Teaching AI about human knowledge

Supervised learning is great — it's data collection that's broken

Ines Montani

Explosion AI

Explosion AI is a digital studio specialising in Artificial Intelligence and Natural Language Processing.

spaCy — Open-source library for industrial-strength Natural Language Processing

THINC — spaCy's next-generation Machine Learning library for deep learning with text

prodigy — A radically efficient data collection and annotation tool, powered by active learning

Data Store — *Coming soon:* pre-trained, customisable models for a variety of languages and domains

Machine Learning is “programming by example”

- annotations let us specify the **output** we’re looking for
- draw examples from the **same distribution** as runtime inputs
- **goal:** system’s prediction given some input matches label a human would have assigned

```
def train_tagger(examples):
    W = defaultdict(lambda: zeros(n_tags))
    for (word, prev, next), human_tag in examples:
        scores = W[word] + W[prev] + W[next]
        guess = scores.argmax()
        if guess != human_tag:
            for feat in (word, prev, next):
                W[feat][guess] -= 1
                W[feat][human_tag] += 1
```

examples = words, tags, contexts

Example: Training a simple part-of-speech tagger with the perceptron algorithm

```
def train_tagger(examples):
    W = defaultdict(lambda: zeros(n_tags))
    for (word, prev, next), human_tag in examples:
        scores = W[word] + W[prev] + W[next]
        guess = scores.argmax()
        if guess != human_tag:
            for feat in (word, prev, next):
                W[feat][guess] -= 1
                W[feat][human_tag] += 1
```

the weights we'll train

Example: Training a simple part-of-speech tagger with the perceptron algorithm

```
def train_tagger(examples):
    W = defaultdict(lambda: zeros(n_tags))
    for (word, prev, next), human_tag in examples:
        scores = W[word] + W[prev] + W[next]
        guess = scores.argmax()
        if guess != human_tag:
            for feat in (word, prev, next):
                W[feat][guess] -= 1
                W[feat][human_tag] += 1
```

score tag given weight & context

Example: Training a simple part-of-speech tagger with the perceptron algorithm

```
def train_tagger(examples):
    W = defaultdict(lambda: zeros(n_tags))
    for (word, prev, next), human_tag in examples:
        scores = W[word] + W[prev] + W[next]
        guess = scores.argmax()
        if guess != human_tag:
            for feat in (word, prev, next):
                W[feat][guess] -= 1
                W[feat][human_tag] += 1
```

get the best-scoring tag

Example: Training a simple part-of-speech tagger with the perceptron algorithm

```
def train_tagger(examples):
    W = defaultdict(lambda: zeros(n_tags))
    for (word, prev, next), human_tag in examples:
        scores = W[word] + W[prev] + W[next]
        guess = scores.argmax()
        if guess != human_tag:
            for feat in (word, prev, next):
                W[feat][guess] -= 1
                W[feat][human_tag] += 1
```

decrease score for bad tag in this context

increase score for good tag in this context

Example: Training a simple part-of-speech tagger with the perceptron algorithm

“Regular” programming

the part you work on

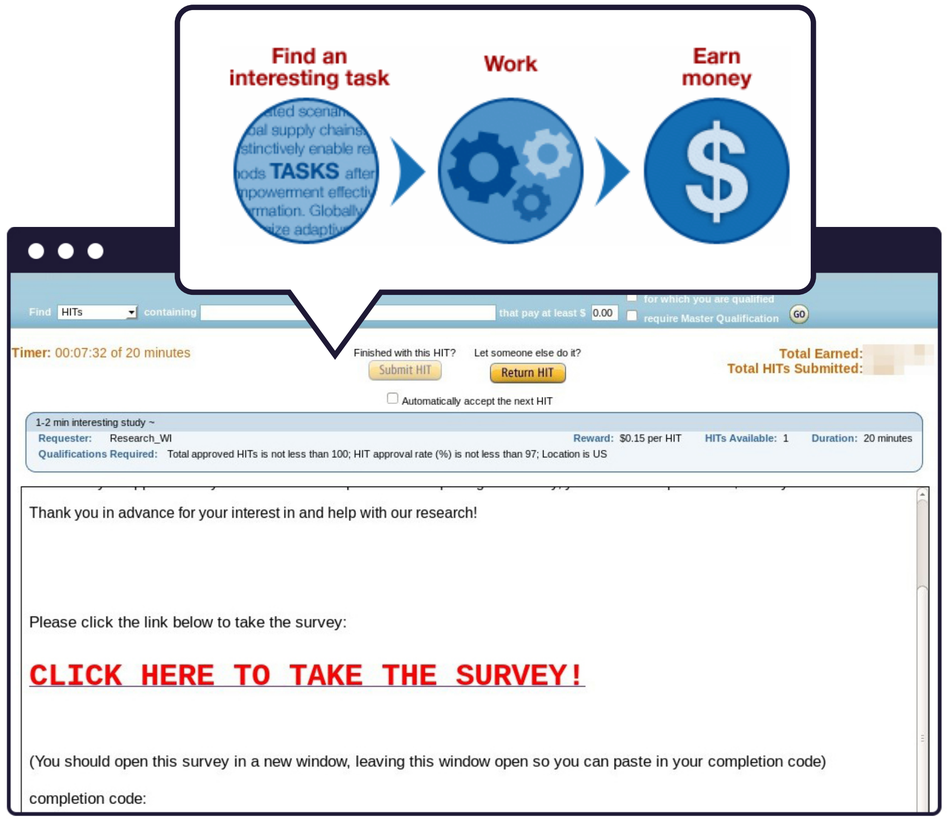
“Regular” programming

Machine Learning

the part you should work on

Where human knowledge in AI *really* comes from

- Mechanical Turk
- **human annotators**
- ~$5 per hour
- boring tasks
- low incentives



Images: Amazon Mechanical Turk, depressing.org

Don’t expect great data if you’re boring the shit out of underpaid people.

Solution #1

Ask simple questions, even for complex tasks

- better annotation **speed**
- better, easier-to-measure **reliability**
- in theory: **any task** can be broken down into a sequence of simpler or even binary decisions


94
Things you find in a pencil case
pen
22
Pen
7
32
Pencil
22
Eraser
6
chalk
Q W E R T Y U I O P
A S D F G H J K L
Z X C V B N M
123
space
Send

No scopes in that one either, but the M1 PRODUCT was a killmachine.

SOURCE: Reddit SECTION: badcompany2 SCORE: 0.74

SOURCE: Unsplash BY: Kirk Morales URL: unsplash.com/@knation

Thanks for your great work – really made my day!

- happy
- sad
- angry
- neutral

POSITIVE

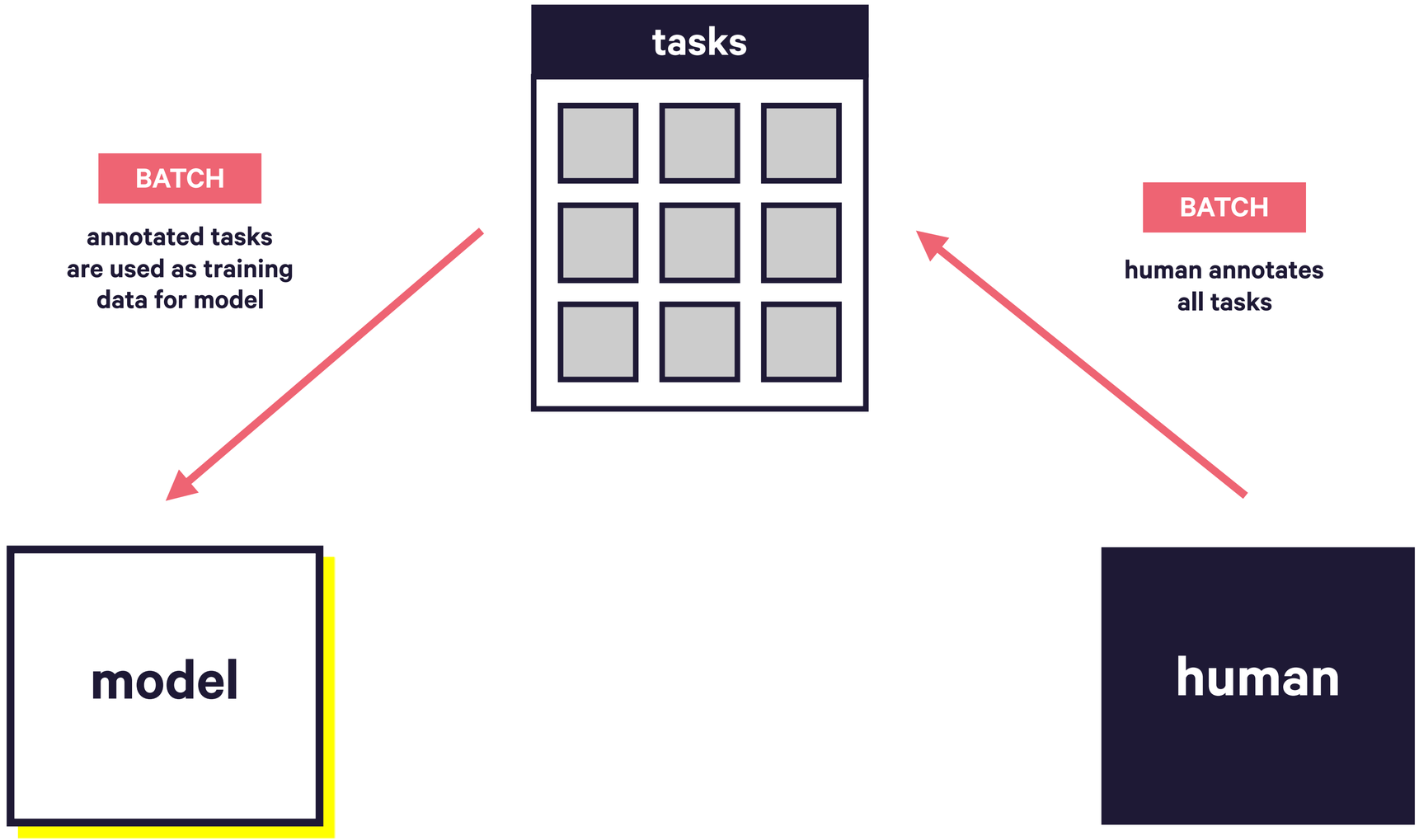
Yahoo Decides to Release a Rosy Forecast

SOURCE: The New York Times

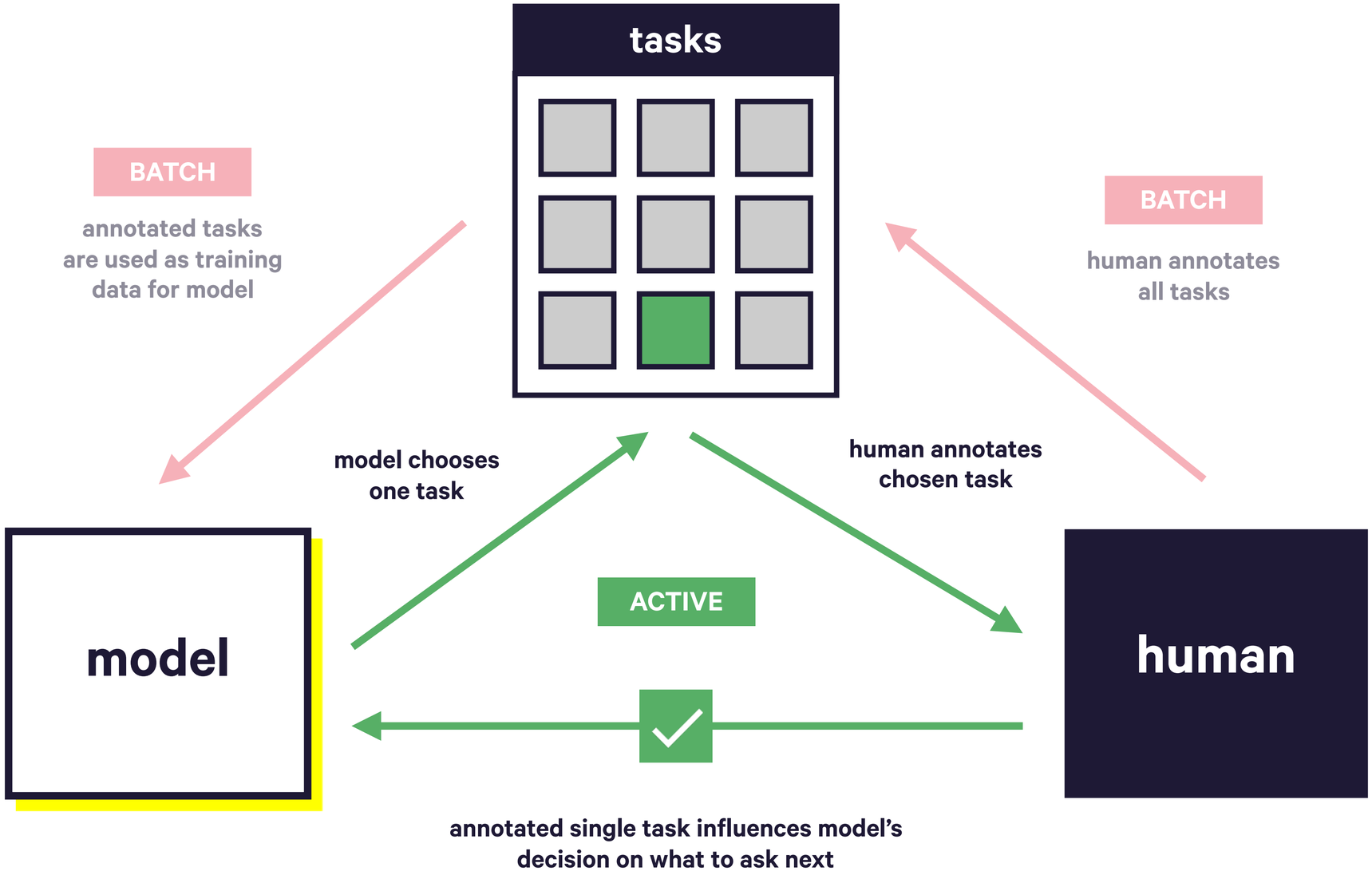
Solution #2

UX-driven data collection with active learning

- assist human with **good UX** and **task structure**
- the things that are hard for the computer are usually easy for the human, and vice versa
- don’t waste time on what the model already knows, ask human about what the model is **most interested in**



Batch learning vs. active learning approach to annotation and training

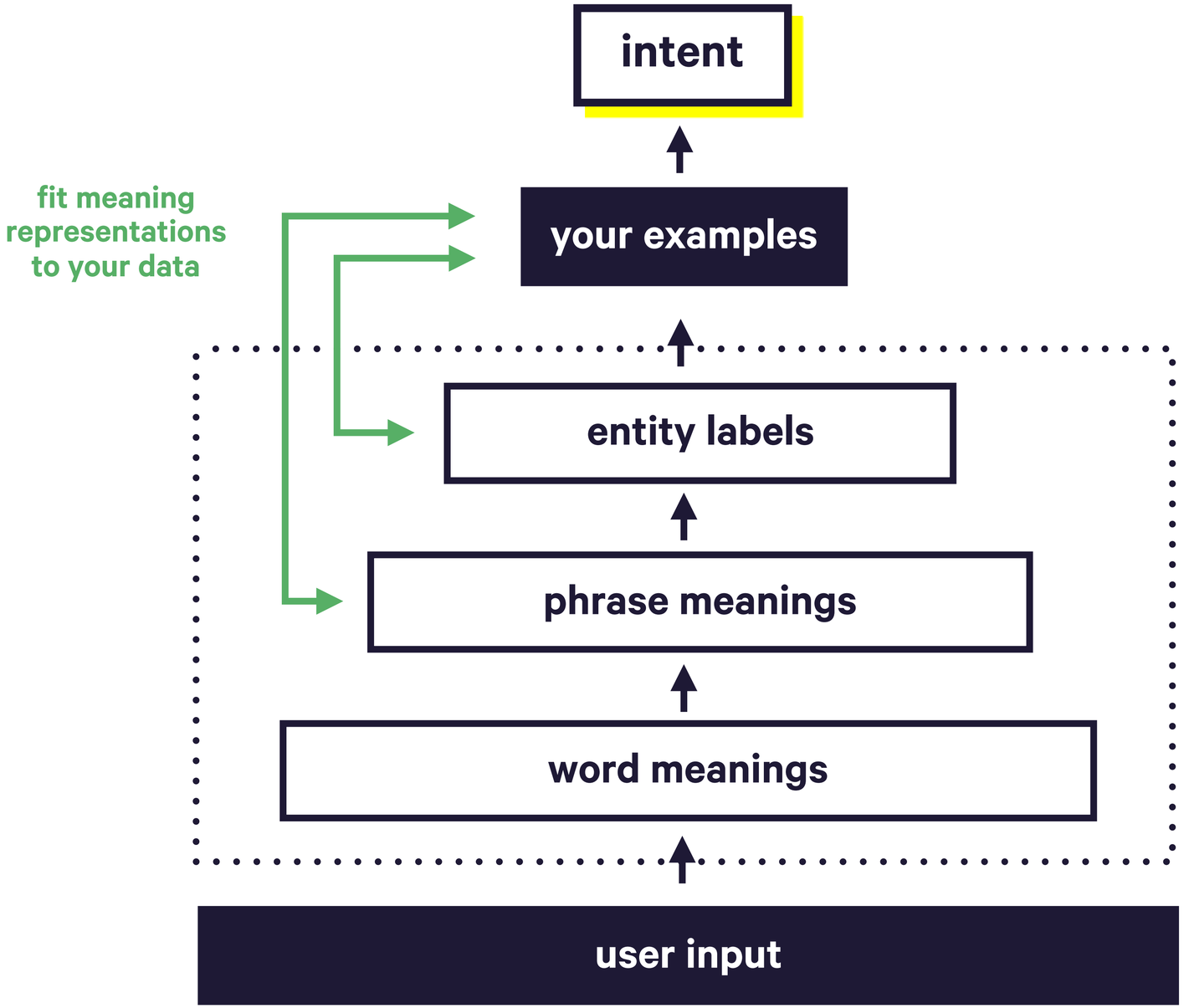


Batch learning vs. active learning approach to annotation and training

Solution #3

Import knowledge with pre-trained models

- start off with **general information** about the language, the world etc.
- **fine-tune** and improve to fit custom needs
- big models can work with **little training data**
- **backpropagate** error signals to correct model



“whats the best way to catalinas”

If you can master annotation…

If you can master annotation…

- … you can try out **more ideas** quickly. Most ideas *don't* work – but some succeed wildly.
- … fewer projects will fail. Figure out **what works** *before* trying to scale it up.
- … you can build entirely **custom solutions** and nobody can lock you in.

Thanks!

💥 **Explosion AI**
explosion.ai

📲 **Follow us on Twitter**
@_inesmontani
@explosion_ai