# Interactive Text Ranking with Bayesian Optimization: A Case Study on Community QA and Summarization

**Edwin Simpson**[1,2] **Yang Gao**[1,3] **Iryna Gurevych**[1]

[1]Ubiquitous Knowledge Processing Lab, Technische Universitaẗ Darmstadt,
https://www.informatik.tu-darmstadt.de/
[2]Dept. of Computer Science, University of Bristol, edwin.simpson@bristol.ac.uk
[3]Dept. of Computer Science, Royal Holloway, University of London, yang.gao@rhul.ac.uk

## Abstract

For many NLP applications, such as question answering and summarization, the goal is to select the best solution from a large space of candidates to meet a particular user's needs. To address the lack of user or task-specific training data, we propose an interactive text ranking approach that actively selects pairs of candidates, from which the user selects the best. Unlike previous strategies, which attempt to learn a ranking across the whole candidate space, our method uses Bayesian optimization to focus the user's labeling effort on high quality candidates and integrate prior knowledge to cope better with small data scenarios. We apply our method to community question answering (cQA) and extractive multidocument summarization, finding that it significantly outperforms existing interactive approaches. We also show that the ranking function learned by our method is an effective reward function for reinforcement learning, which improves the state of the art for interactive summarization.

## 1 Introduction

Many text ranking tasks are highly specific to an individual user's topic of interest, which presents a challenge for NLP systems that have not been trained to solve that user's problem. Consider ranking summaries or answers to non-factoid questions: A good solution requires understanding the topic and the user's information needs (Liu and Agichtein, 2008; López et al., 1999). We address this by proposing an interactive text ranking approach that efficiently gathers user feedback and combines it with predictions from pretrained, generic models.

To minimize the amount of effort the user must expend to train a ranker, we learn from pairwise preference labels, in which the user compares two candidates and labels the best one. Pairwise preference labels can often be provided faster than ratings or class labels (Yang and Chen, 2011; Kingsley and Brown, 2010; Kendall, 1948), can be used to rank candidates using learning-to-rank (Joachims, 2002), preference learning (Thurstone, 1927), or best–worst scaling (Flynn and Marley, 2014), or to train a *reinforcement learning (RL)* agent to find the optimal solution (Wirth et al., 2017).

To reduce the number of labels a user must provide, a common solution is *active learning (AL)*. AL learns a model by iteratively acquiring labels: At each iteration, it trains a model on the labels collected so far, then uses an *acquisition function* to quantify the value of querying the user about a particular pair of candidates. The system then chooses the pairs with the highest values, and instructs the user to label them. The acquisition function implements one of many different strategies to minimize the number of interaction rounds, such as reducing *uncertainty* (Settles, 2012) by choosing informative labels that help learn the model more quickly.

Many active learning strategies, such as the pairwise preference learning method of Gao et al. (2018), aim to learn a good ranking model for all candidates, for example, by querying the annotator about candidates whose rank is most uncertain. However, we often need to find and rank only a small set of *good* candidates to present to the user. For instance, in question answering, irrelevant answers should not be shown to the user, so their precise ordering is unimportant and users should not waste time ranking them. Therefore, by reducing

Transactions of the Association for Computational Linguistics, vol. 8, pp. 759–775, 2020. https://doi.org/10.1162/tacl_a_00344
Action Editor: Preslav Nakov. Submission batch: 2/2020; Revision batch: 7/2020; Published 12/2020.

uncertainty for all candidates, uncertainty-based AL strategies may waste labels on sorting poor candidates.

Here, we propose an interactive method for ranking texts that replaces the standard uncertainty-based acquisition functions with acquisition functions for *Bayesian optimization (BO)* (Močkus, 1975; Brochu et al., 2010). In general, BO aims to find the maximum of a function while minimizing the number of queries to an oracle. Here, we use BO to maximize a ranking function that maps text documents to scores, treating the user as a noisy oracle. Our BO active learning strategy minimizes the number of labels needed to find the best candidate, in contrast to uncertainty-based strategies that attempt to learn the entire ranking function. This makes BO better suited to tasks such as question answering, summarization, or translation, where the aim is to find the best candidate and those with low quality can simply be disregarded rather than ranked precisely. In this paper, we define two BO acquisition functions for interactive text ranking.

While our approach is designed to adapt a model to a highly specialized task, generic models can provide hints to help us avoid low-quality candidates. Therefore, we learn the ranking function itself using a Bayesian approach, which integrates prior predictions from a generic model that is not tailored to the user. Previous interactive text ranking methods either do not exploit prior information (Baldridge and Osborne, 2004; P.V.S and Meyer, 2017; Lin and Parikh, 2017; Siddhant and Lipton, 2018), combine heuristics with user feedback after active learning is complete (Gao et al., 2018), or require expensive re-training of a non-Bayesian method (Peris and Casacuberta, 2018). Here, we show how BO can use prior information to expedite interactive text ranking. The interactive learning process is shown in Algorithm 1 and examples of our system outputs are shown in Figures 1 and 2.

```
Input: candidate texts $\boldsymbol{x}$ with feature vectors $\phi(\boldsymbol{x})$
Initialize ranking model $m$;
Set the training data $\boldsymbol{D} = \varnothing$;
while $|\boldsymbol{D}| < max\_interactions$ do
    For each pair of texts $(x_a, x_b)$ in $\boldsymbol{x}$, compute $v = acquisition(\phi(x_a), \phi(x_b), \boldsymbol{D}, m)$;
    Set $\boldsymbol{P}_i$ to be the set of $batch\_size$ pairs with the highest values of $v$;
    Obtain labels $\boldsymbol{y}_i$ from user for pairs $\boldsymbol{P}_i$ ;
    Add $\boldsymbol{y}_i$ and $\boldsymbol{P}_i$ to $\boldsymbol{D}$ ;
    Train model $m$ on the training set $\boldsymbol{D}$ ;
end
Output: Return the trained model $m$ and/or its final ranked list of candidate texts in $\boldsymbol{x}$.
```

**Algorithm 1:** Interactive text ranking process with preference learning.

**Q:** *Does whiskey go bad by freezing?*
**A1:** It is to cool it down without diluting it–ice cubes would melt. And yes, you could simply cool the entire bottle, but it wouldn't look that fancy. Note that some purists would wrinkle their noses and insist that whisky is best enjoyed at room temperature and perhaps with a small dash of spring water. And I'm soooo not going into a whisky vs. whiskey debate here.
**A2: Putting strong spirits in the freezer should not harm them.** The solubility of air gases increases at low temperature, which is why you see bubbles as it warms up. **Drinks with a lower alcohol content will be affected in the freezer.** There is potential to freeze water out of anything with an alcohol content of 28% or lower. Many people use the freezer to increase the alcohol content of their home brew in UK, by freezing water out of it–the alcohol stays in the liquid portion.

Figure 1: Example from the Stack Exchange Cooking topic. Candidate answer A1 selected without user interaction by COALA (Rücklé et al., 2019); A2 chosen by our system (GPPL with IMP) after 10 user interaction. A2 answers the question (boldfaced texts) but A1 fails.

Our contributions are (1) a Bayesian optimization methodology for interactive text ranking that integrates prior predictions with user feedback, (2) acquisition functions for Bayesian optimization with pairwise labels, and (3) empirical evaluations on community question answering (cQA) and extractive multidocument summarization, which show that our method brings substantial improvements in ranking and summarization performance (e.g., for cQA, an average 25% increase in answer selection accuracy over the next-best method with 10 rounds of user interaction). We release the complete experimental software for future work.[1]

[1] `https://github.com/UKPLab/tacl2020-interactive-ranking`.

**(a):** A third leading advocate of the China Democracy Party who has been in custody for a month, Wang Youcai, was accused of “inciting the overthrow of the government,” the Hong Kong-based Information Center of Human Rights and Democratic Movement in China reported. China’s central government ordered the arrest of a prominent democracy campaigner and may use his contacts with exiled Chinese dissidents to charge him with harming national security, a colleague said Wednesday. One leader of a suppressed new political party will be tried on Dec. 17 on a charge of colluding with foreign enemies of China “to incite the subversion of state power,” according to court documents given to his wife on Monday.

**(b):** The arrests of Xu and Qin at their homes Monday night and the accusations against them and Wang were the sharpest action Chinese leaders have taken since dissidents began pushing to set up and legally register the China Democracy Party in June. Hours before China was expected to sign a key U.N. human rights treaty and host British Prime Minister Tony Blair, police hauled a prominent human rights campaigner in for questioning Monday. With attorneys locked up, harassed or plain scared, two prominent dissidents will defend themselves against charges of subversion Thursday in China’s highest-profile dissident trials in two years. Wang was a student leader in the 1989 Tiananmen Square democracy demonstrations.

**(c):** On the eve of China’s signing the International Covenant of Civil and Political Rights (ICCPR) in October 1998, police detained Chinese human rights advocate Qin Yongmin for questioning. Eight weeks after signing the ICCPR, Chinese police arrested Qin and an associate in the China Democracy Party (CDP), Xu Wenli, without stating charges. Another CDP leader already in custody, Wang Youcai, was accused of "inciting the overthrow of the government". Qin and Wang went to trial in December for inciting subversion. Police pressure on potential defense attorneys forced the accused to mount their own defenses. Xu Wenli had not yet been charged.

Figure 2: Example summaries for DUC’04 produced by RL (see Section 5.4) with a reward function learnt from 100 user interactions using (a) the BT, UNC method of Gao et al. (2018) and (b) our GPPL, IMP method. (c) is a model summary written by an expert. Each color indicates a particular news event or topic, showing where it occurs in each summary. Compared to (a), summary (b) covers more of the events discussed in the reference, (c).

## 2 Related Work

**Interactive Learning in NLP.** Previous work has applied active learning to tasks involving ranking or optimising generated text, including summarization (P.V.S and Meyer, 2017), visual question answering (Lin and Parikh, 2017), and translation (Peris and Casacuberta, 2018). For summarization, Sokolov et al. (2016), Lawrence and Riezler (2018) and Singh et al. (2019), train reinforcement learners by querying the user directly for rewards, which requires in the order of $10^5$ interactions. Gao et al. (2018) dramatically reduce the number of user interactions to the order of to $10^2$ by using active learning to learn a reward function for RL, an approach proposed by Lopes et al. (2009). These previous works use *uncertainty sampling* strategies, which query the user about the candidates with the most uncertain rankings to try to learn all candidates’ rankings with a high degree of confidence. We instead propose to find good candidates using an optimization strategy. Siddhant and Lipton (2018) carried out a large empirical study of uncertainty sampling for sentence classification, semantic role labeling and named entity recognition, finding that exploiting model uncertainty estimates provided by Bayesian neural networks improved performance. Our approach also exploits Bayesian uncertainty estimates.

**BO for Preference Learning.** Bayesian approaches using Gaussian processes (GPs) have previously been used to reduce errors in NLP tasks involving sparse or noisy labels (Cohn and Specia, 2013; Beck et al., 2014), making them well-suited to learning from user feedback. *Gaussian process preference learning (GPPL)* (Chu and Ghahramani 2005) enables GP inference with pairwise preference labels. Simpson and Gurevych (2018) introduced scalable inference for GPPL using stochastic variational inference (SVI) (Hoffman et al., 2013), which outperformed SVM and LSTM methods at ranking arguments by convincingness. They included a study on active learning with pairwise labels, but tested GPPL only with uncertainty sampling, not BO. Here, we adapt GPPL to summarization and cQA, show how to integrate prior predictions, and propose a BO framework for GPPL that facilitates interactive text ranking.

Brochu et al. (2008) proposed a BO approach for pairwise comparisons but applied the approach only to a material design use case with a very simple feature space. González et al. (2017) proposed alternative BO strategies for pairwise preferences, but their approach requires expensive sampling to estimate the utilities, which is too slow for an interactive setting. Yang and Klabjan (2018) also propose BO with pairwise preferences, but again, inference is expensive, the method is only tested with fewer than ten features, and it uses an inferior *probability of improvement* strategy (see Snoek et al., 2012). Our GPPL-based framework permits much faster inference even when the input vector has more than 200 features, and hence allows rapid selection of new pairs when querying users.

Ruder and Plank (2017) use BO to select training data for transfer learning in NLP tasks such as sentiment analysis, POS tagging, and parsing. However, unlike our interactive text ranking approach, their work does not involve pairwise comparisons and is not interactive, as the optimizer learns by training and evaluating a model on the selected data. In summary, previous work has not yet devised BO strategies for GPPL or suitable alternatives for interactive text ranking.

## 3 Background on Preference Learning

Popular preference learning models assume that users choose a candidate from a pair with probability $p$, where $p$ is a function of the candidates' *utilities* (Thurstone, 1927). Utility is defined as the value of a candidate to the user, that is, it quantifies how well that instance meets their needs. When candidates have similar utilities, the user's choice is close to random, while pairs with very different utilities are labeled consistently. Such models include the Bradley–Terry model (BT) (Bradley and Terry, 1952; Luce, 1959; Plackett, 1975), and the Thurstone–Mosteller model (Thurstone, 1927; Mosteller, 1951).

BT defines the probability that candidate $a$ is preferred to candidate $b$ as follows:

$$p(y_{a,b}) = \left(1+\exp\left(\boldsymbol{w}^T\phi(a)-\boldsymbol{w}^T\phi(b)\right)\right)^{-1} \quad (1)$$

where $y_{a,b} = a \succ b$ is a binary preference label, $\phi(a)$ is the feature vector of $a$ and $\boldsymbol{w}^T$ is a weight parameter that must be learned. To learn the weights, we treat each pairwise label as two data points: The first point has input features $\boldsymbol{x} = \phi(a) - \phi(b)$ and label $y$, and the second point is the reverse pair, with $\boldsymbol{x} = \phi(b) - \phi(a)$ and label $1-y$. Then, we use standard techniques for logistic regression to find the weights $\boldsymbol{w}$ that minimize the L2-regularized cross entropy loss. The resulting linear model can be used to predict labels for any unseen pairs, or to estimate candidate utilities, $f_a = w^T\phi(a)$, which can be used for ranking.

**Uncertainty (UNC).** At each active learning iteration, the learner requests training labels for candidates that maximize the acquisition function. P.V.S and Meyer (2017) proposed an *uncertainty sampling* acquisition function for interactive document summarization, which defines the uncertainty about a single candidate's utility, $u$, as follows:

$$u(a|\boldsymbol{D}) = \begin{cases} p(a|\boldsymbol{D}) & \text{if } p(a|\boldsymbol{D}) \leq 0.5 \\ 1-p(a|\boldsymbol{D}) & \text{if } p(a|\boldsymbol{D}) > 0.5, \end{cases} \quad (2)$$

where $p(a|\boldsymbol{D}) = (1 + \exp(-f_a))^{-1}$ is the probability that $a$ is a good candidate and $\boldsymbol{w}$ is the set of BT model weights trained on the data collected so far, $\boldsymbol{D}$, which consists of pairs of candidate texts and pairwise preference labels. For pairwise labels, Gao et al. (2018) define an acquisition function, which we refer to here as *UNC*, which selects the pair of candidates $(a, b)$ with the two highest values of $u(a|\boldsymbol{D})$ and $u(b|\boldsymbol{D})$.

UNC is intended to focus labeling effort on candidates whose utilities are uncertain. If the learner is uncertain about whether candidate $a$ is a good candidate, $p(a|\boldsymbol{D})$ will be close to 0.5, so $a$ will have a higher chance of being selected. Unfortunately, it is also possible for $p(a|\boldsymbol{D})$ to be close to 0.5 even if $a$ has been labeled many times if $a$ is a candidate of intermediate utility. Therefore, when using UNC, labeling effort may be wasted re-labeling mediocre candidates.

The problem is that BT cannot distinguish two types of uncertainty. The first is *aleatoric* uncertainty due to the inherent unpredictability of the phenomenon we wish to model (Senge et al., 2014). For example, when predicting the outcome of a coin toss, we model the outcome as random. Similarly, given two equally preferable items, we assume that the user assigns a preference label randomly. It does not matter how much training data we observe: if the items are equally good, we are uncertain which one the user will choose.

The second type is *epistemic* uncertainty due to our lack of knowledge, which can be reduced by acquiring more training data, as this helps us to learn the model's parameters with higher confidence. BT does not quantify aleatoric and epistemic uncertainty separately, unlike Bayesian approaches (Jaynes, 2003), so we may repeatedly select items with similar utilities that do not require more labels. To rectify this shortcoming, we replace BT with a Bayesian model that both estimates the utility of a candidate and quantifies the epistemic uncertainty in that estimate.

| Learner | BT | BT | GPPL | GPPL | GPPL | GPPL | GPPL |
|---|---|---|---|---|---|---|---|
| Strategy | random | UNC | random | UNPA | EIG | IMP | TP |
| Considers epistemic uncertainty | N | Y | N | Y | Y | Y | Y |
| Ignores aleatoric uncertainty | N | N | N | N | Y | Y | Y |
| Supports warm-start | N | N | Y | Y | Y | Y | Y |
| Focus on finding best candidate | N | N | N | N | N | Y (greedy) | Y (balanced) |

Table 1: Characteristics of active preference learning strategies. TP balances finding best candidate with exploration.

**Gaussian Process Preference Learning** Because BT does not quantify epistemic uncertainty in the utilities, we turn to a Bayesian approach, GPPL. GPPL uses a GP to provide a nonlinear mapping from document feature vectors to utilities, and assumes a Thurstone–Mosteller model for the pairwise preference labels. Whereas BT simply estimates a scalar value of $f_a$ for each candidate, $a$, GPPL outputs a posterior distribution over the utilities, $\boldsymbol{f}$, of all candidate texts, $\boldsymbol{x}$:

$$p(\boldsymbol{f}|\boldsymbol{\phi}(\boldsymbol{x}), \boldsymbol{D}) = \mathcal{N}(\hat{\boldsymbol{f}}, \boldsymbol{C}), \tag{3}$$

where $\hat{\boldsymbol{f}}$ is a vector of posterior mean utilities and $\boldsymbol{C}$ is the posterior covariance matrix of the utilities. The entries of $\hat{\boldsymbol{f}}$ are predictions of $f_a$ for each candidate given $\boldsymbol{D}$, and the diagonal entries of $\boldsymbol{C}$ represent posterior variance, which can be used to quantify uncertainty in the predictions. Thus, GPPL provides a way to separate candidates with uncertain utility from those with middling utility but many pairwise labels. In this paper, we infer the posterior distribution over the utilities using the scalable SVI method detailed by Simpson and Gurevych (2020).

## 4 Interactive Learning with GPPL

We now define four acquisition functions for GPPL that take advantage of the posterior covariance, $\boldsymbol{C}$, to account for uncertainty in the utilities. Table 1 summarises these acquisition functions.

**Pairwise Uncertainty (UNPA).** Here we propose a new adaptation of uncertainty sampling to pairwise labeling with the GPPL model. Rather than evaluating each candidate individually, as in UNC, we select the pair whose label is most uncertain. UNPA selects the pair with label probability $p(y_{a,b})$ closest to 0.5, where, for GPPL:

$$p(y_{a,b}) = \Phi\left(\frac{\hat{f}_a - \hat{f}_b}{\sqrt{1+v}}\right), \tag{4}$$

$$v = \boldsymbol{C}_{a,a} + \boldsymbol{C}_{b,b} - 2\boldsymbol{C}_{a,b}, \tag{5}$$

where $\Phi$ is the probit likelihood and $\hat{f}_a$ is the posterior mean utility for candidate $a$. Through $\boldsymbol{C}$, this function accounts for correlations between candidates' utilities and epistemic uncertainty in the utilities. However, for two items with similar expected utilities, $\hat{f}_a$ and $\hat{f}_b$, the $p(y_{a,b})$ is close to 0.5, that is, it has high aleatoric uncertainty. Therefore, whereas UNPA will favor candidates with uncertain utilities, it may still waste labeling effort on pairs with similar utilities but low uncertainty.

**Expected Information Gain (EIG).** We now define a second acquisition function for active learning with GPPL, which has previously been adapted to GPPL by Houlsby et al. (2011) from an earlier information-theoretic strategy (MacKay, 1992). EIG greedily reduces the epistemic uncertainty in the GPPL model by choosing pairs that maximize *information gain*, which quantifies the information a pairwise label provides about the utilities, $\boldsymbol{f}$. Unlike UNPA, this function avoids pairs that only have high aleatoric uncertainty. The information gain for a pairwise label, $y_{a,b}$, is the reduction in entropy of the distribution over the utilities, $\boldsymbol{f}$, given $y_{a,b}$. Houlsby et al. (2011) note that this can be more easily computed if it is reversed using a method known as Bayesian active learning by disagreement (BALD), which computes the reduction in entropy of the label's distribution given $\boldsymbol{f}$. Because we do not know the value of $\boldsymbol{f}$, we take the *expected* information gain $\boldsymbol{I}$ with respect to $\boldsymbol{f}$:

$$\mathrm{I}(y_{a,b}, \boldsymbol{f}; \boldsymbol{D}) = \mathrm{H}(y_{a,b}|\boldsymbol{D}) - \mathbb{E}_{\boldsymbol{f}|\boldsymbol{D}}[\mathrm{H}(y_{a,b}|\boldsymbol{f})], \tag{6}$$

where H is Shannon entropy. Unlike the related *pure exploration* strategy (González et al., 2017), Equation 6 can be computed in closed form, so does not need expensive sampling.

**Expected Improvement (IMP).** The previous acquisition functions for AL are uncertainty-based, and spread labeling effort across all items whose utilities are uncertain. However, for tasks such as summarization or cQA, the goal is to find the best candidates. While it is important to distinguish between good and optimal candidates, it is wasted effort to compare candidates that we are already confident are not the best, even if their utilities are still uncertain. We propose to address this using an acquisition function for BO that estimates the *expected improvement* (Močkus, 1975) of a candidate, $a$, over our current estimated best solution, $b$, given current pairwise labels, $\boldsymbol{D}$. Here, we provide the first closed-form acquisition function that uses expected improvement for pairwise preference learning.

We define *improvement* as the quantity $\max\{0, f_a - f_b\}$, where $b$ is our current best item and $a$ is our new candidate. Because the values of $f_a$ and $f_b$ are uncertain, we compute the *expected* improvement as follows. First, we estimate the posterior distribution over the candidates' utilities, $\mathcal{N}(\hat{\boldsymbol{f}}, \boldsymbol{C})$, then find the current best utility: $\hat{f}_b = \max_i\{\hat{f}_i\}$. For any candidate $a$, the difference $f_a - f_b$ is Gaussian-distributed as it is a sum of Gaussians. The probability that this is larger than zero is given by the cumulative density function, $\Phi(z)$, where $z = \frac{\hat{f}_a - \hat{f}_b}{\sqrt{v}}$. We use this to derive expected improvement, which results in the following closed form equation:

$$\text{Imp}(a; \boldsymbol{D}) = \sqrt{v} z \Phi(z) + \sqrt{v} \mathcal{N}(z; 0, 1), \quad (7)$$

This weights the probability of finding a better solution, $\Phi(z)$, by the amount of improvement, $\sqrt{v}z$. Both terms account for how close $\hat{f}_a$ is to $\hat{f}_b$ through $z$, as a larger distance causes $z$ to be more negative, which decreases both the probability $\Phi(z)$ and the density $\mathcal{N}(z; 0, 1)$. Expected improvement also accounts for the uncertainty in both utilities through the posterior standard deviation, $\sqrt{v}$, which scales both terms. All candidates have positive expected improvement, as there is a non-zero probability that labeling them will lead to a new best item; otherwise, the current best candidate remains, and improvement is zero.

To select pairs of items, the IMP strategy greedily chooses the current best item and the item with the greatest expected improvement. Through the consideration of posterior uncertainty, IMP trades off *exploration* of unknown candidates with *exploitation* of promising candidates. In contrast, uncertainty-based strategies are pure exploration.

**Thompson Sampling with Pairwise Labels (TP).** Expected improvement is known to over-exploit in some cases (Qin et al., 2017): It chooses where to sample based on the current distribution, so if this distribution underestimates the mean and variance of the optimum, it may never be sampled. To increase exploration, we propose a strategy that uses Thompson sampling (Thompson, 1933). Unlike IMP, which is deterministic, TP introduces random exploration through sampling. TP is similar to dueling-Thompson sampling for continuous input domains (González et al., 2017), but uses an information gain step (described below) and samples from a pool of discrete candidates.

We select an item using Thompson sampling as follows: First draw a sample of candidate utilities from their posterior distribution, $\boldsymbol{f}_{\text{thom}} \sim \mathcal{N}(\hat{\boldsymbol{f}}, \boldsymbol{C})$, then choose the item $b$ with the highest score in the sample. This sampling step depends on a Bayesian approach to provide a posterior distribution from which to sample. Sampling means that while candidates with high expected utilities have higher values of $f_{\text{thom}}$ in most samples, other candidates may also have the highest score in some samples. As the number of samples $\rightarrow \infty$, the number of times each candidate is chosen is proportional to the probability that it is the best candidate.

To create a pair of items for preference learning, the TP strategy computes the expected information gain for all pairs that include candidate $b$, and chooses the pair with the maximum. This strategy is less greedy than IMP as it allows more learning about uncertain items through both the Thompson sampling step and the information gain step. However, compared to EIG, the first step focuses effort more on items with potentially high scores.

**Using Priors to Address Cold Start.** In previous work on summarization (Gao et al., 2018), the BT model was trained from a *cold start*, namely, with no prior knowledge or pretraining. Then, after active learning was complete, the predictions from the trained model were combined with prior predictions based on heuristics by taking an average of the normalized scores from both methods. We propose to use such prior predictions to determine an *informative prior* for GPPL, enabling the active learner to make more informed choices of candidates to label at the start of the active learning process, thereby alleviating the cold-start problem.

We integrate pre-computed predictions as follows. Given a set of prior predictions, $\boldsymbol{\mu}$, from a heuristic or pre-trained model, we set the prior mean of the Gaussian process to $\boldsymbol{\mu}$ before collecting any data, so that the candidate utilities have the prior $p(\boldsymbol{f}|\phi(\boldsymbol{x})) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{K})$, where $\boldsymbol{K}$ is a hyper-parameter. Given this setup, AL can now take the prior predictions into account when choosing pairs of candidates for labeling.

## 5 Experiments

We perform experiments on three tasks to test our interactive text ranking approach: (1) Community question answering (cQA)–identify the best answer to a given question from a pool of candidate answers; (2) Rating extractive multidocument summaries according to a user's preferences; (3) Generating an extractive multidocument summary by training a reinforcement learner with the ranking function from 2 as a reward function. Using interactive learning to learn the reward function rather than the policy reduces the number of user interactions from many thousands to 100 or less. These tasks involve highly specialized questions or topics where generic models could be improved with user feedback. For the first two tasks, we simulate the interactive process in Algorithm 1. The final task uses the results of this process.

**Datasets.** Both the cQA and multidocument summarization datasets were chosen because the answers and candidate summaries in these datasets are multisentence documents that take longer for users to read compared to tasks such as factoid question-answering. We expect our methods to have the greatest impact in this type of long-answer scenario by minimizing user interaction time.

For cQA, we use datasets consisting of questions posted on StackExchange in the communities *Apple*, *Cooking*, and *Travel*, along with their accepted answers and candidate answers taken from related questions (Rücklé et al., 2019). Each accepted answer was marked by the user who posted the question, so reflects that user's own opinion. Dataset statistics are shown in Table 2.

| cQA Topics | #questions | #accepted answers | #candidate answers |
|---|---|---|---|
| Apple | 1,250 | 1,250 | 125,000 |
| Cooking | 792 | 792 | 79,200 |
| Travel | 766 | 766 | 76,600 |
| Summarization Datasets | #topics | #model summaries | #docs |
| DUC 2001 | 30 | 90 | 300 |
| DUC 2002 | 59 | 177 | 567 |
| DUC 2004 | 50 | 150 | 500 |

Table 2: Dataset statistics for summarization and cQA.

For summarization, we use the DUC datasets,[2] which contain model summaries written by experts for collections of documents related to a narrow topic. Each topic has three model summaries, each written by a different expert and therefore reflecting different opinions about what constitutes a good summary. Compared with single-document summarization, the challenging multidocument case is an ideal testbed for interactive approaches, because the diversity of themes within a collection of documents makes it difficult to identify a single, concise summary suitable for all users.

**Priors and Input Vectors.** We use our interactive approach to improve a set of prior predictions provided by a pretrained method. For cQA, we first choose the previous state-of-the-art for long answer selection, *COALA* (Rücklé et al., 2019),which estimates the relevance of answers to a question by extracting *aspects* (e.g., $n$-grams or syntactic structures) from the question and answer texts using CNNs, then matching and aggregating the aspects. For each topic, we train an instance of COALA on the training split given by Rücklé et al. (2019), then run the interactive process on the test set, that is, the dataset in Table 2, to simulate a user interactively refining the answers selected for their question. As inputs for the BT and GPPL models, we use the COALA feature vectors: For each question, COALA extracts aspects from the question and its candidate answers; each dimension of an answer's 50-dimensional feature vector encodes how well the answer covers one of the extracted aspects.

[2] http://duc.nist.gov/.

Next we apply our interactive approach to refine predictions from the current state of the art (Xu et al., 2019), which we refer to as *BERT-cQA*. This method places two dense layers with 100 and 10 hidden units on top of BERT (Devlin et al., 2019). As inputs to BERT, we concatenate the question and candidate answer and pad sequences to 512 tokens (4% QA pairs are over-length and are truncated). The whole model is fine-tuned on the StackExchange training sets, the same as COALA. In our simulations, we use the fine-tuned, final-layer [CLS] embeddings with 768 dimensions as inputs to BT and GPPL for each question-answer pair.

As prior predictions for summary ratings we first evaluate *REAPER*, a heuristic evaluation function described by Ryang and Abekawa (2012). We obtain *bigram+* feature vectors for candidate summaries by augmenting bag-of-bigram embeddings with additional features proposed by Rioux et al. (2014). The first 200 dimensions of the feature vector have binary values to indicate the presence of each of the 200 most common bigrams in each topic after tokenizing, stemming and applying a stop-list. The last 5 dimensions contain the following: the fraction of the 200 most common bigrams that are present in the document (coverage ratio); the fraction of the 200 most common bigrams that occur more than once in the document (redundancy ratio); document length divided by 100 (length ratio); the sum over all extracted sentences of the reciprocal of the position of the extracted sentence in its source document (extracted sentence position feature); a single bit to indicate if document length exceeds the length limit of 100 tokens. The same features are used for both tasks (2) learning summary ratings and (3) reinforcement learning.

We also test prior predictions from a state-of-the-art summary scoring method, *SUPERT* (Gao et al., 2020), which uses a variant of BERT that has been fine-tuned on news articles to obtain 1024-dimensional contextualized embeddings of a summary. To score a summary, SUPERT extracts a pseudo-reference summary from the source documents, then compares its embedding with that of the test summary. With the SUPERT priors we compare bigram+ feature vectors and the SUPERT embeddings as input to BT and GPPL for task (2).

**Interactive Methods.** As baselines, we test BT as our preference learner with random selection and the UNC active learning strategy, and GPPL as the learner with random selection. We also combine GPPL with the acquisition functions described in Section 4, UNPA, EIG, IMP, and TP. For random sampling, we repeat each experiment ten times.

**Simulated Users.** In tasks (1) and (2), we simulate a user's preferences with a noisy oracle based on the user-response models of Viappiani and Boutilier (2010). Given gold standard scores for two documents, $g_a$ and $g_b$, the noisy oracle prefers document $a$ with probability $p(y_{a,b}|g_a, g_b) = (1 + \exp(\frac{g_b - g_a}{t}))^{-1}$, where $t$ is a parameter that controls the noise level. Both datasets contain model summaries or gold answers, but no gold standard scores. We therefore estimate gold scores by computing a ROUGE score of the candidate summary or answer, $a$, against the model summary or gold answer, $m$. For cQA, we take the ROUGE-L score as a gold score, as it is a well-established metric for evaluating question answering systems (e.g., Nguyen et al., 2016; Bauer et al., 2018; Indurthi et al., 2018) and set $t = 0.3$, which results in annotation accuracy of 83% (the fraction of times the pairwise label corresponds to the gold ranking).

For summarization, we use $t = 1$, which gives noisier annotations with 66% accuracy, reflecting the greater difficulty of choosing between two summaries. This corresponds to accuracies of annotators found by Gao et al. (2019) when comparing summaries from the same datasets. As gold for summarization, we combine ROUGE scores using the following formula, previously found to correlate well with human preferences on a comparable summarization task (P.V.S and Meyer, 2017):

$$g_a \approx R_{comb} = \frac{\text{ROUGE}_1(a,m)}{0.47} + \frac{\text{ROUGE}_2(a,m)}{0.22} + \frac{\text{ROUGE}_{\text{SU4}}(a,m)}{0.18}. \quad (8)$$

Following Gao et al. (2019), we normalize the gold scores $g_a$ to the range $[0, 10]$.

## 5.1 Warm-start Using Prior Information

We compare two approaches to integrate the prior predictions of utilities computed before acquiring user feedback. As a baseline, *sum* applies a weighted mean to combine the prior predictions with posterior predictions learned using GPPL or BT. Based on preliminary experiments, we weight

| Strategy | Prior | Datasets | | |
|---|---|---|---|---|
| *Accuracy for cQA with COALA priors* | | | | |
| *#interactions=10* | | Apple | Cooking | Travel |
| random | sum | .245 | .341 | .393 |
| random | prior | **.352** | **.489** | **.556** |
| UNPA | sum | **.293** | **.451** | .423 |
| UNPA | prior | .290 | .392 | **.476** |
| IMP | sum | .373 | .469 | .466 |
| IMP | prior | **.615** | **.750** | **.784** |
| *NDCG@1% for summarization with REAPER priors* | | | | |
| *#interactions=20* | | DUC'01 | DUC'02 | DUC'04 |
| random | sum | **.595** | **.623** | **.652** |
| random | prior | .562 | .590 | .600 |
| UNPA | sum | .590 | .628 | **.650** |
| UNPA | prior | **.592** | **.635** | .648 |
| IMP | sum | .618 | .648 | .683 |
| IMP | prior | **.654** | **.694** | **.702** |

Table 3: The effect of integrating pre-computed predictions as Bayesian priors vs. taking a weighted mean of pre-computed and posterior predictions.

| Learner | Strategy | Apple | | Cooking | | Travel | |
|---|---|---|---|---|---|---|---|
| | | acc | N5 | acc | N5 | acc | N5 |
| COALA | | .318 | .631 | .478 | .696 | .533 | .717 |
| *COALA prior, #interactions=10* | | | | | | | |
| BT | random | .272 | .589 | .368 | .614 | .410 | .644 |
| BT | UNC | .233 | .573 | .308 | .597 | .347 | .619 |
| GPPL | random | .352 | .642 | .489 | .699 | .556 | .722 |
| GPPL | UNPA | .290 | .591 | .392 | .631 | .476 | .656 |
| GPPL | EIG | .302 | .628 | .372 | .671 | .469 | .692 |
| GPPL | TP | .274 | .592 | .353 | .636 | .414 | .675 |
| GPPL | IMP | **.615** | **.714** | **.750** | **.753** | **.784** | **.774** |
| BERT-cQA | | .401 | .580 | .503 | .625 | .620 | .689 |
| *BERT-cQA prior, #interactions=10* | | | | | | | |
| BT | random | .170 | .626 | .228 | .637 | .315 | .676 |
| BT | UNC | .129 | .580 | .181 | .583 | .326 | .618 |
| GPPL | random | .407 | .593 | .510 | .594 | .631 | .657 |
| GPPL | EIG | .080 | .559 | .140 | .552 | .095 | .526 |
| GPPL | IMP | **.614** | **.715** | **.722** | **.731** | **.792** | **.744** |

Table 4: Interactive text ranking for cQA. N5=NDCG@5, acc=accuracy.

the prior and posterior predictions equally. *Prior* sets the prior mean of GPPL to the value of the prior predictions, as described in Section 4. Our hypothesis is that *prior* will provide more information at the start of the interactive learning process and help the learner to select more informative pairs.

Table 3 presents results of a comparison on a subset of strategies, showing that *prior* results in higher performance in many cases. Based on the results of these experiments, we apply *prior* to all further uses of GPPL.

### 5.2 Community Question Answering

We hypothesize that the prior ranking given by COALA can be improved by incorporating a small amount of user feedback for each question. Our interactive process aims to find the best answer to a specific question, rather than learning a model that transfers to new questions, hence preferences are sampled for questions in the *test* splits.

To evaluate the top-ranked answers from each method, we compute accuracy as the fraction of top answers that match the gold answers. We also compare the five highest-ranked solutions to the gold answers using *normalized discounted cumulative gain* (NDCG@5) with ROUGE-L as the relevance score. NDCG@k evaluates the relevance of the top $k$ ranked items, putting more weight on higher-ranked items (Järvelin and Kekäläinen, 2002).

The results in the top half of Table 4 show that with only 10 user interactions, most methods are unable to improve performance over pre-trained COALA. UNC, UNPA, EIG, and TP are outperformed by random selection and IMP ($p \ll .01$ using a two-tailed Wilcoxon signed-rank test).

To see whether the methods improve given more feedback, Figure 3 plots NDCG@5 against number of interactions. Whereas IMP performance increases substantially, random selection improves only very slowly. Early interactions cause a performance drop with UNPA, EIG, and TP. This is unlikely to be caused by noise in the cQA data, because preference labels are generated using ROUGE-L scores computed against the gold answer. The drop is because uncertainty-based methods initially sample many low-quality candidates with high uncertainty. This increases the predicted utility of the preferred candidate in each pair, sometimes exceeding better candidates that were ranked higher by the prior, pushing them out of the top five. Performance rises once the uncertainty of mediocre candidates has been reduced and stronger candidates are selected. Both BT methods start from a worse initial position but improve consistently, as their initial samples are not biased by the prior predictions, although UNC remains worse than random.

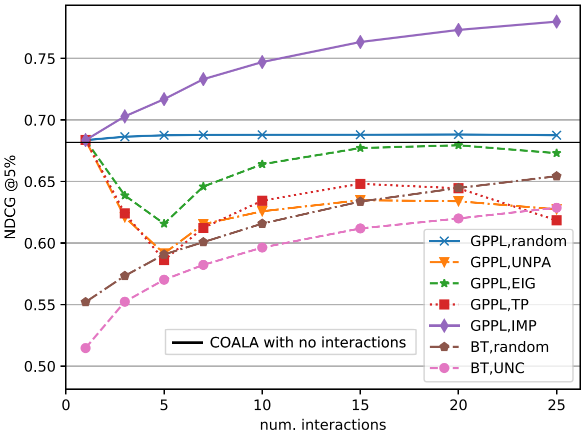

Figure 3: NDCG@5 with increasing interactions, COALA prior, mean across 3 cQA topics.

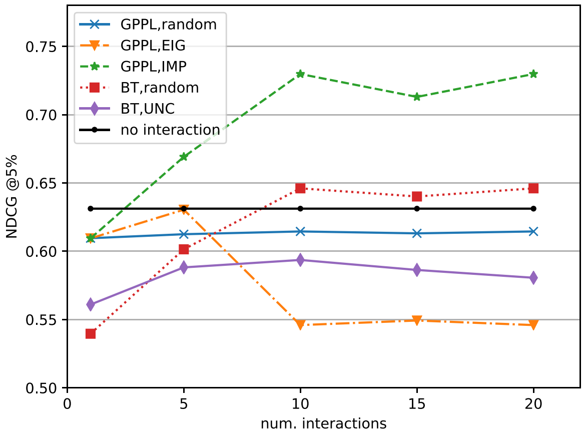

Figure 4: NDCG@5 with increasing number of interactions. BERT-cQA prior. Mean across 3 cQA topics.

The bottom half of Table 4 and Figure 4 show results for key methods with BERT-cQA priors and embeddings. The initial predictions by BERT-cQA have higher accuracy than COALA but lower NDCG@5. BERT-based models better account for question and answer semantics, leading to higher accuracy, but place less emphasis on lexical similarity, which reduces the ROUGE-L scores of top-ranked answers and consequently, NDCG@5. While IMP remains the most successful method, the end result is not a clear improvement over COALA, with a collapse in accuracy for the uncertainty-based EIG and both BT methods. As with COALA, these uncertainty-based methods focus initially on middling candidates, but due to the sparsity of the data with high-dimensional BERT-cQA embeddings, more samples are required to reduce their uncertainty before these methods start to sample strong candidates. The flexibility of the GP model means that it is particularly affected by data sparsity, hence the poor performance of EIG.

### 5.3 Interactive Summary Rating

We apply interactive learning to refine a ranking over candidate summaries given prior information. For each topic, we create 10,000 candidate summaries with fewer than 100 words each, which are constructed by uniformly selecting sentences at random from the input documents. To determine whether some strategies benefit from more samples, we test each active learning method with between 10 and 100 user interactions with noisy simulated users. The method is fast enough for interactive scenarios: on a standard Intel desktop workstation with a quad-core CPU and no GPU, updates to GPPL at each iteration require around one second.

We evaluate the quality of the 100 highest-ranked summaries using NDCG@1%, and compute the Pearson correlation, $r$, between the predicted utilities for all candidates and the combined ROUGE scores (Eq. (8)). Unlike NDCG@1%, $r$ does not focus on higher-ranked candidates but considers the utilities for all candidates. Hence we do not expect that IMP or TP, which optimize the highest-ranked candidates, will have the highest $r$.

With REAPER priors, bigram+ features and 20 interactions, the top part of Table 5 shows a clear advantage to IMP in terms of NDCG@1%, which outperforms the previous state of the art, BT-UNC (significant with $p \ll .01$ on all datasets). In terms of $r$, IMP is out-performed by TP (significant with $p \ll .01$ on all datasets), which appears more balanced between finding the best candidate and learning the ratings for all candidates. UNPA improves slightly over random sampling for both metrics, while EIG is stronger due to a better focus on epistemic uncertainty. Unlike IMP, TP does not always outperform EIG on NDCG@1%.

Figure 5 shows the progress of each method with increasing numbers of interactions on DUC'01. The slow progress of the BT baselines is clear, illustrating the advantage the Bayesian methods have as a basis for active learning by incorporating uncertainty estimates and prior predictions.

| Learner | Strategy | DUC'01 | | DUC'02 | | DUC'04 | |
|---|---|---|---|---|---|---|---|
| | | N1 | r | N1 | r | N1 | r |
| REAPER | | .539 | .262 | .573 | .278 | .597 | .322 |
| *REAPER prior, bigram+ features, #interactions=20* | | | | | | | |
| BT | rand. | .596 | .335 | .626 | .358 | .659 | .408 |
| BT | UNC | .609 | .340 | .641 | .365 | .674 | .415 |
| GPPL | rand. | .558 | .248 | .590 | .266 | .603 | .289 |
| GPPL | UNPA | .592 | .307 | .635 | .370 | .648 | .397 |
| GPPL | EIG | .634 | .327 | .665 | .383 | .675 | .404 |
| GPPL | TP | .629 | **.378** | .665 | **.403** | .690 | **.453** |
| GPPL | IMP | **.654** | .303 | **.694** | .345 | **.702** | .364 |
| SUPERT | | .602 | .382 | .624 | .400 | .657 | .438 |
| *SUPERT prior, bigram+ features, #interactions=20* | | | | | | | |
| BT | rand. | .633 | .415 | .654 | **.438** | .684 | **.483** |
| BT | UNC | .550 | .277 | .561 | .287 | .588 | .334 |
| GPPL | rand. | .601 | .351 | .630 | .377 | .657 | .419 |
| GPPL | EIG | .633 | .365 | .662 | .399 | .671 | .435 |
| GPPL | TP | .649 | **.417** | .668 | .437 | .698 | .479 |
| GPPL | IMP | **.653** | .322 | **.696** | .374 | **.717** | .407 |
| *SUPERT prior, SUPERT embeddings, #interact.=20* | | | | | | | |
| GPPL | IMP | .624 | .297 | .630 | .284 | .653 | .339 |
| *SUPERT prior, bigram+ features, #interactions=100* | | | | | | | |
| GPPL | IMP | .668 | .308 | **.788** | .466 | **.815** | .521 |
| *SUPERT prior, SUPERT embeddings, #interact.=100* | | | | | | | |
| BT | rand. | .661 | **.466** | .696 | **.504** | .727 | **.543** |
| BT | UNC | .634 | .420 | .656 | .453 | .678 | .495 |
| GPPL | rand. | .594 | .354 | .617 | .387 | .643 | .415 |
| GPPL | EIG | .611 | .372 | .647 | .415 | .682 | .471 |
| GPPL | IMP | **.728** | .376 | **.752** | .407 | **.769** | .447 |

Table 5: Interactive Summary Rating. N1= NDCG@1%, r=Pearson's correlation coefficient. Bold indicates best result for each prior and number of interactions.

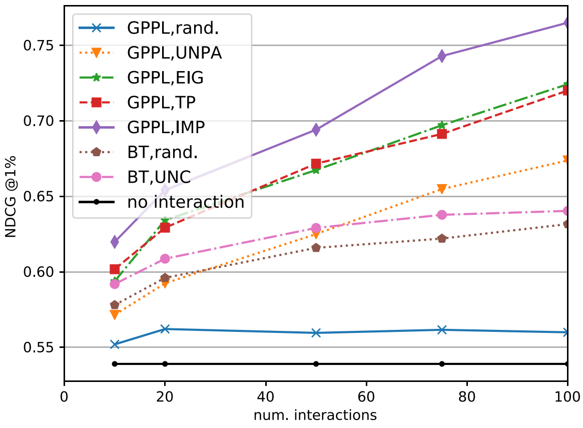

Figure 5: DUC'01, REAPER prior, bigram+ features, changes in NDCG@1% with increasing interactions.

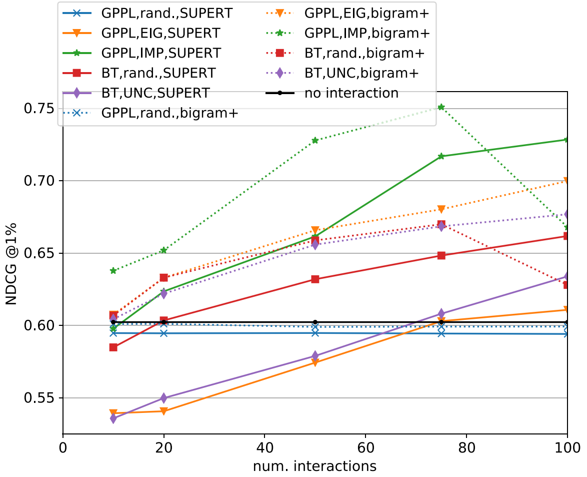

Figure 6: DUC'01, SUPERT prior, changes in NDCG@1% with increasing number of interactions.

The lower part of Table 5 and Figure 6 confirm the superior NDCG@1% scores of IMP with the stronger SUPERT priors. However, while pretrained SUPERT outperforms REAPER, the results after 20 rounds of interaction with bigram+ features are almost identical, suggesting that user feedback helps mitigate the weaker pretrained model. With only 20 interactions, bigram+ features work better than SUPERT embeddings as input to our interactive learners, even with the best-performing method, IMP, since there are fewer features and the model can cope better with limited labeled data. With 100 interactions, SUPERT embeddings provide superior performance as there are sufficient labels to leverage the richer input embeddings.

### 5.4 RL for Summarization

We now investigate whether our approach also improves performance when the ranking function is used to provide rewards for a reinforcement learner. Our hypothesis is that it does not matter whether the rewards assigned to bad candidates are correct, as long as they are distinguished from good candidates, as this will prevent the policy from choosing bad candidates to present to the user.

To test the hypothesis, we simulate a *flat-bottomed* reward function for summarization on DUC'01: First, for each topic, we set the rewards for the 10,000 sampled summaries (see § 5.3) to the gold standard, $R_{comb}$ (Eq. (8), normalized to $[0, 10]$). Then, we set the rewards for a varying

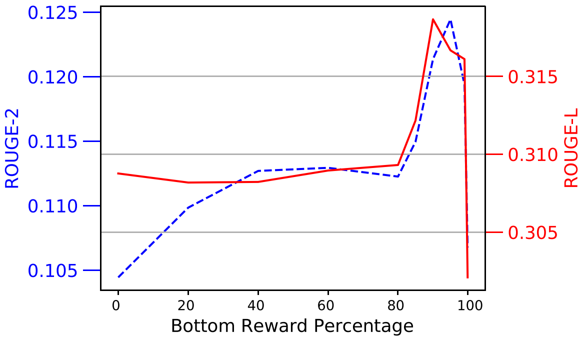

Figure 7: Performance of RL on DUC'01 when the rewards for the bottom $x\%$ summaries are flattened to one. Dashed line = ROUGE-2, solid line = ROUGE-L.

percentage of the lowest-ranked summaries to 1.0 (the flat bottom). We train the reinforcement learner on the flat-bottomed rewards and plot ROUGE scores for the proposed summaries in Figure 7. The performance of the learner actually increases as candidate values are flattened until around 90% of the summaries have the same value. This supports our hypothesis that the user's labeling effort should be spent on the top candidates.

We now use the ranking functions learned in the previous summary rating task as rewards for reinforcement learning. As examples, we take the rankers learned using SUPERT priors with bigram+ features and 20 interactions and with SUPERT embeddings and 100 interactions. We replicate the RL setup of Gao et al. (2018) for interactive multidocument summarization, which previously achieved state-of-the-art performance using the BT learner with UNC. The RL agent models the summarization process as follows: there is a current state, represented by the current draft summary; the agent uses a *policy* to select a sentence to be concatenated to the current draft summary or to terminate the summary construction. During the learning process, the agent receives a reward after terminating, which it uses to update its policy to maximize these rewards. The model is trained for 5,000 episodes (i.e., generating 5,000 summaries and receiving their rewards), then the policy is used to produce a summary. We compare the produced summary to a human-generated model summary using ROUGE. By improving the reward function, we hypothesize that the quality of the resulting summary will also improve.

Table 6 shows that the best-performing method from the previous tasks, IMP, again produces a strong improvement over the previous state of the art, BT with UNC (significant with $p \ll 0.01$ in all cases), as well as GPPL with EIG. With 20 interactions and bigram+ features, EIG also outperforms BT-UNC, indicating the benefits of the Bayesian approach, but this is less clear with SUPERT embeddings, where the high-dimensional embedding space may lead to sparsity problems for the GP. The standard deviation in performance over multiple runs of RL is <0.004 for all metrics, datasets, and methods, suggesting that the advantage gained by using IMP is robust to randomness in the RL algorithm. The results confirm that gains in NDCG@1% made by BO over uncertainty-based strategies when learning the utilities translate to better summaries produced by reinforcement learning in a downstream task.

## 5.5 Limitations of User Simulations

By testing our interactive process with simulated users, we were able to compare numerous methods with a fixed labeling error rate. The user labels were simulated using data from real individuals: the gold answers for cQA were chosen by the user who posed the question, and the three model summaries for each topic in the DUC datasets were each authored by a different individual. While this work shows the promize of BO, further work is needed to test specific NLP applications with real end users. Our experiments illustrate plausible applications where users compare texts of up to 100 words and gain substantial performance advantages. Other applications require a broader study of reading and labeling time versus performance benefits and user satisfaction. It may also be possible to select chunks of longer documents for the user to compare, rather than reading whole documents.

Another dimension to consider is that real users may make systematic, rather than random errors. However, in the applications we foresee, it is accepted that their preference labels will often diverge from any established gold standard, as users adapt models to their own information needs. Future work may therefore apply interactive approaches to more subjective NLP tasks, such as adapting a summary to more personal information needs.

| #inter | Learner | Features | Strat | DUC’01 | | | | DUC’02 | | | | DUC’04 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| -actions | | | -egy | R1 | R2 | RL | RSU4 | R1 | R2 | RL | RSU4 | R1 | R2 | RL | RSU4 |
| 0 | SUPERT | N/A | none | .324 | .061 | .252 | .097 | .345 | .070 | .270 | .107 | .375 | .086 | .293 | .128 |
| 20 | BT | bigrams+ | UNC | .335 | .072 | .265 | .104 | .364 | .086 | .286 | .120 | .390 | .101 | .307 | .136 |
| 20 | GPPL | bigrams+ | rand. | .324 | .064 | .252 | .097 | .358 | .081 | .281 | .115 | .383 | .095 | .302 | .131 |
| 20 | GPPL | bigrams+ | EIG | .346 | .073 | .269 | .110 | .377 | .095 | .295 | .126 | .394 | .106 | .310 | .137 |
| 20 | GPPL | bigrams+ | IMP | **.355** | **.086** | **.277** | **.114** | **.385** | **.103** | **.300** | **.130** | **.419** | **.122** | **.331** | **.154** |
| 100 | BT | SUPERT | UNC | .337 | .072 | .264 | .104 | .366 | .086 | .284 | .118 | .377 | .090 | .297 | .128 |
| 100 | GPPL | SUPERT | rand. | .317 | .057 | .247 | .092 | .344 | 071 | .270 | .107 | .372 | .087 | .292 | .124 |
| 100 | GPPL | SUPERT | EIG | .331 | .070 | .259 | .101 | .367 | .088 | .287 | .120 | .394 | .103 | .309 | .136 |
| 100 | GPPL | SUPERT | IMP | **.370** | **.100** | **.293** | **.123** | **.406** | **.118** | **.316** | **.140** | **.422** | **.130** | **.337** | **.155** |

Table 6: RL for summarization: ROUGE scores of final summaries, mean over 10 repeats with different random seeds. Once the rewards are fixed, the performance of RL is stable: standard deviation of each result is $< 0.004$.

## 6 Conclusions

We proposed a novel approach to interactive text ranking that uses Bayesian optimization (BO) to identify top-ranked texts by acquiring pairwise feedback from a user and applying Gaussian process preference learning (GPPL). Our experiments showed that BO significantly improves the accuracy of answers chosen in a cQA task with small amounts of feedback, and leads to summaries that better match human-generated model summaries when used to learn a reward function for reinforcement learning.

Of two proposed Bayesian optimization strategies, we found that expected improvement (IMP) outperforms Thompson sampling (TP) if the goal is to optimize the proposed best solution. TP may require a larger number of interactions due to its random sampling step. IMP is effective in both cQA and summarization tasks, but has the strongest impact on cQA with only 10 interactions. This may be due to the greater sparsity of candidates in cQA (100 versus 10,000 for summarization), which allows them to be more easily discriminated by the model, given good training examples. Further evaluation with real users is required to gauge the quantity of feedback needed in a particular domain.

When using high-dimensional BERT embeddings as inputs, GPPL requires more labels to achieve substantial improvements. Future work may therefore investigate recent dimensionality reduction methods (Raunak et al., 2019). We found that performance improves when including prior predictions as the GPPL prior mean but it is unclear how best to estimate confidence in the prior predictions—here we assume equal confidence in all prior predictions. Future work could address this by adapting the GPPL prior covariance matrix to kick-start BO. The method is also currently limited to a single set of prior predictions: In future we intend to integrate predictions from several models.

## Acknowledgments

This work was supported by the German Research Foundation (DFG) as part of the EVIDENCE project (grant GU 798/27-1) and the PEER project (grant GU 798/28-1). We would like to thank the reviewers and journal editors for their extremely helpful feedback.

## References

Jason Baldridge and Miles Osborne. 2004. Active learning and the total cost of annotation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 9–16, Barcelona, Spain. Association for Computational Linguistics.

Lisa Bauer, Yicheng Wang, and Mohit Bansal. 2018. Commonsense for generative multi-hop question answering tasks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4220–4230, Brussels, Belgium. Association for Computational Linguistics. **DOI:** `https://doi.org/10.18653/v1/D18-1454`

Daniel Beck, Trevor Cohn, and Lucia Specia. 2014. Joint emotion analysis via multi-task

Gaussian processes. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1798–1803. Association for Computational Linguistics. **DOI:** https://doi.org/10.3115/v1/D14-1190

Ralph Allan Bradley and Milton E. Terry. 1952. Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*, 39(3/4):324–345. **DOI:** https://doi.org/10.1093/biomet/39.3-4.324

Eric Brochu, Vlad M. Cora, and Nando De Freitas. 2010. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*.

Eric Brochu, Nando de Freitas, and Abhijeet Ghosh. 2008. Active preference learning with discrete choice data. In *Advances in Neural Information Processing Systems*, pages 409–416.

Wei Chu and Zoubin Ghahramani. 2005. Preference learning with Gaussian processes. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 137–144. ACM. **DOI:** https://doi.org/10.1145/1102351.1102369

Trevor Cohn and Lucia Specia. 2013. Modelling annotator bias with multi-task Gaussian processes: An application to machine translation quality estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 32–42. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pretraining of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Terry N. Flynn and A. A. J. Marley. 2014. Best–worst scaling: Theory and methods. In Stephane Hess and Andrew Daly, editors, *Handbook of Choice Modelling*, pages 178–201. Edward Elgar Publishing, Cheltenham, UK. **DOI:** https://doi.org/10.4337/9781781003152.00014

Yang Gao, Christian M. Meyer, and Iryna Gurevych. 2018. APRIL: Interactively learning to summarise by combining active preference learning and reinforcement learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4120–4130, Brussels, Belgium. Association for Computational Linguistics. **DOI:** https://doi.org/10.18653/v1/D18-1445

Yang Gao, Christian M. Meyer, and Iryna Gurevych. 2019. Preference-based interactive multi-document summarisation. *Information Retrieval Journal*, pages 1–31.

Yang Gao, Wei Zhao, and Steffen Eger. 2020. SUPERT: Towards new frontiers in unsupervised evaluation metrics for multi-document summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1347–1354, Online. Association for Computational Linguistics. **DOI:** https://doi.org/10.18653/v1/2020.acl-main.124, **PMID:** 31961681

Javier González, Zhenwen Dai, Andreas Damianou, and Neil D. Lawrence. 2017. Preferential Bayesian optimization. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1282–1291.

Matthew D. Hoffman, David M. Blei, Chong Wang, and John William Paisley. 2013. Stochastic variational inference. *Journal of Machine Learning Research*, 14:1303–1347.

Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. 2011. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*.

Sathish Reddy Indurthi, Seunghak Yu, Seohyun Back, and Heriberto Cuayáhuitl. 2018. Cut to the chase: A context zoom-in network for reading comprehension. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 570–575. Brussels, Belgium. Association for Computational Linguistics. **DOI:** https://doi.org/10.18653.18653/v1/D18-1054

Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446. **DOI:** https://doi.org/10.1145/582415.582418

Edwin T. Jaynes. 2003. *Probability Theory: The Logic of Science*, Cambridge University Press. **DOI:** https://doi.org/10.1017/CBO9780511790423

Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 133–142. ACM. **DOI:** https://doi.org/10.1145/775047.775067

Maurice George Kendall. 1948. *Rank Correlation Methods*. Griffin, Oxford, UK.

David C. Kingsley and Thomas C. Brown. 2010. Preference uncertainty, preference refinement and paired comparison experiments. *Land Economics*, 86(3):530–544. **DOI:** https://doi.org/10.3368/le.86.3.530

Carolin Lawrence and Stefan Riezler. 2018. Improving a neural semantic parser by counterfactual learning from human bandit feedback. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1820–1830, Melbourne, Australia. Association for Computational Linguistics. **DOI:** https://doi.org/10.18653/v1/P18-1169

Xiao Lin and Devi Parikh. 2017. Active learning for visual question answering: An empirical study. *arXiv preprint arXiv:1711.01732*.

Yandong Liu and Eugene Agichtein. 2008. You've got answers: Towards personalized models for predicting success in community question answering. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, pages 97–100. Association for Computational Linguistics.

Manuel Lopes, Francisco Melo, and Luis Montesano. 2009. Active learning for reward estimation in inverse reinforcement learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 31–46. Springer. **DOI:** https://doi.org/10.1007/978-3-642-04174-7_3

Manuel J. Maña López, Manuel de Buenaga Rodríguez, and José María Gómez Hidalgo. 1999. Using and evaluating user directed summaries to improve information access. In *International Conference on Theory and Practice of Digital Libraries*, pages 198–214. Springer. **DOI:** https://doi.org/10.1007/3-540-48155-9_14

R. Duncan Luce. 1959. On the possible psychophysical laws. *Psychological Review*, 66(2): 81–95. **DOI:** https://doi.org/10.1037/h0043178

David J. C. MacKay. 1992. Information-based objective functions for active data selection. *Neural computation*, 4(4):590–604. **DOI:** https://doi.org/10.1162/neco.1992.4.4.590

Jonas Močkus. 1975. On bayesian methods for seeking the extremum. In *Optimization Techniques IFIP Technical Conference*, pages 400–404. Springer. **DOI:** https://doi.org/10.1007/978-3-662-38527-2_55

Frederick Mosteller. 1951. Remarks on the method of paired comparisons: I. The least squares solution assuming equal standard deviations and equal correlations. *Psychometrika*, 16(1): 3–9. **DOI:** https://doi.org/10.1007/BF02313422

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. *choice*, 2640:660.

Álvaro Peris and Francisco Casacuberta. 2018. Active learning for interactive neural machine translation of data streams. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 151–160. Association for Computational Linguistics, Brussels, Belgium. **DOI:** https://doi.org/10.18653/v1/K18-1015

R. L. Plackett. 1975. The analysis of permutations. *Journal of the Royal Statistical Society, Series C*

*(Applied Statistics)*, 24(2):193–202. **DOI:** https://doi.org/10.2307/2346567

Avinesh P.V.S and Christian M. Meyer. 2017. Joint optimization of user-desired content in multi-document summaries by learning from user feedback. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1353–1363, Vancouver, Canada. Association for Computational Linguistics.

Chao Qin, Diego Klabjan, and Daniel Russo. 2017. Improving the expected improvement algorithm. In *Advances in Neural Information Processing Systems*, pages 5381–5391.

Vikas Raunak, Vivek Gupta, and Florian Metze. 2019. Effective dimensionality reduction for word embeddings. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 235–243. Florence, Italy. Association for Computational Linguistics. **DOI:** https://doi.org/10.18653/v1/W19-4328

Cody Rioux, Sadid A. Hasan, and Yllias Chali. 2014. Fear the REAPER: A system for automatic multi-document summarization with reinforcement learning. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 681–690, Doha, Qatar. Association for Computational Linguistics. **DOI:** https://doi.org/10.3115/v1/D14-1075

Sebastian Ruder and Barbara Plank. 2017. Learning to select data for transfer learning with Bayesian optimization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 372–382, Copenhagen, Denmark. Association for Computational Linguistics. **DOI:** https://doi.org/10.18653/v1/D17-1038

Seonggi Ryang and Takeshi Abekawa. 2012. Framework of automatic text summarization using reinforcement learning. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 256–265, Jeju Island, Korea. Association for Computational Linguistics.

Andreas Rücklé, Nafise Sadat Moosavi, and Iryna Gurevych. 2019. COALA: A neural coverage-based approach for long answer selection with small data. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*. AAAI. **DOI:** https://doi.org/10.1609/aaai.v33i01.33016932

Robin Senge, Stefan Bösner, Krzysztof Dembczyński, Jörg Haasenritter, Oliver Hirsch, Norbert Donner-Banzhoff, and Eyke Hüllermeier. 2014. Reliable classification: Learning classifiers that distinguish aleatoric and epistemic uncertainty. *Information Sciences*, 255: 16–29. **DOI:** https://doi.org/10.1016/j.ins.2013.07.030

Burr Settles. 2012. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114. **DOI:** https://doi.org/10.2200/S00429ED1V01Y201207AIM018

Aditya Siddhant and Zachary C. Lipton. 2018. Deep Bayesian active learning for natural language processing: Results of a large-scale empirical study. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2904–2909, Brussels, Belgium. Association for Computational Linguistics. **DOI:** https://doi.org/10.18653/v1/D18-1318

Edwin Simpson and Iryna Gurevych. 2018. Finding convincing arguments using scalable Bayesian preference learning. *Transactions of the Association for Computational Linguistics*, 6:357–371. **DOI:** https://doi.org/10.1162/tacl_a_00026

Edwin Simpson and Iryna Gurevych. 2020. Scalable Bayesian preference learning for crowds. *Machine Learning*, 109(4):689–718. **DOI:** https://doi.org/10.1007/s10994-019-05867-2

Avi Singh, Larry Yang, Kristian Hartikainen, Chelsea Finn, and Sergey Levine. 2019. End-to-end robotic reinforcement learning without reward engineering. *arXiv preprint arXiv:1904.07854*. **DOI:** https://doi.org/10.15607/RSS.2019.XV.073

Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. 2012. Practical Bayesian optimization of machine learning algorithms. In *Advances* *in Neural Information Processing Systems*, pages 2951–2959.

Artem Sokolov, Julia Kreutzer, Stefan Riezler, and Christopher Lo. 2016. Stochastic structured prediction under bandit feedback. In *Advances in Neural Information Processing Systems*, pages 1489–1497.

William R. Thompson. 1933. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294. **DOI:** `https://doi.org/10.1093/biomet/25.3-4.285`

Louis L. Thurstone. 1927. A law of comparative judgment. *Psychological Review*, 34(4): 273–286. **DOI:** `https://doi.org/10.1037/h0070288`

Paolo Viappiani and Craig Boutilier. 2010. Optimal Bayesian recommendation sets and myopically optimal choice query sets. In *Advances in Neural Information Processing Systems*, pages 2352–2360.

Christian Wirth, Riad Akrour, Gerhard Neumann, and Johannes Fürnkranz. 2017. A survey of preference-based reinforcement learning methods. *The Journal of Machine Learning Research*, 18(1):4945–4990.

Peng Xu, Xiaofei Ma, Ramesh Nallapati, and Bing Xiang. 2019. Passage ranking with weak supervsion. *arXiv preprint arXiv:1905.05910*.

Jie Yang and Diego Klabjan. 2018. Bayesian active learning for choice models with deep gaussian processes. *arXiv preprint arXiv:1805.01867*.

Yi-Hsuan Yang and Homer H. Chen. 2011. Ranking-based emotion recognition for music organization and retrieval. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):762–774. **DOI:** `https://doi.org/10.1109/TASL.2010.2064164`