A Shared Task on Bandit Learning for Machine Translation

Artem Sokolov[◇], Julia Kreutzer[*], Kellen Sunderland[◇], Pavel Danchenko[◇], Witold Szymaniak[◇], Hagen Fürstenau[◇], Stefan Riezler[*,‡]

[◇] amazon

[*] UNIVERSITÄT HEIDELBERG ZUKUNFT SEIT 1386

[‡] IWR Interdisciplinary Center for Scientific Computing

WMT17

MT supervision	cost	noise/signal
references	\$\$\$	low
post-edits/quasi-refs	\$\$	middle
feedback	\$	high

MT supervision	cost	noise/signal	downstream fit
references	$$$	low	hard
post-edits/quasi-refs	$$	middle	hard
feedback	$	high	easy/trivial

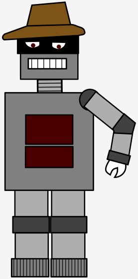
MT supervision	cost	noise/signal	downstream fit
references	\$\$\$	low	hard
post-edits/quasi-refs	\$\$	middle	hard
feedback	\$	high	easy/trivial

Sweet spot for bandit MT:

1. costs drop faster than noise/signal increases ⇒ lots of data
2. downstream performance is hard to wrap into an automatic metric

One-armed bandits – slot-machines:

- pull an arm to play
- get some reward (or none)
- try a new machine or stick to the best among discovered so far?



One-armed Bandit

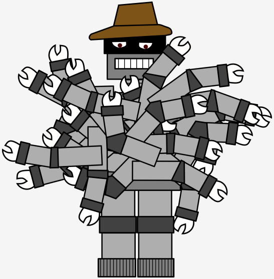
Multi-armed bandits:

- many arms (actions)
- each arm has an unknown reward distribution
- find an arm-picking strategy to maximize total reward

Multi-armed Bandit

Multi-armed bandits for structured prediction (MT):

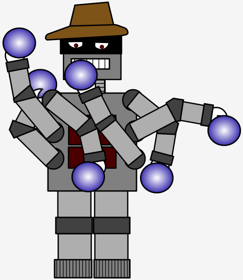
- observe context
  (source sentence)
- pick one out of exponentially many outputs
  (translations)
- each structure results in some reward
  (BLEU)
- tune an arm-picking strategy
  (decoder weights)



Exponential number of arms

Multi-armed bandits for structured prediction (MT):

- observe context
  (source sentence)
- pick one out of exponentially many outputs
  (translations)
- each structure results in some reward
  (BLEU)
- tune an arm-picking strategy
  (decoder weights)

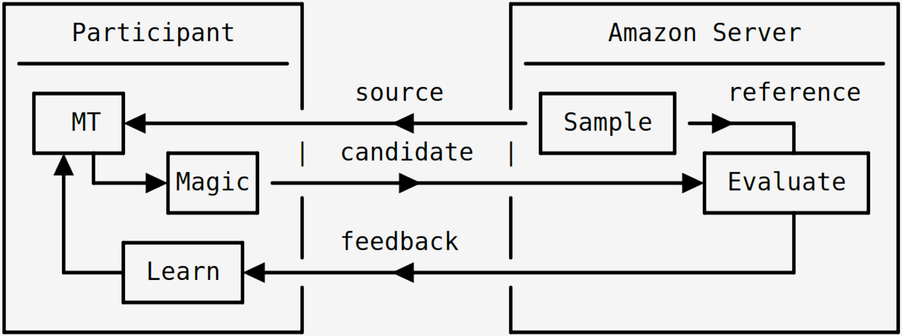


MT - Full info

Multi-armed bandits for structured prediction (MT):

- observe context
  (source sentence)
- pick one out of exponentially many outputs
  (translations)
- each structure results in some reward
  (BLEU)
- tune an arm-picking strategy
  (decoder weights)
- note: only one translation is scored,
  others are not

MT - Partial info



for $t = 0, \ldots, T$ **do**

Request source sentence x_t from service

Propose a translation y_t

Obtain feedback $\Delta(y_t)$ from service

Improve MT model

- DE-EN (pre-processed)
- domain-adaptation: general (WMT17) $\rightarrow$ e-commerce (Amazon)
- all participants received the same sequence of x_t
- feedback was sent-BLEU (plans for more realistic feedback dropped to avoid complicating the task)
- organizers provided the service, client SDK and baselines

- DE-EN (pre-processed)
- domain-adaptation: general (WMT17) $\rightarrow$ e-commerce (Amazon)
- all participants received the same sequence of x_t
- feedback was sent-BLEU (plans for more realistic feedback dropped to avoid complicating the task)
- organizers provided the service, client SDK and baselines

Participants had to do:

1. pick Python or Java
2. download a short client snippet
3. wrap it around an MT system

- DE-EN (pre-processed)
- domain-adaptation: general (WMT17) $\rightarrow$ e-commerce (Amazon)
- all participants received the same sequence of x_t
- feedback was sent-BLEU (plans for more realistic feedback dropped to avoid complicating the task)
- organizers provided the service, client SDK and baselines

Participants had to do:

1. pick Python or Java
2. download a short client snippet
3. wrap it around an MT system

phase	sentences	passes	purpose
mock (since 13 Mar)	40	unlimited	test client API
dev (since 5 Apr)	40k	unlimited	tune hyperparams
train (25 Apr - Jun 9)	1.3M	only one	final evaluation

- **cumulative reward**

$$\sum_{t=1}^{T} \Delta(y_t)$$

- **corpus-BLEU** at regular intervals on an embedded test set:
  - ➡ 700 sent. at 4 locations in 40k dev sent.
  - ➡ 4000 sent. at 12 locations in 1.3M train sent.
- **regret**

$$\frac{1}{T} \sum_{t=1}^{T} \Delta(y_t^*) - \Delta(y_t)$$

(average cumulative reward difference w.r.t. to an in-domain system)

- different domains $\Rightarrow$ high OOV rate
- learning from one-shot feedback
- real-world data:
  - ➡ typos/errors in sources
  - ➡ mixed direction of data
  - ➡ translators improved readability/corrected errors/deleted irrelevancies

- different domains $\Rightarrow$ high OOV rate
- learning from one-shot feedback
- real-world data:
  - ➡ typos/errors in sources
  - ➡ mixed direction of data
  - ➡ translators improved readability/corrected errors/deleted irrelevancies

source	**reference**
schwarz gr.xxl / xxxl	black , size xxl / xxxl
, 147 cm	147 cm
für starke , glänzende nägel	great for strengthen your nails and enhance shine
seemless verarbeitung	seamless processing
maschinenwaschbar bei 30 ° c	machine washable at 30 degrees .
32 unzen volumen	32-ounce capacity
material : 1050 denier nylon .	material : 1050d nylon .
für e-gitarre entworfen	designed for electric guitar

- 8 teams registered
- 4 used the dev service
- 2 started full training:

UNIVERSITÉ PARIS SUD Limsi **Guillaume Wisniewski**

- ➡ UCB-style selection from a pool of MT systems
- ➡ online linear regression to predict rewards
- ➡ additional exploration

UNIVERSITY OF MARYLAND 1856 **Amr Sharaf, Shi Feng, Khanh Nguyen, Kianté Brantley, Hal Daumé**

- ➡ domain-adaptation (Moore-Lewis)
- ➡ online non-linear regression for rewards
- ➡ policy gradient with adaptive control variate (Advantage Actor-Critic)

- NMT (word- and BPE-based; `NeuralMonkey`)
- SMT (dense features; `cdec`)

stochastic approximation of the **expected loss**

$\swarrow \qquad \searrow$

policy-gradient

$$\mathbb{E}_{\tilde{y} \sim p_w(y|x)}[\Delta(\tilde{y})]$$

for $t = 1, \ldots, T$ **do**
Observe x_t
Sample $\tilde{y}_t \sim p_{w_t}(y|x_t)$
Obtain $\Delta(\tilde{y}_t)$

$$w_{t+1} = w_t - \gamma \Delta(\tilde{y}_t) \nabla \log p_{w_t}(\tilde{y}_t|x_t)$$

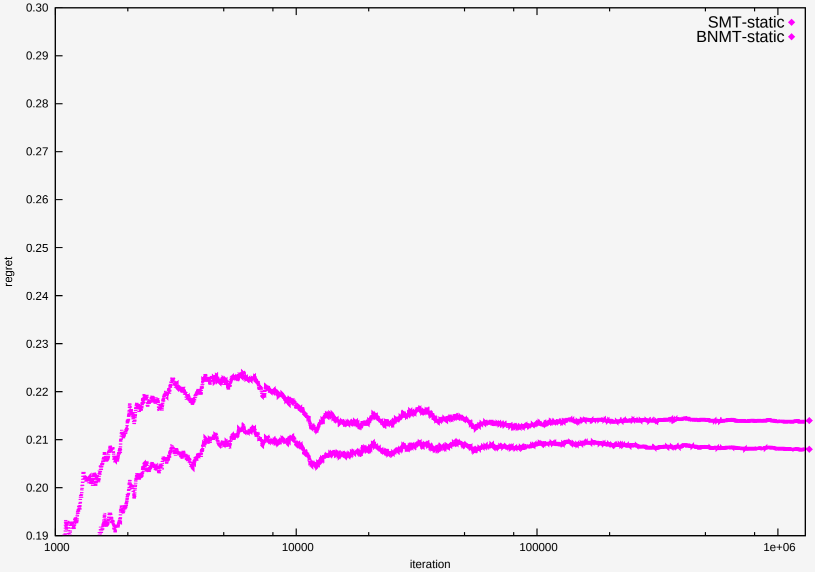
gradient-free

$$\mathbb{E}_{\varepsilon \sim \mathcal{N}(0,1)}[\Delta(\hat{y}(w + \varepsilon))]$$

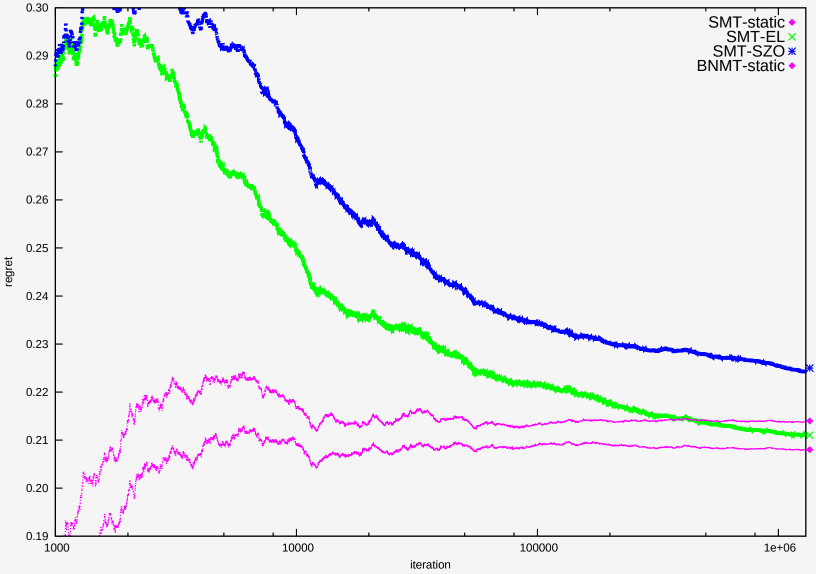
for $t = 1, \ldots, T$ **do**
Observe x_t
Sample $\varepsilon_t \sim \mathcal{N}(0, 1)$
Decode $\hat{y}_t$ with $w_t + \varepsilon_t$
Obtain $\Delta(\hat{y}_t)$

$$w_{t+1} = w_t + \gamma \Delta(\hat{y}_t) \varepsilon_t$$

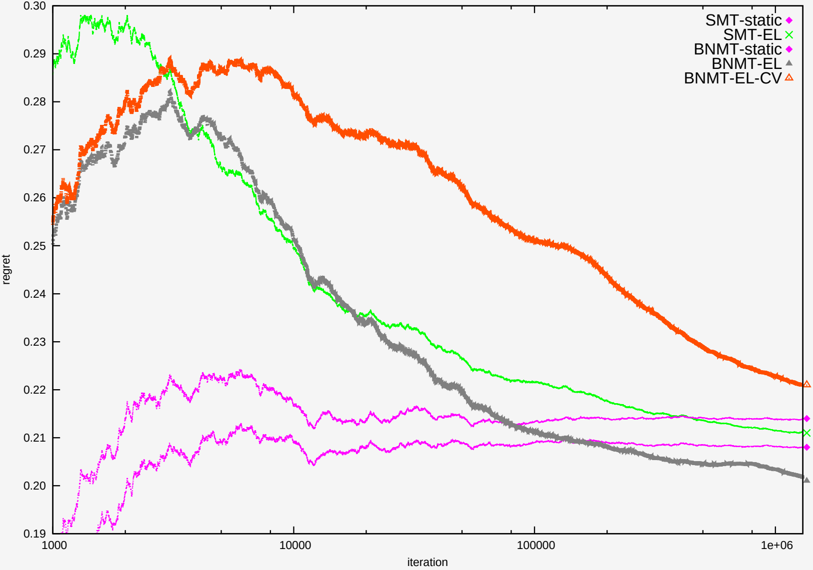
Regret


SMT-static
BNMT-static
regret
iteration
0.30
0.29
0.28
0.27
0.26
0.25
0.24
0.23
0.22
0.21
0.20
0.19
1000
10000
100000
1e+06

Regret

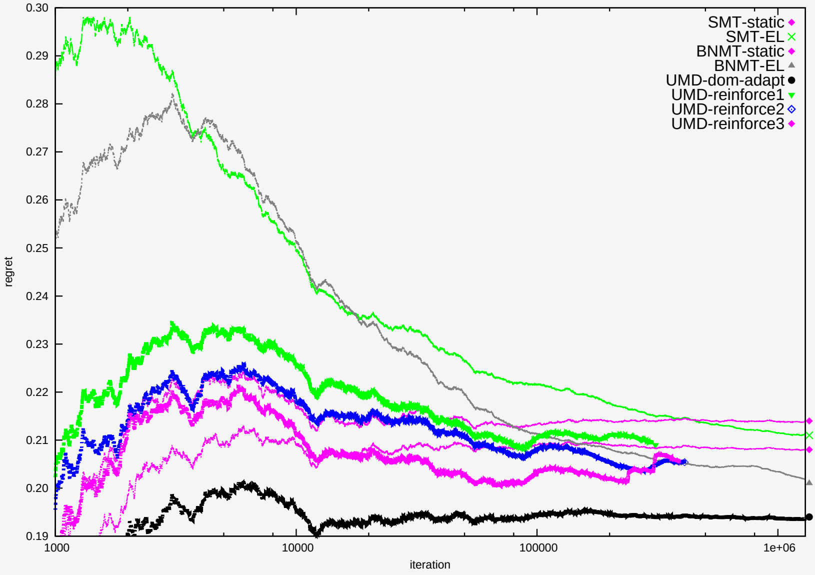


SMT-static
SMT-EL
SMT-SZO
BNMT-static
regret
iteration
0.30
0.29
0.28
0.27
0.26
0.25
0.24
0.23
0.22
0.21
0.20
0.19
1000
10000
100000
1e+06

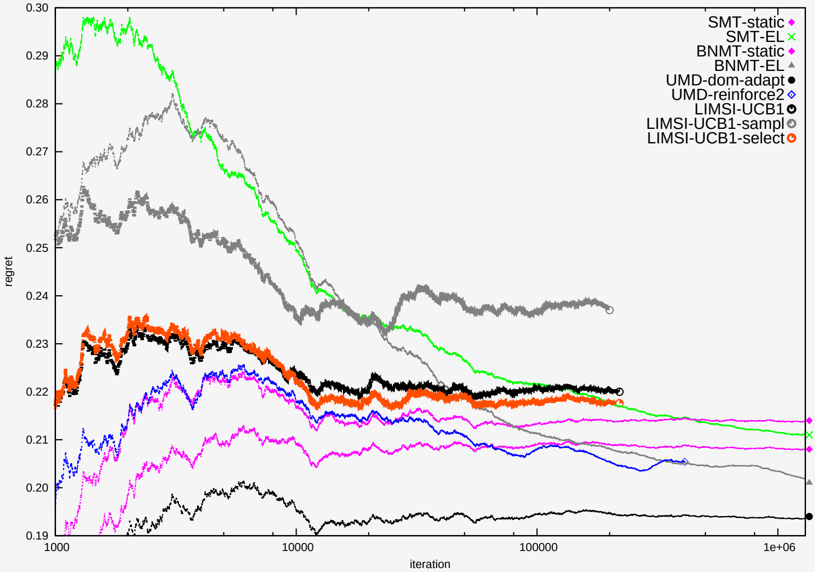


SMT-static
SMT-EL
BNMT-static
BNMT-EL
BNMT-EL-CV
regret
iteration
0.30
0.29
0.28
0.27
0.26
0.25
0.24
0.23
0.22
0.21
0.20
0.19
1000
10000
100000
1e+06

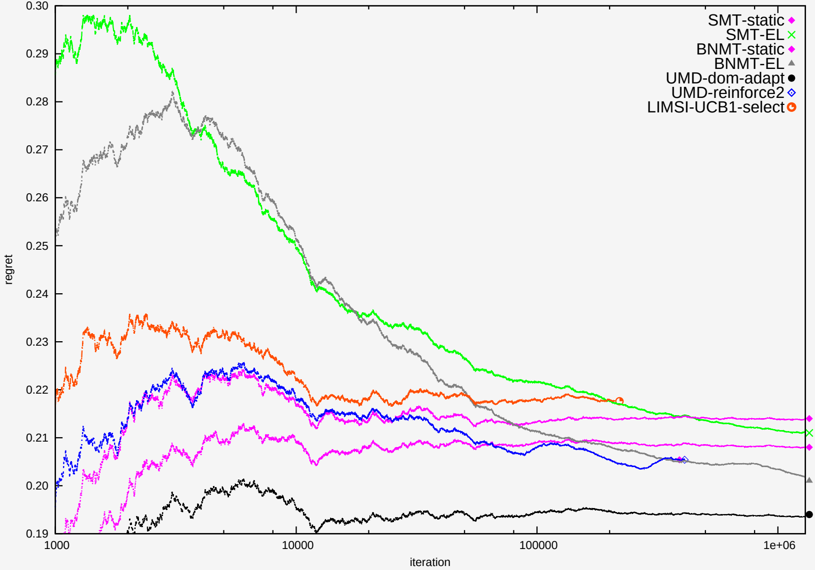
Regret



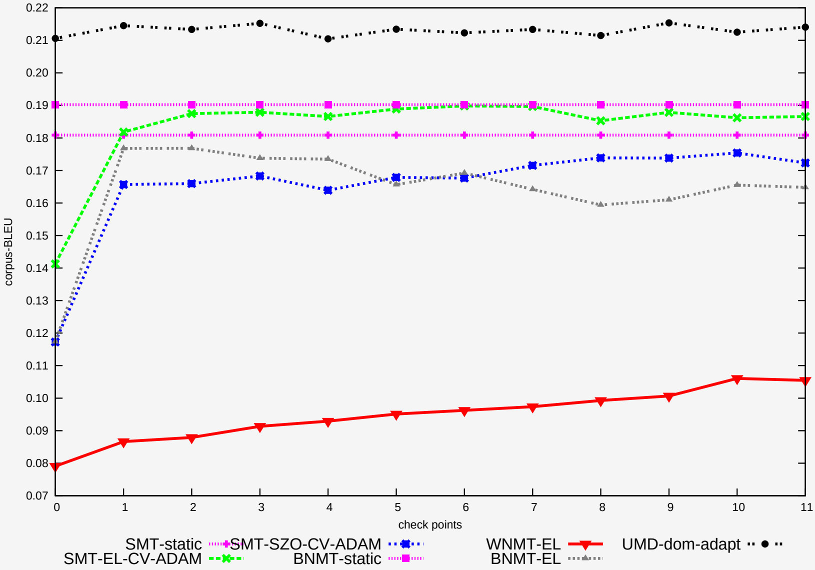
regret
iteration
0.30
0.29
0.28
0.27
0.26
0.25
0.24
0.23
0.22
0.21
0.20
0.19
1000
10000
100000
1e+06
SMT-static
SMT-EL
BNMT-static
BNMT-EL
UMD-dom-adapt
UMD-reinforce1
UMD-reinforce2
UMD-reinforce3

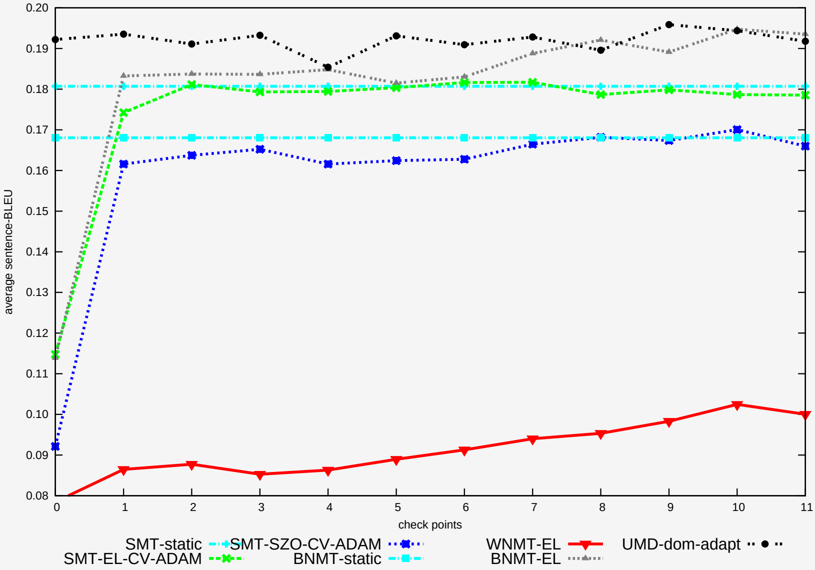


0.30
0.29
0.28
0.27
0.26
0.25
0.24
0.23
0.22
0.21
0.20
0.19
regret
1000
10000
100000
1e+06
iteration
SMT-static
SMT-EL
BNMT-static
BNMT-EL
UMD-dom-adapt
UMD-reinforce2
LIMSI-UCB1
LIMSI-UCB1-sampl
LIMSI-UCB1-select


SMT-static
SMT-EL
BNMT-static
BNMT-EL
UMD-dom-adapt
UMD-reinforce2
LIMSI-UCB1-select
regret
iteration
0.30
0.29
0.28
0.27
0.26
0.25
0.24
0.23
0.22
0.21
0.20
0.19
1000
10000
100000
1e+06

Checkpoints (corpus-BLEU)




0.20
0.19
0.18
0.17
0.16
0.15
0.14
0.13
0.12
0.11
0.10
0.09
0.08
average sentence-BLEU
0
1
2
3
4
5
6
7
8
9
10
11
check points
SMT-static
SMT-SZO-CV-ADAM
WNMT-EL
UMD-dom-adapt
SMT-EL-CV-ADAM
BNMT-static
BNMT-EL

Static system:

- ★ UMD domain adaptation got the best BLEU and lowest regret

Static system:

- ★ UMD domain adaptation got the best BLEU and lowest regret

Learning systems:

- cumulative reward:
  - ★ BNMT-EL is the only to beat its static NMT baseline
    - +lowest regret of all learning systems
    - +best sent-BLEU overall
  - ★ SMT-EL-CV came very close
- corpus-BLEU:
  - ★ SMT-EL-CV improves over its SMT baseline
  - ➡ none of the submissions show monotonic learning curves (or are too short)

Static system:

- ★ UMD domain adaptation got the best BLEU and lowest regret

Learning systems:

- cumulative reward:
  - ★ BNMT-EL is the only to beat its static NMT baseline
    - +lowest regret of all learning systems
    - +best sent-BLEU overall
  - ★ SMT-EL-CV came very close
- corpus-BLEU:
  - ★ SMT-EL-CV improves over its SMT baseline
  - ➡ none of the submissions show monotonic learning curves (or are too short)

Conclusion:

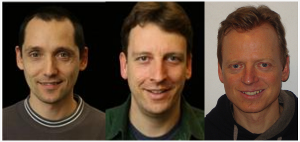
- difficult task (even with all simplifications)
- difficult data
- we need more research!:)

Acknowledgments

funding

task idea, NMT baselines

API, SDK,
leaderboard & operation

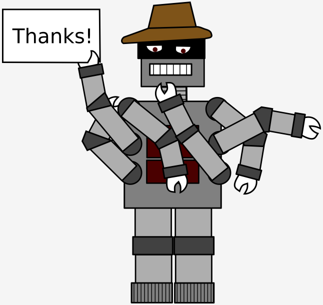


task design, data,
SMT baselines, etc.

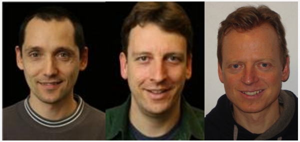
funding

amazon DFG

task idea, NMT baselines
and bandit pictures! →



API, SDK,
leaderboard & operation



task design, data,
SMT baselines, etc.