpeed-Chat/training/utils/data/data_utils.py (2023-09-16).
25. *HardwareRequirements for LLaMA and Llama-2 Local Use (GPU, CPU, RAM).* en-US. 2023-07. https://www.hardware-corner.net/guides/computer-to-run-llama-ai-model/ (2023-10-02).
26. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D. & Sutskever, I. Language Models Are Unsupervised Multitask Learners. en. *published on GitHub* (2019).

Sources VIII

27. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G. & Askell, A. Language Models Are Few-Shot Learners. *Advances in neural information processing systems* **33,** 1877–1901 (2020).

28. OpenAI. *GPT-4 Technical Report.* 2023. `https://cdn.openai.com/papers/gpt-4.pdf` (2023-03-14).

29. *Convolutional Neural Networks (CNN): Step 4 - Full Connection - Blogs - SuperDataScience | Machine Learning | AI | Data Science Career | Analytics | Success.* 2018-08. `https://www.superdatascience.com/blogs/convolutional-neural-networks-cnn-step-4-full-connection` (2023-10-07).

30. Ouyang, L. *et al.* Training Language Models to Follow Instructions with Human Feedback. *arXiv:2203.02155.* doi:`10.48550/arXiv.2203.02155`. arXiv: `2203.02155 [cs]`. `http://arxiv.org/abs/2203.02155` (2023-02-16) (2022-03).

Sources IX

31. Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S. & Amodei, D. Deep Reinforcement Learning from Human Preferences. *Advances in neural information processing systems* **30** (2017).
32. *ChatGPT: KI ist jetzt der natürlichen Ignoranz gewachsen - Onlineportal von IT Management.* de-DE. 2023-01. https://www.it-daily.net/it-sicherheit/cloud-security/chatgpt-ki-ist-jetzt-der-natuerlichen-ignoranz-gewachsen (2023-05-13).
33. *What Is The Difference Between InstructGPT And ChatGPT?.* en-US. 2023-05. https://www.theinsaneapp.com/2023/05/instructgpt-vs-chatgpt.html (2023-05-13).
34. Bai, Y. *et al.* Constitutional AI: Harmlessness from AI Feedback. *arXiv:2212.08073.* arXiv: 2212.08073 [cs]. http://arxiv.org/abs/2212.08073 (2023-05-11) (2022-12).

Glossary I

causal language model A causal language model predicts the likelihood of the next token based on a sequence of tokens (input). By sampling one of the predicted tokens and appending it to the input, output can be generated autoregressively. This in contrast to e.g. a masked language model. 34–37, 104

Falcon One of the LLMs used. Created by the Technology Innovation Institute (TII). 38–43, 86–92

GPT2 The second generation **G**enerative **P**retrained **T**ransformer LM from OpenAI [26]. 105

GPT3 The third generation **G**enerative **P**retrained **T**ransformer LM from OpenAI [27]. 105

GPT4 The fourth generation **G**enerative **P**retrained **T**ransformer LM from OpenAI [28]. Currently their most capable model. 86–92, 105

Glossary II

HuggingFace American deep learning ecosystem startup, having created the well established `transformers` framework which provides useful abstractions of most existing open-access Machine Learning models. 74–78

LLaMa A LLM from Meta. 38–43, 104, 105

LLaMa 2 One of the LLMs used. It is the successor of LLaMa, also created by Meta. 38–43

masked language model A masked language model predicts all masked (often missing) tokens in a sequence based on the context provided by the surrounding tokens. This in contrast to e.g. a causal language model. 86–92, 103

Meta Previously known as Facebook, Meta is a deep learning powerhose and regularly open-sources new state-of-the-art machine learning models. 104

Glossary III

Microsoft Tech Giant, well-known for its operating system. Microsoft recently started intensive cooperation with OpenAI through a $10 Billion USD investment, and started integrating GPT4 and other models throughout their services. 72

OpenAI American AI company, trailblazer at the frontier of scaling deep learning architectures and corresponding algorithmic breakthroughs. Their currently most well-known models are the Generative Pretrained Transformer (GPT) family of models, particularly GPT2, GPT3 and GPT4. 103, 105

Technology Innovation Institute Abu Dhabi-based machine learning research institute. 103

Vicuna One of the LLMs used. Based on LLaMa. 38–43

Acronyms I

BERT Bidirectional Encoder Representation from Transformers 18

GPT Generative Pretrained Transformer 105

GQA Grouped Query Attention 25–30

LLM Large Language Model 3–7, 20–24, 80–83, 86–92, 103–105, 108

LM Language Model 18, 103

ML Machine Learning 2

MLP Multi-Layer Perceptron 108

NER Named Entity Recognition 10, 18, 19, 86–92

NLP Natural Language Processing 9, 10

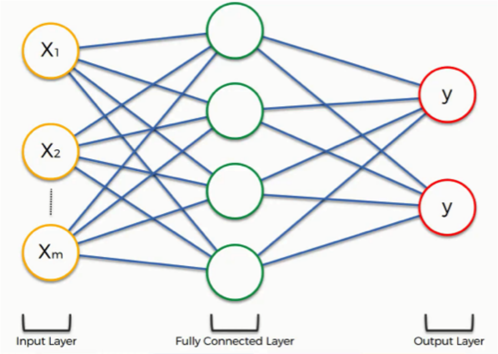
Acronyms II

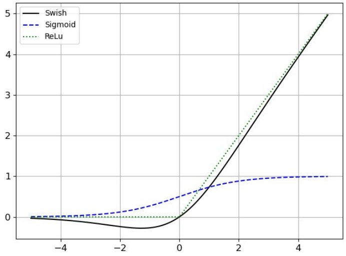
ReLU Rectified Linier Unit 25–30, 108

RoPE Rotary Positional Encoding 25–30

SwiGLU Swish Gated Linear Unit 25–30, 108

Multi-Layer Perceptron





Multi-Layer Perceptron (MLP) with one fully connected layer.
Alternative names include 'dense', 'fully connected' and 'mlp' layer.
Figure from [29].

Common activation function:
ReLU, or recently for LLMs:
SwiGLU.

InstructGPT: Following Instructions

“In human evaluations on our prompt distribution, outputs from the 1.3B parameter InstructGPT model are preferred to outputs from the 175B GPT-3, despite having 100x fewer parameters. Moreover, InstructGPT models show improvements in truthfulness and reductions in toxic output generation while having minimal performance regressions on public NLP datasets.” Ouyang et. al. 2022 [30]

Reinforcement Learning from Human Feedback

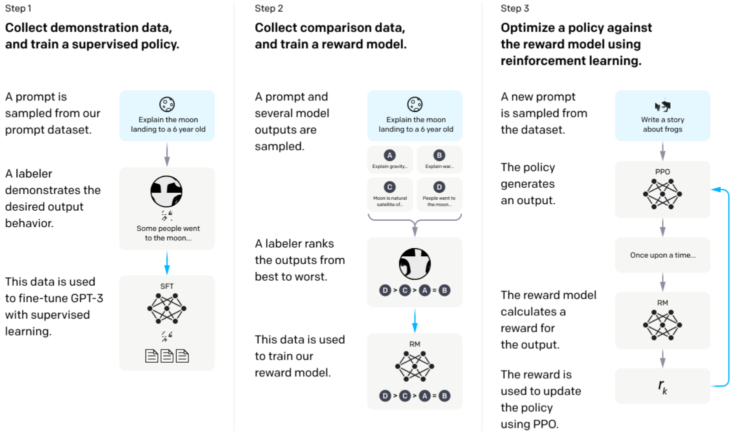


Image Source: [30]

RLHF originated from [31]

ChatGPT

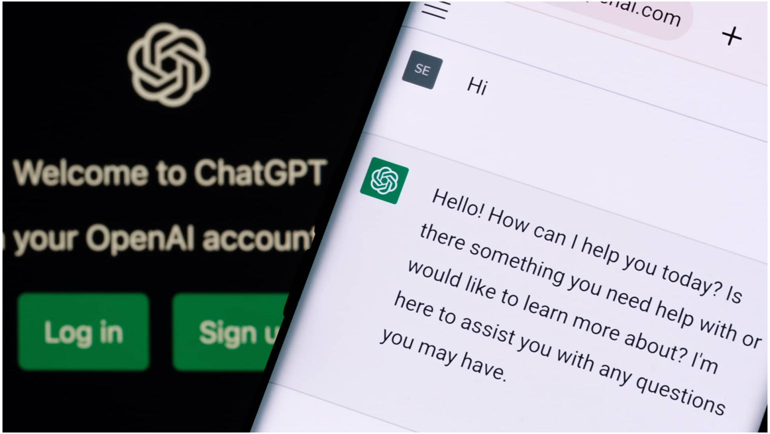


Image Source: [32]

ChatGPT Training Steps

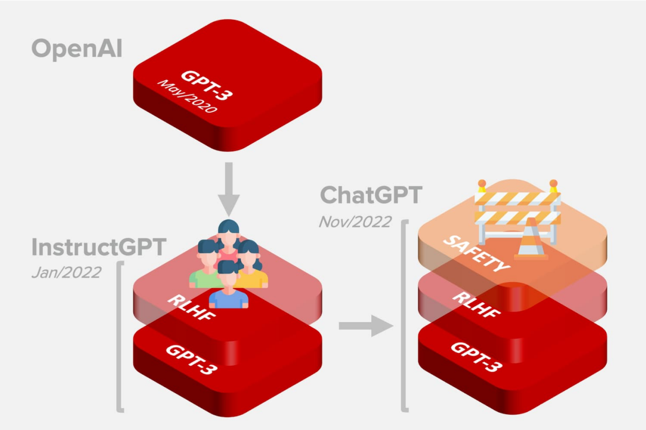


Image Source: [33]

Constitutional AI

1. Prompt LLM with questions illiciting ethically questionable responses
2. Ask it to "rewrite this to be more ethical"
3. Fine-Tune to prefer rewritten response
4. Repeat a few times

Constitutional AI

1. Prompt LLM with questions illiciting ethically questionable responses
2. Ask it to "rewrite this to be more ethical"
3. Fine-Tune to prefer rewritten response
4. Repeat a few times

Constitutional AI

1. Prompt LLM with questions illiciting ethically questionable responses
2. Ask it to "rewrite this to be more ethical"
3. Fine-Tune to prefer rewritten response
4. Repeat a few times

Constitutional AI

1. Prompt LLM with questions illiciting ethically questionable responses
2. Ask it to "rewrite this to be more ethical"
3. Fine-Tune to prefer rewritten response
4. Repeat a few times

Constitutional AI

1. Prompt LLM with questions illiciting ethically questionable responses
2. Ask it to "rewrite this to be more ethical"
3. Fine-Tune to prefer rewritten response
4. Repeat a few times

Constitutional Results

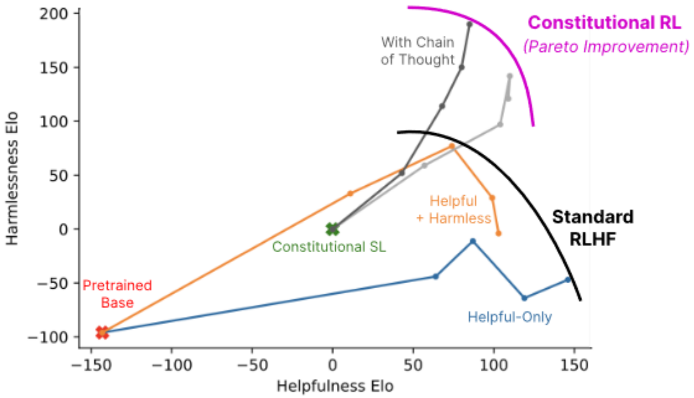


Image Source: [34]