

PRACTICAL TIPS FOR BOOTSTRAPPING INFORMATION EXTRACTION PIPELINES



You
Developer



Matthew Honnibal  Explosion

SPACY

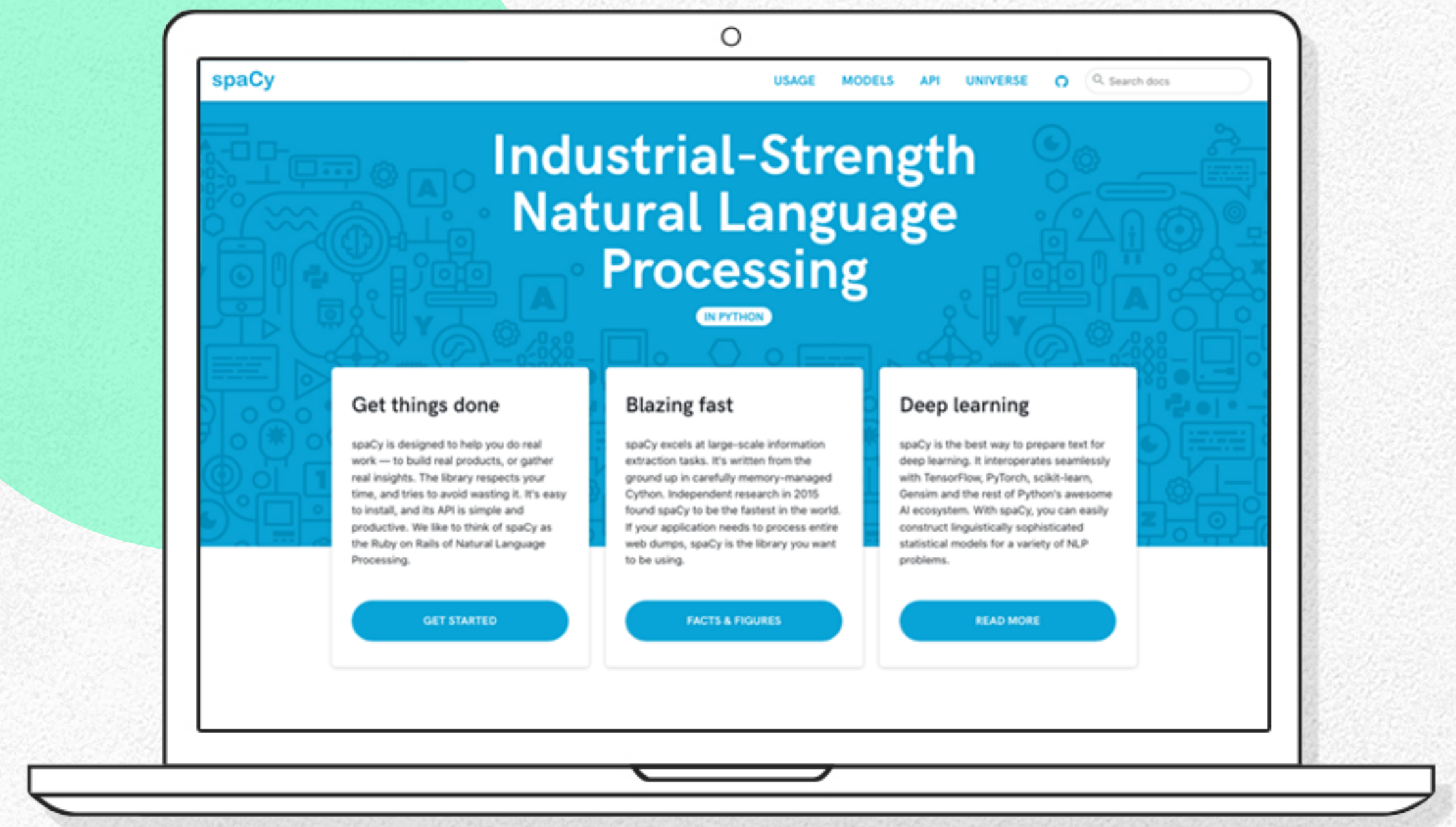
Open-source library for
industrial-strength natural
language processing

 spacy.io

EXPLOSION

250M+

downloads



SPACY

Open-source library for
industrial-strength natural
language processing

 spacy.io

EXPLOSION

250M+

downloads

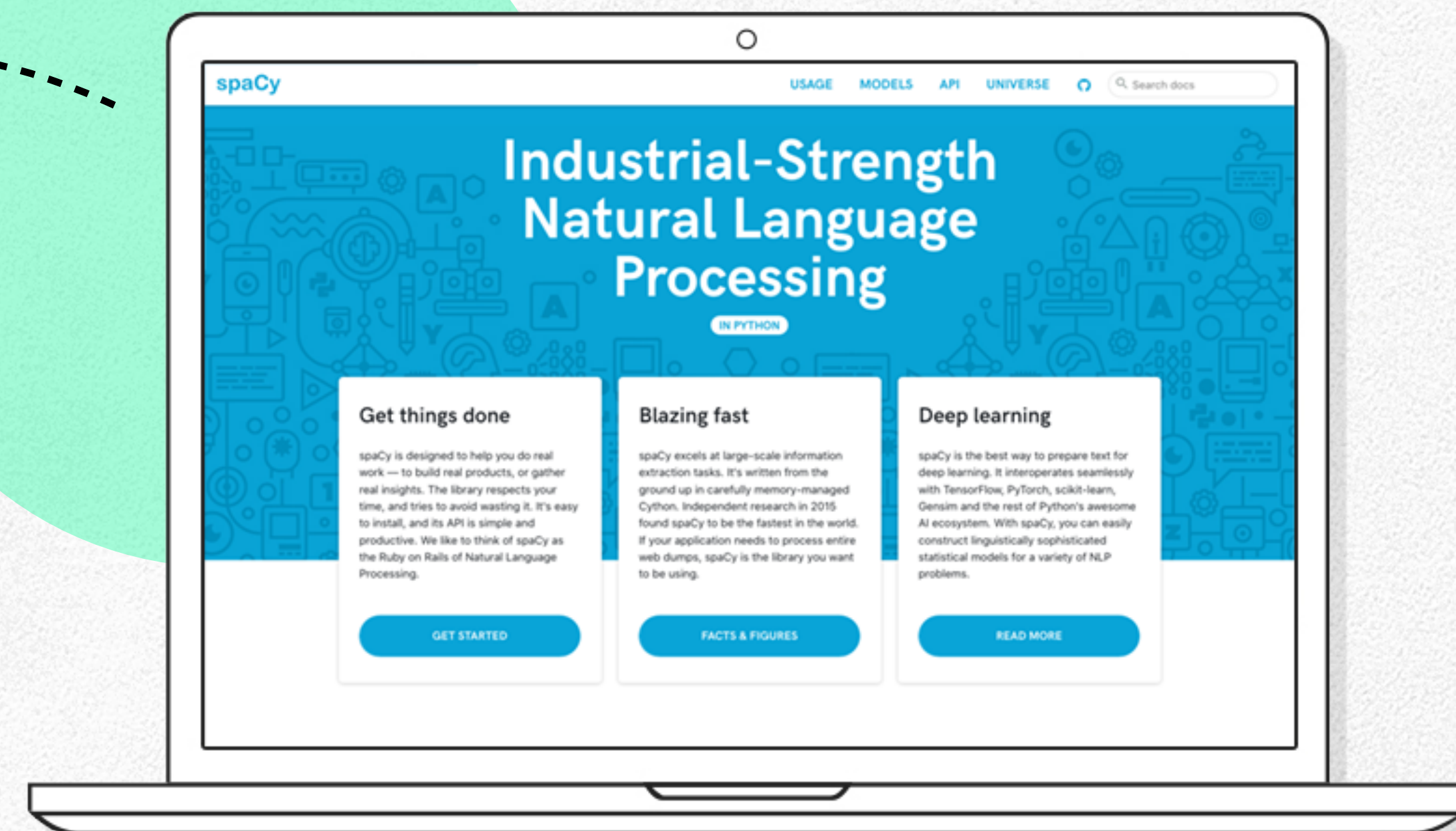
ChatGPT can
write spaCy code!

Write patterns to match US addresses using spaCy's SpanRuler

Sure, here are some patterns to match US addresses using spaCy's SpanRuler class:

1. Match street addresses:

```
sql
pattern = [{"TEXT": {"REGEX": "^[0-9]+[a-zA-Z]?(-[0-9]+[a-zA-Z]?)?$",
  "IS_SPACE": True},
  {"IS_SPACE": True, "OP": "+"},
  {"IS_SPACE": True},
  {"TEXT": {"REGEX": "^(?i)(street|st|avenue|ave|road|rd|boulevard
```



PRODIGY

Modern scriptable annotation
tool for machine learning
developers

 prodigy.ai

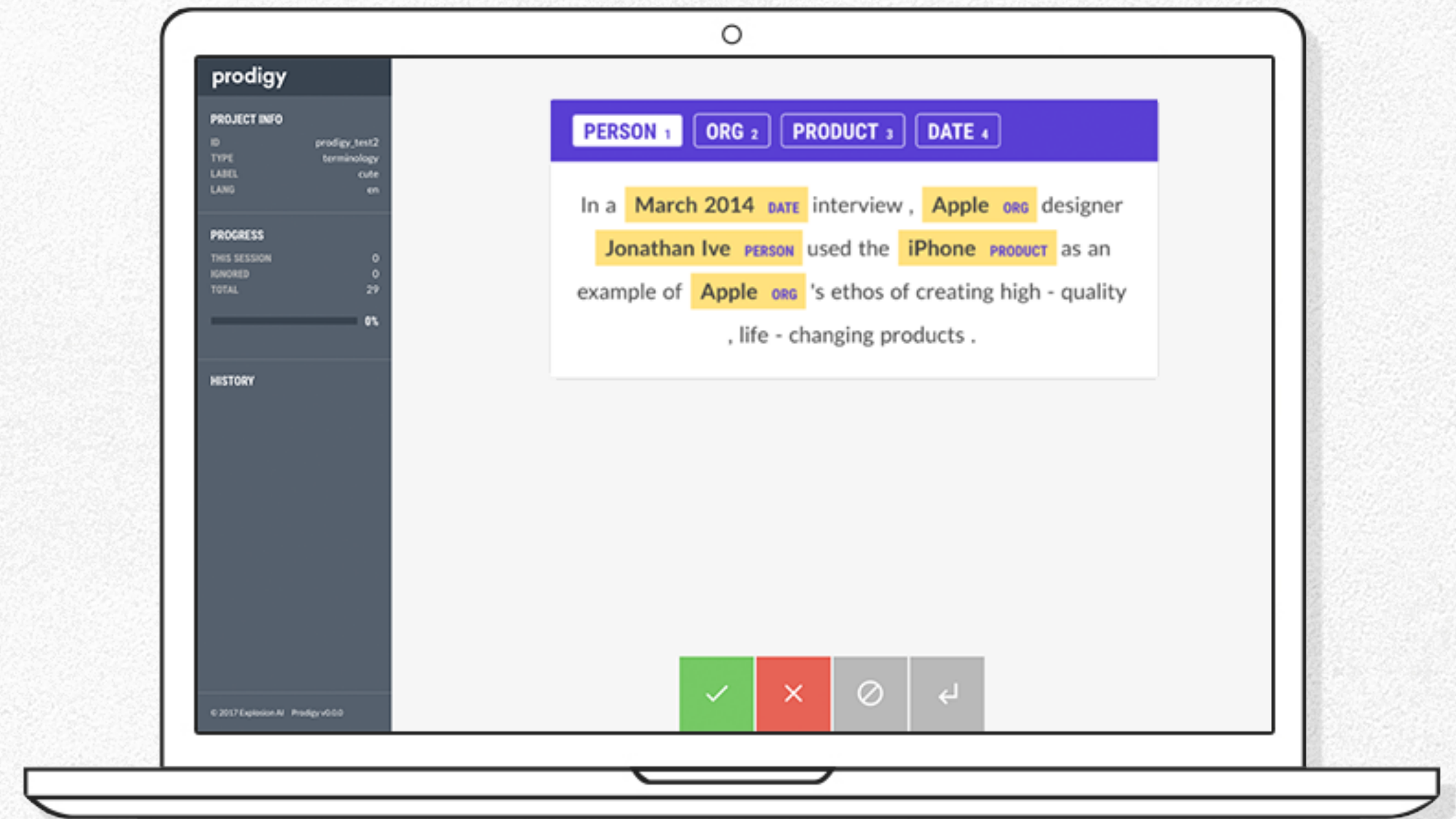


10K+

users

900+

companies



PRODIGY

Modern scriptable annotation
tool for machine learning
developers

 prodigy.ai

EXPLOSION

10K+

users

900+

companies

```
@prodigy.recipe("my-custom-recipe")
def my_custom_recipe(dataset, view_id="text"):
    stream = load_my_custom_stream()

    def update(examples):
        print(f"Received {len(examples)} annotations!")

    return {
        "dataset": dataset,
        "view_id": view_id,
        "stream": stream,
        "update": update
    }
```

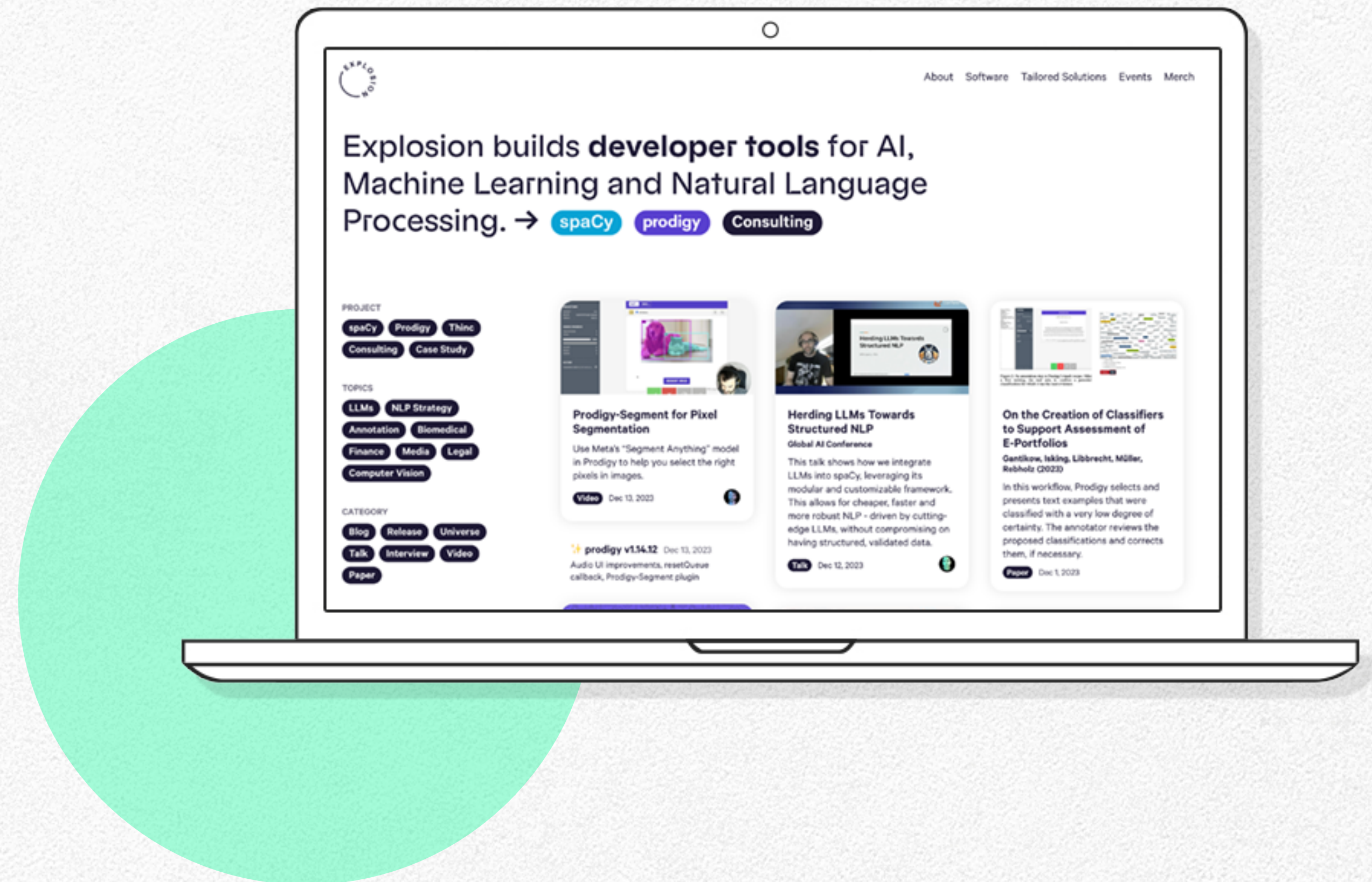


Alex Smith
Developer



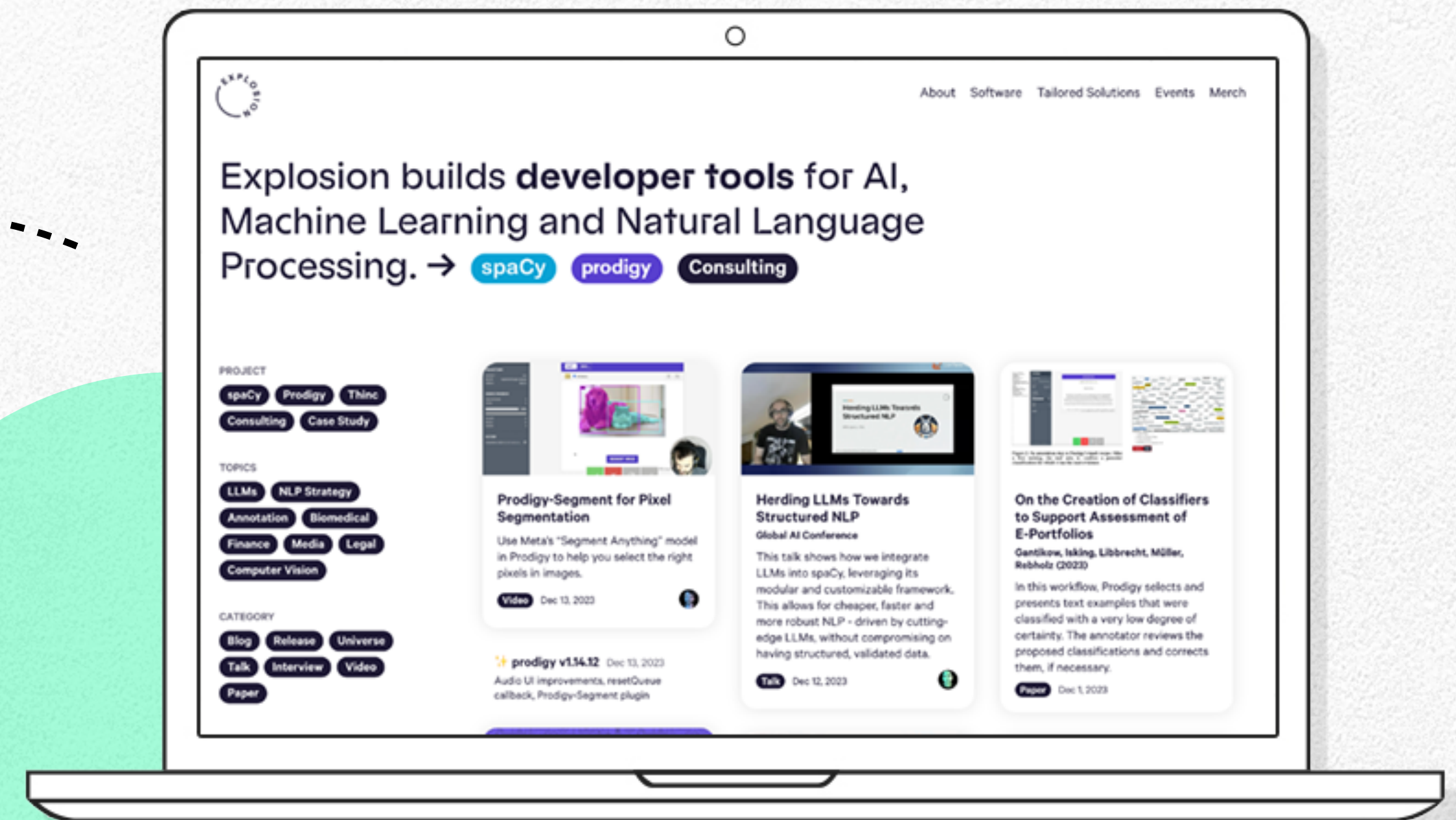
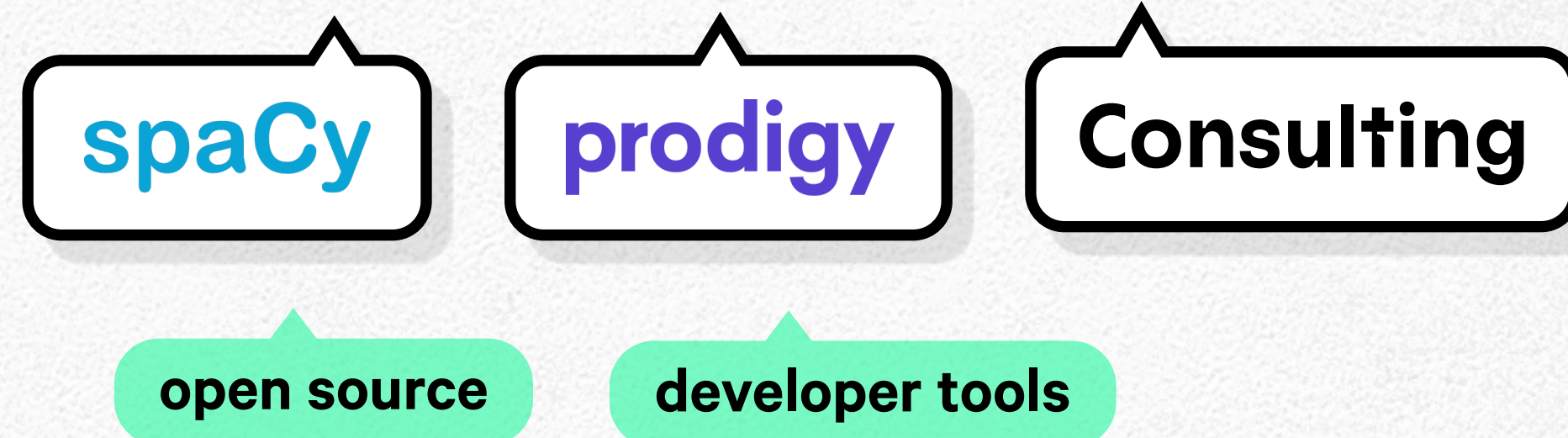
BACK TO OUR ROOTS

We're back to running Explosion as a smaller, independent-minded and self-sufficient company.



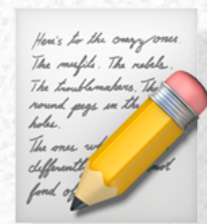
BACK TO OUR ROOTS

We're back to running Explosion as a smaller, independent-minded and self-sufficient company.



WHAT I MEAN BY INFORMATION EXTRACTION

WHAT I MEAN BY INFORMATION EXTRACTION



Turn text into data. Make a database from earnings reports, or skills in job postings, or product feedback in social media – many more.

EXPLOSION

WHAT I MEAN BY INFORMATION EXTRACTION



Turn text into data. Make a database from earnings reports, or skills in job postings, or product feedback in social media – many more.



Lots of subtasks. Text classification, named entity recognition, entity linking, relation extraction can all be part of an information extraction pipeline.

WHAT I MEAN BY INFORMATION EXTRACTION



Turn text into data. Make a database from earnings reports, or skills in job postings, or product feedback in social media – many more.



Lots of subtasks. Text classification, named entity recognition, entity linking, relation extraction can all be part of an information extraction pipeline.



Mostly static schema. Most people are solving one problem at a time, so that's what I'll focus on.

Database

```
{
  "text": "Hooli raises $5m to revolutionize search, led by ACME Ventures",
  "label": "FUNDRAISING",
  "entities": [
    {
      "text": "Hooli",
      "label": "COMPANY",
      "org_name": "Hooli, Inc.",
      "crunchbase": "hooli",
      "db_id": 5923214
    },
    {
      "text": "ACME Ventures",
      "label": "COMPANY",
      "org_name": "ACME Ventures",
      "crunchbase": "acmevc",
      "db_id": 1681056
    },
    {
      "text": "$5m",
      "label": "MONEY",
      "value": 5000000,
      "currency": "USD"
    }
  ],
  "relations": [
    { "head_id": 1681056, "child_id": 5923214, "label": "INVESTOR" }
  ]
}
```

“Hooli raises \$5m to revolutionize search, led by ACME Ventures”

Database

```
{
  "text": "Hooli raises $5m to revolutionize search, led by ACME Ventures",
  "label": "FUNDRAISING",
  "entities": [
    {
      "text": "Hooli",
      "label": "COMPANY",
      "org_name": "Hooli, Inc.",
      "crunchbase": "hooli",
      "db_id": 5923214
    },
    {
      "text": "ACME Ventures",
      "label": "COMPANY",
      "org_name": "ACME Ventures",
      "crunchbase": "acmevc",
      "db_id": 1681056
    },
    {
      "text": "$5m",
      "label": "MONEY",
      "value": 5000000,
      "currency": "USD"
    }
  ],
  "relations": [
    { "head_id": 1681056, "child_id": 5923214, "label": "INVESTOR" }
  ]
}
```

named entity recognition

COMPANY

“Hooli raises \$5m to revolutionize search, led by ACME Ventures”

COMPANY

Database

```
{
  "text": "Hooli raises $5m to revolutionize search, led by ACME Ventures",
  "label": "FUNDRAISING",
  "entities": [
    {
      "text": "Hooli",
      "label": "COMPANY",
      "org_name": "Hooli, Inc.",
      "crunchbase": "hooli",
      "db_id": 5923214
    },
    {
      "text": "ACME Ventures",
      "label": "COMPANY",
      "org_name": "ACME Ventures",
      "crunchbase": "acmevc",
      "db_id": 1681056
    },
    {
      "text": "$5m",
      "label": "MONEY",
      "value": 5000000,
      "currency": "USD"
    }
  ],
  "relations": [
    { "head_id": 1681056, "child_id": 5923214, "label": "INVESTOR" }
  ]
}
```

named entity recognition

currency normalization

COMPANY MONEY

“Hooli raises \$5m to revolutionize search, led by ACME Ventures”

COMPANY

Database

```

{
  "text": "Hooli raises $5m to revolutionize search, led by ACME Ventures",
  "label": "FUNDRAISING",
  "entities": [
    {
      "text": "Hooli",
      "label": "COMPANY",
      "org_name": "Hooli, Inc.",
      "crunchbase": "hooli",
      "db_id": 5923214
    },
    {
      "text": "ACME Ventures",
      "label": "COMPANY",
      "org_name": "ACME Ventures",
      "crunchbase": "acmevc",
      "db_id": 1681056
    },
    {
      "text": "$5m",
      "label": "MONEY",
      "value": 5000000,
      "currency": "USD"
    }
  ],
  "relations": [
    { "head_id": 1681056, "child_id": 5923214, "label": "INVESTOR" }
  ]
}

```

named entity recognition

entity disambiguation

custom database lookup

currency normalization

COMPANY 5923214

MONEY

“Hooli raises \$5m to revolutionize search, led by ACME Ventures”

COMPANY 1681056

Database

```
{
  "text": "Hooli raises $5m to revolutionize search, led by ACME Ventures",
  "label": "FUNDRAISING",
  "entities": [
    {
      "text": "Hooli",
      "label": "COMPANY",
      "org_name": "Hooli, Inc.",
      "crunchbase": "hooli",
      "db_id": 5923214
    },
    {
      "text": "ACME Ventures",
      "label": "COMPANY",
      "org_name": "ACME Ventures",
      "crunchbase": "acmevc",
      "db_id": 1681056
    },
    {
      "text": "$5m",
      "label": "MONEY",
      "value": 5000000,
      "currency": "USD"
    }
  ],
  "relations": [
    {
      "head_id": 1681056,
      "child_id": 5923214,
      "label": "INVESTOR"
    }
  ]
}
```

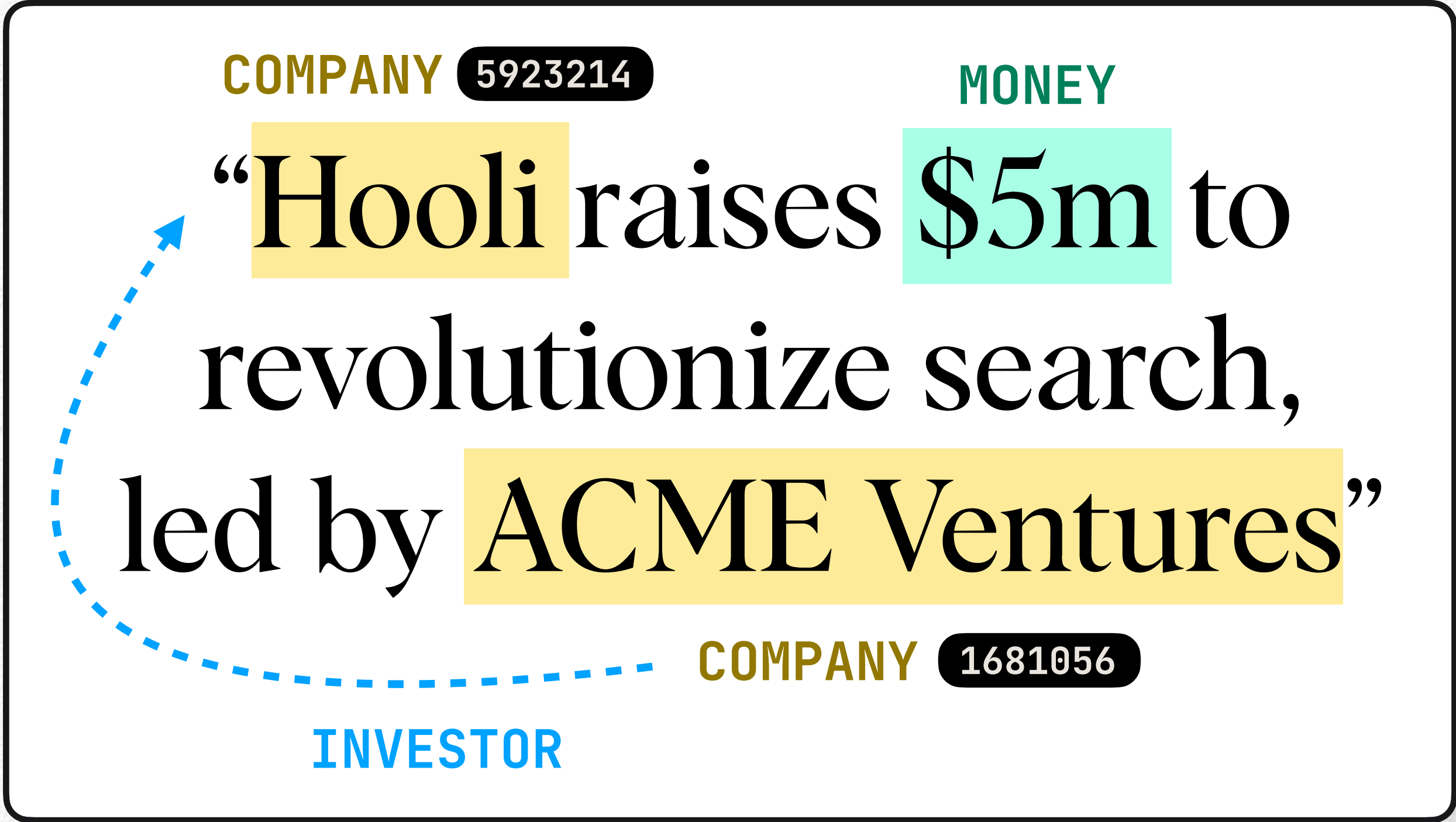
named entity recognition

entity disambiguation

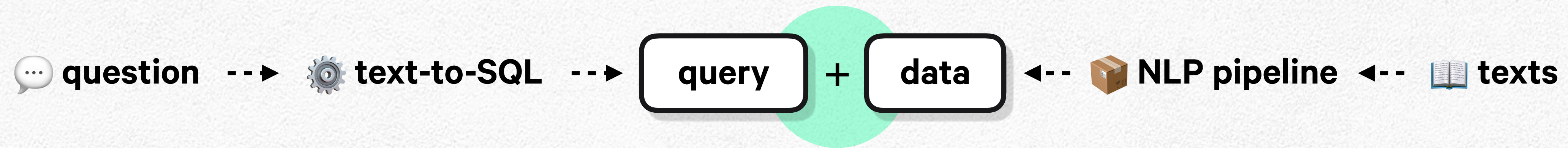
custom database lookup

currency normalization

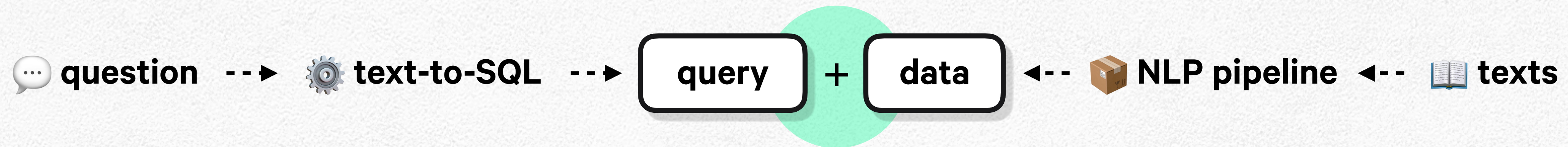
entity relation extraction



RIE: RETRIEVAL VIA INFORMATION EXTRACTION

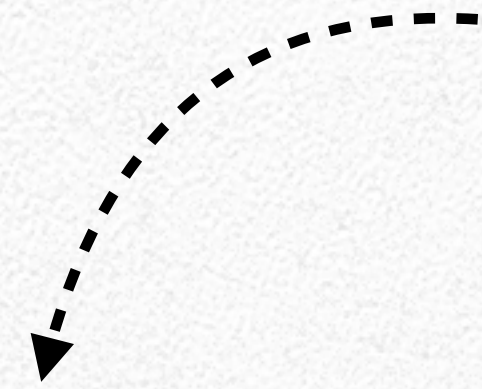


RIE: RETRIEVAL VIA INFORMATION EXTRACTION



RAG: RETRIEVAL-AUGMENTED GENERATION



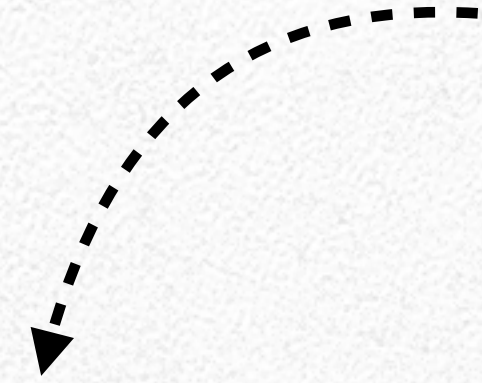


TALK OUTLINE



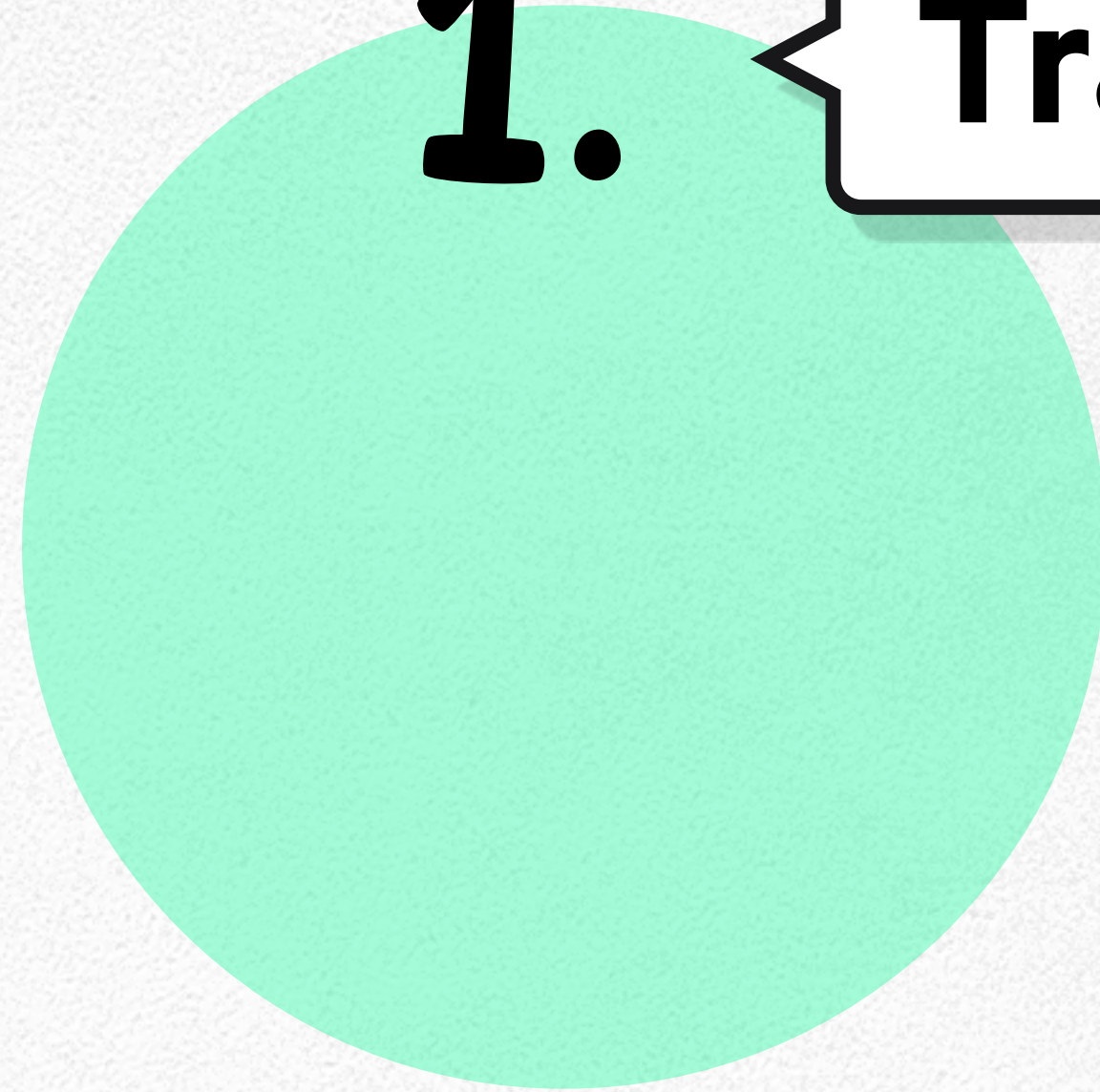


TALK OUTLINE



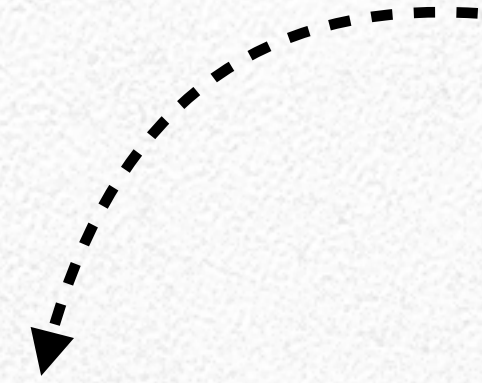
1.

Training tips





TALK OUTLINE

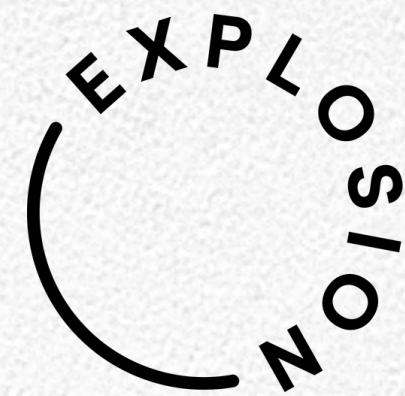


1.

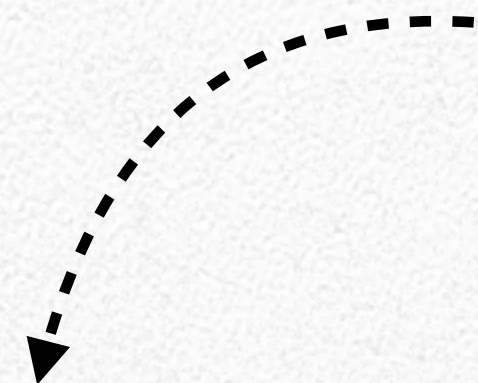
Training tips

2.

Modelling tips



TALK OUTLINE



1.

Training tips

2.

Modelling tips

3.

Data annotation tips

**SUPERVISED
LEARNING
IS STILL VERY
STRONG**

Example data is super powerful.

SUPERVISED LEARNING IS STILL VERY STRONG

Example data can
do things that
instructions can't.

Example data is super powerful.

SUPERVISED LEARNING IS STILL VERY STRONG

Example data can
do things that
instructions can't.

In-context
learning can't use
examples scalably.

Example data is super powerful.

KNOW YOUR ENEMIES

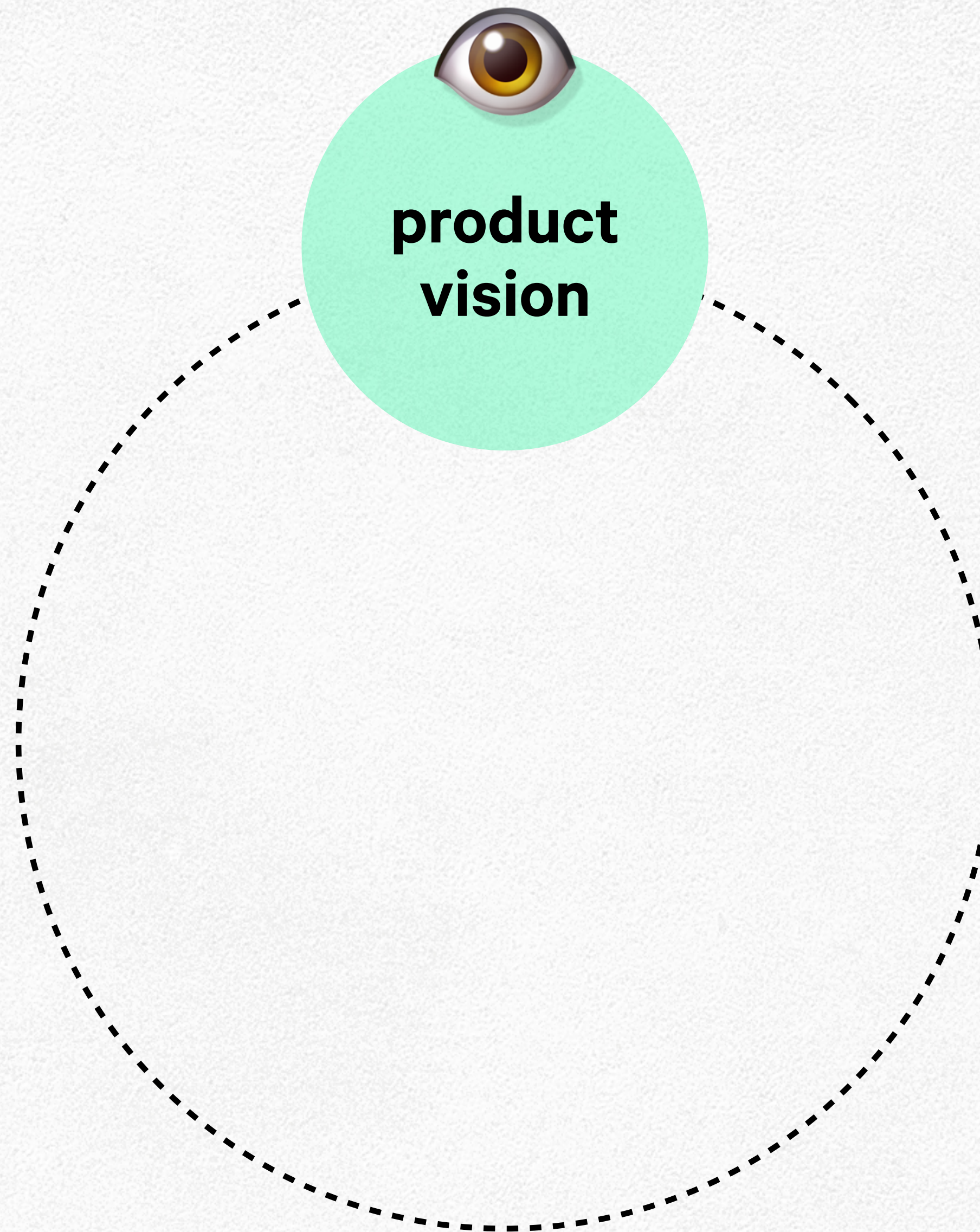
What makes
supervised
learning hard?



KNOW YOUR ENEMIES

What makes supervised learning hard?

chicken-and-egg problem

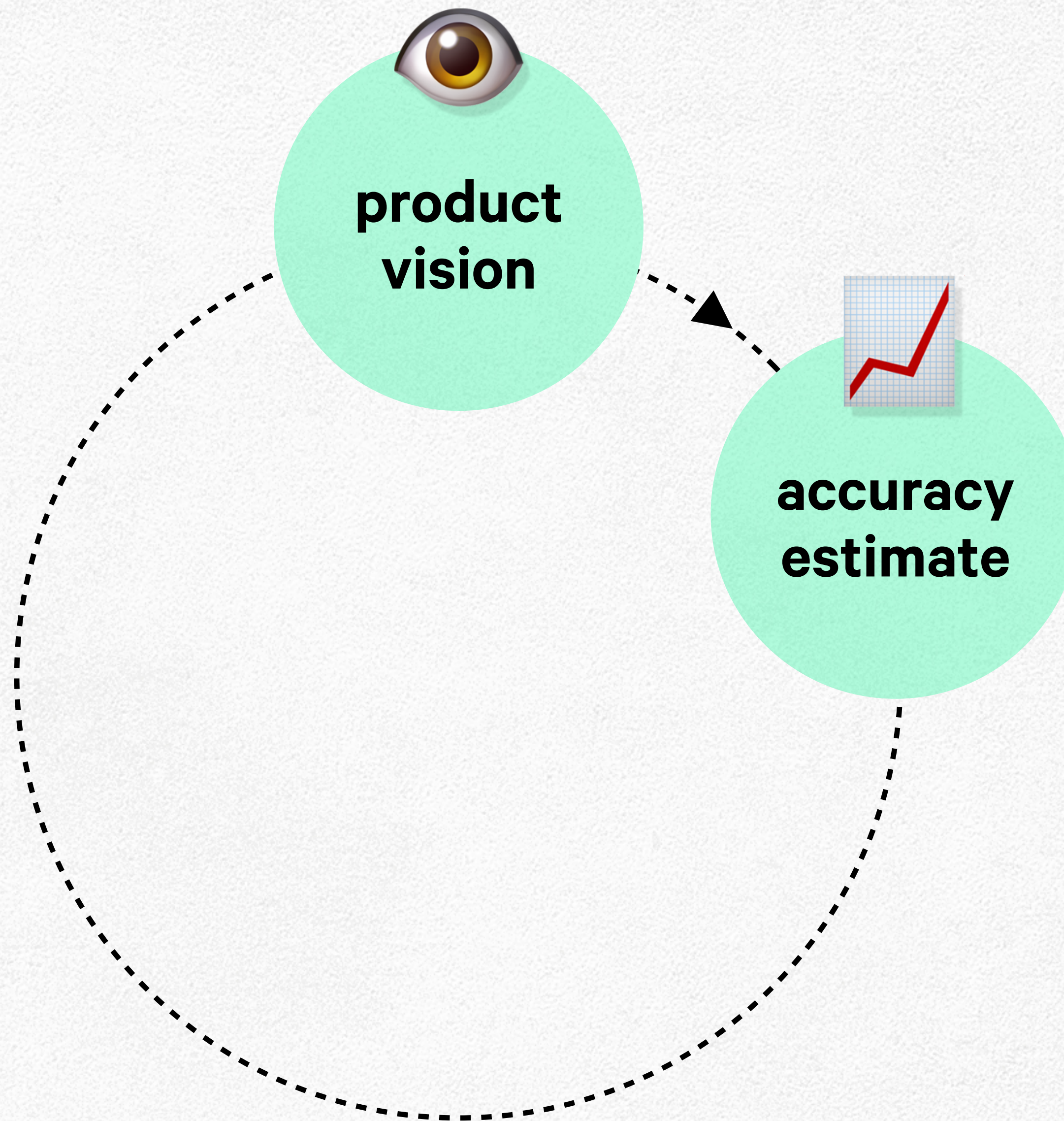


EXPLOSION

KNOW YOUR ENEMIES

What makes supervised learning hard?

chicken-and-egg problem

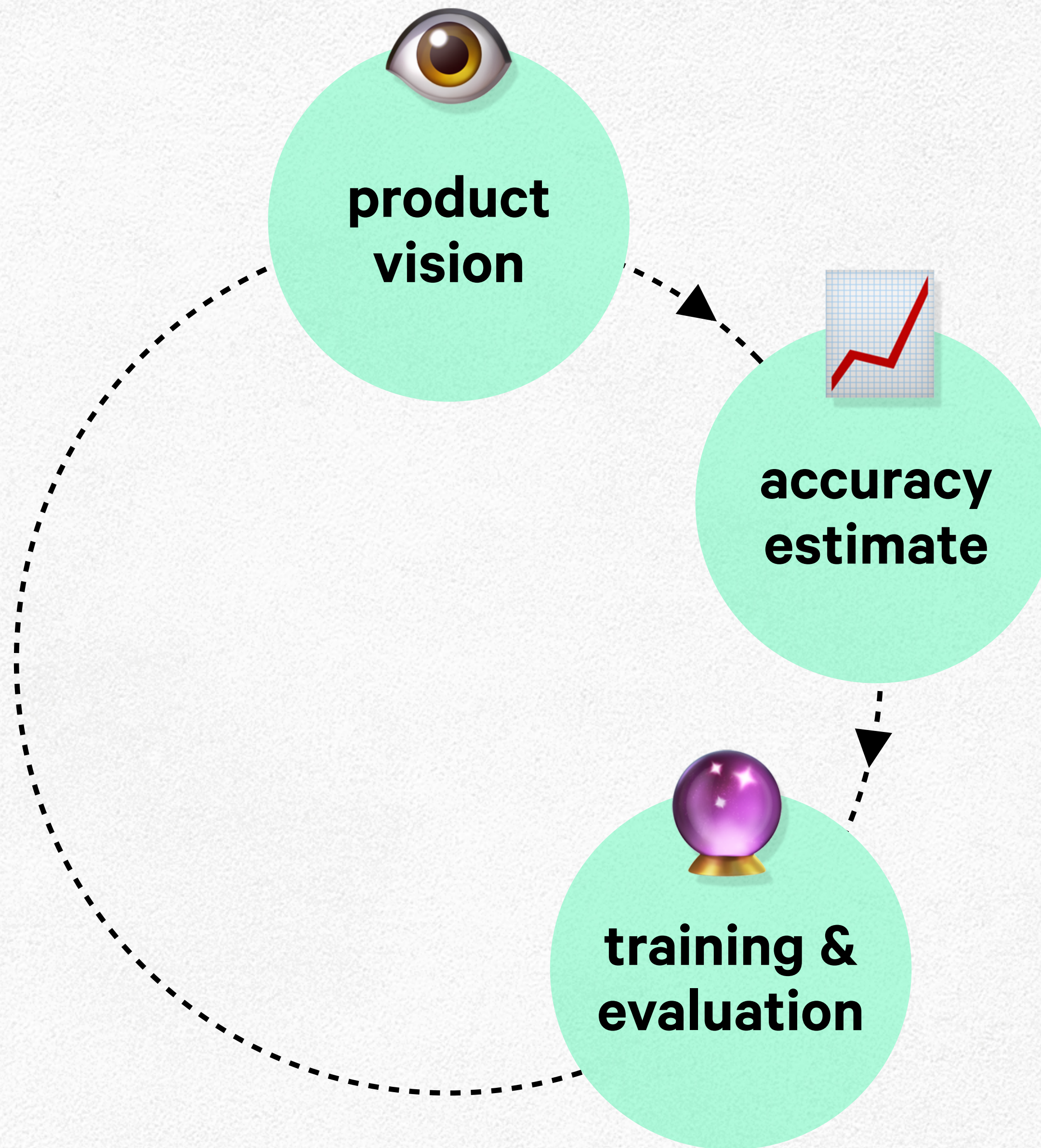


EXPLOSION

KNOW YOUR ENEMIES

What makes supervised learning hard?

chicken-and-egg problem

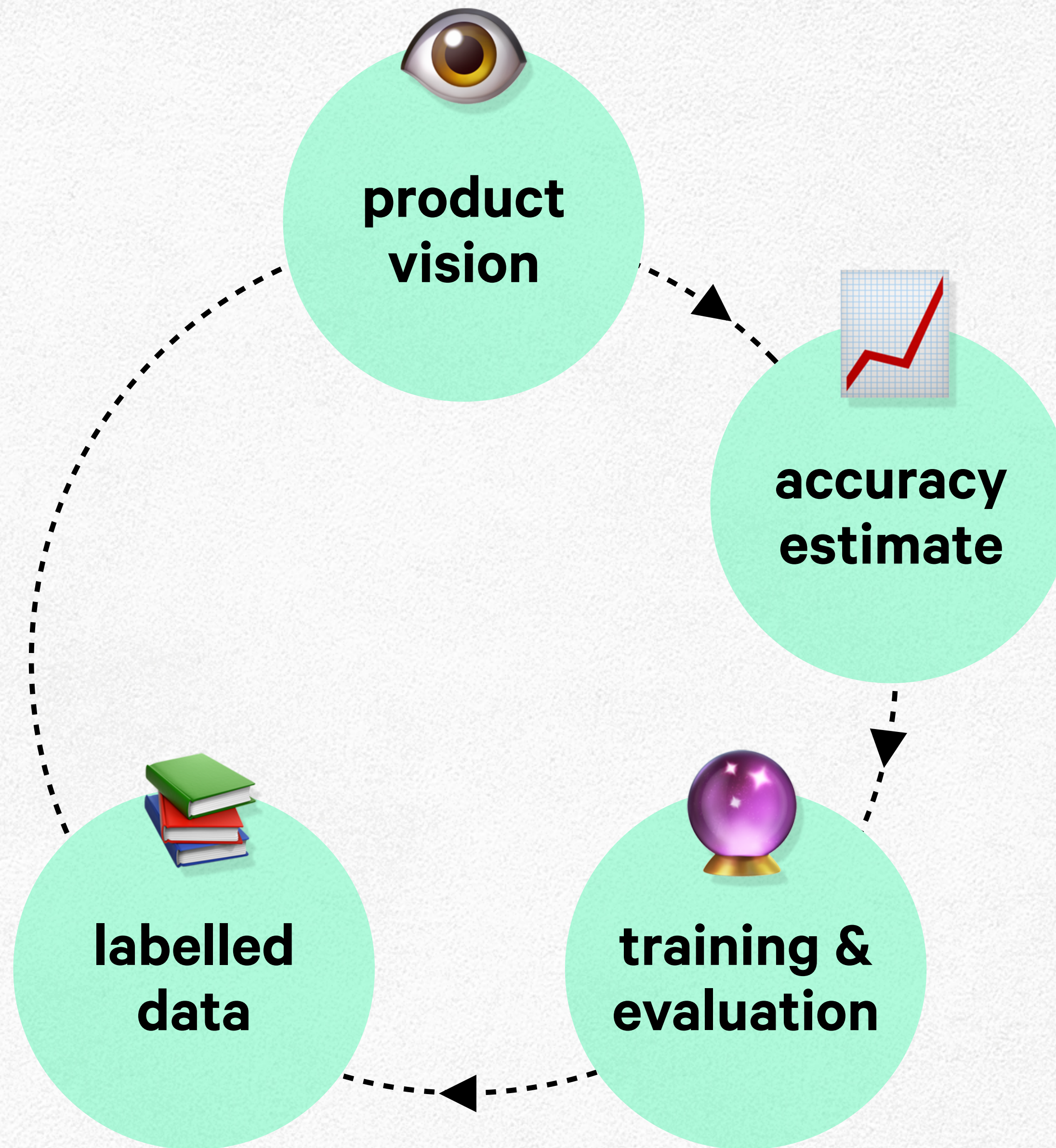


EXPLOSION

KNOW YOUR ENEMIES

What makes supervised learning hard?

chicken-and-egg problem

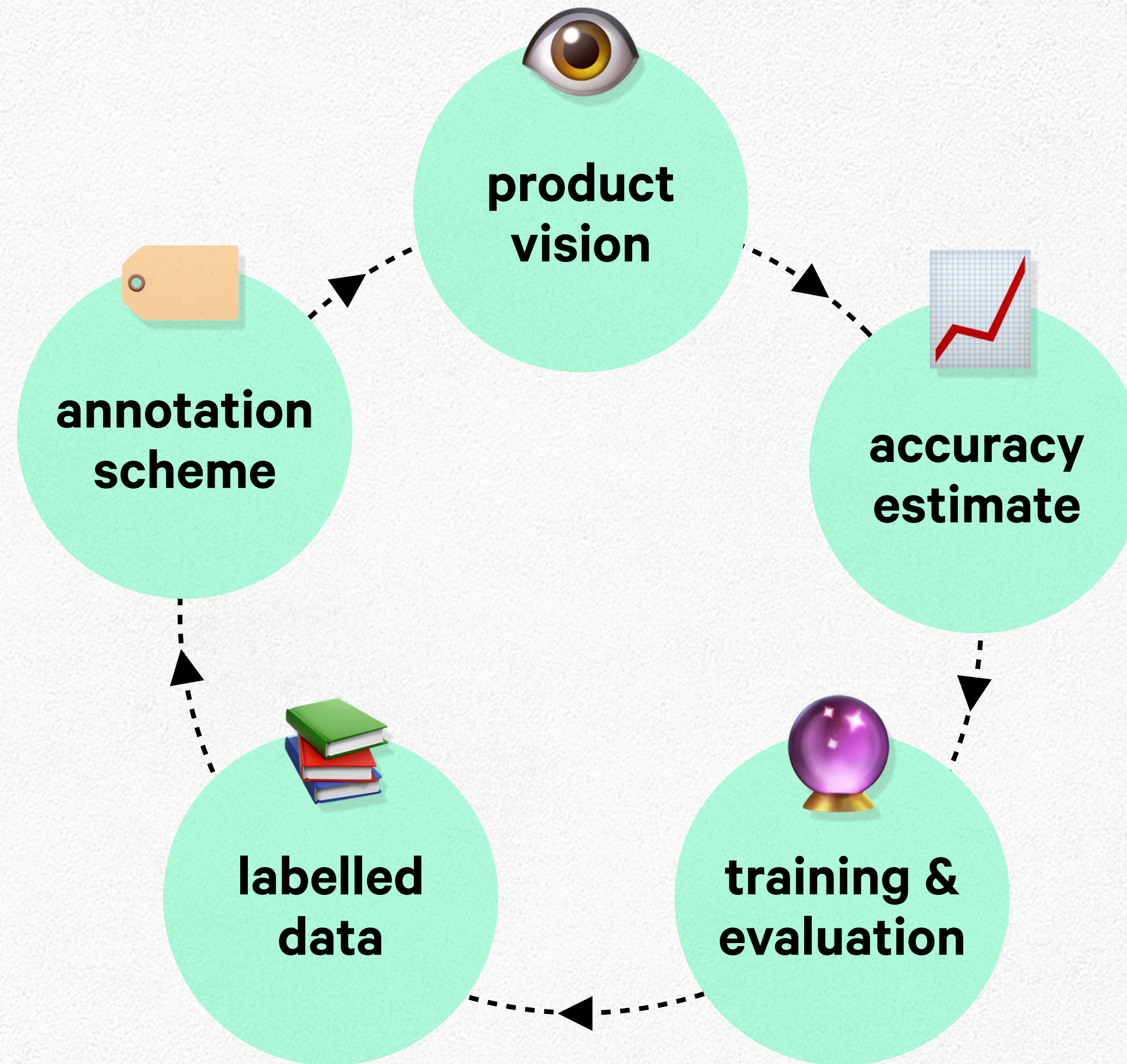


EXPLOSION

KNOW YOUR ENEMIES

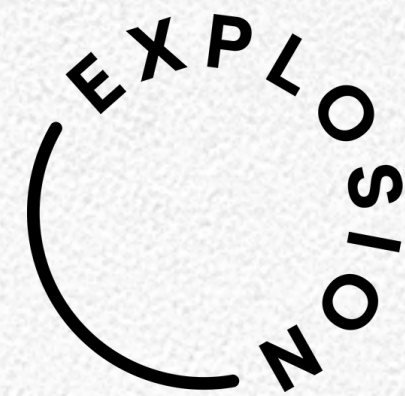
What makes supervised learning hard?

chicken-and-egg problem



EXPLOSION

**RESULTS ARE
HARD TO
INTERPRET**



**RESULTS ARE
HARD TO
INTERPRET**



Model doesn't train at all.
Is the data messed up somehow?

EXPLOSION

RESULTS ARE HARD TO INTERPRET



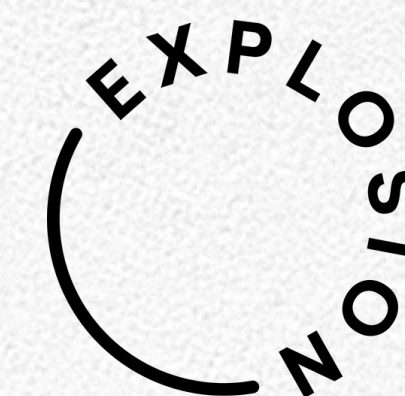
Model doesn't train at all.
Is the data messed up somehow?



Model learns barely better than chance.
Could be data, hyper-parameters, modelling...

EXPLOSION

RESULTS ARE HARD TO INTERPRET



Model doesn't train at all.
Is the data messed up somehow?

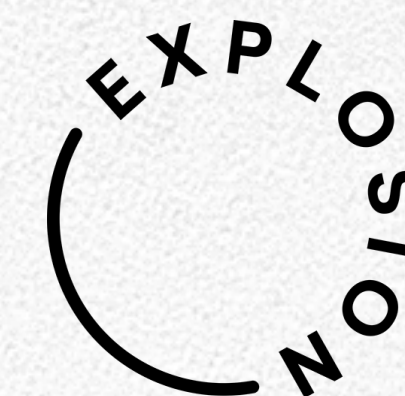


Model learns barely better than chance.
Could be data, hyper-parameters, modelling...



Results are decent!
But can it be better? How do I know if I'm missing out?

RESULTS ARE HARD TO INTERPRET



Model doesn't train at all.
Is the data messed up somehow?



Model learns barely better than chance.
Could be data, hyper-parameters, modelling...



Results are decent!
But can it be better? How do I know if I'm missing out?



Results are too good to be true.
Probably messed up the data...

PRACTICAL



TIPS

Training

FORM AND FALSIFY HYPOTHESES



HYPOTHESIS

◀ This is the bit that's broken.



HYPOTHESIS

◀ This is the bit that's broken.

QUESTION

◀ If this bit is broken, what should I expect to see?

HYPOTHESIS

◀ This is the bit that's broken.

QUESTION

◀ If this bit is broken, what should I expect to see?

TEST

◀ Is that what actually happens?



"I can't connect to this site."

HYPOTHESIS

This is the bit that's broken.

QUESTION

If this bit is broken, what should I expect to see?

TEST

Is that what actually happens?

SOLUTION MINDSET

“I can’t connect to this site.”

“Maybe it’ll work if I reconnect to the wi-fi or if I restart my router.”



HYPOTHESIS

This is the bit that’s broken.

QUESTION

If this bit is broken, what should I expect to see?

TEST

Is that what actually happens?

SOLUTION MINDSET

“I can’t connect to this site.”

SCIENTIFIC MINDSET



“Maybe it’ll work if I reconnect to the wi-fi or if I restart my router.”

“If the problem is between me and the site, other sites won’t load either. If the problem is between me and the router, I won’t be able to ping it.”

HYPOTHESIS

This is the bit that’s broken.

QUESTION

If this bit is broken, what should I expect to see?

TEST

Is that what actually happens?

EXAMPLES OF DEBUGGING TRAINING



EXAMPLES OF DEBUGGING TRAINING



What happens if I train on a tiny amount of data?
Does the model converge?

EXAMPLES OF DEBUGGING TRAINING



**What happens if I train on a tiny amount of data?
Does the model converge?**



**What happens if I randomize the training labels?
Does the model still learn?**

EXAMPLES OF DEBUGGING TRAINING



**What happens if I train on a tiny amount of data?
Does the model converge?**



**What happens if I randomize the training labels?
Does the model still learn?**



Are my model weights changing at all during training?

EXAMPLES OF DEBUGGING TRAINING



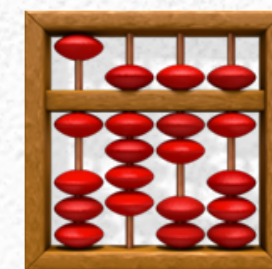
**What happens if I train on a tiny amount of data?
Does the model converge?**



**What happens if I randomize the training labels?
Does the model still learn?**



Are my model weights changing at all during training?

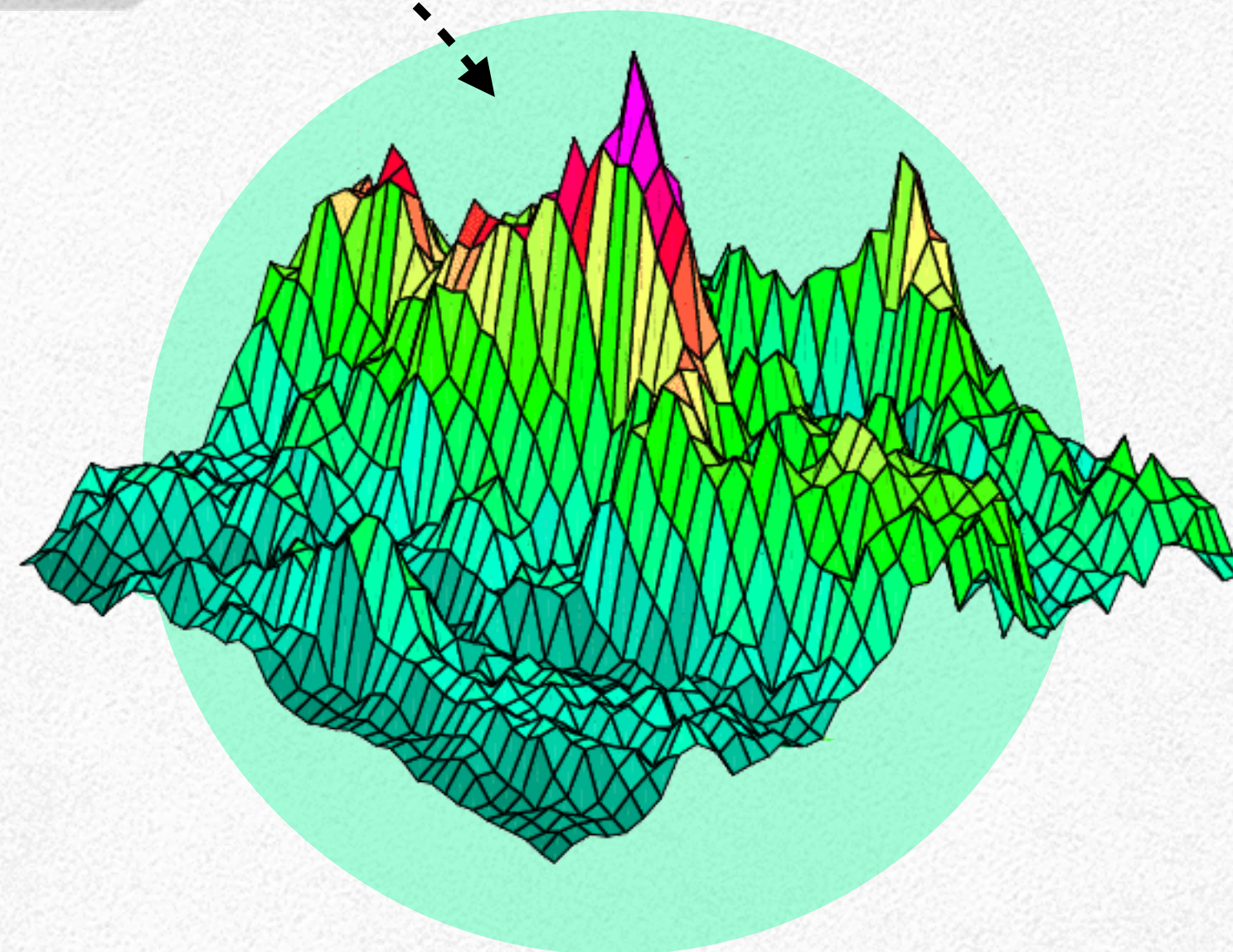


What's the mean and variance of my gradients?

**PRIORITIZE
ROBUSTNESS
NOT ACCURACY**



Better needs to look **better**.
You need it to not be like this:

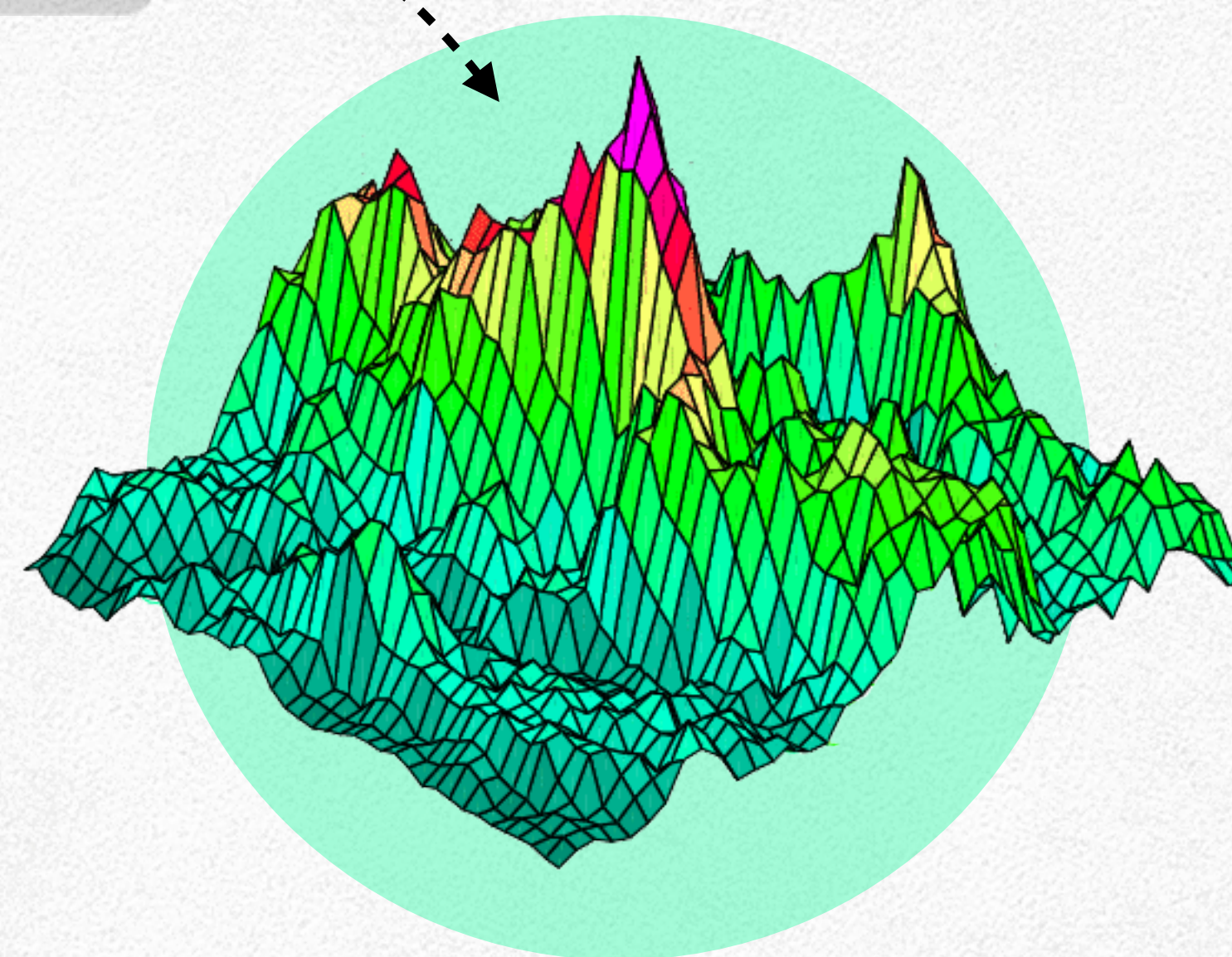




Better needs to look **better**.
You need it to not be like this:



Larger models are
often **less practical**.





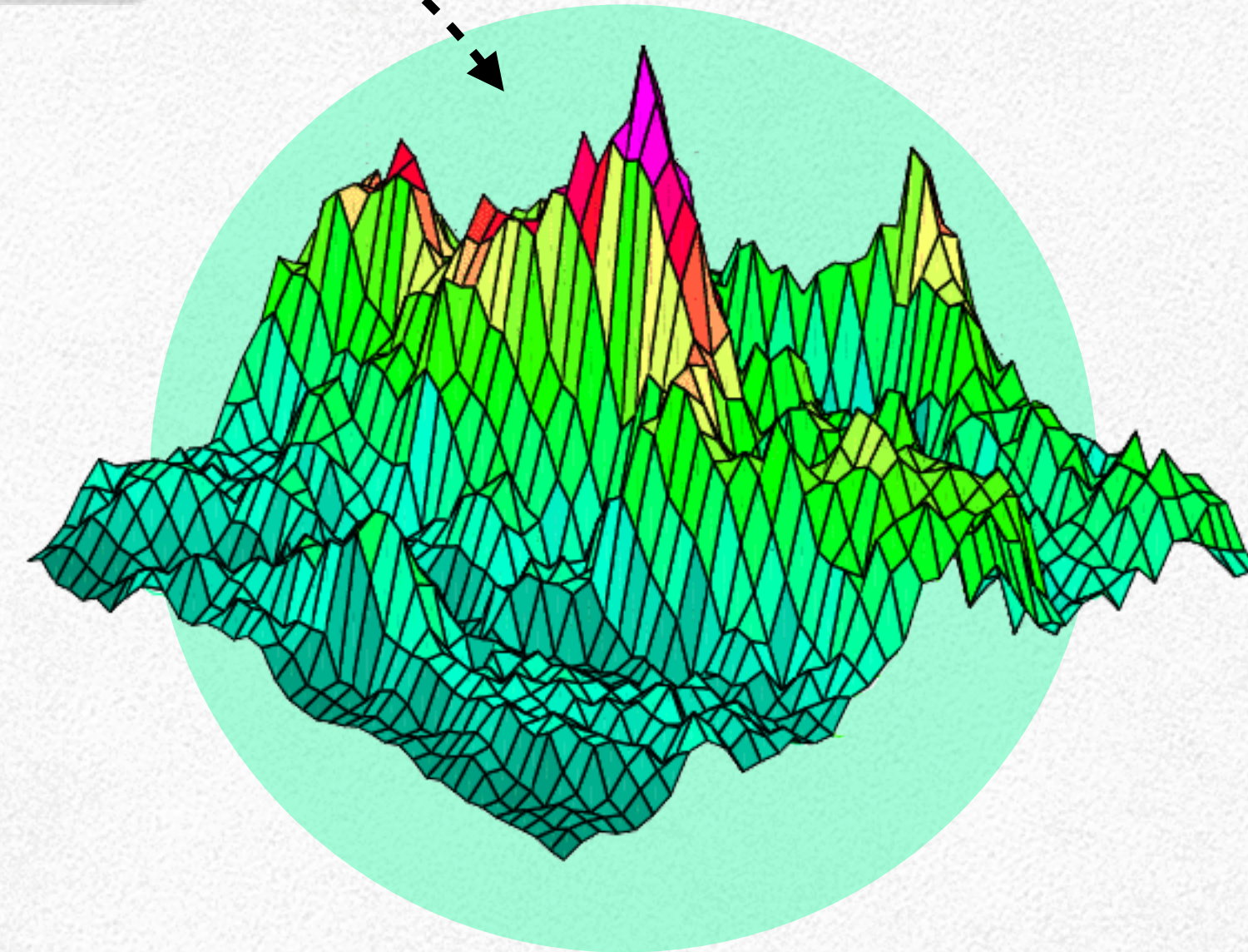
Better needs to look **better**.
You need it to not be like this:



Larger models are
often **less practical**.



You need it to work
with **small samples**.





Better needs to look **better**.
You need it to not be like this:



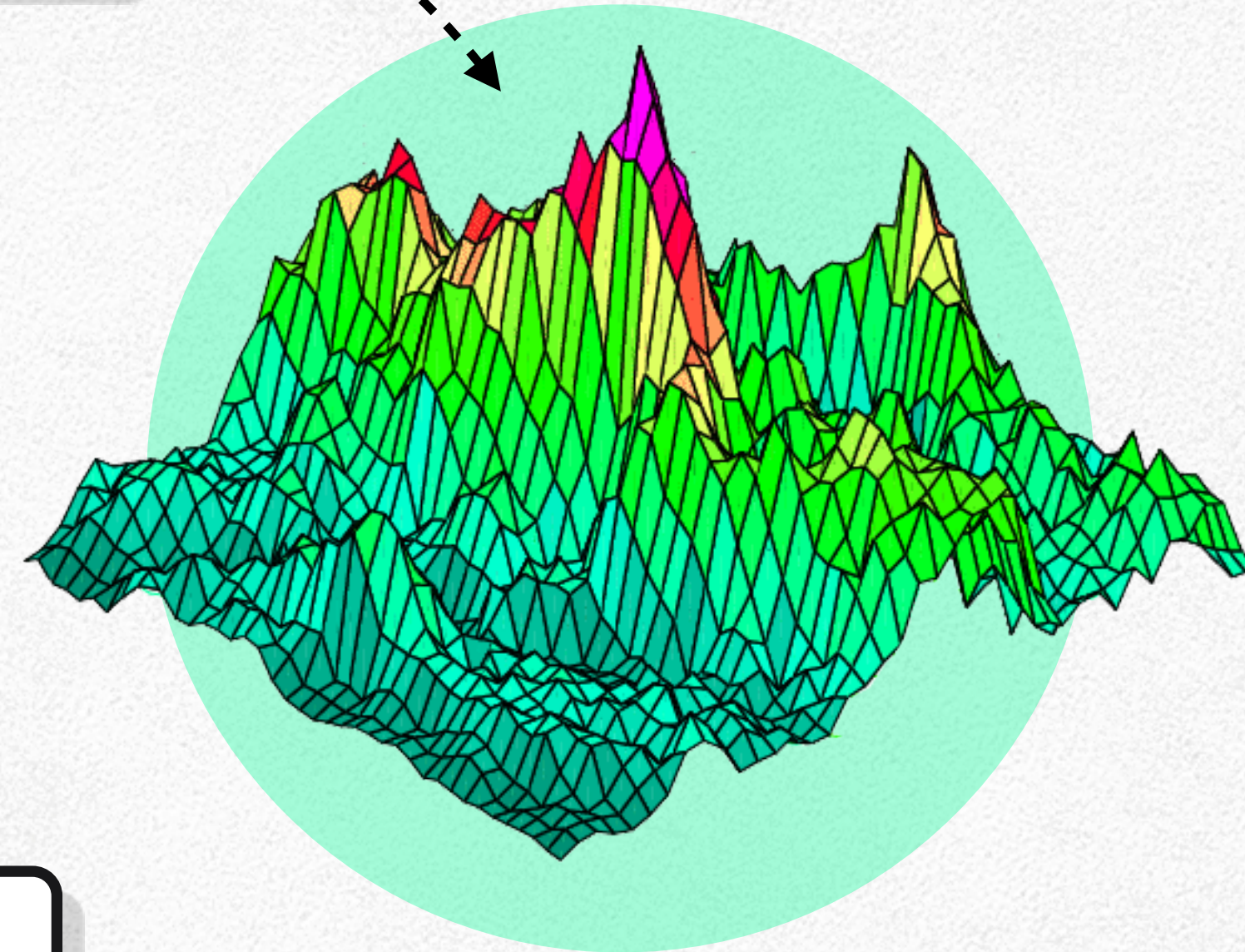
Larger models are
often **less practical**.



You need it to work
with **small samples**.



Large models are less stable
with **small batch sizes**.



PRACTICAL



TIPS

Modelling

ITERATE ON
YOUR DATA AND
SCALE DOWN

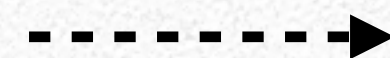
PROTOTYPE



PROTOTYPE

💬 prompt ----->

📖 text ----->



task-
specific
output

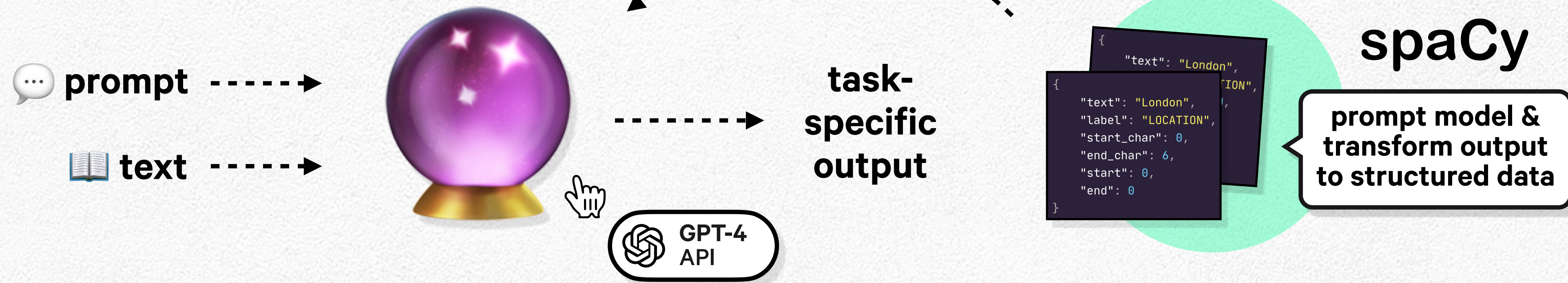


spaCy

prompt model &
transform output
to structured data



PROTOTYPE

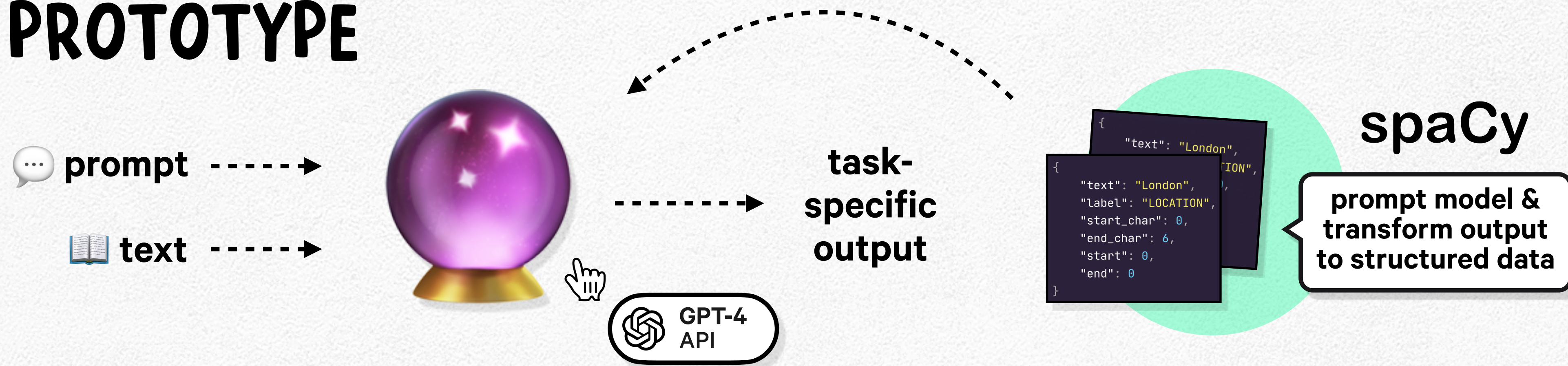


PRODUCTION

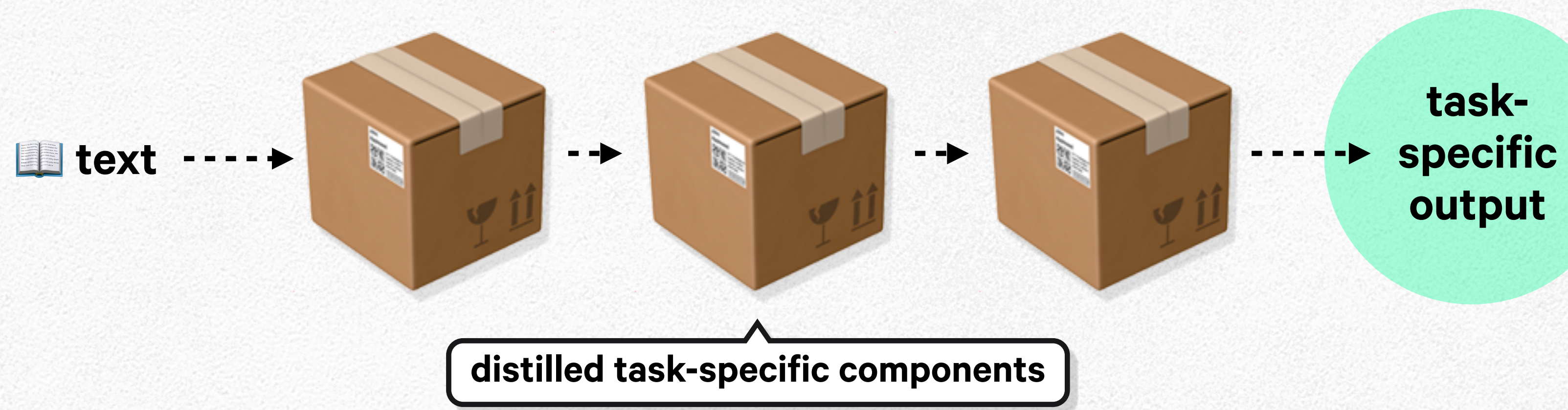
 **text**

**task-specific
output**

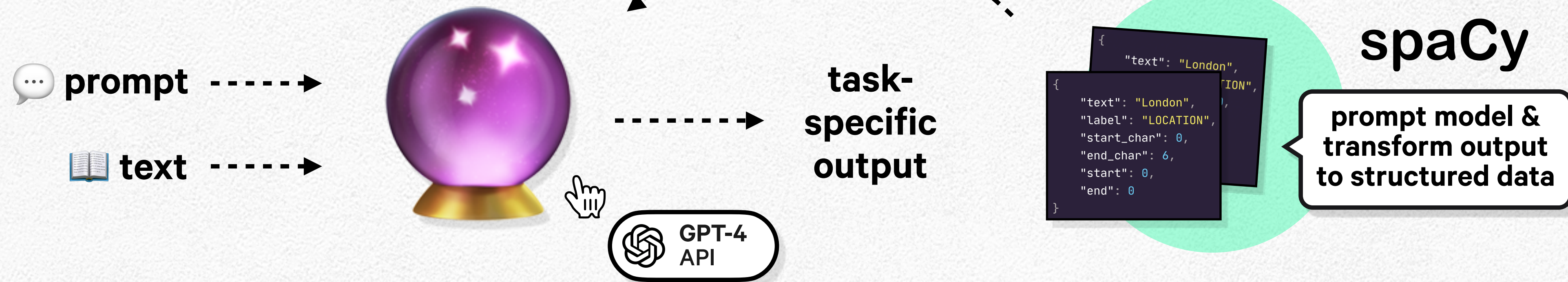
PROTOTYPE



PRODUCTION

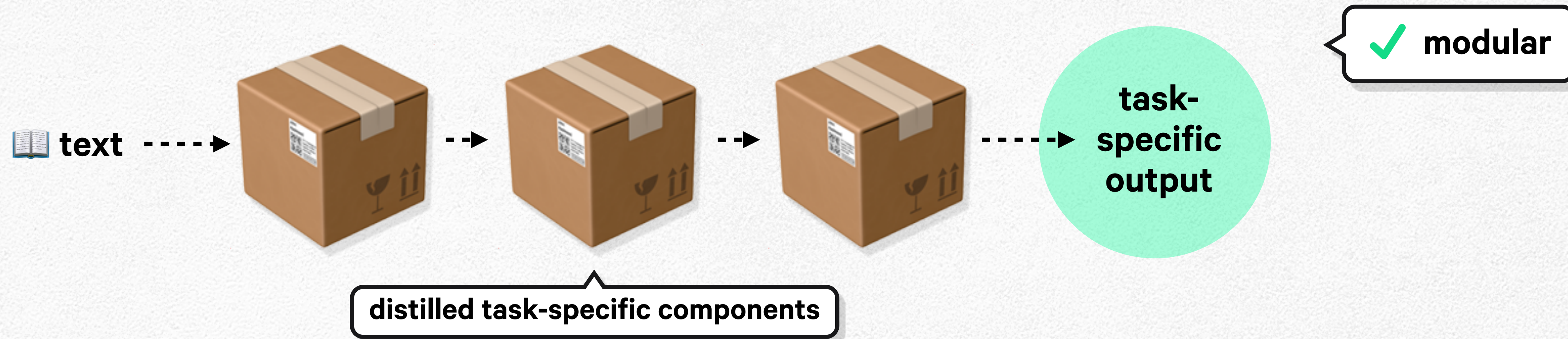


PROTOTYPE

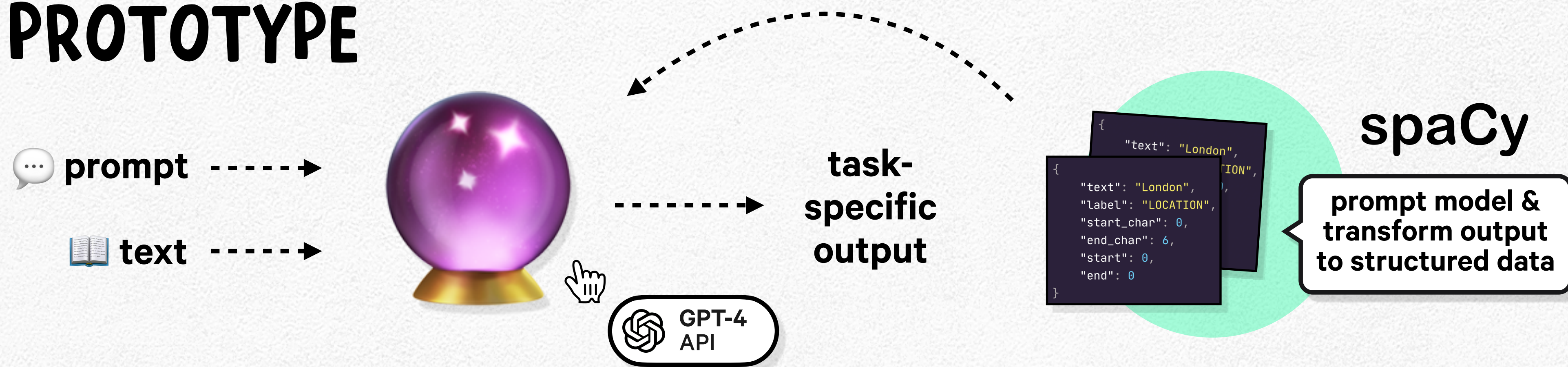


EXPLOSION

PRODUCTION

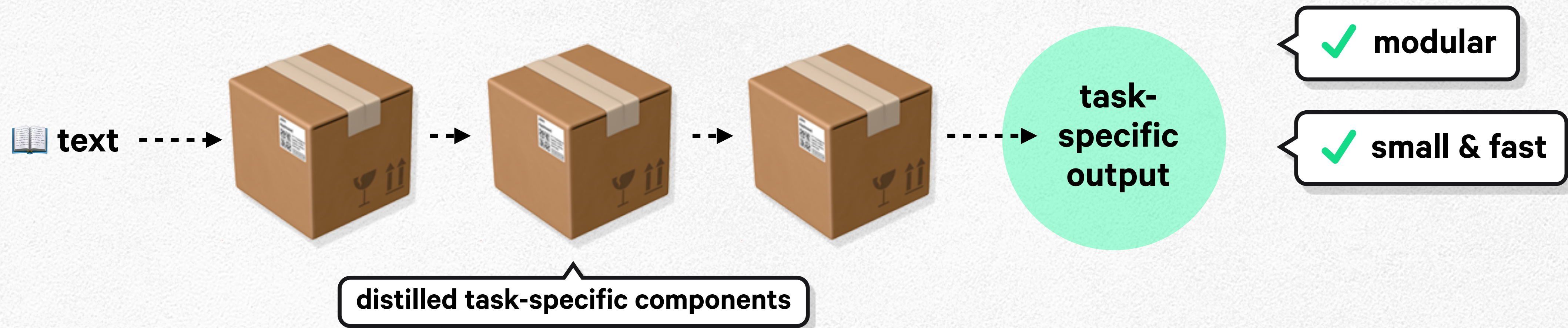


PROTOTYPE

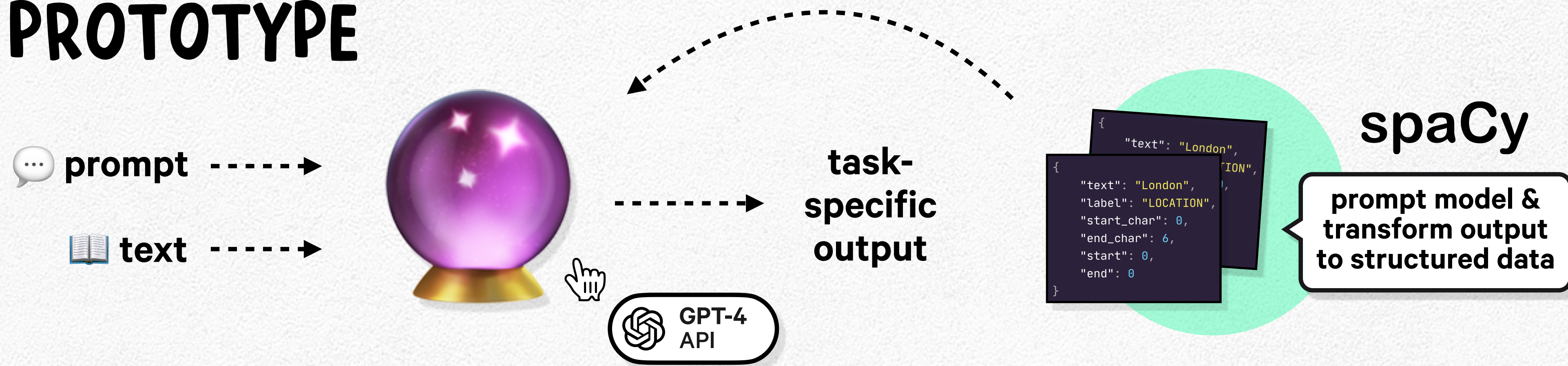


EXPLOSION

PRODUCTION

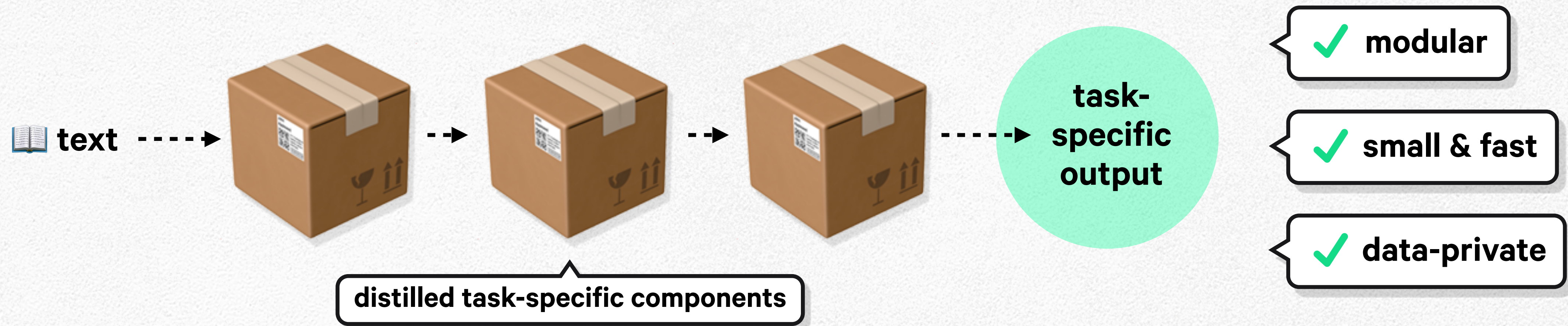


PROTOTYPE



EXPLOSION

PRODUCTION





```
config.cfg

[components.llm]
factory = "llm"

[components.llm.model]
@llm_models = "spacy.Azure.v1"

[components.llm.task]
@llm_tasks = "spacy.NER.v3"
labels = ["bid_price", "buyer", "charterparty_options", "currency"]

[components.llm.task.label_definitions]
bid_price = "the bid price when a bid/ask spread is given in a heard"
buyer = "the entity buying the item/activity"
charterparty_options = "the options available for the charterparty"
currency = "the money system used for a reported price"
```



spaCy component



```
config.cfg

[components.llm]
factory = "llm"

[components.llm.model]
@llm_models = "spacy.Azure.v1"

[components.llm.task]
@llm_tasks = "spacy.NER.v3"
labels = ["bid_price", "buyer", "charterparty_options", "currency"]

[components.llm.task.label_definitions]
bid_price = "the bid price when a bid/ask spread is given in a heard"
buyer = "the entity buying the item/activity"
charterparty_options = "the options available for the charterparty"
currency = "the money system used for a reported price"
```



spaCy component

model and provider



```
config.cfg

[components.llm]
factory = "llm"

[components.llm.model]
@llm_models = "spacy.Azure.v1"

[components.llm.task]
@llm_tasks = "spacy.NER.v3"
labels = ["bid_price", "buyer", "charterparty_options", "currency"]

[components.llm.task.label_definitions]
bid_price = "the bid price when a bid/ask spread is given in a heard"
buyer = "the entity buying the item/activity"
charterparty_options = "the options available for the charterparty"
currency = "the money system used for a reported price"
```



spaCy component

model and provider



task definition and labels

Named Entity Recognition, Text Classification, Relation Extraction, ...

```
config.cfg

[components.llm]
factory = "llm"

[components.llm.model]
@llm_models = "spacy.Azure.v1"

[components.llm.task]
@llm_tasks = "spacy.NER.v3"
labels = ["bid_price", "buyer", "charterparty_options", "currency"]

[components.llm.task.label_definitions]
bid_price = "the bid price when a bid/ask spread is given in a heard"
buyer = "the entity buying the item/activity"
charterparty_options = "the options available for the charterparty"
currency = "the money system used for a reported price"
```



spaCy component

model and provider



task definition and labels

Named Entity Recognition, Text Classification, Relation Extraction, ...

label definitions to use in prompt

```
config.cfg

[components.llm]
factory = "llm"

[components.llm.model]
@llm_models = "spacy.Azure.v1"

[components.llm.task]
@llm_tasks = "spacy.NER.v3"
labels = ["bid_price", "buyer", "charterparty_options", "currency"]

[components.llm.task.label_definitions]
bid_price = "the bid price when a bid/ask spread is given in a heard"
buyer = "the entity buying the item/activity"
charterparty_options = "the options available for the charterparty"
currency = "the money system used for a reported price"
```



spaCy component

model and provider



task definition and labels

Named Entity Recognition, Text Classification, Relation Extraction, ...

label definitions to use in prompt

```
config.cfg

[components.llm]
factory = "llm"

[components.llm.model]
@llm_models = "spacy.Azure.v1"

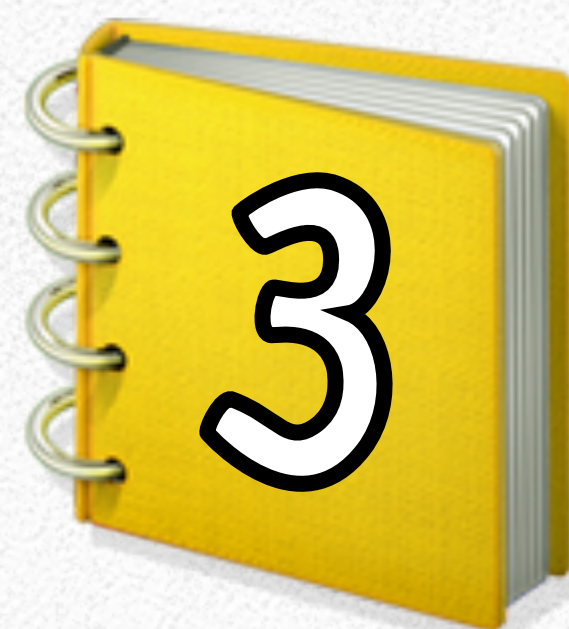
[components.llm.task]
@llm_tasks = "spacy.NER.v3"
labels = ["bid_price", "buyer", "charterparty_options", "currency"]

[components.llm.task.label_definitions]
bid_price = "the bid price when a bid/ask spread is given in a heard"
buyer = "the entity buying the item/activity"
charterparty_options = "the options available for the charterparty"
currency = "the money system used for a reported price"
```

example from **S&P Global** case study

🌐 explosion.ai/blog/sp-global-commodities

PRACTICAL



TIPS

Data annotation

How much data do you need?



How much data do you need?



TRAINING

Prodigy

```
===== Train curve diagnostic =====
Training 4 times with 25%, 50%, 75%, 100% of the data
```

%	Score	ner
----	-----	-----
0%	0.00	0.00
25%	0.31 ▲	0.31 ▲
50%	0.44 ▲	0.44 ▲
75%	0.43 ▼	0.43 ▼
100%	0.56 ▲	0.56 ▲

How much data do you need?

EXPLOSION

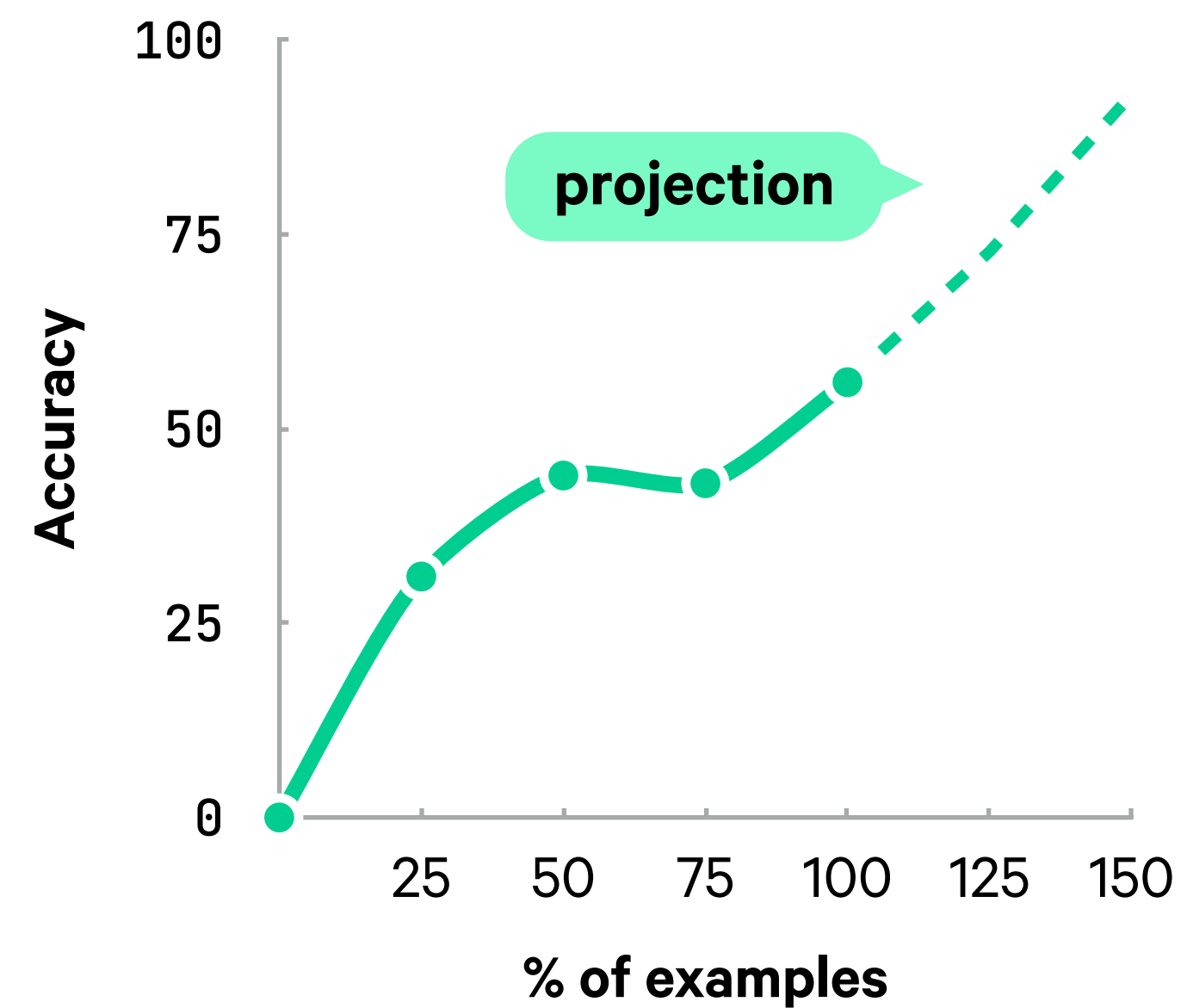
TRAINING

Prodigy

===== Train curve diagnostic =====

Training 4 times with 25%, 50%, 75%, 100% of the data

%	Score	ner
0%	0.00	0.00
25%	0.31 ▲	0.31 ▲
50%	0.44 ▲	0.44 ▲
75%	0.43 ▼	0.43 ▼
100%	0.56 ▲	0.56 ▲



How much data do you need?

EXPLOSION

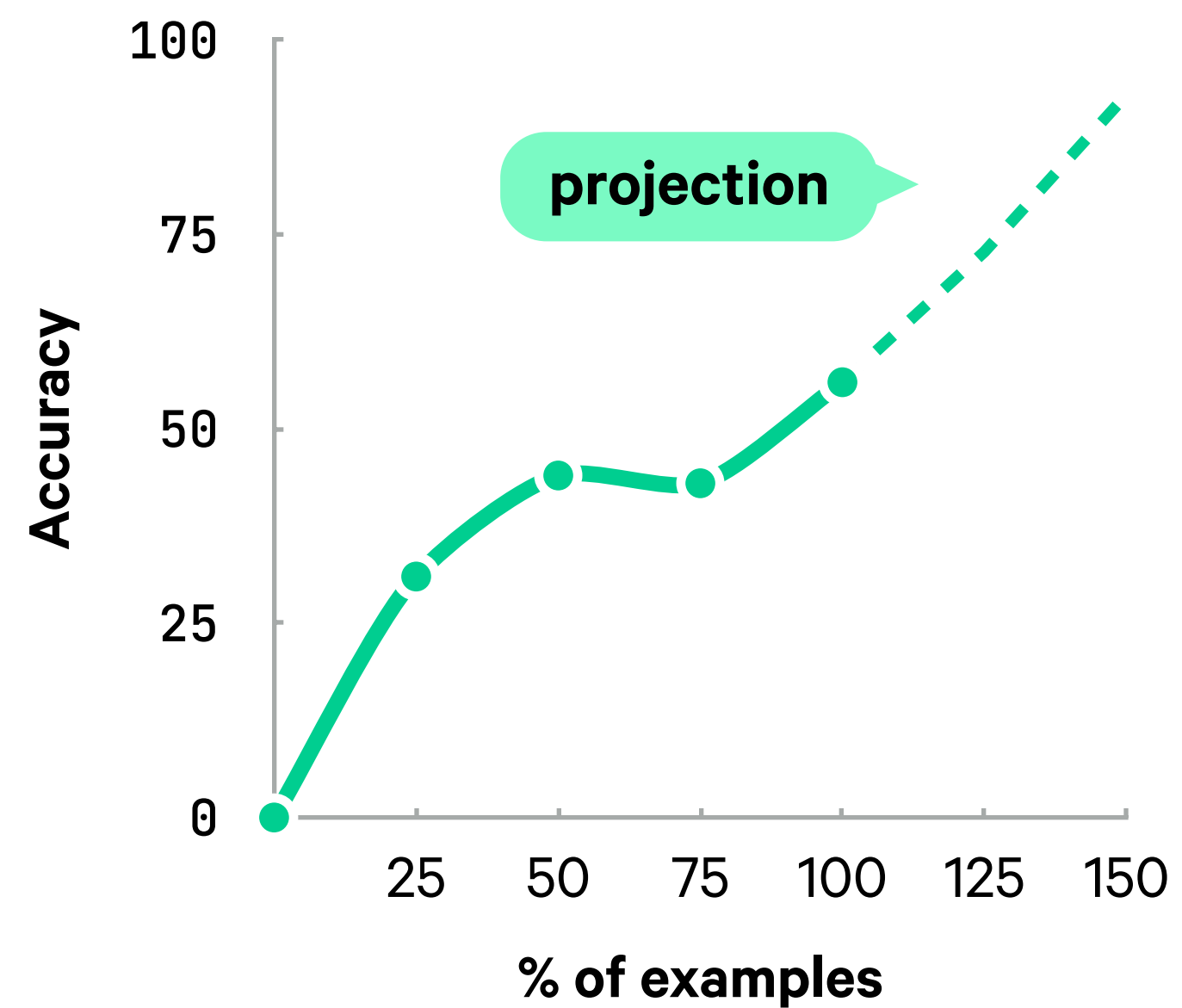
TRAINING

Prodigy

===== Train curve diagnostic =====

Training 4 times with 25%, 50%, 75%, 100% of the data

%	Score	ner
0%	0.00	0.00
25%	0.31 ▲	0.31 ▲
50%	0.44 ▲	0.44 ▲
75%	0.43 ▼	0.43 ▼
100%	0.56 ▲	0.56 ▲



EVALUATION



You need enough data to avoid reporting meaningless precision.

How much data do you need?

EXPLOSION

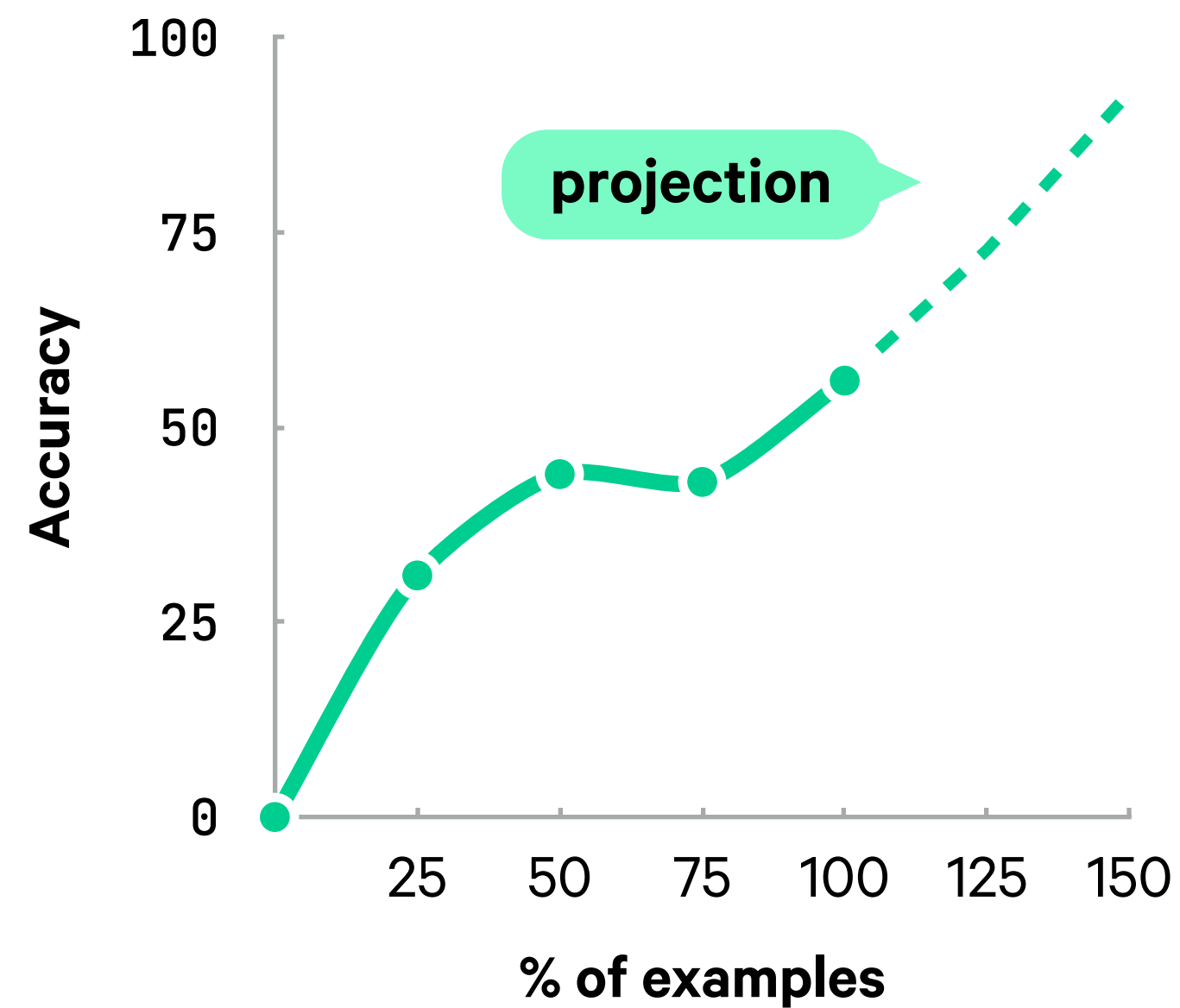
TRAINING

Prodigy

===== Train curve diagnostic =====

Training 4 times with 25%, 50%, 75%, 100% of the data

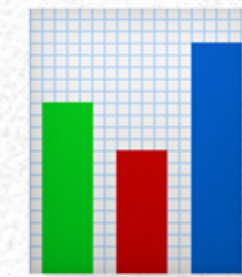
%	Score	ner
0%	0.00	0.00
25%	0.31 ▲	0.31 ▲
50%	0.44 ▲	0.44 ▲
75%	0.43 ▼	0.43 ▼
100%	0.56 ▲	0.56 ▲



EVALUATION



You need enough data to avoid reporting **meaningless precision**.



Ten samples per significant figure is a good rule of thumb.

How much data do you need?

EXPLOSION

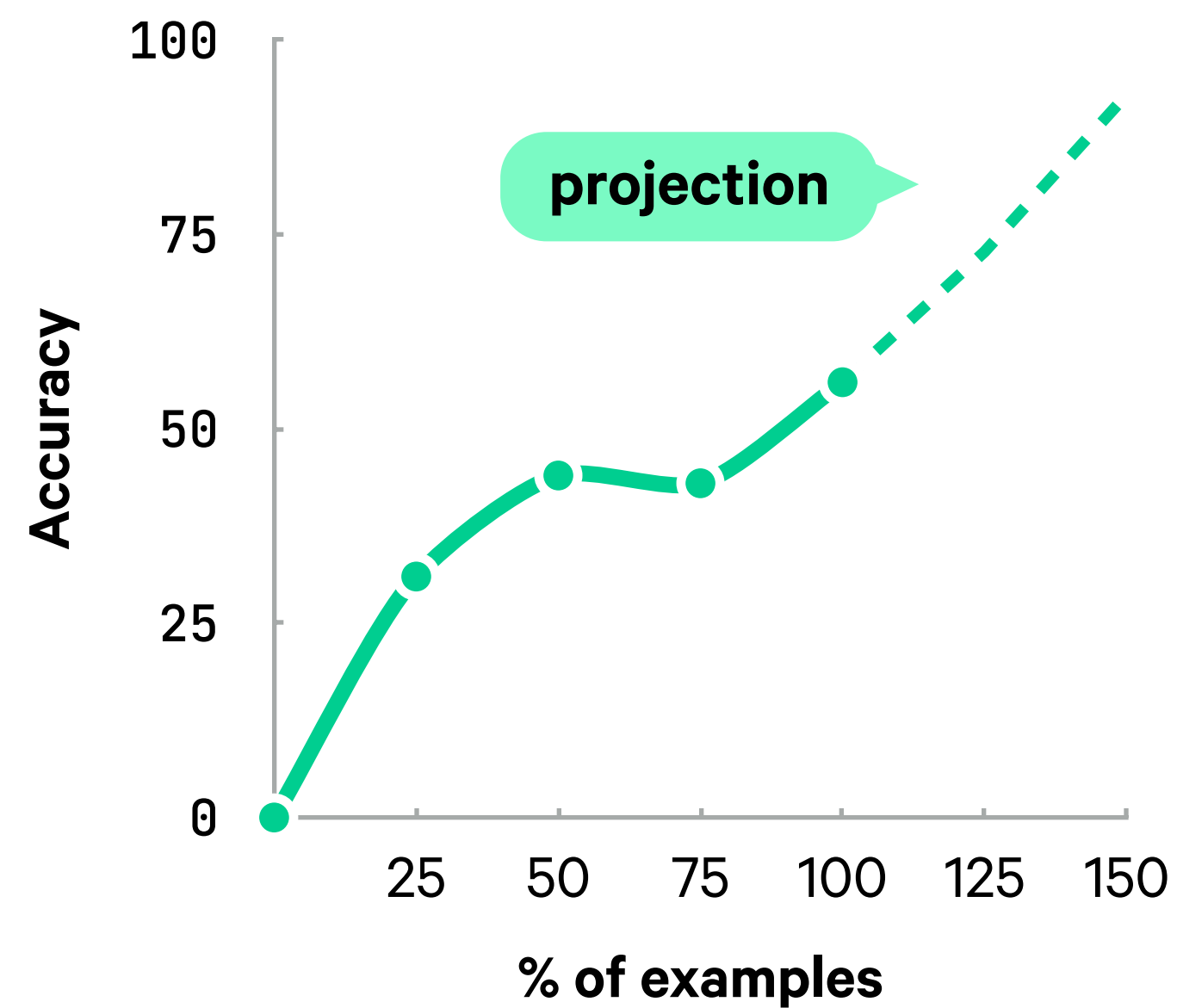
TRAINING

Prodigy

===== Train curve diagnostic =====

Training 4 times with 25%, 50%, 75%, 100% of the data

%	Score	ner
0%	0.00	0.00
25%	0.31 ▲	0.31 ▲
50%	0.44 ▲	0.44 ▲
75%	0.43 ▼	0.43 ▼
100%	0.56 ▲	0.56 ▲



EVALUATION



You need enough data to avoid reporting **meaningless precision**.



Ten samples per significant figure is a good rule of thumb.

1,000 samples is pretty good
– enough for 94% vs. 95%.

KEEP TASKS SMALL



KEEP TASKS SMALL



GOOD



```
for i in range(rows):  
    access_data(array[i])
```

BAD



```
for j in range(columns):  
    access_data(array[:, j])
```



KEEP TASKS SMALL

Humans have a cache, too!



GOOD



```
for i in range(rows):  
    access_data(array[i])
```

BAD



```
for j in range(columns):  
    access_data(array[:, j])
```

KEEP TASKS SMALL

Humans have a cache, too!

EXPLOSION

GOOD

```
for i in range(rows):  
    access_data(array[i])
```

BAD

```
for j in range(columns):  
    access_data(array[:, j])
```

DO THIS

```
for annotation_type in annotation_types:  
    for example in examples:  
        annotate(example, annotation_type)
```

NOT THIS

```
for example in examples:  
    for annotation_type in annotation_types:  
        annotate(example, annotation_type)
```

USE **MODEL ASSISTANCE**



USE **MODEL ASSISTANCE**



Suggest annotations however you can. Rule-based, initial trained model, an LLM – or a combination of all.

USE **MODEL ASSISTANCE**



Suggest annotations however you can. Rule-based, initial trained model, an LLM – or a combination of all.



Suggestions improve efficiency. Common cases are common, so getting them preset speeds up annotation a lot.

USE MODEL ASSISTANCE



Suggest annotations however you can. Rule-based, initial trained model, an LLM – or a combination of all.



Suggestions improve efficiency. Common cases are common, so getting them preset speeds up annotation a lot.

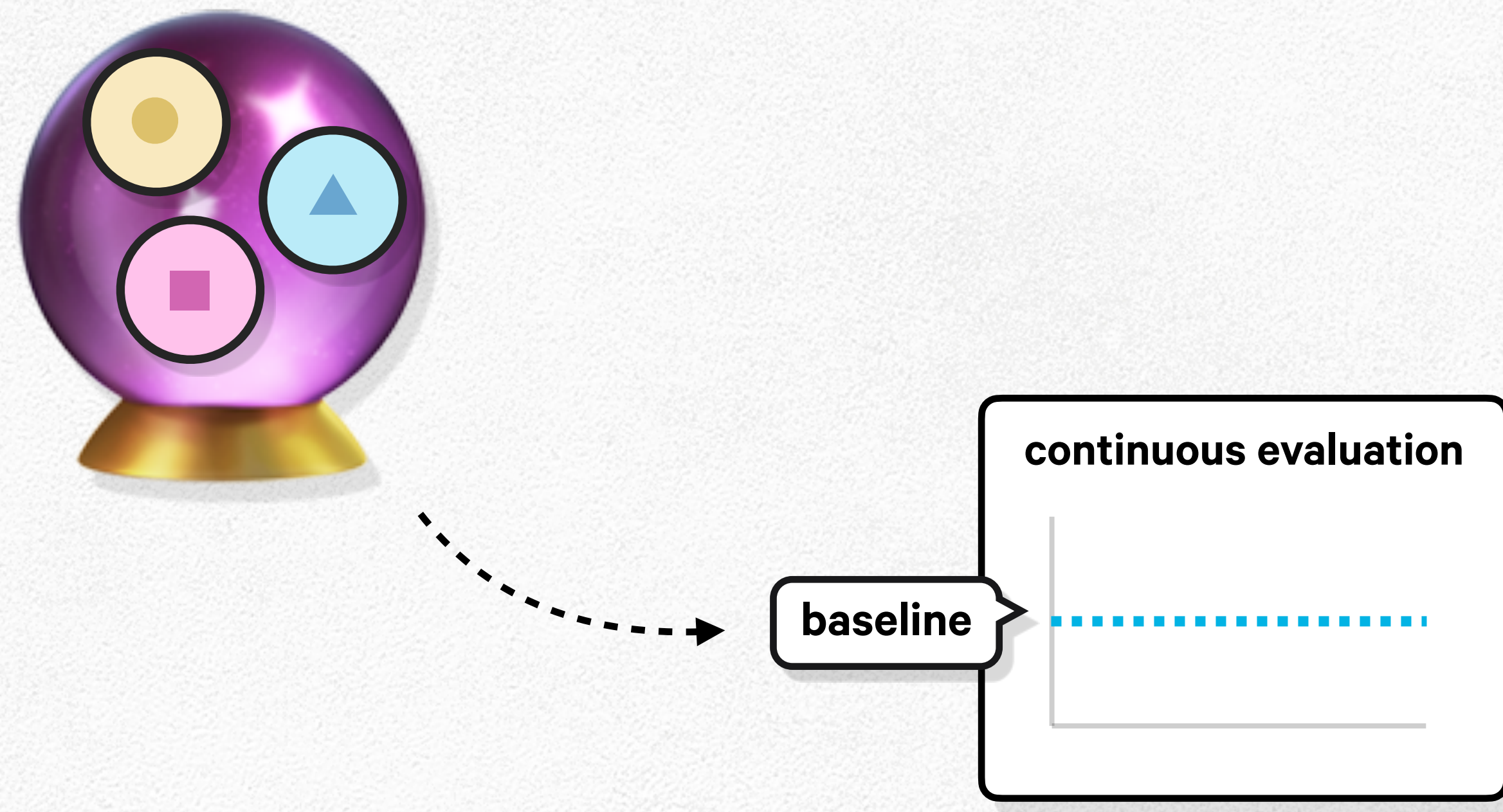


Suggestions improve accuracy. You need the common cases to be annotated consistently. Humans suck at this.

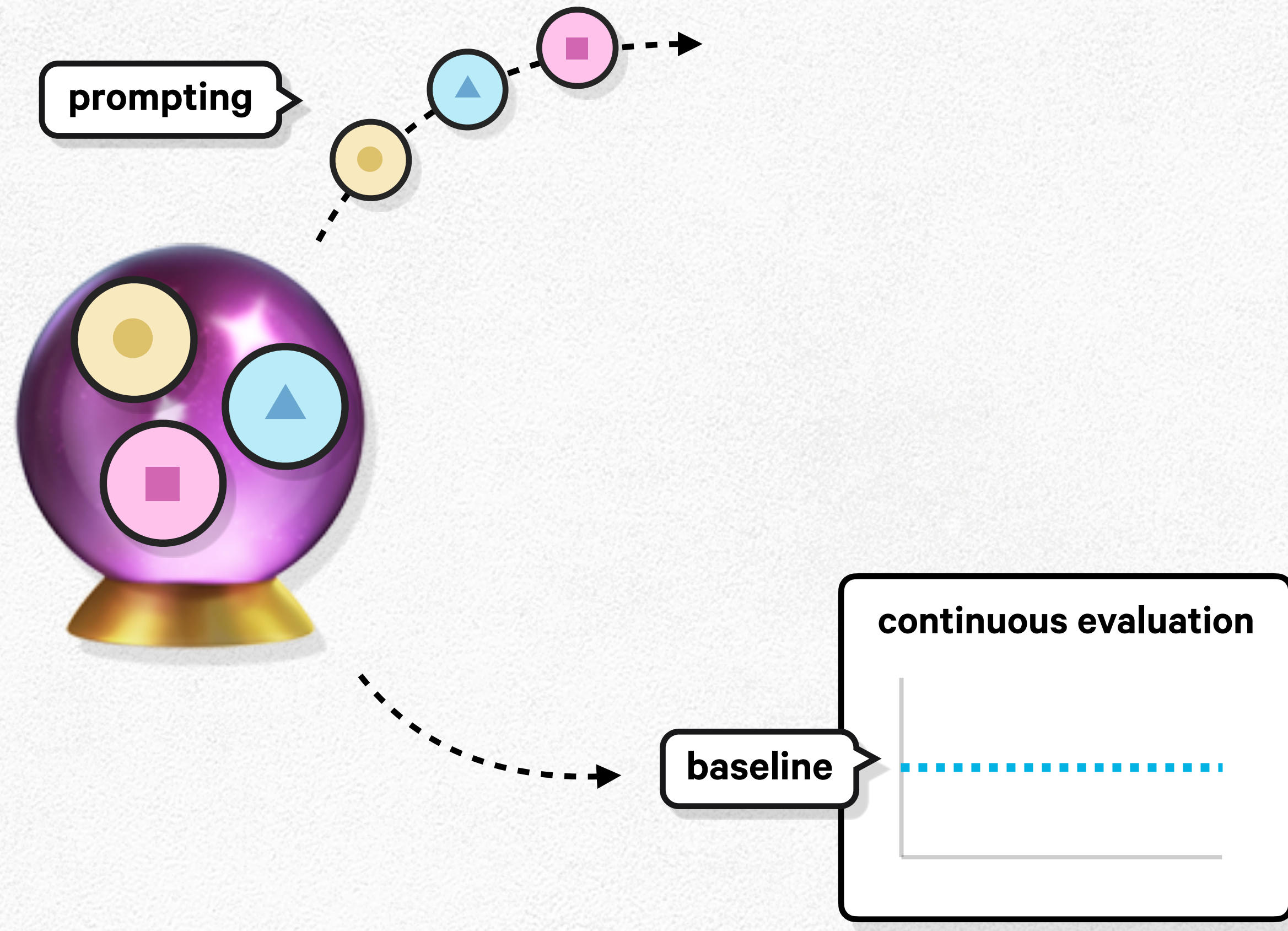
HUMAN IN THE LOOP



HUMAN IN THE LOOP

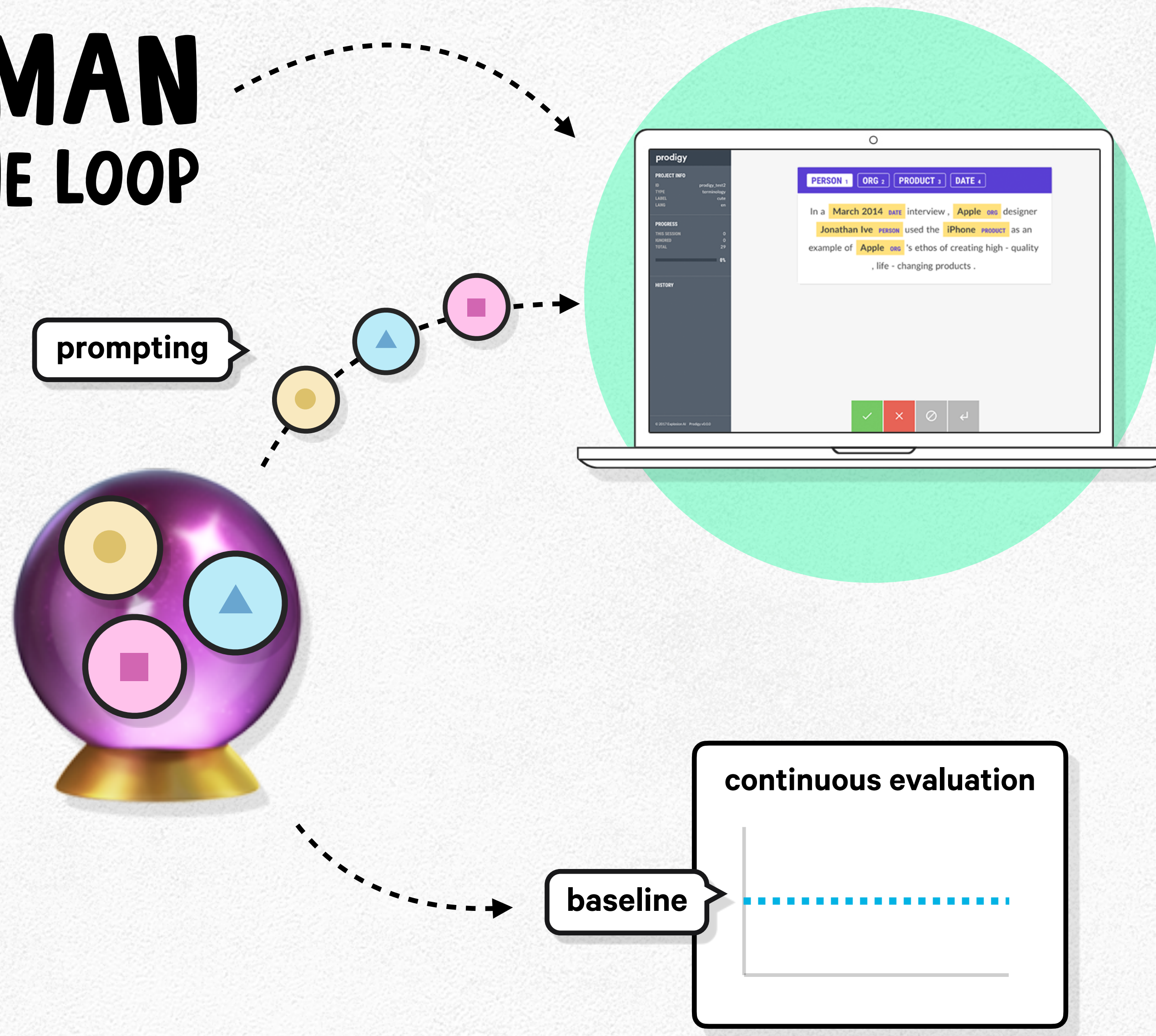


HUMAN IN THE LOOP



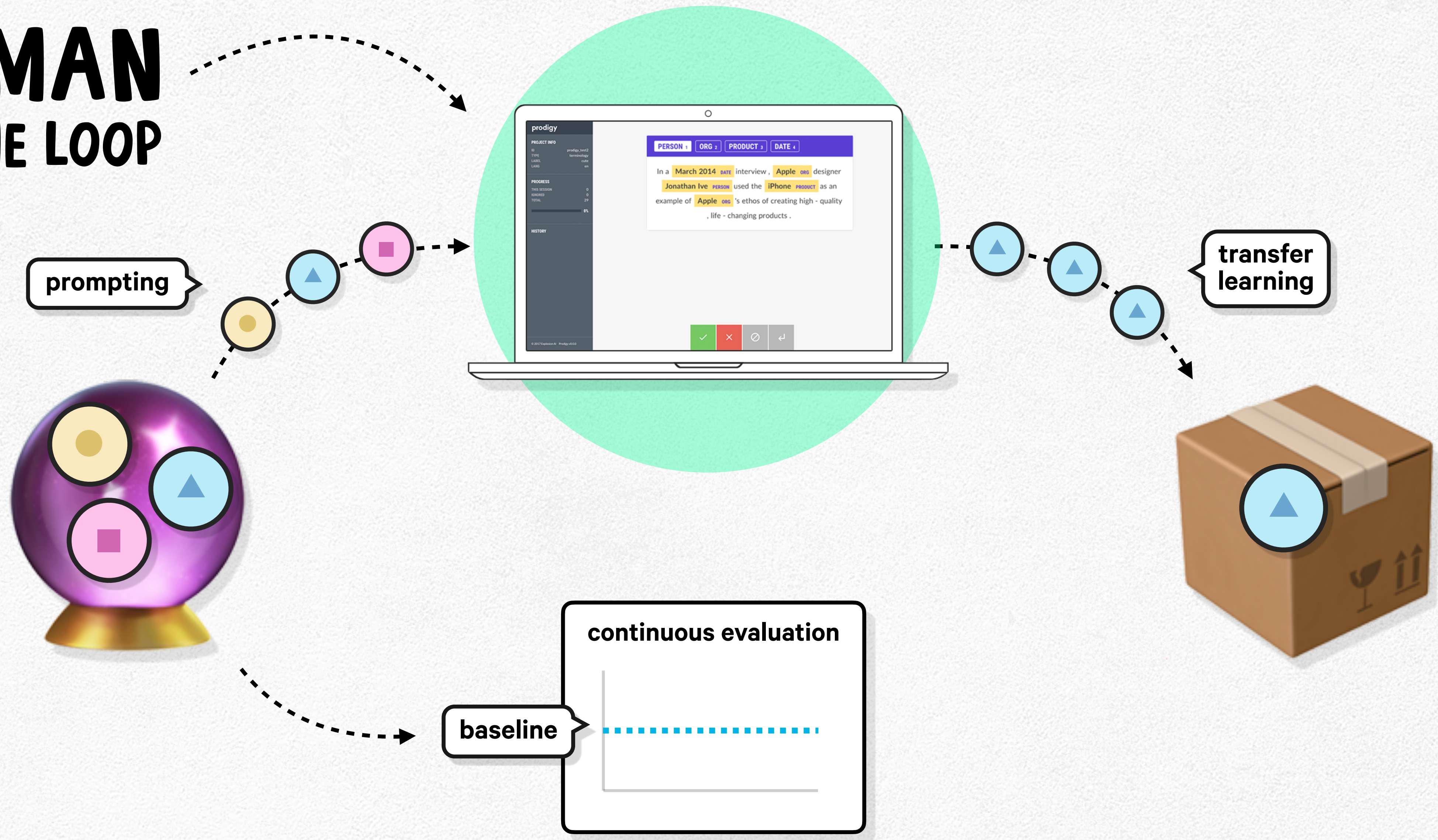
HUMAN IN THE LOOP

EXPLOSION



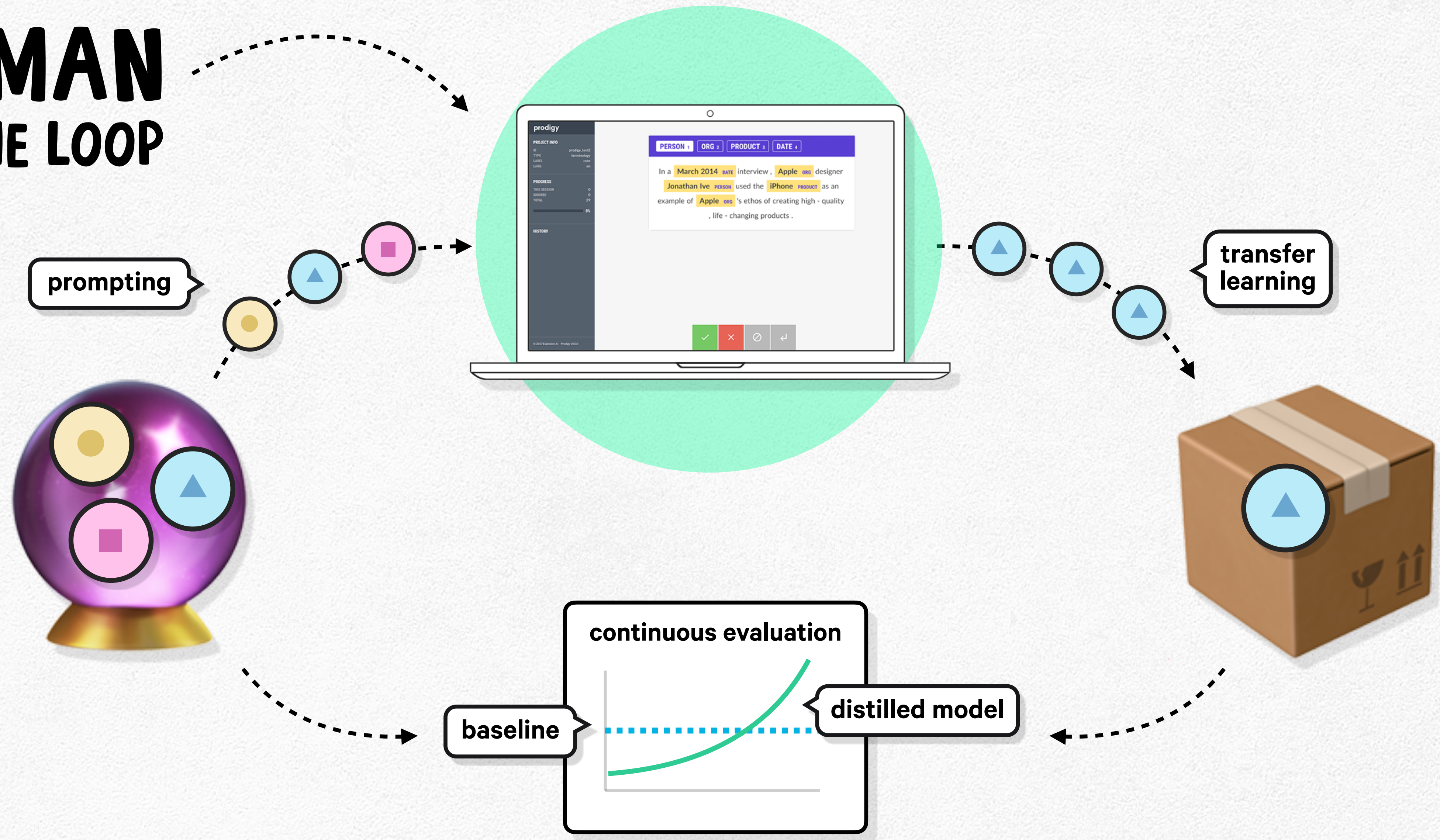
HUMAN IN THE LOOP

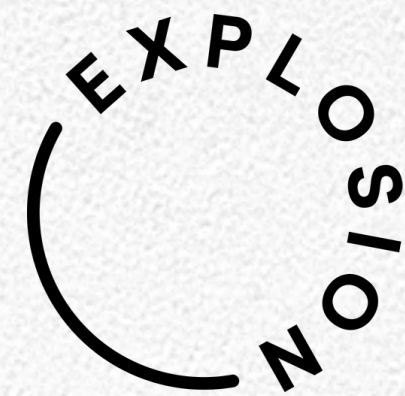
EXPLOSION



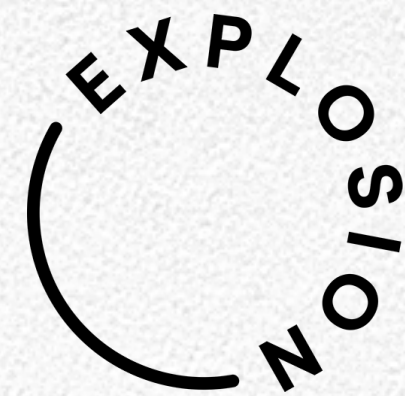
HUMAN IN THE LOOP

EXPLOSION



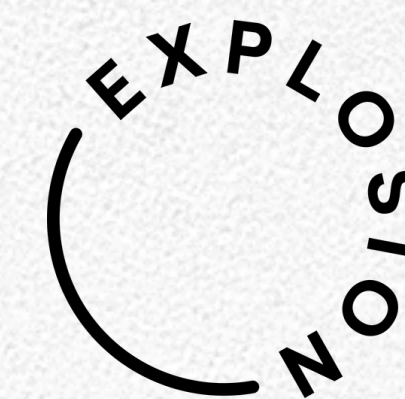


```
$ prodigy ner.llm.correct todo_eval  
./config.cfg ./examples.jsonl
```



recipe function
with workflow

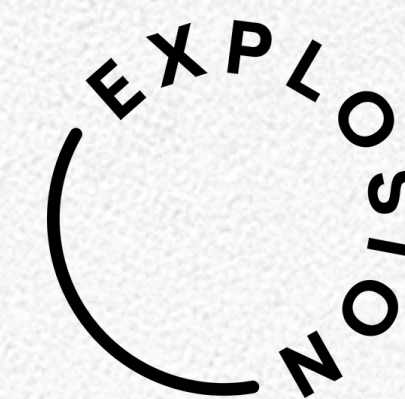
```
$ prodigy ner.llm.correct todo_eval  
./config.cfg ./examples.jsonl
```



recipe function
with workflow

dataset to save
annotations to

```
$ prodigy ner.llm.correct todo_eval  
./config.cfg ./examples.jsonl
```

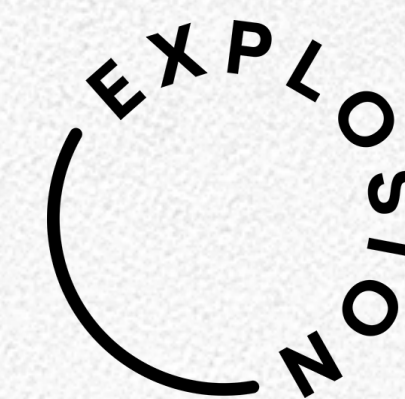


recipe function
with workflow

dataset to save
annotations to

```
$ prodigy ner.llm.correct todo_eval  
./config.cfg ./examples.jsonl
```

```
[components.llm.model]  
@llm_models = "spacy.GPT-4.v2"
```



recipe function
with workflow

dataset to save
annotations to

```
$ prodigy ner.llm.correct todo_eval  
./config.cfg ./examples.jsonl
```

raw data

```
[components.llm.model]  
@llm_models = "spacy.GPT-4.v2"
```



recipe function
with workflow

dataset to save
annotations to

```
$ prodigy ner.llm.correct todo_eval  
./config.cfg ./examples.jsonl
```

raw data

✨ Starting the web server at localhost:8080...
Open the app and start annotating!

```
[components.llm.model]  
@llm_models = "spacy.GPT-4.v2"
```





recipe function
with workflow

dataset to save
annotations to

```
$ prodigy ner.llm.correct todo_eval  
./config.cfg ./examples.jsonl
```

raw data

✨ Starting the web server at localhost:8080...
Open the app and start annotating!

```
[components.llm.model]  
@llm_models = "spacy.GPT-4.v2"
```



ANNOTATION STARTS AT HOME

What is a quote?

A **quotation** is the repetition of a sentence, phrase, or passage from speech or text that someone has said or written. In oral speech, it is the representation of an utterance that is introduced by a quotative marker, such as a verb of saying. For example: John said: "I saw Mary today". In written text, quotations are signaled by quotation marks.

The model is trained to recognise **3 entities**:

- SOURCE** - the speaker, which might be a person, an organisation etc.
- CUE** - usually a verb phrase, indicating the act of speech or expression
- CONTENT** - the direct quote, including quotation marks

SOURCE and/or **CUE** must always be accompanied by **CONTENT**, but **CONTENT** can exist without a **SOURCE** or **CUE**.

Examples of different quotation styles:

What is a quote?

A **quotation** is the repetition of a sentence, phrase, or passage from speech or text that someone has said or written. In oral speech, it is the representation of an utterance that is introduced by a quotative marker, such as a verb of saying. For example: John said: "I saw Mary today". In written text, quotations are signaled by quotation marks.

The model is trained to recognise **3 entities**:

- SOURCE** - the speaker, which might be a person, an organisation etc.
- CUE** - usually a verb phrase, indicating the act of speech or expression
- CONTENT** - the direct quote, including quotation marks

SOURCE and/or **CUE** must always be accompanied by **CONTENT**, but **CONTENT** can exist without a **SOURCE** or **CUE**.

Examples of different quotation styles:

- 1. The cue:** **SOURCE** **CUE** **CONTENT** (e.g. "I saw Mary today").
- 2. The cue:** **SOURCE** **CUE** **CONTENT** (e.g. "I saw Mary today").
- 3. The cue:** **SOURCE** **CUE** **CONTENT** (e.g. "I saw Mary today").
- 4. The cue:** **SOURCE** **CUE** **CONTENT** (e.g. "I saw Mary today").
- 5. The cue:** **SOURCE** **CUE** **CONTENT** (e.g. "I saw Mary today").
- 6. The cue:** **SOURCE** **CUE** **CONTENT** (e.g. "I saw Mary today").
- 7. The cue:** **SOURCE** **CUE** **CONTENT** (e.g. "I saw Mary today").
- 8. The cue:** **SOURCE** **CUE** **CONTENT** (e.g. "I saw Mary today").
- 9. The cue:** **SOURCE** **CUE** **CONTENT** (e.g. "I saw Mary today").
- 10. The cue:** **SOURCE** **CUE** **CONTENT** (e.g. "I saw Mary today").
- 11. The cue:** **SOURCE** **CUE** **CONTENT** (e.g. "I saw Mary today").
- 12. The cue:** **SOURCE** **CUE** **CONTENT** (e.g. "I saw Mary today").

What is a quote?

A **quotation** is the repetition of a sentence, phrase, or passage from speech or text that someone has said or written. In oral speech, it is the representation of an utterance that is introduced by a quotative marker, such as a verb of saying. For example: John said: "I saw Mary today". In written text, quotations are signaled by quotation marks.

The model is trained to recognise **3 entities**:

- SOURCE** - the speaker, which might be a person, an organisation etc.
- CUE** - usually a verb phrase, indicating the act of speech or expression
- CONTENT** - the direct quote, including quotation marks

SOURCE and/or **CUE** must always be accompanied by **CONTENT**, but **CONTENT** can exist without a **SOURCE** or **CUE**.

Examples of different quotation styles:

- 1. The cue:** **SOURCE** **CUE** **CONTENT** (e.g. "I saw Mary today").
- 2. The cue:** **SOURCE** **CUE** **CONTENT** (e.g. "I saw Mary today").
- 3. The cue:** **SOURCE** **CUE** **CONTENT** (e.g. "I saw Mary today").
- 4. The cue:** **SOURCE** **CUE** **CONTENT** (e.g. "I saw Mary today").
- 5. The cue:** **SOURCE** **CUE** **CONTENT** (e.g. "I saw Mary today").
- 6. The cue:** **SOURCE** **CUE** **CONTENT** (e.g. "I saw Mary today").
- 7. The cue:** **SOURCE** **CUE** **CONTENT** (e.g. "I saw Mary today").
- 8. The cue:** **SOURCE** **CUE** **CONTENT** (e.g. "I saw Mary today").
- 9. The cue:** **SOURCE** **CUE** **CONTENT** (e.g. "I saw Mary today").
- 10. The cue:** **SOURCE** **CUE** **CONTENT** (e.g. "I saw Mary today").
- 11. The cue:** **SOURCE** **CUE** **CONTENT** (e.g. "I saw Mary today").
- 12. The cue:** **SOURCE** **CUE** **CONTENT** (e.g. "I saw Mary today").

What is a quote?

A **quotation** is the repetition of a sentence, phrase, or passage from speech or text that someone has said or written. In oral speech, it is the representation of an utterance that is introduced by a quotative marker, such as a verb of saying. For example: John said: "I saw Mary today". In written text, quotations are signaled by quotation marks.

The model is trained to recognise **3 entities**:

- SOURCE** - the speaker, which might be a person, an organisation etc.
- CUE** - usually a verb phrase, indicating the act of speech or expression
- CONTENT** - the direct quote, including quotation marks

SOURCE and/or **CUE** must always be accompanied by **CONTENT**, but **CONTENT** can exist without a **SOURCE** or **CUE**.

Examples of different quotation styles:

- 1. The cue:** **SOURCE** **CUE** **CONTENT** (e.g. "I saw Mary today").
- 2. The cue:** **SOURCE** **CUE** **CONTENT** (e.g. "I saw Mary today").
- 3. The cue:** **SOURCE** **CUE** **CONTENT** (e.g. "I saw Mary today").
- 4. The cue:** **SOURCE** **CUE** **CONTENT** (e.g. "I saw Mary today").
- 5. The cue:** **SOURCE** **CUE** **CONTENT** (e.g. "I saw Mary today").
- 6. The cue:** **SOURCE** **CUE** **CONTENT** (e.g. "I saw Mary today").
- 7. The cue:** **SOURCE** **CUE** **CONTENT** (e.g. "I saw Mary today").
- 8. The cue:** **SOURCE** **CUE** **CONTENT** (e.g. "I saw Mary today").
- 9. The cue:** **SOURCE** **CUE** **CONTENT** (e.g. "I saw Mary today").
- 10. The cue:** **SOURCE** **CUE** **CONTENT** (e.g. "I saw Mary today").
- 11. The cue:** **SOURCE** **CUE** **CONTENT** (e.g. "I saw Mary today").
- 12. The cue:** **SOURCE** **CUE** **CONTENT** (e.g. "I saw Mary today").

What is a quote?

A **quotation** is the repetition of a sentence, phrase, or passage from speech or text that someone has said or written. In oral speech, it is the representation of an utterance that is introduced by a quotative marker, such as a verb of saying. For example: John said: "I saw Mary today". In written text, quotations are signaled by quotation marks.

The model is trained to recognise **3 entities**:

- SOURCE** - the speaker, which might be a person, an organisation etc.
- CUE** - usually a verb phrase, indicating the act of speech or expression
- CONTENT** - the direct quote, including quotation marks

SOURCE and/or **CUE** must always be accompanied by **CONTENT**, but **CONTENT** can exist without a **SOURCE** or **CUE**.

Examples of different quotation styles:

- 1. The cue:** **SOURCE** **CUE** **CONTENT** (e.g. "I saw Mary today").
- 2. The cue:** **SOURCE** **CUE** **CONTENT** (e.g. "I saw Mary today").
- 3. The cue:** **SOURCE** **CUE** **CONTENT** (e.g. "I saw Mary today").
- 4. The cue:** **SOURCE** **CUE** **CONTENT** (e.g. "I saw Mary today").
- 5. The cue:** **SOURCE** **CUE** **CONTENT** (e.g. "I saw Mary today").
- 6. The cue:** **SOURCE** **CUE** **CONTENT** (e.g. "I saw Mary today").
- 7. The cue:** **SOURCE** **CUE** **CONTENT** (e.g. "I saw Mary today").
- 8. The cue:** **SOURCE** **CUE** **CONTENT** (e.g. "I saw Mary today").
- 9. The cue:** **SOURCE** **CUE** **CONTENT** (e.g. "I saw Mary today").
- 10. The cue:** **SOURCE** **CUE** **CONTENT** (e.g. "I saw Mary today").
- 11. The cue:** **SOURCE** **CUE** **CONTENT** (e.g. "I saw Mary today").
- 12. The cue:** **SOURCE** **CUE** **CONTENT** (e.g. "I saw Mary today").

The Guardian case study
explosion.ai/blog/guardian

prodigy

PROJECT INFO

DATASET

PROGRESS

SOURCE PROGRESS

HISTORY

12:32 PM | Team Prodigy Data Annotation

No tasks available. Make sure to save your progress. If working in multi-annotator scenario, try reloading the page to see if more tasks have become accessible.

Dr. Rebecca Blinco

Patrick Deane

Freya Rosary

Lambert

Nneka Okigbo

Rotational Labs

TRAINING

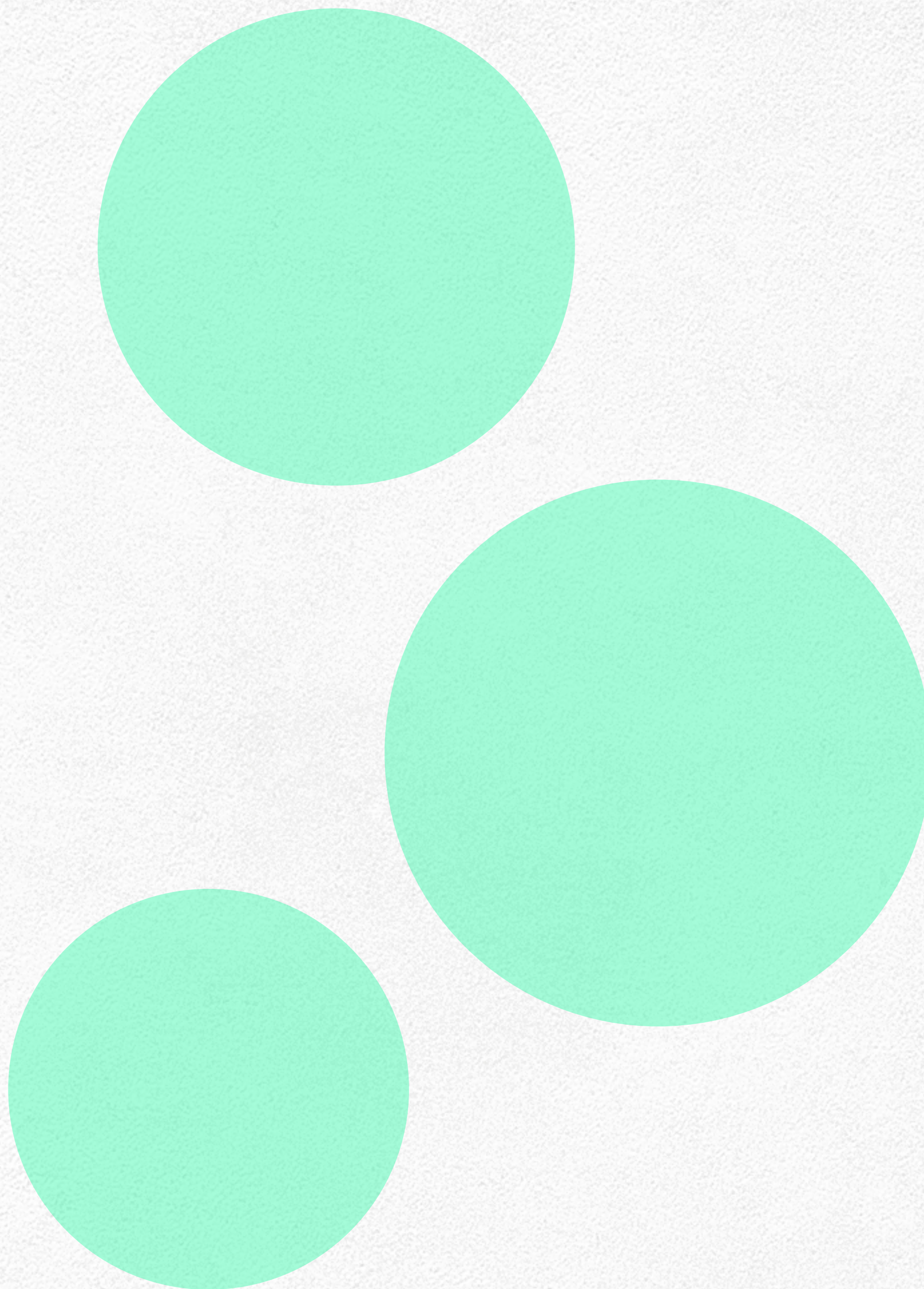


TIPS

DATA



TIPS



EXPLOSION

MODELLING



TIPS

TRAINING



TIPS

Form and falsify hypotheses.

EXPLOSION
NOISE

MODELLING



TIPS

DATA



TIPS

TRAINING



TIPS

Form and falsify hypotheses.

Prioritize robustness.

DATA



TIPS

MODELLING



TIPS

EXPLOSION
NOISE

TRAINING



TIPS

Form and falsify hypotheses.

Prioritize robustness.

Scale down and iterate.

MODELLING



TIPS

DATA



TIPS

EXPLOSION
NOISE

TRAINING



TIPS

Form and falsify hypotheses.

Prioritize robustness.

Scale down and iterate.

Imagine you're the model.

DATA



TIPS

MODELLING



TIPS

EXPLOSION
NOISE

TRAINING



TIPS

Form and falsify hypotheses.

Prioritize robustness.

Scale down and iterate.

Imagine you're the model.

Finish the pipeline to production.

MODELLING



TIPS

DATA



TIPS

EXPLOSION
NOISE

TRAINING



TIPS

Form and falsify hypotheses.

Prioritize robustness.

Scale down and iterate.

Imagine you're the model.

Finish the pipeline to production.

Be agile and annotate yourself.

DATA



TIPS

MODELLING



TIPS

EXPLOSION
NOISE

TRAINING



TIPS

Form and falsify hypotheses.

Prioritize robustness.

Scale down and iterate.

Imagine you're the model.

Finish the pipeline to production.

DATA



TIPS

Be agile and annotate yourself.

Keep tasks small.

MODELLING



TIPS

EXPLOSION
NOISE

TRAINING



TIPS

Form and falsify hypotheses.

Prioritize robustness.

Scale down and iterate.

Imagine you're the model.

Finish the pipeline to production.

DATA



TIPS

Be agile and annotate yourself.

Keep tasks small.

Use model assistance.

MODELLING



TIPS

EXPLOSION
NOISE



THANK YOU!

✦ **Explosion** explosion.ai

🌙 **spaCy** spacy.io

✦ **Prodigy** prodigy.ai

🐦 **Twitter** @honnibal

🐙 **Mastodon** @honnibal@sigmoid.social

🦋 **Bluesky** @honnibal.bsky.social

📁 **LinkedIn**

