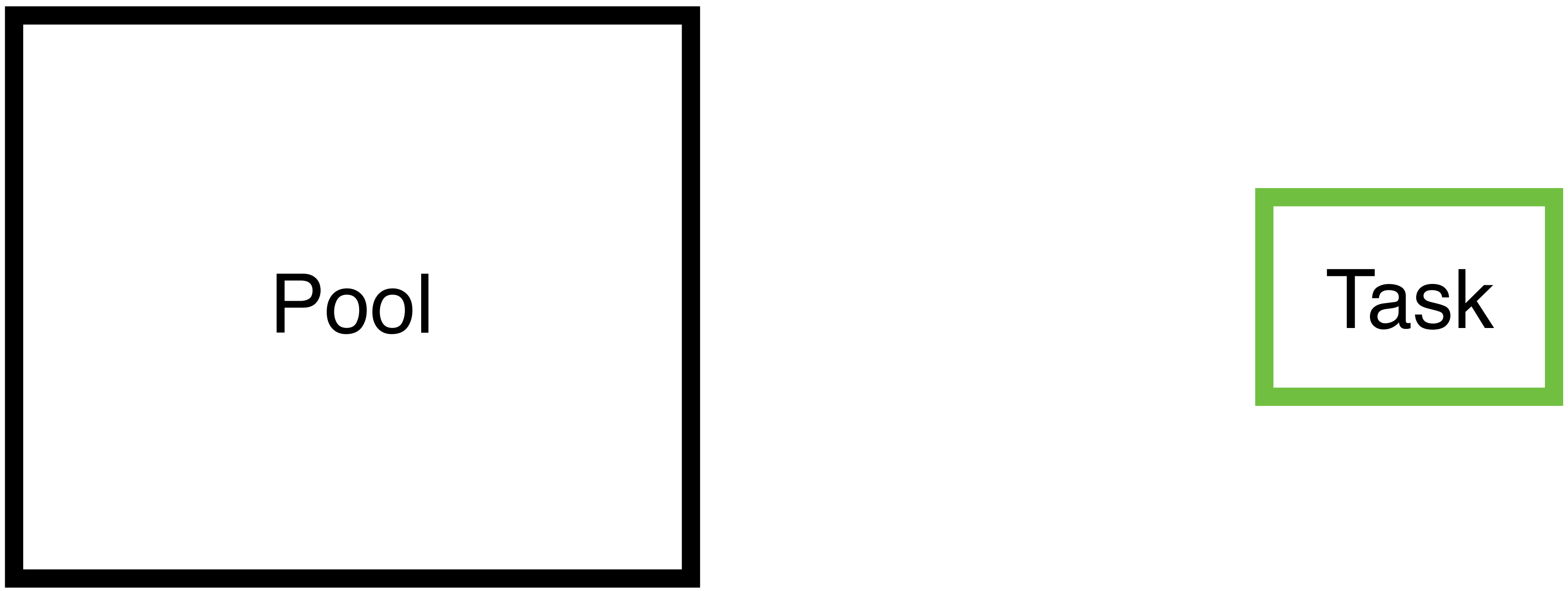
Data Selection
with
Cluster-Based Language Difference Models
and
Cynical Selection

Lucía Santamaría
lucsan@amazon.com

Amittai Axelrod
amittai@amazon.com

Domain Adaptation



Lots of data

Data I care about

Domain Adaptation

Pool

Lots of data

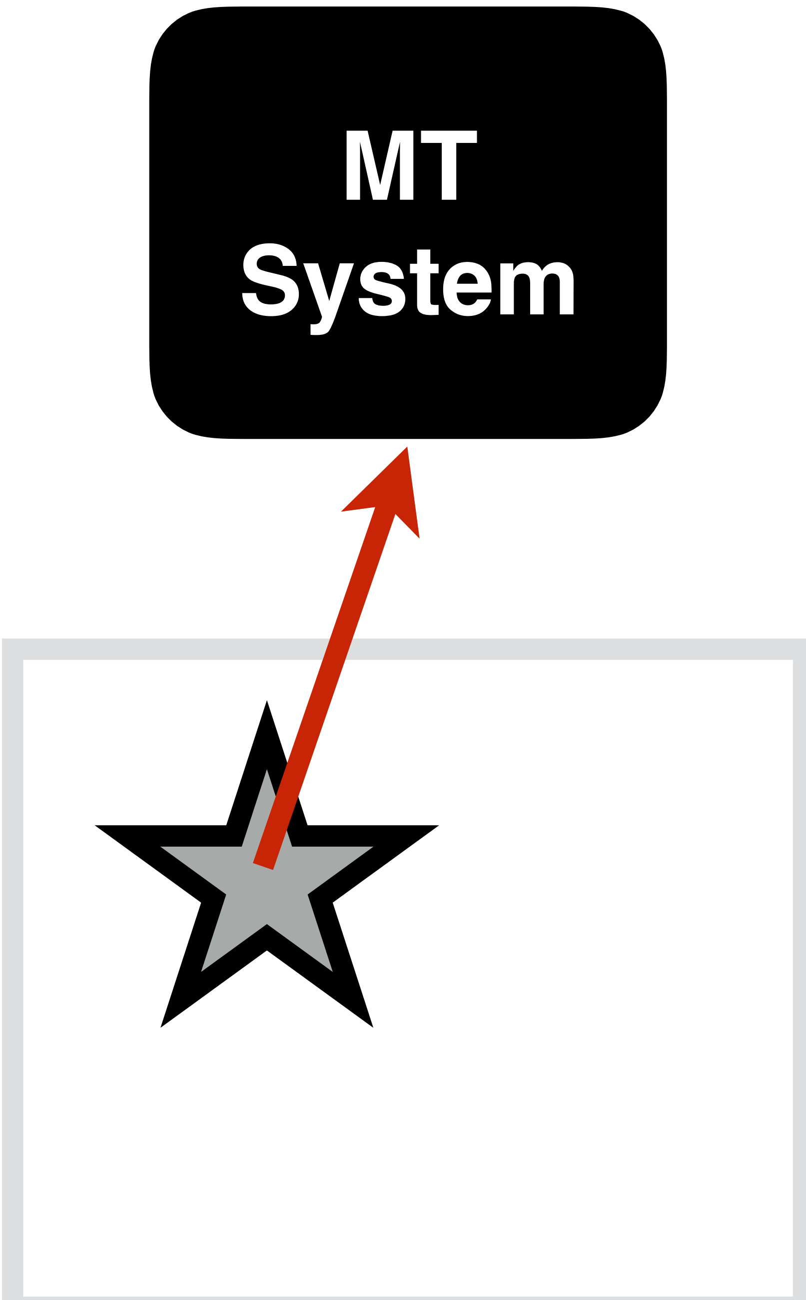
Task

Data I care about

Use this…

…for that

Data Selection



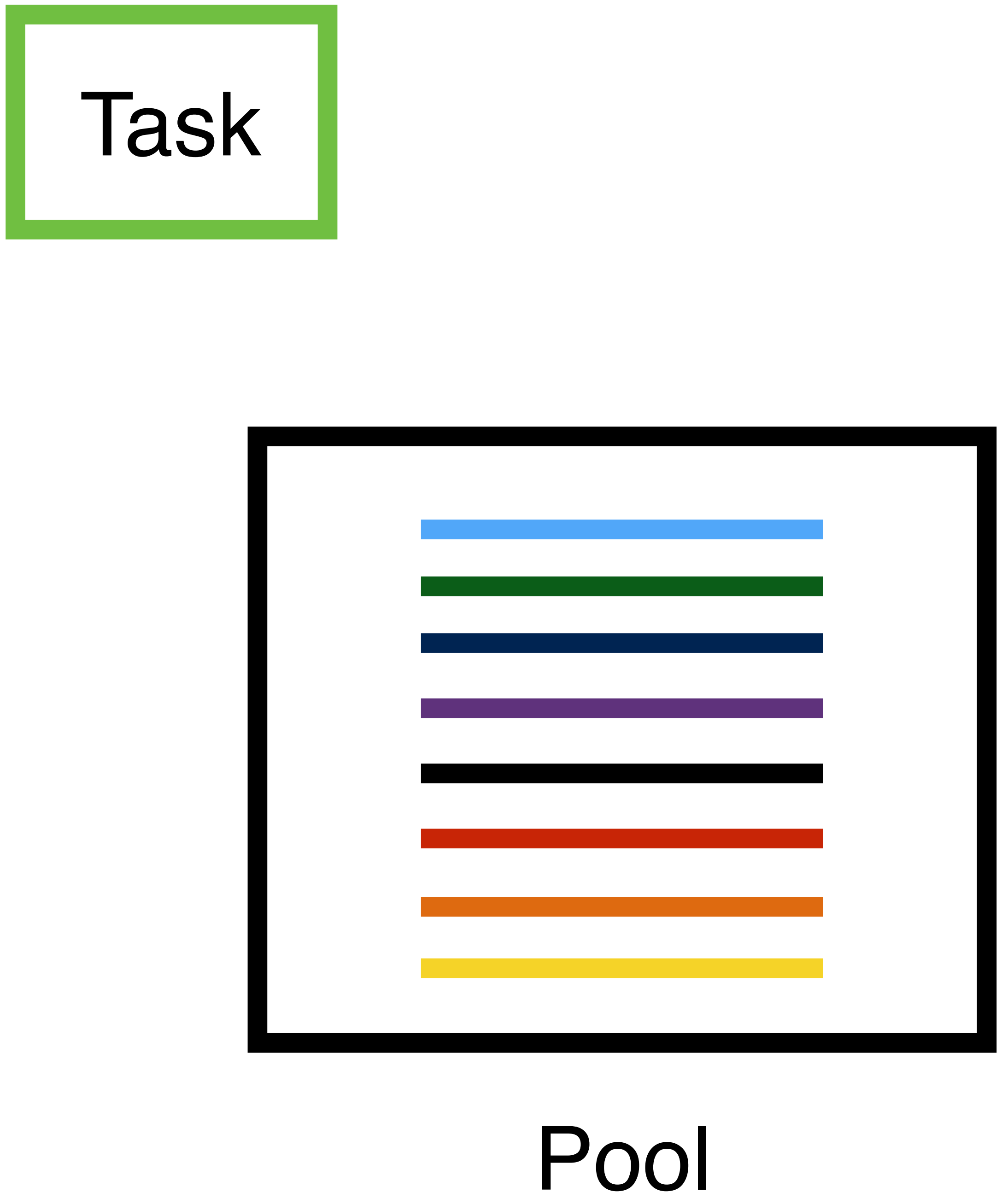
- Use same MT toolkit, with better input!
- Outdated: "There's no data like more data."

↓

There's no data like **relevant** data!

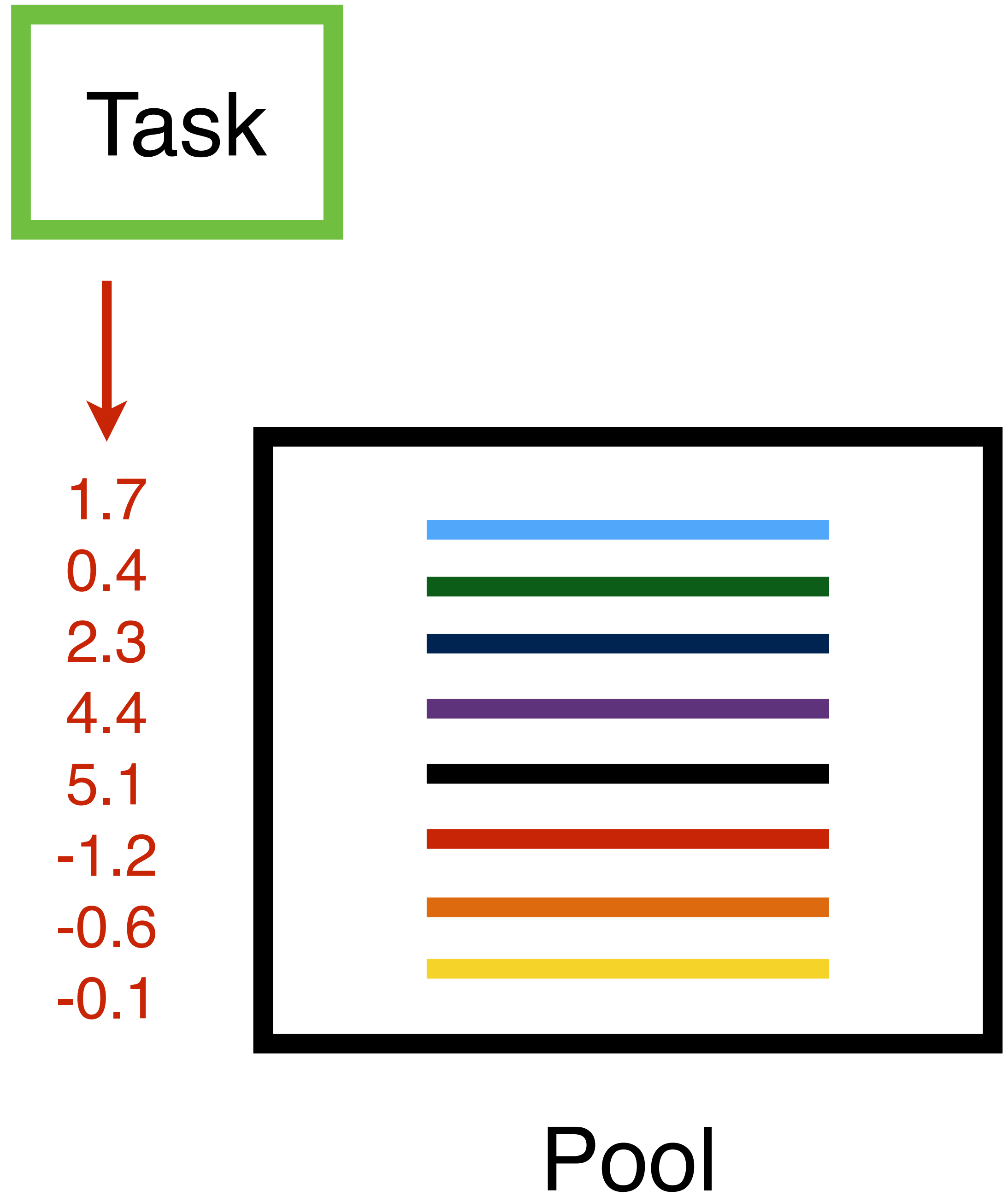
Data Selection Process

- Compute similarity of sentences in pool to the task corpus
- Sort pool sentences by score
- Select (keep) some
- Build task-specific MT system



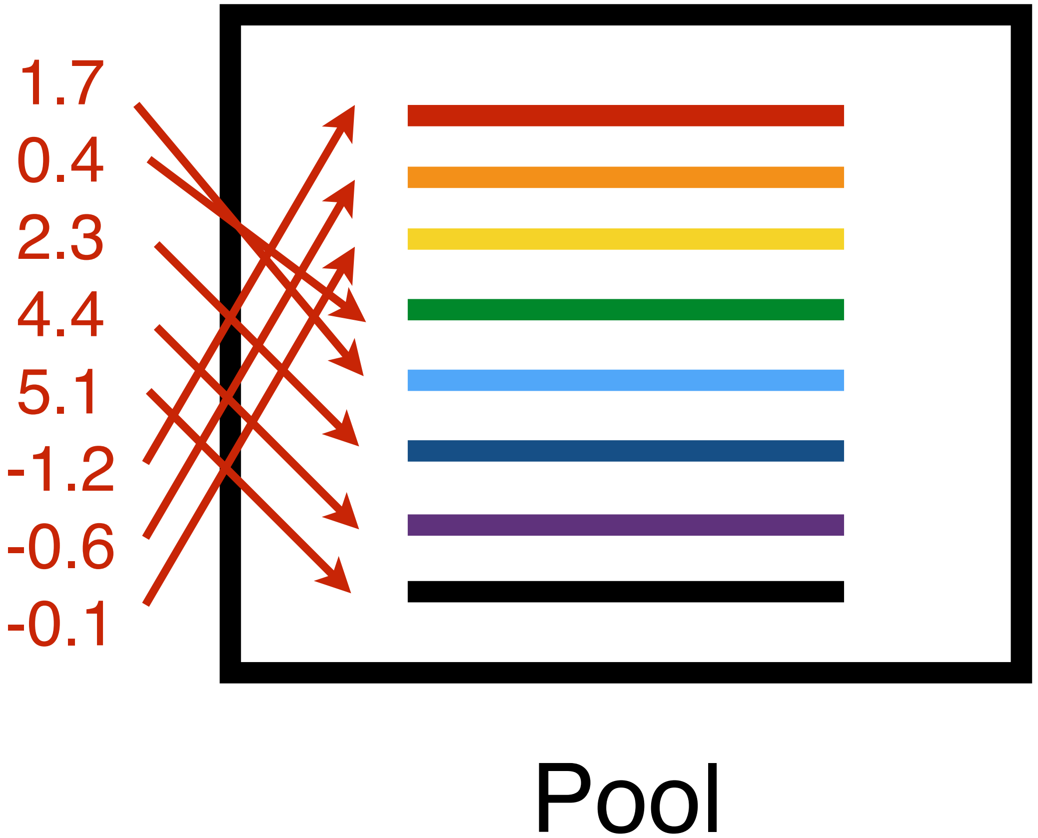
Data Selection Process

- Compute similarity of sentences in pool to the task corpus
- Sort pool sentences by score
- Select (keep) some
- Build task-specific MT system



Data Selection Process

- Compute similarity of sentences in pool to the task corpus
- Sort pool sentences by score
- Select (keep) some
- Build task-specific MT system



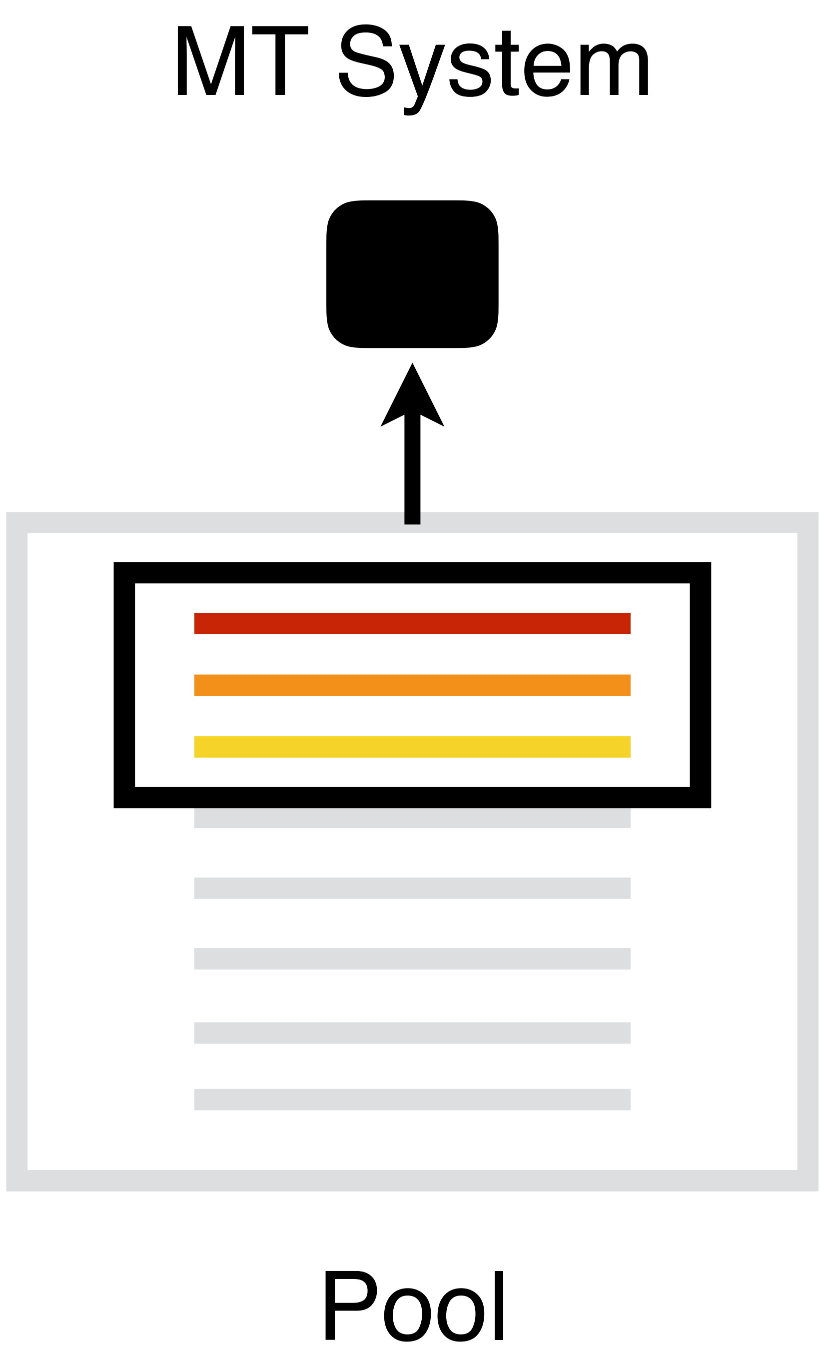
Data Selection Process

- Compute similarity of sentences in pool to the task corpus
- Sort pool sentences by score
- Select (keep) some
- Build task-specific MT system

Pool

Data Selection Process

- Compute similarity of sentences in pool to the task corpus
- Sort pool sentences by score
- Select (keep) some
- Build task-specific MT system



- Data selection:

  There's no data like relevant data!

Cross-Entropy Difference

$$\underset{s \in Pool}{\operatorname{argmin}} \; H_{LM_{Task}}(s) - H_{LM_{Pool}}(s)$$

- (Sometimes called "Moore-Lewis method")
- This prefers sentences that both:
  - Are <u>like</u> the target task
  - Are <u>unlike</u> the pool average.

Class-Based Moore-Lewis

We only want the relationship between two texts.

We don't need to model either of them separately.

Deriving Intuition

From definition of cross-entropy difference:

$$\begin{aligned}
\text{score}(s) &= H_{LM_{Task}} - H_{LM_{Pool}} \\
&= -\frac{1}{N}\sum_{w \in s} \log LM_{Task}(w) - -\frac{1}{N}\sum_{w \in s} \log LM_{Pool}(w) \\
&= -\frac{1}{N}\sum_{w \in s} [\log LM_{Task}(w) - \log LM_{Pool}(w)] \\
&\propto \sum_{w \in s} \log \frac{LM_{Task}(w)}{LM_{Pool}(w)}
\end{aligned}$$

$$\boxed{\text{score}(s) \propto \sum_{w \in s} \log \frac{P_{Task}(w)}{P_{Pool}(w)}}$$

unigram frequency ratio

Not All Words Are Equal

- Scores depend on word probability ratio.
  - <u>Rare</u> word statistics aren't trustworthy
    (count close to 0)
  - <u>Fair</u> words don't affect the score
    (ratio close to 1)
  - <u>Biased</u> words matter the most
    (ratio close to 0 or very large)
- Move bias information into the corpus!

Aggregating Statistics

- Bias info alone does not change word statistics
- Need to also take some information <u>out</u>!
- Collapse words-of-a-kind together:

  Replace words with Brown cluster labels
  (fully unsupervised, for any language)

- [previously @IWSLT 2015: POS tags]

Marking Bias Explicitly

- Replace word with its cluster label,
  and add a suffix indicating unigram frequency ratio

- “ozeanen” —> “862/+”
  Brown cluster #862,
  1 order of magnitude more common in the Task

- “…” —> “3/-“
  Brown cluster #3,
  1 order of magnitude more common in the Pool

Marking Bias Explicitly

- “ozeanen” —> “862/+”
  Brown cluster #862,
  1 order of magnitude more common in the Task

- Now each corpus knows about the other one!

- Bias (ratio) changes for each Task/Pool pair,
  allows for nuance in relationship

Cluster-based Methodology

- Drop-in addition to existing method!
  1. Compute Brown clusters for the corpora
  2. Compute vocab statistics and ratios
  3. Transform text
  4. Do cross-entropy difference data selection
  5. Put words back in, and carry on!

- Data selection:

  There's no data like relevant data!

- Language difference models:

  model each corpus <u>relative to the other one</u>

Experimental Setup

- German --> English translation

- Task: TED, 218 k lines (4 m tokens)
  Pool: WMT, 17.6 m lines (235 m tokens)

- Vocab (De): 1.1m
  Vocab (En): 900k

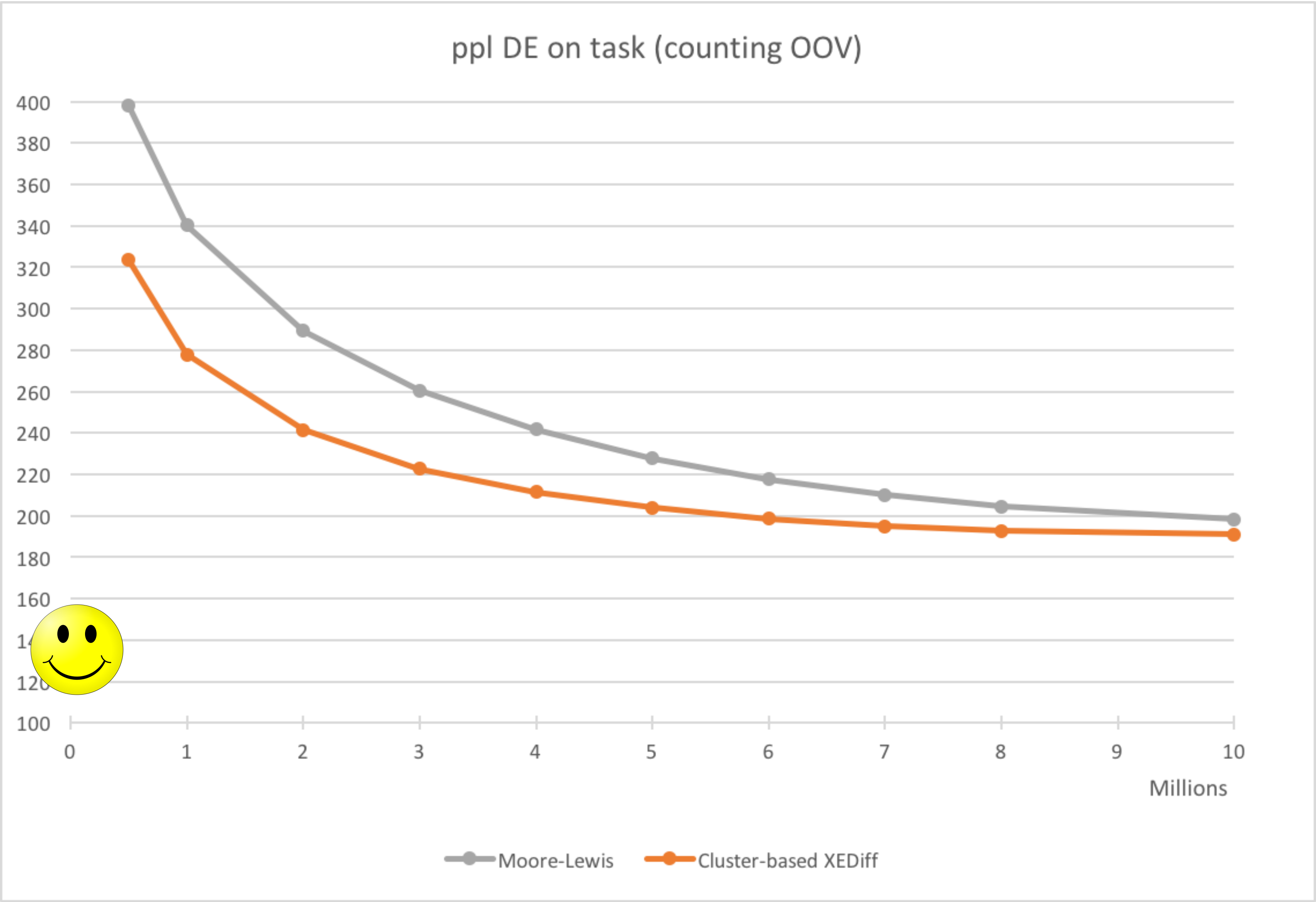
- 1,000 Brown clusters x 8 bias labels,
  collapsing lexicon to < 3000 types

In-Domain Perplexity

- cluster-based:

  -20 ppl

-

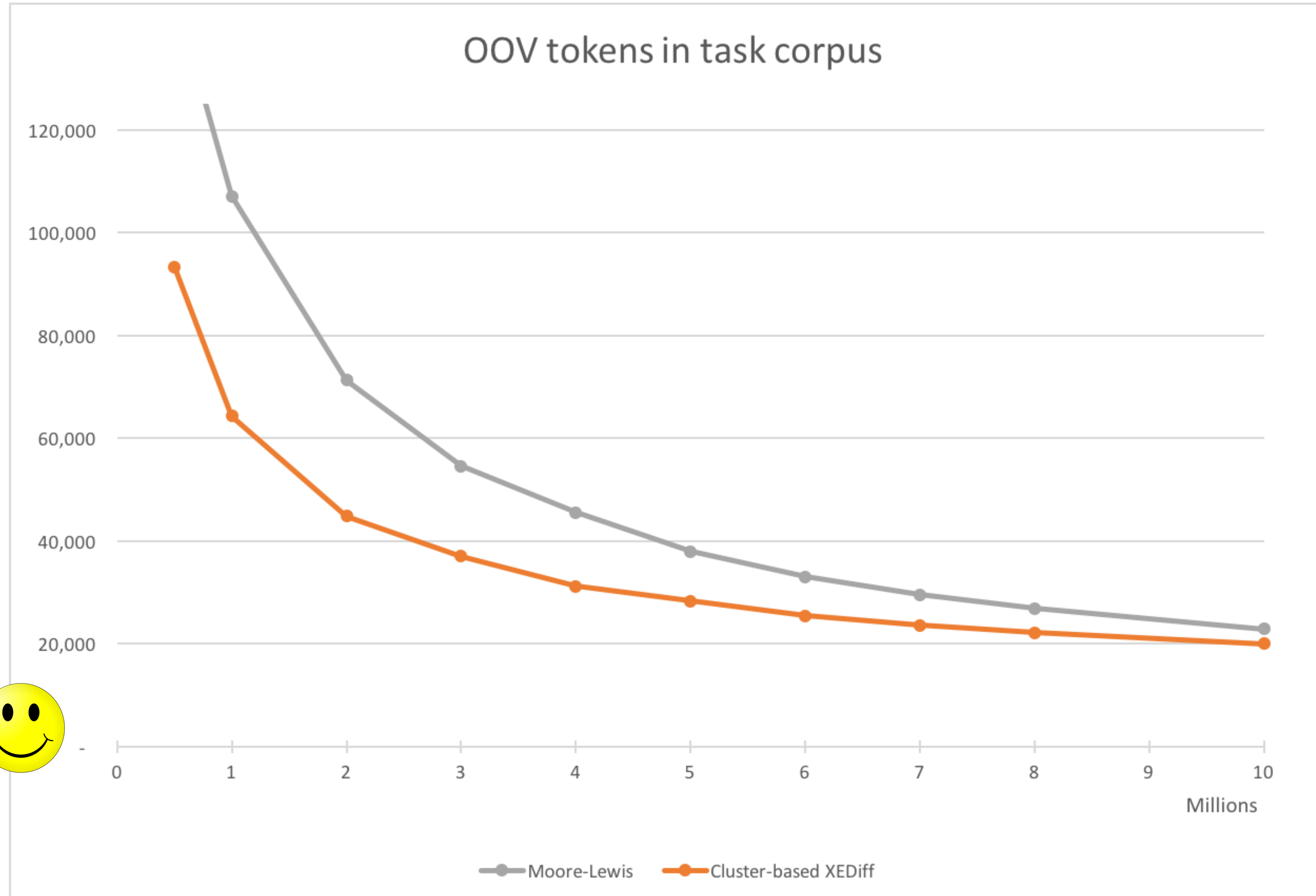


In-Domain Lexical Coverage

- cluster-based:

  -33% oov

-



Moore-Lewis Limitations

Cross-entropy difference…

… treats Task and Pool as the opposing ends of a single spectrum "IN = Good, OUT = Bad"

… Not guaranteed to model nor cover in-domain data. LM_{TASK} likes them; this is necessary, but not sufficient.

… No intuition as to how many sentences to select. Grid search doesn't count.

Moore-Lewis Limitations

Cross-entropy difference…

… treats Task and Pool as the opposing ends of a single spectrum "IN = Good, OUT = Bad"

… Not guaranteed to model nor cover in-domain data. LM_{TASK} likes them; this is necessary, but not sufficient.

… No intuition as to how many sentences to select. Grid search doesn't count.

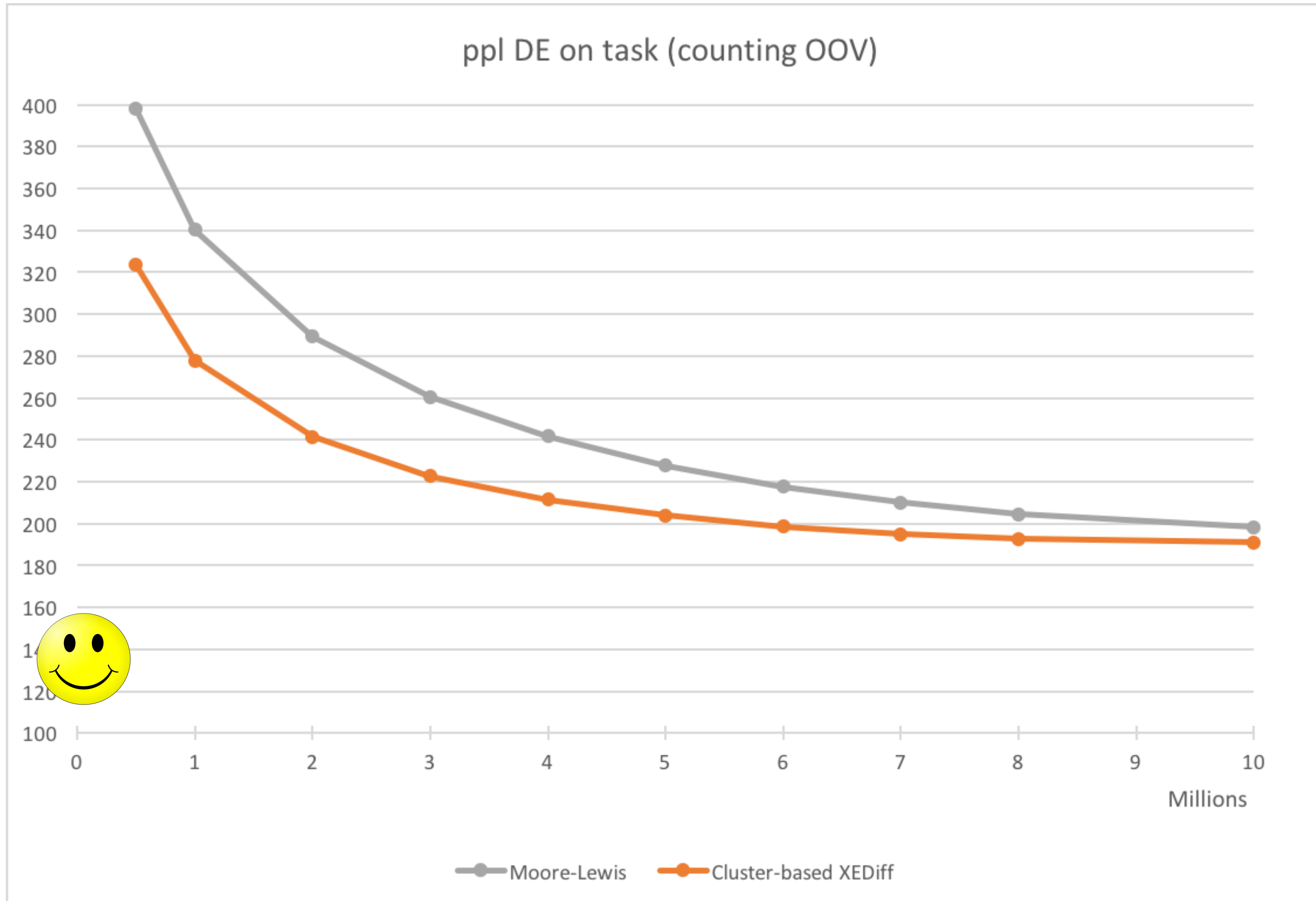
BUT IT WORKS — WHY CHANGE?

In-Domain Perplexity

- cluster-based:

  -20 ppl

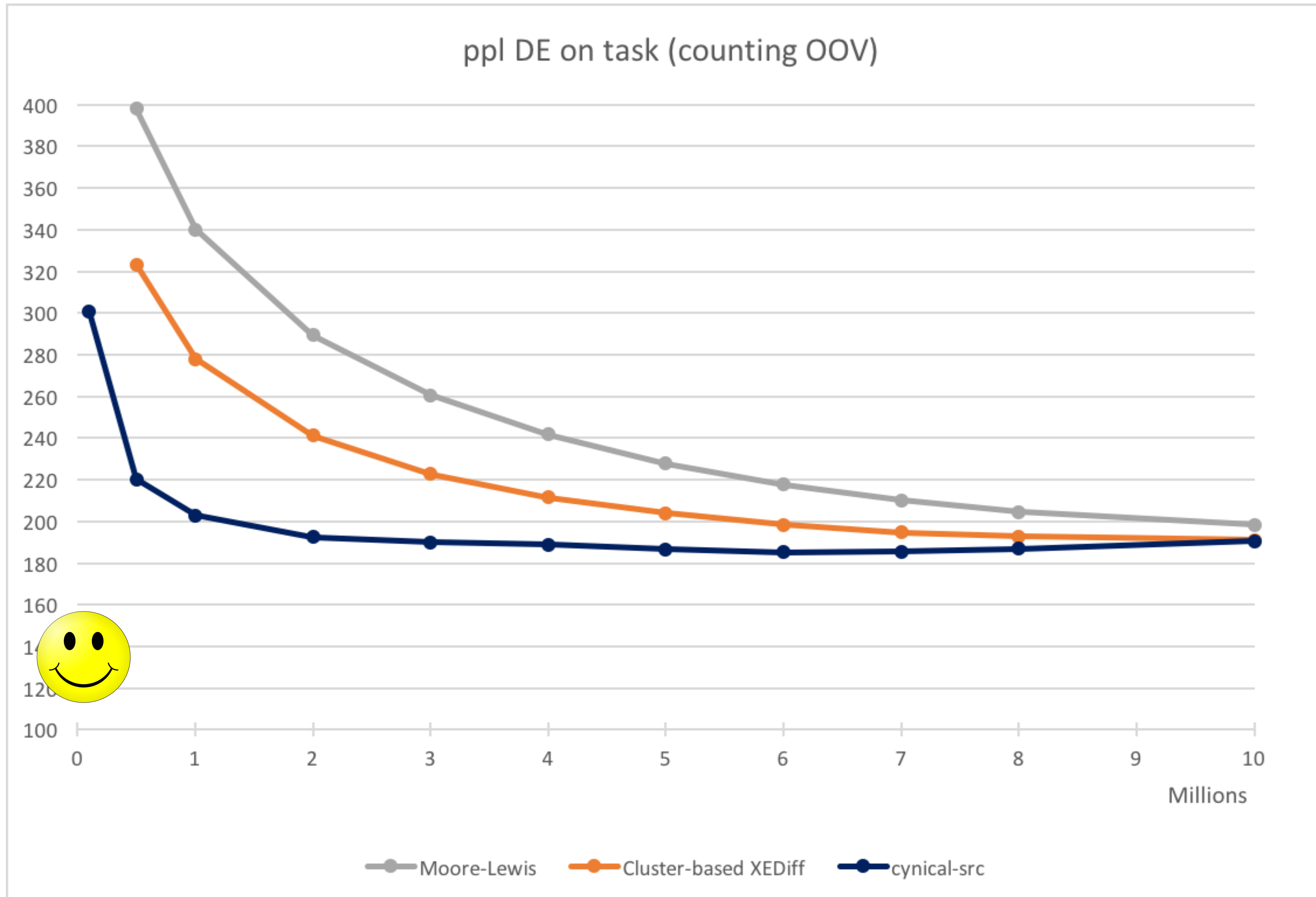
-



In-Domain Perplexity

- cluster-based:

  -20 ppl



- cynical:

  -30 ppl

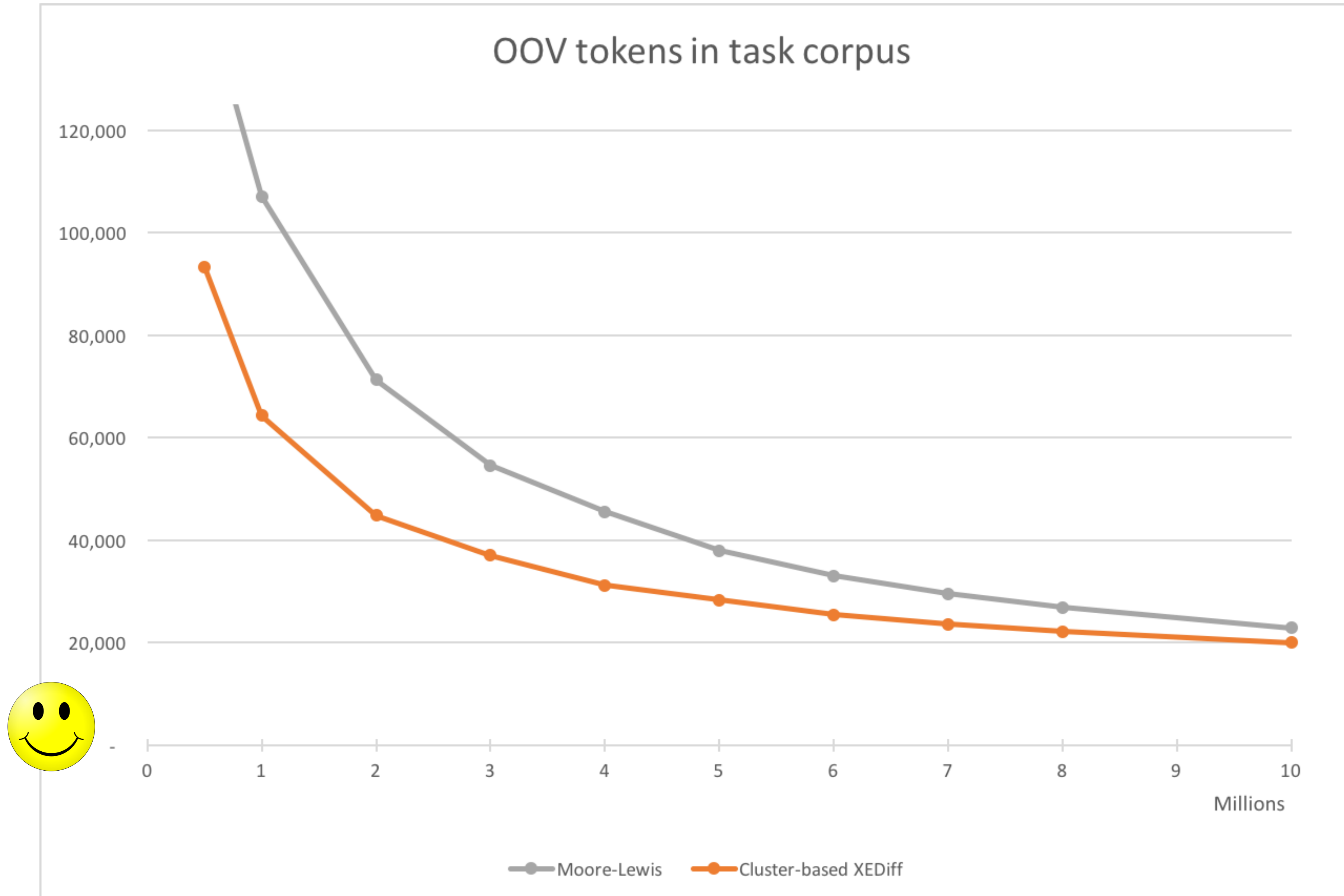
…and -66% less data.

In-Domain Lexical Coverage

- cluster-based:

  -33% oov

-



In-Domain Lexical Coverage

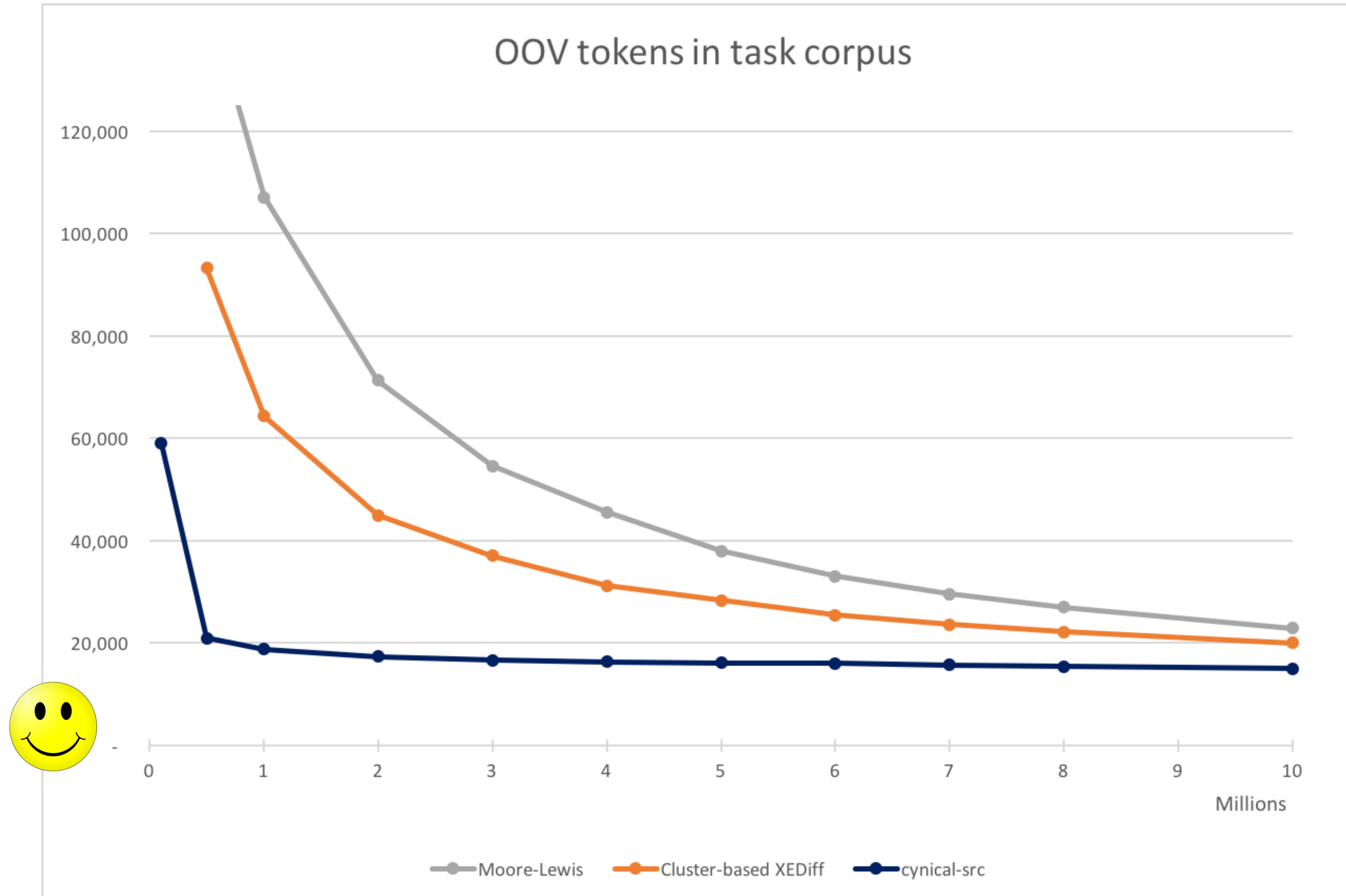
- cluster-based:

  -33% oov

- cynical:

  -85% oov

…and -83% less data.



Cynical Motivation

Cross-entropy difference…

… treats Task and Pool as the opposing ends of a single spectrum "IN = Good, OUT = Bad"

… Not guaranteed to model nor cover in-domain data. LM_{TASK} likes them; this is necessary, but not sufficient.

… No intuition as to how many sentences to select. Grid search doesn't count.

BUT IT WORKS — WHY CHANGE? …AH.

Cynical Data Selection

- “an incremental greedy selection scheme based on relative entropy, which selects a sentence if adding it to the already selected set of sentences reduces the relative entropy with respect to the in-domain data distribution” [Sethy et al, 2006]

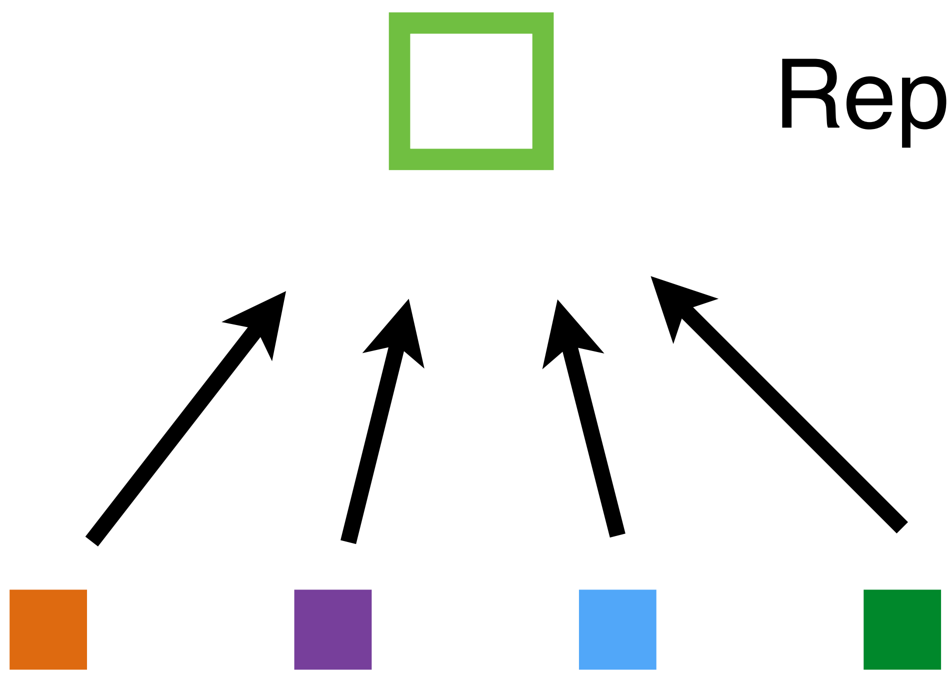
- incrementally grow the training corpus only based on how **useful** the data is

- “does it help me **now**?”

Cynical Data Selection

- How many bits of information would we learn
  if we added this line to our corpus?
- Only add sentences that can be **proven** to make the model better.
- Pick most informative lines first.

Cynical Selection Process

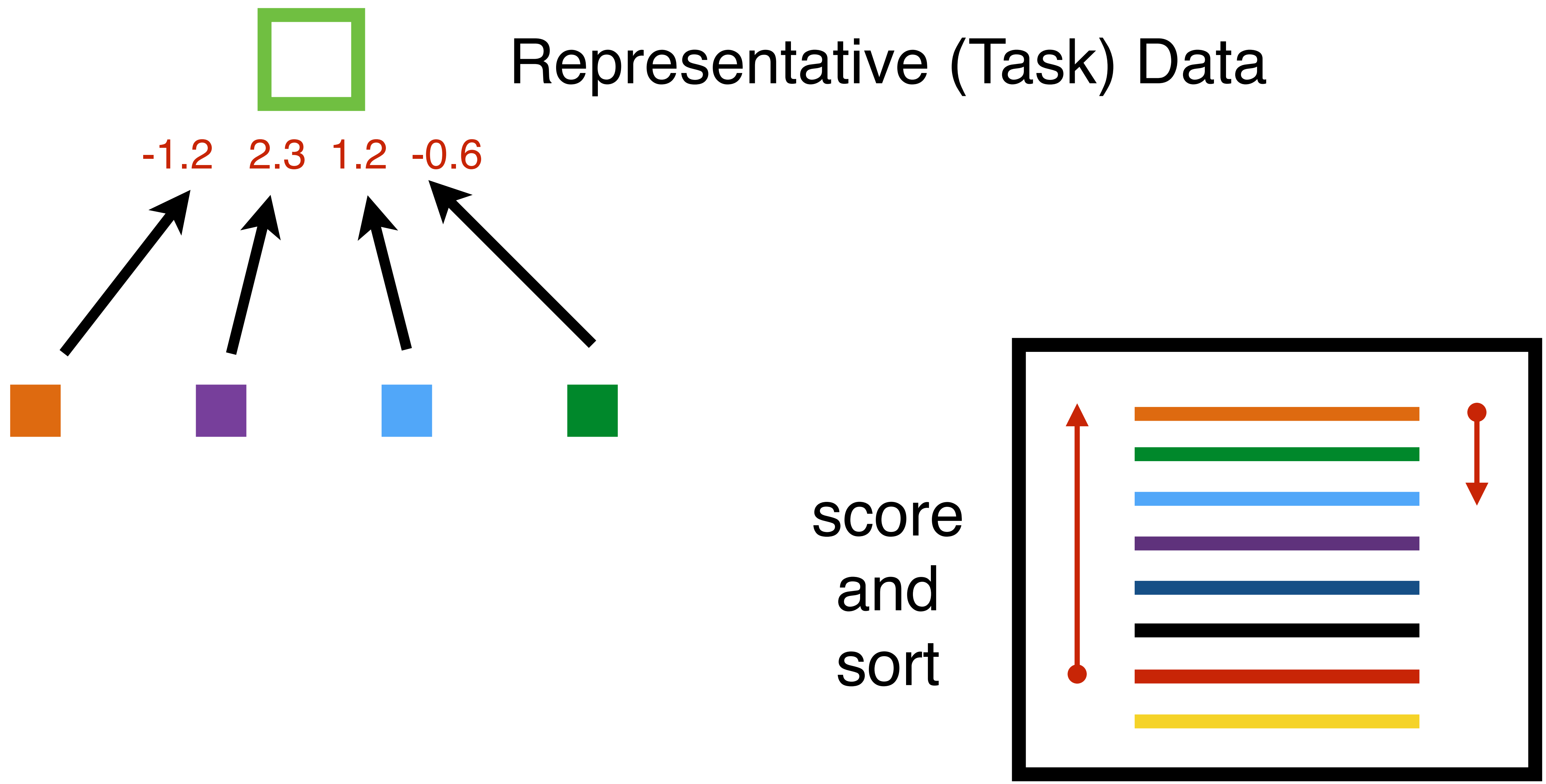


Representative (Task) Data

Can it model?

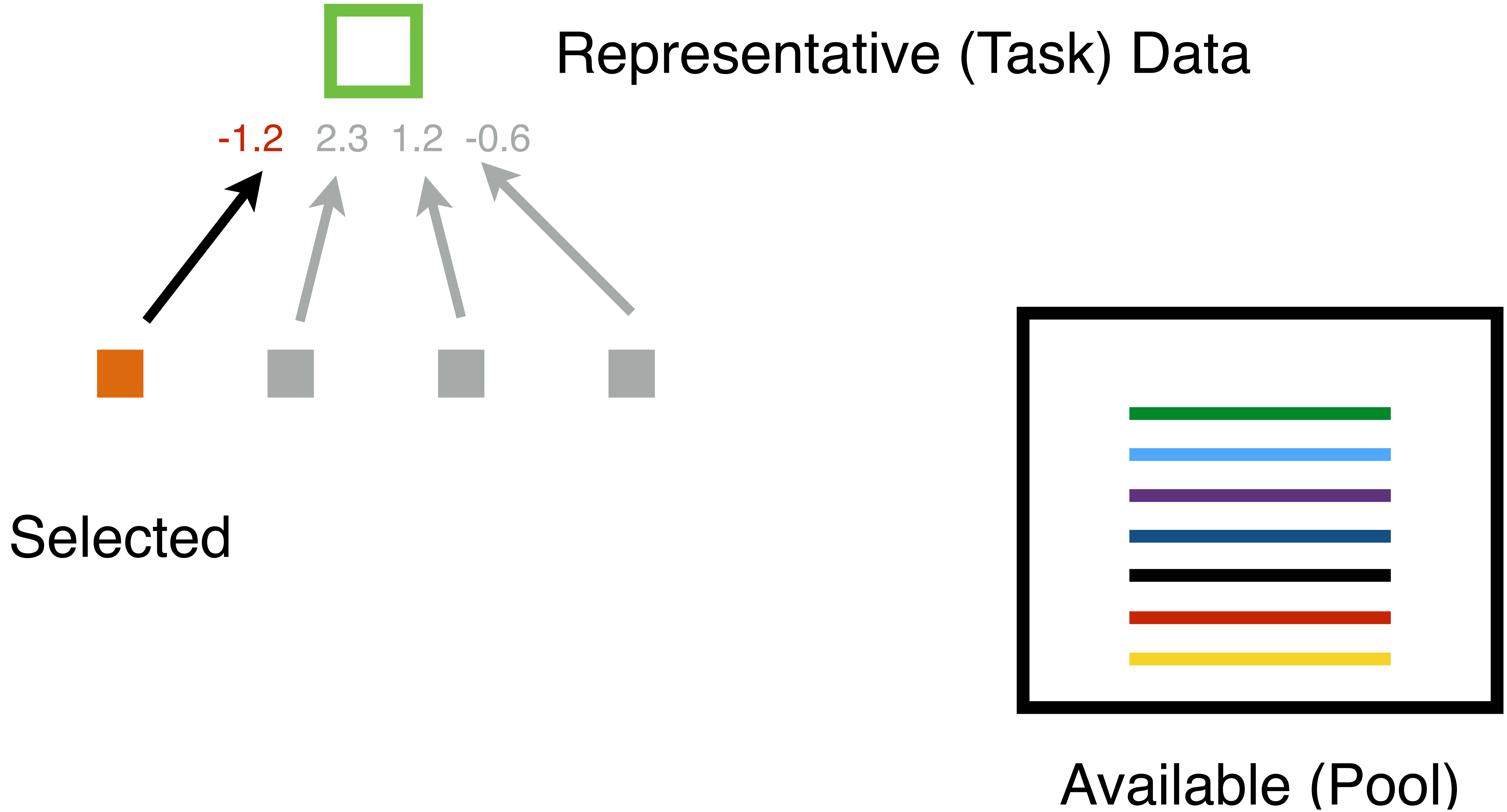
Available (Pool)

Cynical Selection Process

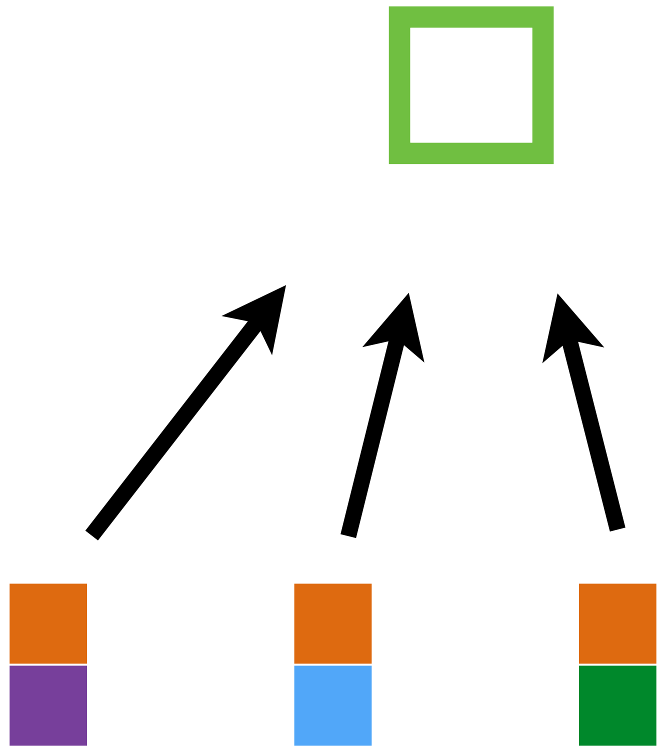


Available (Pool)

Cynical Selection Process



Cynical Selection Process

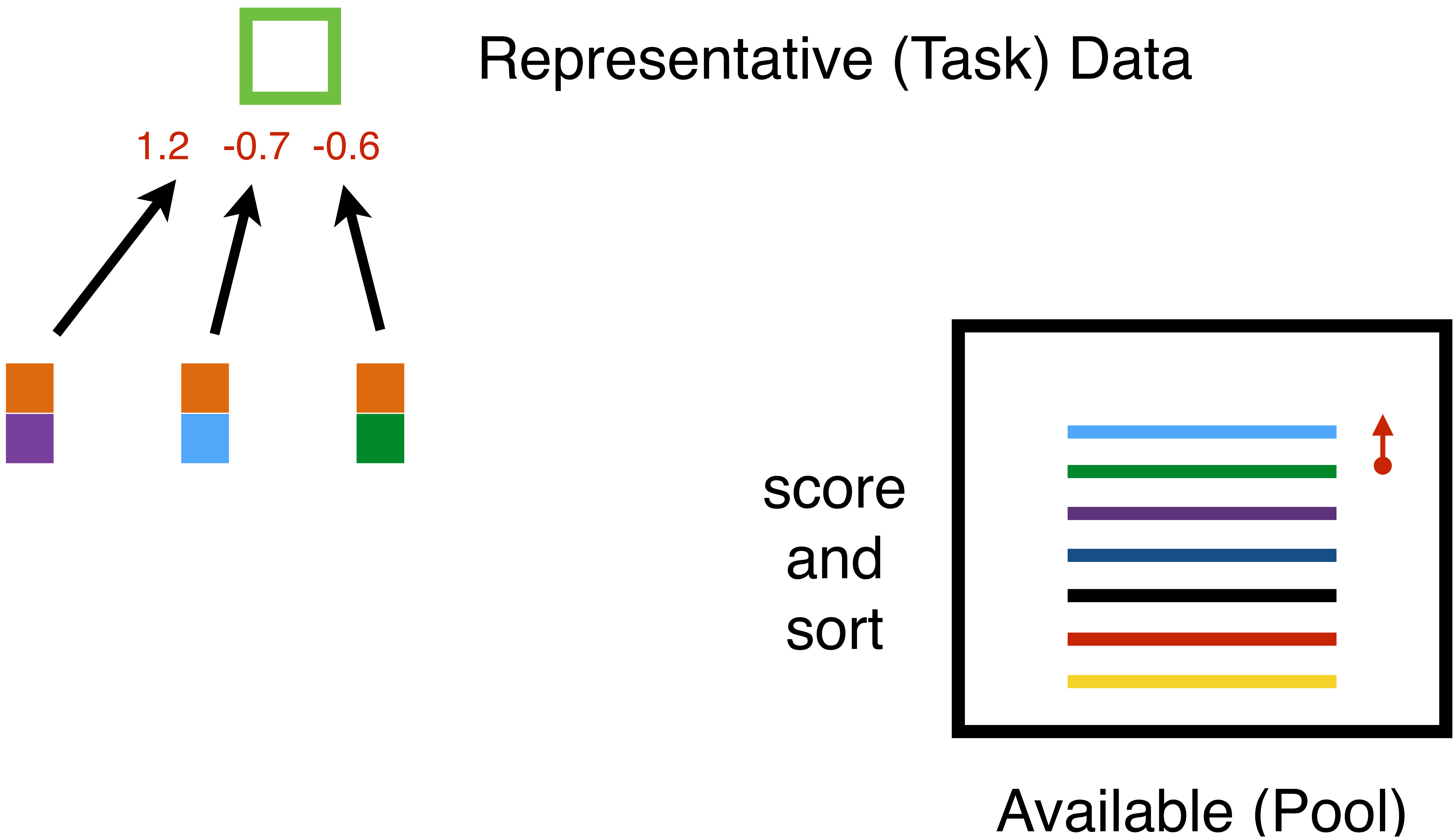


Representative (Task) Data

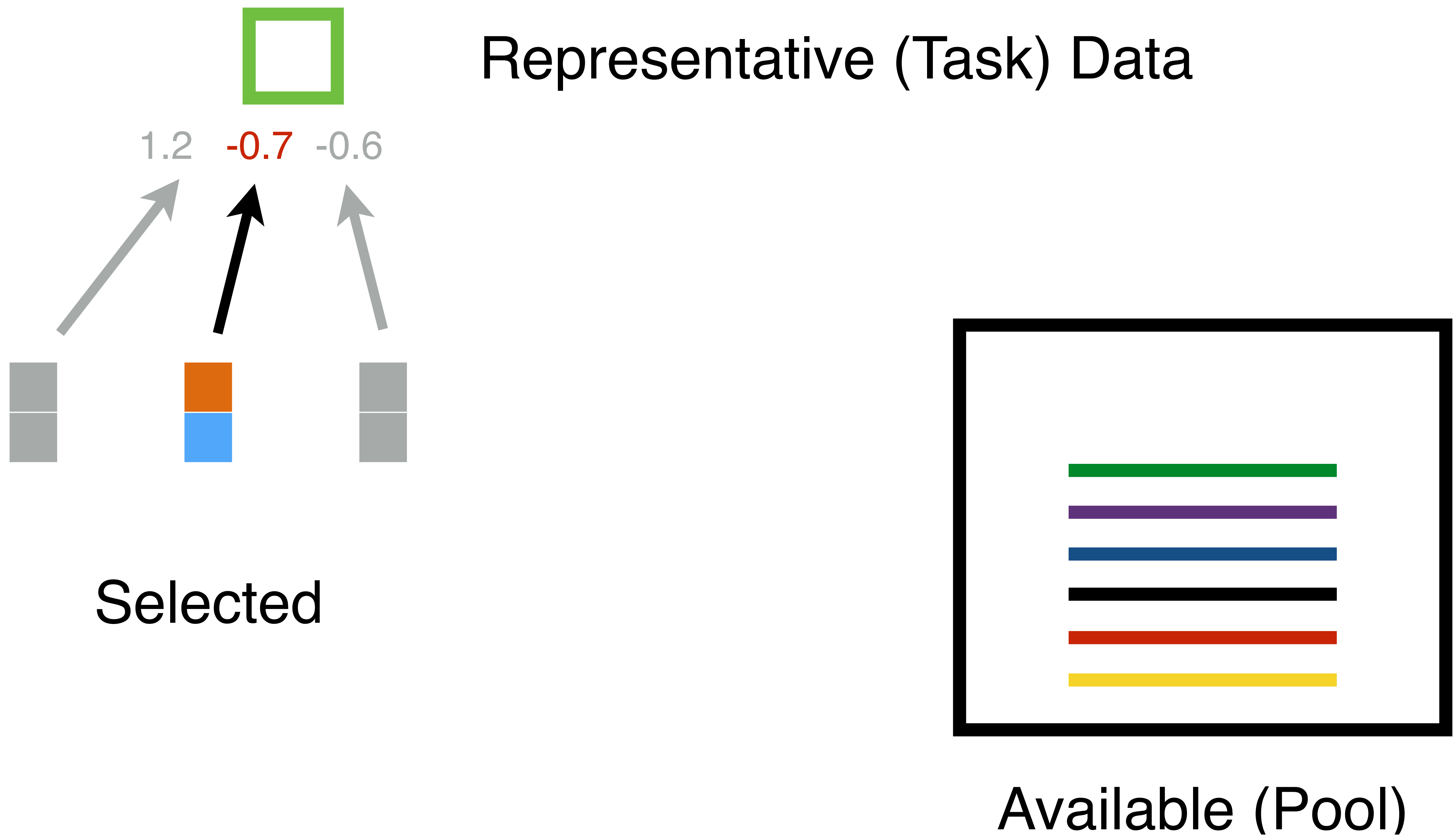
Can it model?

Available (Pool)

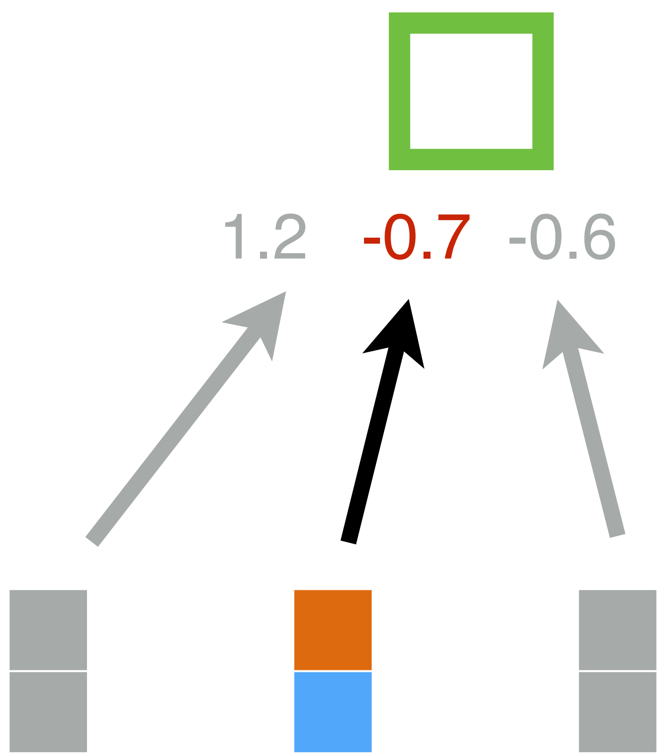
Cynical Selection Process



Cynical Selection Process



Cynical Selection Process



Representative (Task) Data

Where does this number come from?

What does it mean?

Available (Pool)

Quantifying Subset Score

- Pick any *n* lines. Is that subset any good?
- See how well they model the task !
- Cross-entropy between subset and the task is:

$$H_n(\mathrm{REPR}) = -\sum_{v \in V_{\mathrm{REPR}}} \boxed{\frac{C_{\mathrm{REPR}}(v)}{W_{\mathrm{REPR}}} \log \frac{C_n(v)}{W_n}}$$

(bits of entropy)

P(n) log Q(n)

Inductive Step

- After picking *n* lines, how do we pick line *n+1* ?
- Need to score all potential choices.
- We already computed H_n , so

$$H_{n+1} = H_n + \underset{n \to n+1}{\Delta H}$$

Decompose as:

$$H_{n+1} = H_n + \underset{n \to n+1}{\text{Penalty}} + \underset{n \to n+1}{\text{Gain}}$$

Greedy Cross-Entropy Delta

[Sethy/Georgiou/Narayanan 2006]

$$\underset{n\rightarrow n+1}{\Delta H} = H_{n+1} - H_n$$

$$= \left(- \sum_{v \in V_{\text{REPR}}} \frac{C_{\text{REPR}}(v)}{W_{\text{REPR}}} \log \frac{C_{n+1}(v)}{W_{n+1}} \right) - \left(- \sum_{v \in V_{\text{REPR}}} \frac{C_{\text{REPR}}(v)}{W_{\text{REPR}}} \log \frac{C_n(v)}{W_n} \right)$$

$$\underset{n\rightarrow n+1}{\Delta H} = \boxed{\underbrace{\log \frac{W_n + w_{n+1}}{W_n}}_{Penalty}} + \boxed{\underbrace{\sum_{v \in V_{\text{REPR}}} \frac{C_{\text{REPR}}(v)}{W_{\text{REPR}}} \log \frac{C_n(v)}{C_n(v) + c_{n+1}(v)}}_{Gain}}$$

Penalty Term

$$\underbrace{\log \frac{W_n + w_{n+1}}{W_n}}_{Penalty}$$

All words in line

- Each word we add increases the penalty for the line
- Bias towards shorter sentences
- Penalty for line decreases over time

Gain Term

Only relevant words

$$\underbrace{\sum_{v \in V_{\text{REPR}}} \frac{C_{\text{REPR}}(v)}{W_{\text{REPR}}} \log \frac{C_n(v)}{C_n(v) + c_{n+1}(v)}}_{Gain}$$

- Rewards each word in line that is also in the task
- Bigger reward for higher-probability words
- Bias towards longer sentences
- Gain of line also decreases over time

Selection Criterion

$$\underset{n \to n+1}{\Delta H} = \underset{n \to n+1}{\text{Penalty}} + \underset{n \to n+1}{\text{Gain}}$$

- Computable separately
- Cheap to update
- Approximations are upper bounds:
  - Easy to sort
  - High precision (no bad lines with good scores)

Selection Criterion

$$\underset{n \to n+1}{\Delta H} = \underset{n \to n+1}{\text{Penalty}} + \underset{n \to n+1}{\text{Gain}}$$

- Delta H < 0
  This line adds information (lowers entropy). Select it!

- Delta H > 0
  This sentence makes your model dumber. Leave it!

- Delta H starts < 0, and increases over time.
  Score passes zero when it runs out of useful sentences. Ok to stop!

Naïve Algorithm

- Picking n+1:
  - Compute the Delta H score for each sentence remaining in AVAIL.
  - Sort the sentences in AVAIL by Delta H
  - Select sentence with the best (lowest) score.
  - Remove it from AVAIL.
- Loop

Why Not Do It Like That?

- (*i.e.* "Why wasn't this done in 2006?")
- N iterations $O(N)$
  - Each updating W_{AVAIL} words $O(N)$
    and sort N lines to find best $+ O(N \log N)$
- Total: $= O(N^2 + N^2 \log N) > O(N^2)$
- No thanks!

Implementation

- naïve iterative greedy selection: at least $O(N^2)$
- "Perfect is the enemy of Good"
- What if we just want 'good' and not 'best' ?
- Doable in $O(N \log N)$

Not All Words Are Equal

- Sentence gain score decomposes into word scores:

$$\underset{n \to n+1}{\text{Gain}} = \sum_{v \in v_{n+1}} \underset{n \to n+1}{\text{Gain}}(v)$$

- Dominated by one or two of the word (type) terms, because of Zipfian distribution

“Good Enough”

- What about word with the best gain estimate?
- It will help to <u>eventually</u> add a line with that word.
- We will pick many sentences—
  no harm in adding now.

“Good Enough”

- Pick best sentence containing
  the word with best gain.

- Might not be best sentence, but is good sentence.

- Reduces # lines to evaluate at each step.

Lowering Complexity

- N iterations — N
  - Each updating V words, sorting V words to find best — $V + V \log V$
  - Update and sort AVAIL(v') lines — $N^{1/3} + N^{1/3} \log N^{1/3}$
- Total: — $O(N\,V \log V)$

Squish Lexicon

- Cynical Selection complexity depends on size of V
- Reduce lexicon with insight from Class-based Moore-Lewis
- Focus on words biased towards TASK and away from AVAIL
- Collapse all other words into 5 classes
- Final vocabulary: ~30k words.

“Tractable Enough”

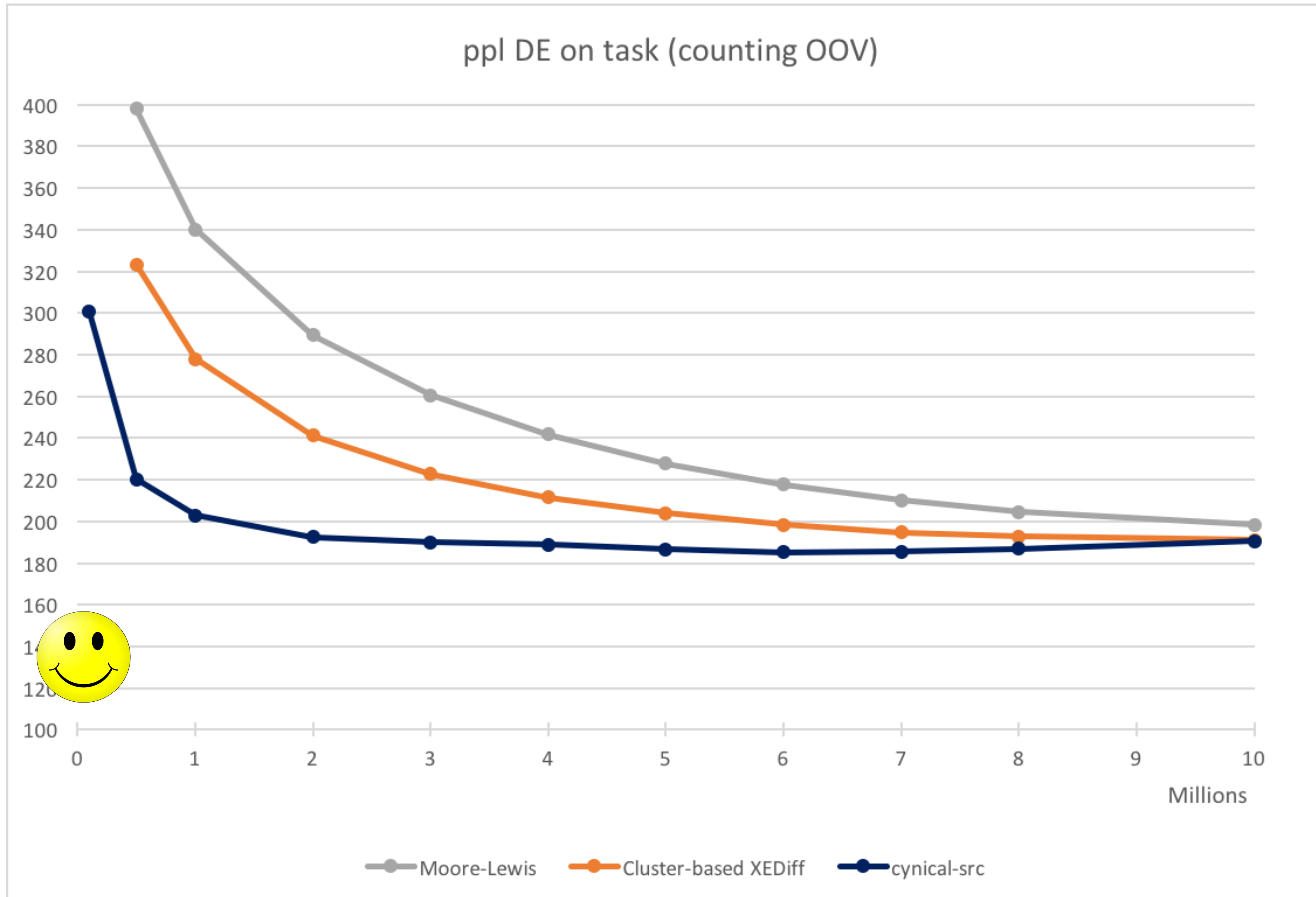
- Complexity of O(N log N) achievable with reduced lexicon and dynamically-sized batching.
- Still super-linear!
  But not terribly (experiments ran in 0.5 - 1 day)
- Also, do not need to run to completion —
  stop when estimated entropy gain stays above 0.

Algorithm in Practice

- Picking n+1:
  - Compute Word Gain Estimate (WGE) for all V
  - Sort, then select word v’ with best WGE
  - Compute the Delta H score for each sentence in AVAIL(v’)
    (set of sentences remaining in AVAIL that contain v’)
  - Sort AVAIL(v’) by Delta H
  - Select sentence with the best (lowest) score.
  - Remove it from AVAIL.
- Loop until best Delta H > 0 for all words.

In-Domain Perplexity

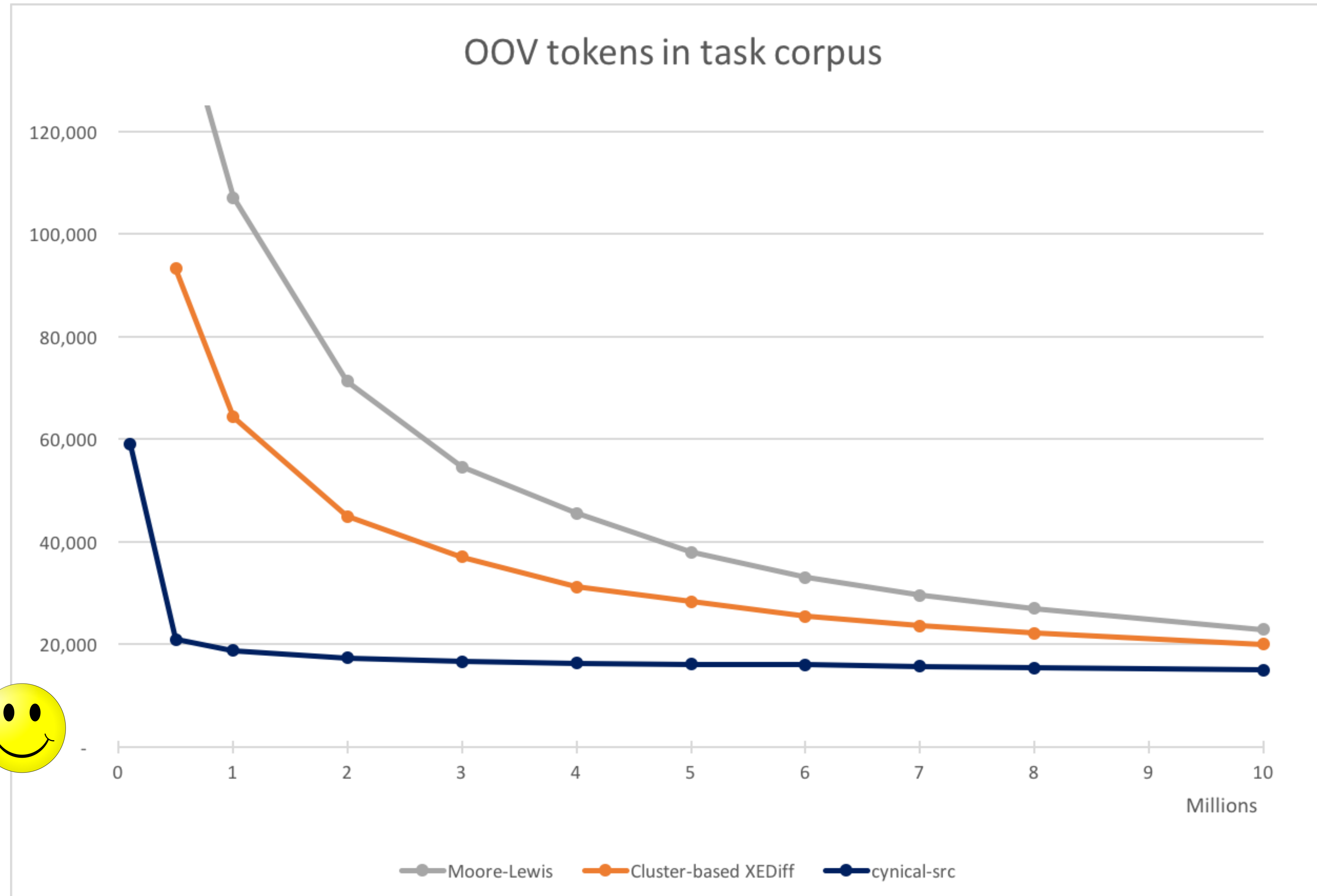
- cluster-based:
  -20 ppl



- cynical:
  -30 ppl
  …and -66% less data.

In-Domain Lexical Coverage

- cluster-based:
  -33% oov



- cynical:
  -85% oov
  …and -83% less data.

Summary

- Data selection:

  There's no data like useful data!

- If you use Moore-Lewis, upgrade to class-based.

  Always better, runs on tiny computers.

- Cynical gives same MT results,

  with much smaller systems

  and near-perfect coverage.

  Always better, runs on big computers.

Thanks!

Lucía Santamaría : lucsan@amazon.com

Cynical Selection available:

https://github.com/amittai/cynical

(totally open-source, MIT license, Amazon is not responsible for my bugs)

[this slide intentionally left blank]