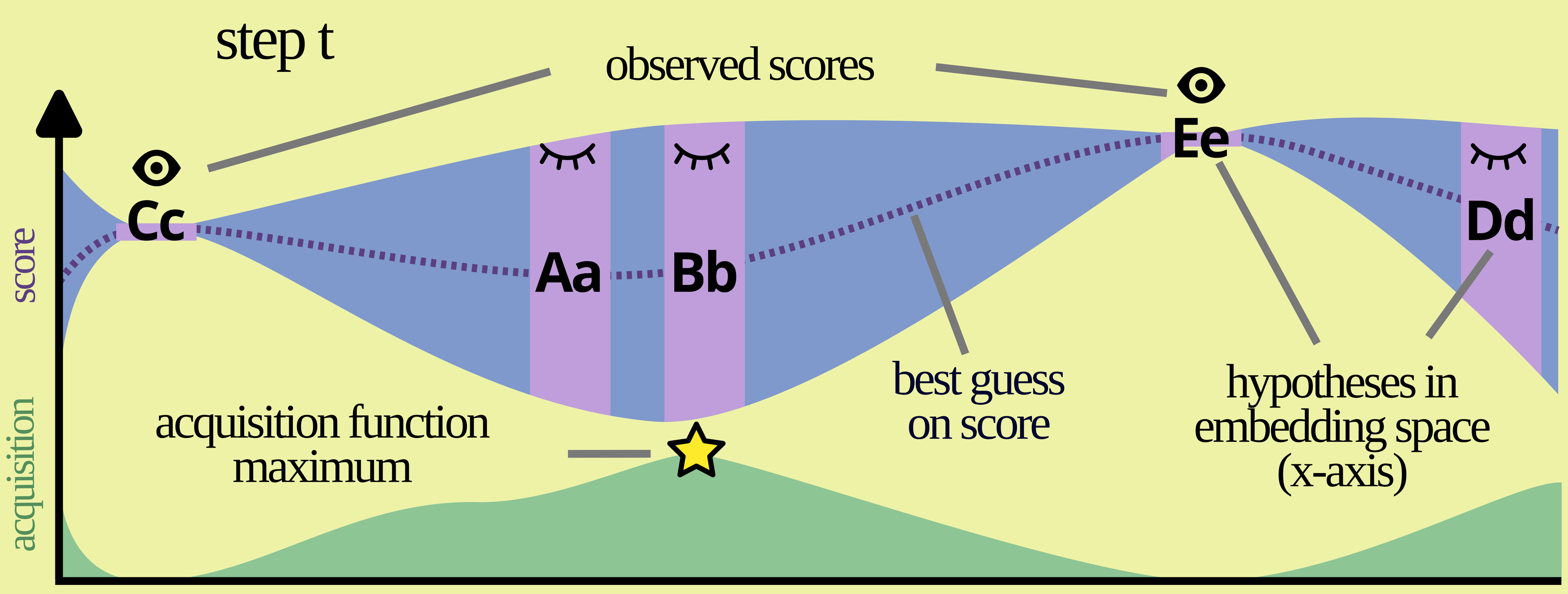
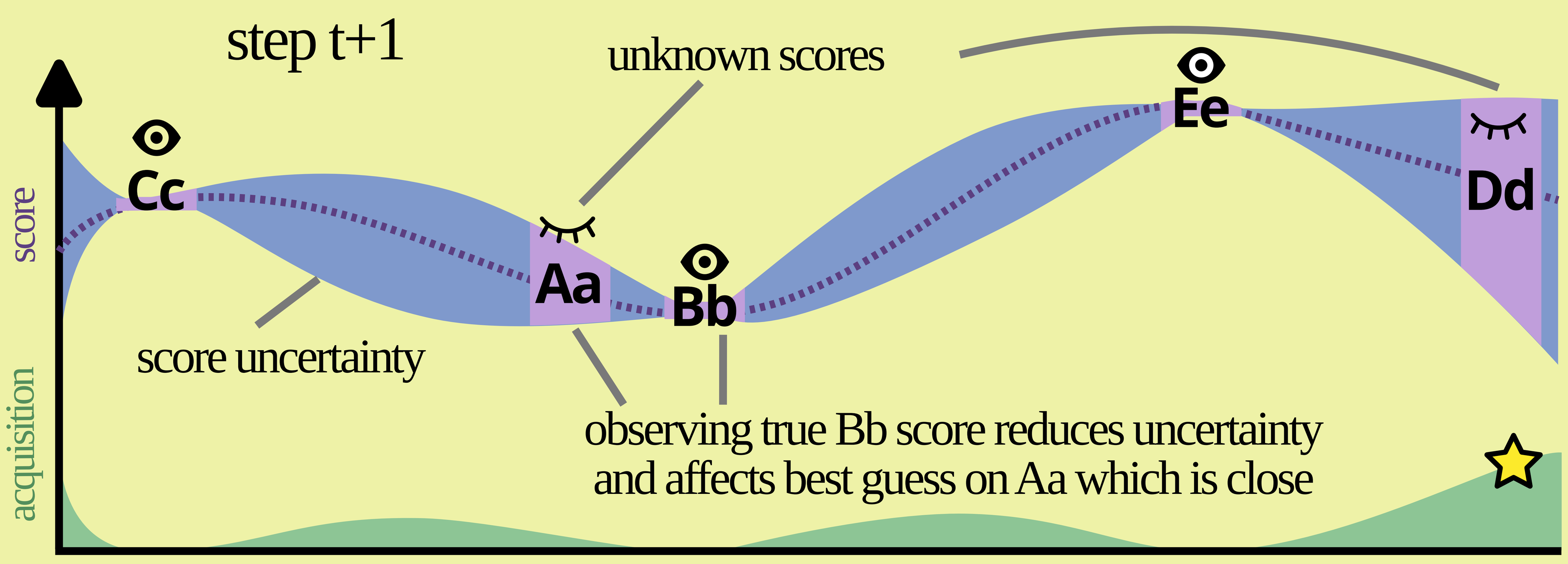
A Bayesian Optimization Approach to Machine Translation Reranking

Julius Cheng, Maike Züfle, Vilém Zouhar, Andreas Vlachos





Problem: MT produces 100 translations; how to find the best one?
Idea 0: Use MT logprobs to select the most probable one. ⇝ low quality
Idea 1: Use quality estimation metric. ⇝ expensive
Idea 2: Don't evaluate all candidates. Estimate unknown scores with existing scores or cheap proxy metrics.
Idea 3: Search efficiently with Gaussian Process and BayesOpt.

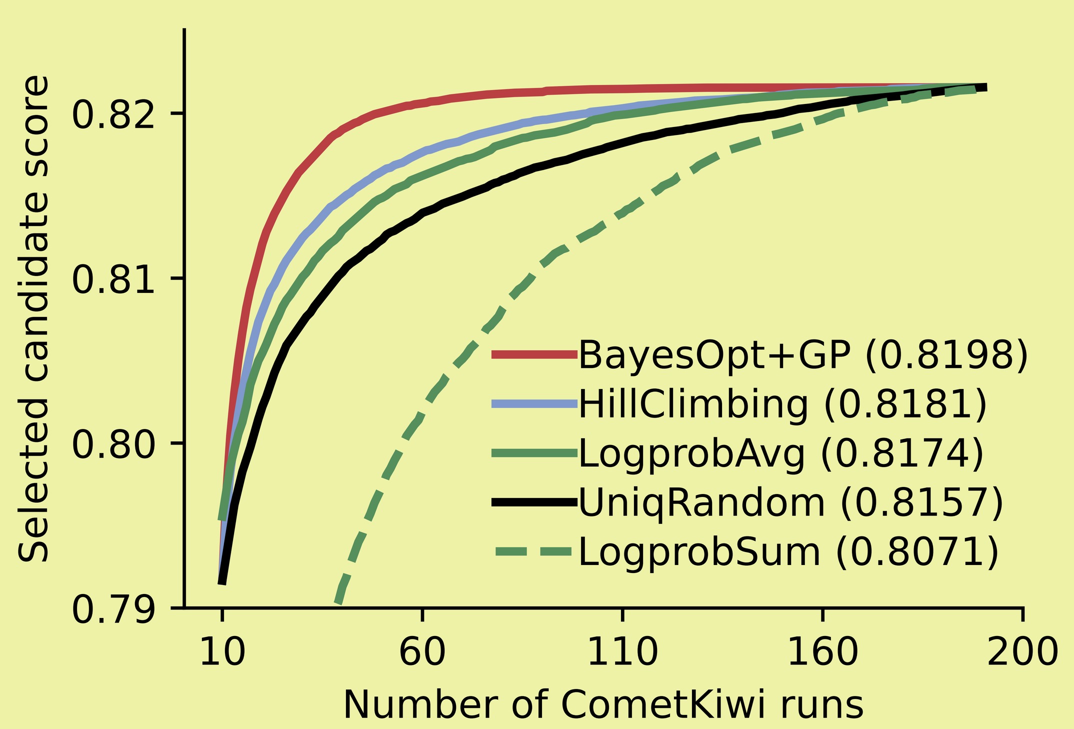
while $|C_{\text{seen}}| < \text{Budget}$:
$\quad y_{\text{next}} = \arg\max_{y \in C - C_{\text{seen}}} \text{ExpectedImprovement}(y, C_{\text{seen}})$
$\quad C_{\text{seen}} = C_{\text{seen}} \cup \{y_{\text{next}}\}$ **and compute** $\text{score}(y_{\text{next}})$
return $\arg\max_{y \in C_{\text{seen}}} \text{score}(y)$

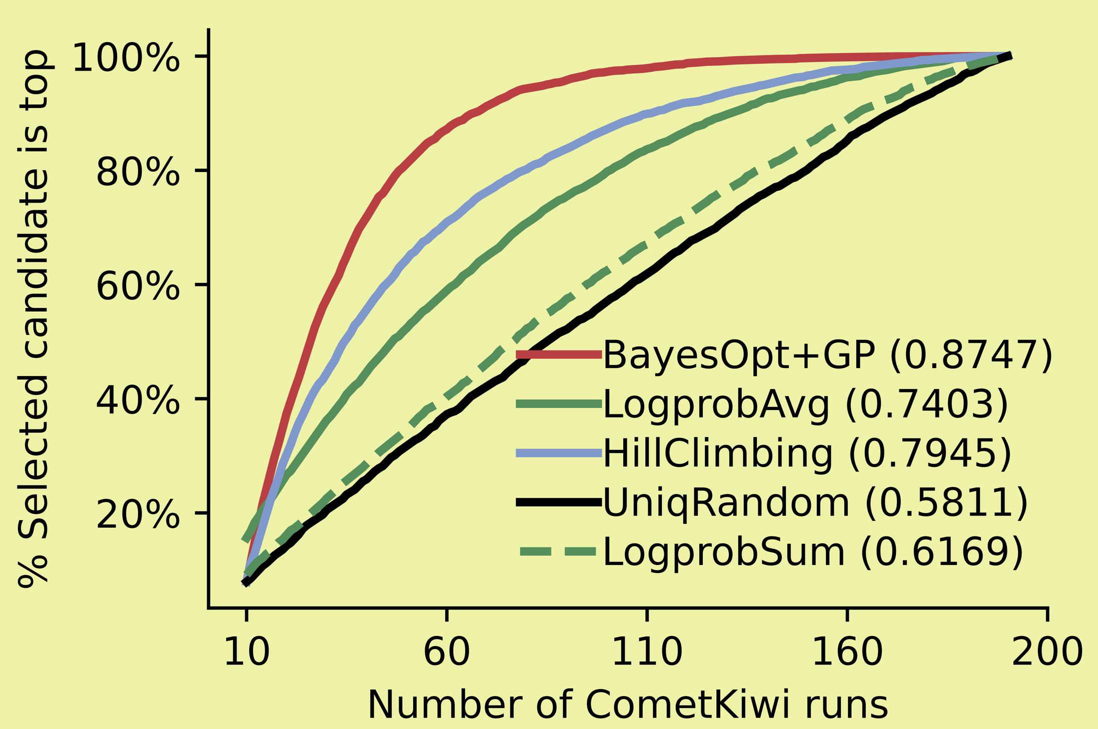
Iteratively select the most promising candidate to score. Return the best one seen so far.

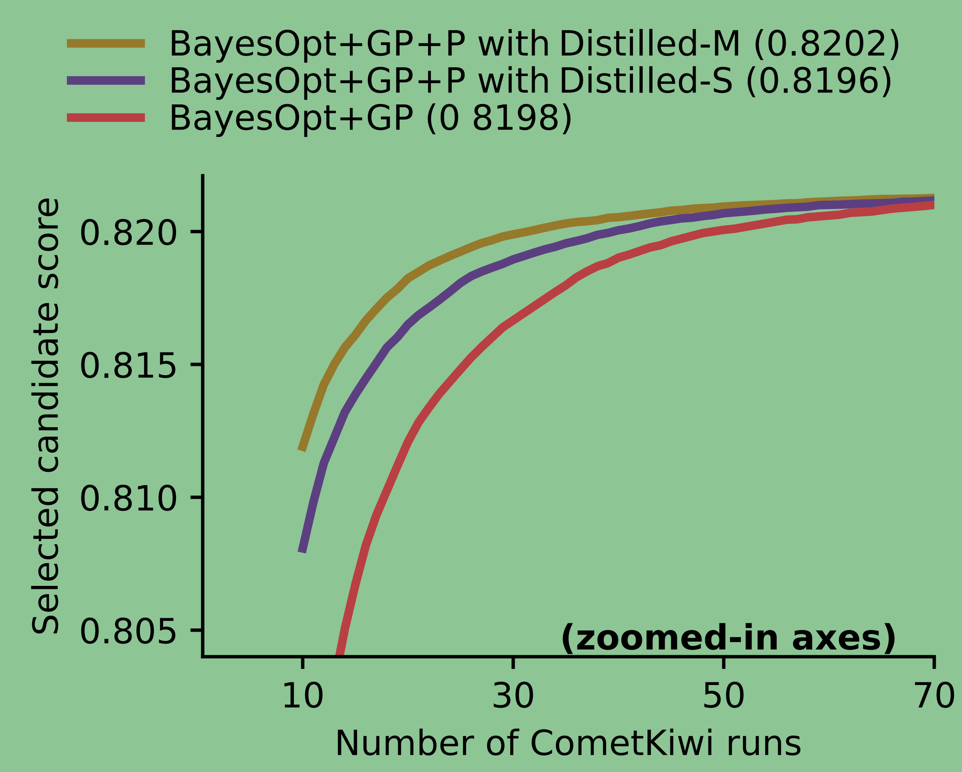
Compute expected improvement based on scores revealed so far for other candidates and their distance to y

$\mathcal{K}(y, y') = \exp\left(-\frac{\|\text{embd}(y) - \text{embd}(y')\|^2}{2w^2}\right)$
def $\text{ExpectedImprovement}(y, C_{\text{seen}})$:
$\quad y_{\text{best}} = \arg\max_{y \in C_{\text{seen}}} \text{score}(y)$
$\quad \sigma(y), \mu(y) = \text{GaussianProcess}(\mathcal{K}, y, C_{\text{seen}})$
$\quad z = (\text{score}(y_{\text{best}}) - \mu(y))/\sigma(y)$
return $\sigma(y)(z \cdot \text{cdf}(z) + \text{pdf}(z))$

Find near-best candidates with much fewer scorings







Baselines:
UniqRandom: score B random candidates & select max.
Logprob: score top-B candidates based on logprob.
HillClimbing: iteratively score closest to current best.
BayesOpt+GP: our algorithm

Upgrade:
Bootstrap selection with cheaper but less accurate scorers (small COMET quality estimation).